

**Πανεπιστήμιο Πειραιώς**

**Τμήμα Μεταπτυχιακών Σπουδών  
Δικτυοκεντρικά Συστήματα**



**Ανάπτυξη δικτυακών πληροφοριακών  
συστημάτων με χρήση τεχνολογιών OLAP -  
Online Analytical Processing**

**Υπεύθυνος: Καθηγητής Παναγιώτης Δεμέστιχας**

**Όνομα: Κατερίνα Ισκιούπη**  
**A.M.: ME / 0461**  
**E-mail: katerina.iskioupi@gmail.com**

## Ευχαριστίες

Η ολοκλήρωση της συγκεκριμένης εργασίας εκτός από τη δική μου προσωπική προσπάθεια οφείλεται και σε ανθρώπους που με στήριξαν και με βοήθησαν. Θα ήθελα να ευχαριστήσω τον καθηγητή μου κύριο Δεμέστιχα Παναγιώτη για την υπομονή του. Την εταιρεία EXIS I.T όπου χάριν σε αυτήν μπόρεσα να έχω το κατάλληλο υλικό και λογισμικό για να υλοποιήσω τη συγκεκριμένη εργασία. Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον κύριο Σεραφείμ Κοτρώτσο για τις πολύτιμες συμβουλές και την Ελένη Αντωνοπούλου για τη στήριξη και την ανοχή που είχε όλο αυτό το καιρό. Τέλος, θα ήθελα να ευχαριστήσω τον ξάδερφο μου Μάρκο Γουλιέλμο για την υπομονή του και τον χρόνο του.

## Περιεχόμενα

Ευχαριστίες.....	2
Περιεχόμενα.....	3
Κατάλογος Πινάκων .....	5
Κατάλογος Σχημάτων.....	6
Περίληψη .....	7
<b>1. Εισαγωγή .....</b>	<b>8</b>
<b>2. Συστήματα Data – Warehousing .....</b>	<b>10</b>
2.1. Εισαγωγή.....	10
2.2. Data Warehouse .....	11
2.3. Data Mining .....	17
2.4. Τρόπος εισαγωγής δεδομένων σε DW .....	19
2.5. Σχεσιακό Μοντέλο .....	22
2.6. Ερωτήματα σε πολυδιάστατη σχεδίαση.....	24
ROLAP .....	26
MOLAP.....	27
HOLAP .....	29
2.7. Επίλογος.....	30
<b>3. ORACLE .....</b>	<b>31</b>
3.1. Εισαγωγή.....	31
3.2. Oracle & OLAP .....	32
3.3. Oracle Analytic Workspace Manager .....	35
3.4. Oracle Warehouse Builder (OWB) .....	39
3.5. Oracle Business Intelligence Suite.....	43
3.6. Επίλογος.....	48
<b>4. Μεθοδολογία ανάπτυξης ενός DW .....</b>	<b>49</b>
4.1. Εισαγωγή.....	49
4.2. Ανάλυση Απαιτήσεων.....	50
Επιχειρησιακές Απαιτήσεις.....	51
Απαιτήσεις Χρηστών .....	52
Λεπτομερείς Απαιτήσεις συστήματος.....	52
Ιδιότητες Απαιτήσεων.....	53
4.3. Λογικός Σχεδιασμός.....	53
4.4. Φυσικός Σχεδιασμός .....	56
4.5. Διαδικασία ETL .....	58

Εξαγωγή Δεδομένων .....	59
Επεξεργασία Δεδομένων.....	60
Φόρτωση Δεδομένων .....	61
4.6. Ανάκτηση δεδομένων .....	63
4.7. Επίλογος.....	67
<b>5. Παράδειγμα χρήσης .....</b>	<b>70</b>
5.1. Εισαγωγή.....	70
5.2. Περιγραφή Σεναρίου.....	71
5.3. Ανάλυση αρχικής βάσης δεδομένων - ER .....	72
5.4. Ανάλυση Απαιτήσεων.....	78
5.5. Λογικός Σχεδιασμός.....	81
5.6. Φυσικός Σχεδιασμός .....	84
5.7. Διαδικασία ETL .....	100
5.8. Παραγωγή αναφορών και αναλύσεων .....	107
5.9. Επίλογος.....	111
<b>6. Συμπεράσματα.....</b>	<b>114</b>
<b>7. Αναφορές.....</b>	<b>118</b>
<b>8. Appendix.....</b>	<b>121</b>

## Κατάλογος Πινάκων

Πίνακας 1: Σχεσιακά αντικείμενα δεδομένων: Ορολογία .....	24
Πίνακας 2: Oracle Warehouse Builder – Διαδικασία ETL.....	42
Πίνακας 3: Oracle Warehouse Builder – Ποιότητα δεδομένων .....	42
Πίνακας 4: Διαφοροποίηση Data Warehouse με τα Πληροφοριακά Συστήματα.....	50
Πίνακας 5: Συγκεντρωτικός πίνακας μοντελοποίησης.....	84

## Κατάλογος Σχημάτων

Εικόνα 1: Επίπεδα αφαίρεσης απαιτήσεων .....	51
Εικόνα 2: Παράδειγμα Star σχεδίασης .....	55
Εικόνα 3: Αρχική σελίδα σύνδεσης στη πλατφόρμα Oracle Business Intelligence ....	64
Εικόνα 4: Επιλογή Αναφοράς .....	64
Εικόνα 5: Προβολή αναφοράς .....	65
Εικόνα 6: Δημιουργία νέας αναφοράς .....	66
Εικόνα 7: Προγραμματισμός Αναφοράς (1/2) .....	66
Εικόνα 8: Προγραμματισμός Αναφοράς (2/2) .....	67
Εικόνα 9: Αρχικό Σχήμα Βάσης δεδομένων .....	73
Εικόνα 10: Πίνακας COMPANIES .....	74
Εικόνα 11: Πίνακας STORES .....	75
Εικόνα 12: Πίνακας SALES .....	75
Εικόνα 13: Πίνακας PRODUCTS .....	75
Εικόνα 14: Πίνακας PR_CATEGORIES .....	76
Εικόνα 15: Πίνακας PR_TYPES .....	76
Εικόνα 16: Σχέση COMPANIES – STORES .....	77
Εικόνα 17: Σχέση ST_ADDRESSES - STORES .....	77
Εικόνα 18: Σχέση STORES - SALES .....	77
Εικόνα 19: Σχέση PRODUCTS – SALES .....	78
Εικόνα 20: Σχέσεις PRODUCTS - PR_CATEGORIES & PR_CATEGORIES - PR_TYPES .....	78
Εικόνα 21: Σχεδίαση Star .....	82
Εικόνα 22: Δημιουργία Analytic Workspace .....	87
Εικόνα 23: Δημιουργία Dimension .....	87
Εικόνα 24: Δημιουργία Dimension – Ετικέτα General .....	88
Εικόνα 25: Δημιουργία Level .....	88
Εικόνα 26: Δημιουργία Level – Ετικέτα General .....	88
Εικόνα 27: Δημιουργία Hierarchy .....	89
Εικόνα 28: Mapping .....	94
Εικόνα 29: Mapping dimension Company με τον σχεσιακό πίνακα της βάσης .....	95
Εικόνα 30: Δημιουργία Cube Sales – Ετικέτα General .....	96
Εικόνα 31: Δημιουργία Cube Sales – Ετικέτα Implementation Details .....	97
Εικόνα 32: Δημιουργία Cube – Ετικέτα Rules .....	97
Εικόνα 33: Δημιουργία measure SALES .....	98
Εικόνα 34: Δημιουργία Calculated measure – Βήμα 1 .....	99
Εικόνα 35: Δημιουργία Calculated measure – Βήμα 2 .....	99
Εικόνα 36: Επιλογή Maintain Dimension .....	108
Εικόνα 37: Αποτελέσματα με τη χρήση του Analytic Workspace and Manager για το έτος 2007 .....	109
Εικόνα 38: Αποτελέσματα για το δεύτερο τρίμηνο του έτους 2007 και το κατάστημα Πλαίσιο – Αθήνα – Κέντρο – Στουρνάρη 1 (Olap Analytical Workspace Manager) .....	109
Εικόνα 39: Αποτέλεσμα για το δεύτερο τρίμηνο του έτους 2007 και το κατάστημα Πλαίσιο – Αθήνα – Κέντρο – Στουρνάρη 1 (Oracle BI Publisher) .....	110
Εικόνα 40: Παράδειγμα αναφοράς με τη χρήση του Oracle BI Publisher και του MS Office .....	111

## Περίληψη

Στην εργασία αυτή παρουσιάζονται τεχνολογίες οι οποίες σκοπό έχουν να καλύψουν την ανάγκη διαχείρισης δεδομένων σε μεγάλα πληροφοριακά συστήματα που περιέχουν πολύπλοκα και μεγάλα σε όγκο δεδομένα. Συγκεκριμένα, οι τεχνολογίες αυτές σκοπό έχουν να διευκολύνουν την εύκολη πρόσβαση στα δεδομένα αυτά και την ανάλυση τους.

Η χρήση πληροφοριακών συστημάτων με συνδυασμό τις παραπάνω τεχνολογίες καταλήγει στην δημιουργία ενός Data Warehouse. Η υλοποίηση ενός Data Warehouse είναι η λύση που μπορεί να καλύψει τη παραπάνω ανάγκη. Σε αυτή την εργασία θα παρουσιαστεί ο τρόπος υλοποίησης ενός Data Warehouse, ποια βήματα πρέπει να εκτελούνται και με ποιο τρόπο μπορεί ο χρήστης να εκμεταλλευτεί τα δεδομένα που δημιουργούνται. Στη προσπάθεια υλοποίησης του Data Warehouse παρουσιάζονται νέοι όροι και τεχνολογίες όπως η τεχνολογία OLAP - Online Analytical Processing και η διαδικασία ETL - Extract, Transformation Loading. Η όλη υλοποίηση έγινε με τη χρήση της βάσης δεδομένων Oracle 10G καθώς είναι μια από τις πιο ισχυρές και ευρέως γνωστές βάσεις δεδομένων.

Τα αποτελέσματα που παρουσιάζονται αφορούν τόσο την εύκολη χρήση, επεξεργασία και παρουσίαση των δεδομένων.

## ΚΕΦΑΛΑΙΟ 1

### 1. Εισαγωγή

Πλέον κάθε εταιρεία, οργανισμός, επιχείρηση χρησιμοποιούν συστήματα πληροφορικής για τη διαχείριση και την ανάλυση των δεδομένων τους. Με το πέρασμα του χρόνου ο όγκος των δεδομένων αυξάνεται, γεγονός που δυσχεραίνει την παραπάνω διαδικασία. Μια από τις πιο διαδεδομένες λύσεις είναι η δημιουργία ενός Data Warehouse. Η δημιουργία του Data Warehouse μπορεί να γίνει με πολλά εργαλεία του εμπορίου όπως η Oracle, MS SQL Server, MySQL ή ακόμα να χρησιμοποιηθούν έτοιμα όπως είναι το Cognos.

Επιλέχθηκε να παρουσιαστεί η δημιουργία ενός Data Warehouse από την αρχή για να γίνουν αντιληπτά τα βήματα δημιουργίας και οι τεχνολογίες που χρησιμοποιούνται. Συγκεκριμένα, παρουσιάζεται η δημιουργία ενός Data Warehouse με τη χρήση της Oracle 10G καθώς είναι από τις πιο διαδεδομένες βάσεις δεδομένων με πληθώρα εργαλείων και ικανή να υποστηρίξει συστήματα με μεγάλο όγκο δεδομένων.

Στο Κεφάλαιο 2, εισάγεται ο αναγνώστης στις βασικές έννοιες του Data Warehouse όπως το Data Mining, η διαδικασία ETL, το σχεσιακό μοντέλο βάσεων δεδομένων και τέλος η τεχνολογία Olap. Ο συνδυασμός των παραπάνω εννοιών ολοκληρώνει τη δημιουργία και τη χρήση ενός Data Warehouse.

Στο Κεφάλαιο 3, αναφέρονται μερικά στοιχεία για τη βάση δεδομένων Oracle, αφού αυτή είναι που χρησιμοποιήθηκε στο σενάριο χρήσης. Περιγράφονται βασικά εργαλεία της όπως το Olap Analytic Workspace Manager, Oracle Warehouse Builder και τέλος το Oracle Business Intelligence Suite. Η Oracle παρέχει τα εργαλεία αυτά με σκοπό τη πλήρη διαχείριση του Data Warehouse για όλο τον κύκλο ζωής του.

Στο Κεφάλαιο 4, παρουσιάζονται τα βήματα δημιουργίας ενός Data Warehouse. Η διαδικασία ξεκινάει από την ανάλυση απαιτήσεων. Με βάση την ανάλυση απαιτήσεων γίνεται ο λογικός και φυσικός σχεδιασμός. Στην συνέχεια ακολουθεί η διαδικασία φόρτωσης των δεδομένων (ETL) και τέλος η παρουσίαση των αποτελεσμάτων.

Στο Κεφάλαιο 5, παρουσιάζεται πλέον το παράδειγμα χρήσης. Η υλοποίηση έχει γίνει με βάση τα βήματα που περιγράφονται στο Κεφάλαιο 4.



Τέλος, στο Κεφάλαιο 6, αναφέρονται τα συμπεράσματα της εργασίας αυτής. Ένα γενικό συμπέρασμα είναι ότι η χρήση Data Warehouse σε συνδυασμό με τη τεχνολογία Olap έχει καλύτερους χρόνους απόκρισης σε σχέση με τα καθιερωμένα πληροφοριακά συστήματα. Αυτό συμβαίνει γιατί τα δεδομένα μοντελοποιούνται με τέτοιο τρόπο ώστε η τεχνολογία Olap που χρησιμοποιείται να τα επεξεργάζεται πιο γρήγορα, σε αντίθεση με τη μοντελοποίηση ενός σχεσιακού μοντέλου. Ένα ακόμα συμπέρασμα είναι ότι ενώ η Oracle παρέχει πολλά εργαλεία τόσο για τη δημιουργία όσο και για τη χρήση ενός Data Warehouse, το κόστος είναι αρκετά υψηλό όταν χρησιμοποιείται από μικρές/ μεσαίες επιχειρήσεις/ οργανισμούς. Επίσης, η χρήση της Oracle αναγκάζει τις επιχειρήσεις να έχουν εξειδικευμένα άτομα για τη χρήση των εργαλείων που παρέχει καθώς δεν είναι ιδιαίτερα εύχρηστα για απλούς διαχειριστές βάσεων δεδομένων.

## 2. Συστήματα Data – Warehousing<sup>1</sup>

### 2.1. Εισαγωγή

Η διάδοση των συστημάτων πληροφορικής και ειδικότερα η ανάπτυξη των συστημάτων διαχείρισης βάσεων δεδομένων τόσο στον επιχειρηματικό τομέα όσο και στους κοινωφελείς οργανισμούς έχει αυξήσει κατά πολύ τον όγκο και την πολυπλοκότητα των διαχειριζόμενων δεδομένων. Το φαινόμενο που παρατηρείται είναι ότι ενώ υπάρχει πληθώρα δεδομένων και ευκολία πρόσβασης σε αυτά είναι δύσκολη η ανάλυσή τους και η εξαγωγή χρήσιμων συμπερασμάτων για την λήψη αποφάσεων. Ως ενδεδειγμένη λύση για αυτό το πρόβλημα έχει προκύψει η τεχνολογία OLAP – Online Analytical Processing (Άμεσα Συνδεδεμένη Αναλυτική επεξεργασία). Στην τεχνολογία ενσωματώνονται και επιμέρους εξελίξεις όπως ETL – Extract, Transformation and Loading (Εξαγωγή, Μετατροπή, Φόρτωση) και Data Mining (Εξόρυξη Δεδομένων) όπως θα δούμε αργότερα. Στην εργασία αυτή επικεντρωνόμαστε στη νέα βάση δεδομένων Oracle 10G και σε αυτό το κεφάλαιο θα αναλυθούν και θα παρουσιαστούν οι παραπάνω τεχνολογίες.

Στη παράγραφο 2.2 αναλύεται γενικά η τεχνολογία Data Warehouse, δίδεται ο ορισμός και περιγράφονται τα σημαντικότερα χαρακτηριστικά της.

Στη παράγραφο 2.3, παρουσιάζεται η τεχνολογία Data Mining και πώς αυτή είναι άμεσα συνδεδεμένη με τα Data Warehouses. Η τεχνολογία Data Mining είναι μια ευρέως χρησιμοποιημένη προσέγγιση για τον μετασχηματισμό των δεδομένων σε χρήσιμη μορφή, με σκοπό να βοηθήσουν στην εξόρυξη περιεκτικής γνώσης για τις συγκεκριμένες περιοχές πληροφοριών που ανήκουν τα δεδομένα [17].

Στη παράγραφο 2.4, θα αναλυθεί η τεχνολογία ETL. Η τεχνολογία ETL είναι η διαδικασία εκείνη η οποία είναι υπεύθυνη για την εξαγωγή δεδομένων από διάφορες πηγές, την μετατροπή και καθαρισμό αυτών των δεδομένων και την εισαγωγή τους σε ένα σύστημα Data Warehouse [18].

---

<sup>1</sup> Data warehouse: Σύστημα Αποθήκης Δεδομένων

Στη παράγραφο 2.5, θα παρουσιαστεί η έννοια του Σχεσιακού Μοντέλου καθώς τόσο η τεχνολογία OLAP όσο και το Data Warehouse χρησιμοποιούν και αποθηκεύουν τα δεδομένα σε σχεσιακές βάσεις δεδομένων.

Στη παράγραφο 2.6, παρουσιάζεται η τεχνολογία OLAP και τα ιδιαίτερα χαρακτηριστικά της που χρησιμοποιούνται στα συστήματα data warehouses. Η τεχνολογία OLAP είναι μια κατηγορία λογισμικού που επιτρέπει στους αναλυτές, τους διευθυντές και τους ανώτερους υπαλλήλους να έχουν τη δυνατότητα να αποκτούν επίγνωση των δεδομένων μέσω της γρήγορης, συνεπούς και διαλογικής πρόσβασης πιθανών πληροφοριών που έχουν μετασχηματιστεί από βασικά δεδομένα για να απεικονίσουν την πραγματική διαστατικότητα της επιχείρησης όπως αυτή είναι κατανοητή από τον χρήστη [19].

Τέλος, θα ακολουθήσει μια παράγραφος στην οποία θα παρουσιαστεί πως συνεργάζονται όλες οι παραπάνω τεχνολογίες μεταξύ τους για την καλύτερη αξιοποίηση του μεγάλου όγκου των πληροφοριών που είναι διαθέσιμα σε ένα επιχειρησιακό περιβάλλον.

## 2.2. Data Warehouse

Στόχος της παραγράφου είναι να παρουσιαστούν τα συστήματα Data warehouse και ο ρόλος τους στα πλαίσια του OLAP. Στα τέλη του 1980, η IBM χρειάστηκε να δημιουργήσει ένα αρχιτεκτονικό πρότυπο για τη ροή της πληροφορίας από τα διάφορα λειτουργικά συστήματα σε αντίστοιχα Decision Support Systems - DSS (συστήματα υποστήριξης αποφάσεων). Τα DSS είναι μια κατηγορία πληροφοριακών συστημάτων που σκοπό έχει την υποστήριξη της διαδικασίας λήψης αποφάσεων από τα αρμόδια στελέχη της διοίκησης [10]. Χωρίς κατάλληλη αρχιτεκτονική αποθήκευσης δεδομένων ένας μεγάλος όγκος πληροφορίας χρειαζόταν για να υποστηρίξει τα μέχρι τότε υπάρχοντα decision support systems. Στις μεγάλες εταιρείες μέχρι τότε τα συστήματα αυτά λειτουργούσαν ανεξάρτητα. Κάθε τέτοιο σύστημα εξυπηρετούσε διαφορετικούς χρήστες αλλά συχνά υπήρχε η ανάγκη να χρησιμοποιήσουν κοινή πληροφορία. Έτσι, για να μπορέσει κάθε σύστημα να έχει τη πληροφορία που χρειαζόταν, έπρεπε από διαφορετικές πηγές να συλλέξει τα

δεδομένα, να τα μετατρέψει και να τα ενοποιήσει. Η παραπάνω διαδικασία πολλές φορές επαναλαμβάνονταν για κάθε σύστημα.

Ένα Data Warehouse σύμφωνα με τον Ralph Kimball [29] είναι ένα αντίγραφο από δεδομένα δομημένα κατά τέτοιο τρόπο ώστε να εκτελούνται εύκολα τα υποερωτήματα και η ανάλυση τους. Ένας όμως πιο γενικός ορισμός είναι ότι ένα Data Warehouse είναι ένα μέρος στο οποίο αποθηκεύονται τα δεδομένα ώστε να είναι δυνατή η προσπέλαση και η ανάλυση τους. Η γενική σχεδίαση των συγκεκριμένων συστημάτων σκοπό έχει να βοηθήσει τους χρήστες να μπορούν να αναλύσουν και να δημιουργήσουν αναφορές πάνω στα δεδομένα.[7].

Ένα Data Warehouse σύστημα είναι μια μεγάλη αποθήκη εταιρικών δεδομένων. Τα δεδομένα αυτά μπορούν να βρίσκονται όλα σε ένα κεντρικό σημείο ή να είναι καταναμημένα σε διάφορα σημεία, τα λεγόμενα “Data Marts” (καταστήματα δεδομένων) έτοιμα για χρήση. Ένα Data Warehouse είναι επίσης μια συλλογή από τεχνολογίες decision making προς διευκόλυνση συγκεκριμένων ομάδων ( διοικητές, διευθυντές, αναλυτές ) στη λήψη άμεσων και ορθότερων αποφάσεων [1].

Ένα βασικό συστατικό συστημάτων data warehouse είναι η διαδικασία ETL (Extract, Transform, Load) η οποία χρησιμοποιείται κατά κόρον σε εργαλεία Business Intelligence (Επιχειρηματικής Νοημοσύνης) που θα αναλυθεί στη παράγραφο 2.4. Τα εργαλεία Business Intelligence χρησιμοποιούνται για την συλλογή, την ενσωμάτωση και την ανάλυση των επιχειρηματικών δεδομένων. Ένα Data Warehouse ενημερώνεται από εσωτερικές ή εξωτερικές πηγές με τη βοήθεια της διαδικασίας ETL ώστε τα δεδομένα να είναι σε μια συνεπή δομή.

Σε ένα Data Warehouse όλα τα ηλεκτρονικά δεδομένα έχουν ένα κοινό τρόπο επεξεργασίας. Αν η διαδικασία ανάλυσης και η διαδικασία παραγωγής αναφορών εκτελεστούν σε δεδομένα που βρίσκονται στις αρχικές πηγές τους, τότε θα ήταν επίπονες σε χρόνο καθώς θα έπρεπε για κάθε πηγή (μοντέλο) να δημιουργηθεί ένας συγκεκριμένος τρόπος ανάκτησης της πληροφορίας. Σε αντίθεση, με τη χρήση των data warehouses τα δεδομένα είναι συνεπή και έχουν τη κατάλληλη μορφή ώστε να γίνεται πιο εύκολα και γρήγορα η ανάλυση και η παραγωγή αναφορών κι αυτό γιατί

όλα τα προβλήματα ελέγχονται και αντιμετωπίζονται πριν τα δεδομένα αποθηκευτούν στη βάση.

Ένα άλλο βασικό χαρακτηριστικό του data warehouse είναι ότι καθώς τα δεδομένα είναι πλέον ανεξάρτητα από την αρχική τους πηγή παραμένουν ακίνδυνα αποθηκευμένα για αρκετά μεγάλο χρονικό διάστημα. [16].

Υπάρχει ποικιλία αρχιτεκτονικών συστημάτων Data Warehouses. Μια προτεινόμενη δομή είναι η παρακάτω [4] η οποία ορίζει ότι το σύστημα θα πρέπει να αποτελείται από τα παρακάτω επίπεδα (layers):

- Operational Database Layer (Επίπεδο Λειτουργικής βάσης): η πηγή των δεδομένων για το σύστημα data warehouse.
- Informational Access Layer (Επίπεδο Πρόσβασης Πληροφορίας): τα δεδομένα που θα χρησιμοποιηθούν για τη παραγωγή αναφορών και την ανάλυση καθώς επίσης και τα αντίστοιχα εργαλεία.
- Data Access layer (Επίπεδο Πρόσβασης Δεδομένων): η διεπαφή μεταξύ του λειτουργικού επιπέδου και του επιπέδου πρόσβασης πληροφορίας.
- Metadata layer (Επίπεδο Μεταδεδομένων): ένας κατάλογος δεδομένων ο οποίος περιγράφει και είναι πιο λεπτομερής από το λειτουργικό κατάλογο συστήματος. Υπάρχουν κατάλογοι που αφορούν ολόκληρο το warehouse αλλά υπάρχει και η δυνατότητα οι κατάλογοι αυτοί να αφορούν τα δεδομένα. Οι συγκεκριμένοι κατάλογοι χρησιμοποιούνται για να δηλώνουν ποια εργαλεία αναφορών και ανάλυσης έχουν πρόσβαση σε ποια δεδομένα.

Ο τρόπος αποθήκευσης των δεδομένων μπορεί να γίνει με δυο τρόπους. Υπάρχει η dimensional (διαστατική) προσέγγιση και η normalized (κανονικοποιημένη) προσέγγιση. Στη dimensional προσέγγιση τα δεδομένα χωρίζονται σε facts (γεγονότα) και dimensions (διαστάσεις). Τα facts είναι οι μετρήσεις που χρησιμοποιούν οι χρήστες για τη λήψη επιχειρησιακών αποφάσεων και συνήθως αναφέρονται σε αριθμούς, ενώ τα dimensions προσδίδουν ιδιότητες που είναι κατάλληλες στα fact δίνοντας τους έτσι μια συγκεκριμένη δομή.

Ένα βασικό πλεονέκτημα της dimensional προσέγγισης είναι ότι με τη συγκεκριμένη δομή οι χρήστες μπορούν να χρησιμοποιήσουν και να καταλάβουν το Data Warehouse πιο εύκολα. Επίσης, η ανάκτηση των δεδομένων από το Data Warehouse τείνει να είναι πολύ πιο γρήγορη. Από την άλλη μεριά, τα μειονεκτήματα της dimensional προσέγγισης είναι ότι η διαδικασία που καθιστά εφικτή την ακεραιότητα των facts και των dimensions είναι περίπλοκη καθώς ενημερώνονται από διαφορετικά λειτουργικά συστήματα και η τροποποίηση της δομής ανάλογα με το τρόπο που λειτουργεί η οργάνωση είναι δύσκολη.

Στην normalized προσέγγιση, τα δεδομένα στο data warehouse αποθηκεύονται με βάση τον κανόνα του Codd (Σχεσιακό Μοντέλο) [20]. Οι γενικές κατηγορίες των δεδομένων απεικονίζονται σε πίνακες στη βάση δεδομένων οι οποίοι ομαδοποιούνται σε θεματικές περιοχές. Χαρακτηριστικά παραδείγματα τέτοιων περιοχών είναι τα δεδομένα των πελατών, τα δεδομένα των προϊόντων κτλ. Το βασικό πλεονέκτημα της συγκεκριμένης αρχιτεκτονικής είναι η εύκολη εισαγωγή νέων δεδομένων στη βάση. Ενώ αντίθετα, βασικό μειονέκτημα της είναι ότι εξαιτίας του αριθμού των πινάκων από τους οποίους αποτελείται είναι δύσκολο για τους χρήστες τόσο να συγχωνεύσουν τα δεδομένα από διαφορετικές πηγές όσο και να τα επεξεργαστούν στη συνέχεια χωρίς να έχουν κατανοήσει πρώτα τη δομή δεδομένων του Data Warehouse.

Όπως έχει ήδη αναφερθεί και στην αρχή της παραγράφου, μια σημαντική απόφαση που θα πρέπει να ληφθεί κατά τον σχεδιασμό του Data Warehouse είναι ποια δεδομένα θα πρέπει να επεξεργαστούν και πώς. Ένα μεγάλο μέρος της υλοποίησης ενός Data Warehouse είναι να εντοπιστούν τα κοινά δεδομένα των διαφορετικών πηγών και να μην επαναληφθούν στη βάση δεδομένων. Γενικά για την διευθέτηση του παραπάνω προβλήματος χρησιμοποιούνται εργαλεία ETL.

Ο Inmon (1991) υποστήριζε την «top-down» (από πάνω προς τα κάτω) τεχνική προσέγγισης για τη σχεδίαση του data warehouse. Τα «ατομικά» δεδομένα, δηλαδή εκείνα τα οποία βρίσκονται στη λιγότερο λεπτομερή μορφή, αποθηκεύονται στο data warehouse. Τα «διαστατικά» δεδομένα, δηλαδή τα δεδομένα εκείνα τα οποία αφορούν τις επιχειρησιακές διαδικασίες ή συγκεκριμένα τμήματα δεν αποθηκεύονται αλλά δημιουργούνται από το Data Warehouse [13].

Μερικά από τα οφέλη που ένα Data Warehouse παρέχει είναι τα ακόλουθα:

- **Κοινό πρότυπο δεδομένων:** Παρέχει ένα κοινό πρότυπο στοιχείων για όλα τα στοιχεία ενδιαφέροντος, ανεξάρτητα από την πηγή των στοιχείων. Αυτό το καθιστά πιο ευλύγιστο στο να παρουσιάζει και να αναλύει τις πληροφορίες σε σχέση με τα δεδομένα που βρίσκονταν σε ανόμοιες πηγές οι οποίες χρησιμοποιήθηκαν για να ανακτήσουν τη πληροφορία όπως τα τιμολόγια πωλήσεων, παραλαβές διαταγής, γενικές δαπάνες κ.λπ.
- **Προετοιμασία δεδομένων:** Πριν τα δεδομένα φορτωθούν στο data warehouse από τις επιμέρους πηγές, οι ασυνέπειες προσδιορίζονται και επιλύονται. Αυτό απλοποιεί πολύ την υποβολή έκθεσης και την ανάλυση.
- **Μεγάλη αποθήκευση:** Τα δεδομένα του data warehouse τα ελέγχουν οι ίδιοι οι χρήστες του data warehouse. Αυτό σημαίνει ότι ακόμα κι αν τα δεδομένα διαγραφούν από την αρχική πηγή μετά τη πάροδο του χρόνου, μπορούν να αποθηκευτούν ακίνδυνα για μεγάλα χρονικά διαστήματα στο data warehouse.
- **Μικρή επιβάρυνση:** Επειδή είναι χωριστά από τα λειτουργικά συστήματα, τα data warehouses παρέχουν την ανάκτηση των στοιχείων χωρίς επιβράδυνση των λειτουργικών συστημάτων.
- **Καλύτερη σύγκριση αποτελεσμάτων:** Στα data warehouses οι εφαρμογές συστημάτων υποστήριξης απόφασης όπως οι εκθέσεις τάσης (π.χ., τα στοιχεία με τις περισσότερες πωλήσεις σε μια ιδιαίτερη περιοχή μέσα στα τελευταία δύο έτη), οι εκθέσεις εξαίρεσης, και οι εκθέσεις που παρουσιάζουν πραγματική απόδοση εναντίον των στόχων.
- **Ολοκλήρωση με άλλα συστήματα:** Τα data warehouses μπορούν να λειτουργήσουν από κοινού και με άλλα συστήματα. Ως εκ τούτου, ενισχύουν την αξία των λειτουργικών επιχειρηματικών εφαρμογών, ειδικότερα στα συστήματα διαχείρισης σχέσης πελατών (CRM).

Όπως όλες οι τεχνολογίες τα συστήματα data warehouse παρουσιάζουν επίσης κάποια μειονεκτήματα:

- **Υψηλό κόστος λειτουργίας:** Κατά τη διάρκεια ζωής ενός data warehouse υπάρχουν υψηλές δαπάνες. Αυτό συμβαίνει γιατί τα συστήματα αυτά δεν είναι στατικά και το κόστος συντήρησής τους είναι υψηλό.
- **Γρήγορη απαξίωση:** Τα data warehouses μπορούν να θεωρηθούν «ξεπερασμένα» πολύ γρήγορα καθώς είναι μεγάλο κόστος για τις εταιρείες να τα τροφοδοτήσουν με τα πιο πρόσφατα δεδομένα.
- **Αστοχίες με προγράμματα:** Πολλές φορές υπάρχει σύγχυση μεταξύ των data warehouses και άλλων λειτουργικών προγραμμάτων. Αυτό συνεπάγεται η ίδια λειτουργία να έχει υλοποιηθεί τόσο στο data warehouse όσο και στο λειτουργικό σύστημα.

Ο παραδοσιακός ρόλος ενός data warehouse είναι να συλλέγει και να οργανώνει τα ιστορικά επιχειρησιακά δεδομένα έτσι ώστε να μπορούν να αναλυθούν με σκοπό να βοηθήσουν τη διαχείριση στη λήψη επιχειρησιακών αποφάσεων. Μέχρι σήμερα, η πρόσβαση σε ένα data warehouse είχε περιοριστεί μόνο από έμπειρους χρήστες βάσεων δεδομένων που θα μπορούσαν να δημιουργήσουν και να εκτελέσουν περίπλοκα ερωτήματα ώστε να ανακτήσουν τη πληροφορία που χρειάζονται και να τη χρησιμοποιήσουν οι υπεύθυνοι αναλυτές κι αυτοί που λαμβάνουν αποφάσεις. Πλέον όμως, η χρήση των data warehouses έχει γίνει ολοένα και πιο έντονη με αποτέλεσμα να υπάρχει ανάγκη πρόσβασης στο data warehouse από τελικούς χρήστες οι οποίοι δεν έχουν γνώσεις βάσης δεδομένων. Αυτή είναι μια βασική διαφορά ενός Data Warehouse από μια απλή βάση δεδομένων ότι δηλαδή απαιτείται από απλούς χρήστες να διαχειριστούν τα δεδομένα και να κάνουν προχωρημένες εταιρικές αναλύσεις.

Μια πολύ γνωστή χρήση των data warehouse είναι στα συστήματα Γεωγραφικών Πληροφοριών (Geographical Information Systems). Οι βασικές απαιτήσεις που πληρούνται από το data warehouse είναι η εύκολη προσπελασιμότητα, το καλοσχεδιασμένο λειτουργικό σύστημα διεπαφής και η δυνατότητα παραμετροποίησης.

Άλλες χρήσεις των συστημάτων data warehouses παρουσιάζονται σε συστήματα λήψης αποφάσεων όπου οργανώνονται έτσι ώστε να αποδίδεται μια γενικότερη εικόνα ολόκληρου του οργανισμού. Χαρακτηριστικό παράδειγμα χρήσης είναι τα



συστήματα Customer Relationship Management - CRM (Συστήματα Διαχείρισης Πελατείας) τα οποία είναι συστήματα που υποστηρίζουν διαδικασίες και δεδομένα για την σχέση που αναπτύσσεται μεταξύ εταιρίας και πελάτη. Στα συστήματα αυτά, οι επιχειρήσεις εντοπίζουν εύκολα τις συμπεριφορές και τις προτιμήσεις των πελατών, τηρούν χρονικό των επαφών μαζί τους και μπορούν να υποστηρίξουν την ανάλυση πληροφοριών για θέματα πωλήσεων και marketing.

Στη παράγραφο αυτή αναλύθηκε η έννοια ενός Data Warehouse. Με την ενσωμάτωση των τεχνολογιών ETL, OLAP και Data Mining ένα τέτοιο σύστημα βοηθά στην αξιοποίηση μεγάλου όγκου πληροφορίας και στην υποστήριξη αναλύσεων. Οι τεχνολογίες ETL, Data Mining και OLAP θα παρουσιαστούν στις παρακάτω παραγράφους αντίστοιχα.

### 2.3. Data Mining

Στο παρόν κεφάλαιο θα αναλύσουμε την ορολογία Data Mining – Εξόρυξη Δεδομένων. Η τεχνολογία αυτή συνδέεται άμεσα με την τεχνολογία OLAP. Το Data Mining είναι μια παραδοσιακή μεθοδολογία ανάλυσης δεδομένων και μπορεί να οριστεί ως η διαδικασία επιλογής, έρευνας και μοντελοποίησης μεγάλου όγκου δεδομένων. Σκοπό έχει την ανακάλυψη προηγούμενων σχεδίων για την απόκτηση ενός επιχειρησιακού πλεονεκτήματος [16]. Καθώς η τεχνολογία Data Mining είναι μια μεθοδολογία ανάλυσης των δεδομένων, χρησιμοποιείται κατά κόρον για business intelligence σε επιχειρήσεις καθώς επίσης και από επιχειρήσεις που ασχολούνται με οικονομικά στοιχεία και αναλύσεις.

Παραδοσιακά, οι αναλυτές εντοπίζουν τη χρήσιμη πληροφορία από τα αποθηκευμένα δεδομένα, αλλά ο αυξανόμενος όγκος των δεδομένων κάνει απαραίτητη την προσέγγιση μέσω υπολογιστή. Τα δεδομένα έχουν αυξηθεί τόσο σε μέγεθος όσο και σε πολυπλοκότητα, που πλέον η ανάλυση τους δεν μπορεί να γίνει άμεσα αλλά έμμεσα και αυτόματα με τη χρήση διαφόρων εργαλείων. Οι σύγχρονες τεχνολογίες των υπολογιστών και των δικτύων έχουν κάνει τη συλλογή και την οργάνωση των δεδομένων ευκολότερη. Παρόλα αυτά τα δεδομένα που εξάγονται πρέπει να μετατραπούν ώστε να είναι χρήσιμα. Με τη χρήση του Data Mining ο

αναλυτής μπορεί να προσδιορίσει τις διάφορες τάσεις, τις βασικές ιδιότητες των επιχειρησιακών διαδικασιών και τις ευκαιρίες που προκύπτουν για την επίτευξη των στόχων [23].

Οι επιχειρήσεις για πολύ καιρό έχουν χρησιμοποιήσει ισχυρούς υπολογιστές που ανιχνεύουν και επεξεργάζονται τον όγκο των δεδομένων. Οι συνεχείς καινοτομίες στη δύναμη επεξεργασίας υπολογιστών, ο δίσκος αποθήκευσης και τα λογισμικά που χρησιμοποιούνται στη στατιστική αυξάνουν εντυπωσιακά την ακρίβεια και την χρησιμότητα της ανάλυσης. Η ανάσυρση δεδομένων, συνήθως, χρησιμοποιείται για να καλύψει δύο διαφορετικές διαδικασίες : την εύρεση και την πρόβλεψη της γνώσης. Η εύρεση της γνώσης παρέχει πληροφορίες που έχουν αναγνωρίσιμη μορφή και μπορούν να γίνουν κατανοητές από ένα απλό χρήστη. Αντίστοιχα, η πρόβλεψη της γνώσης παρέχει προβλέψεις μελλοντικών γεγονότων. Μία άλλη χρήση του Data Mining είναι πάνω σε σύστημα simulation (προσομοίωσης). Με τη βοήθεια των συστημάτων αυτών και τη χρήση της τεχνολογία Data Mining οι αναλυτές μπορούν τόσο να προβλέψουν τυχόν ανεπιθύμητες καταστάσεις ενός μοντέλου / προγράμματος όσο και να βελτιστοποιήσουν τις ήδη υπάρχουσες.

Το Data Mining είναι ευάλωτο στην αντιστοίχιση των αντικειμένων με τον πραγματικό κόσμο κι αυτό γιατί τα δεδομένα από τον πραγματικό κόσμο μπορούν να έχουν άγνωστες και αμοιβαίες σχέσεις οι οποίες δεν απεικονίζονται άμεσα και δεν είναι ευδιάκριτες. Μια αναπόφευκτη αδυναμία της τεχνολογίας αυτής είναι ότι τα κρίσιμα δεδομένα που μπορούν να εξηγούν τις σχέσεις μεταξύ τους δεν παρατηρούνται ποτέ. Εναλλακτικές προσεγγίσεις χρησιμοποιούν μια πειραματική προσέγγιση επιλογής. Τα συστήματα Διαχείρισης Πελατών (CRM) κάνουν χρήση κυρίως του Data mining. Πλέον, δε καλούνται όλοι οι πελάτες για τη προσφορά ενός προϊόντος αλλά επιλέγονται αυτοί που έχουν τις καλύτερες προοπτικές με βάση το Data mining. Περίπλοκες μέθοδοι χρησιμοποιούνται για να βοηθήσουν στην εκστρατεία προώθησης ενός προϊόντος τόσο για το ποια προσφορά καλύπτει ένα πελάτη όσο και με ποιο μέσο είναι καλύτερα να προωθηθεί το προϊόν ώστε να ενημερωθεί το κατάλληλο σύνολο πελατών. Τα στοιχεία που συγκεντρώνονται με το Data mining μπορούν να χρησιμοποιηθούν για τον εντοπισμό ομάδων πελατών και τον τρόπο προώθησης.

Γι αυτό το λόγο, οι επιχειρήσεις οι οποίες χρησιμοποιούν Data mining αναγνωρίζουν ότι ο αριθμός των προβλεπτικών μοντέλων μπορεί πολύ γρήγορα να γίνει μεγάλος. Οι επιχειρήσεις αυτές δε χρησιμοποιούν ένα πρότυπο μόνο για να εντοπίσουν και να προσελκύσουν νέους πελάτες αλλά αντίθετα δημιουργούν διαφορετικούς κανόνες και πρότυπα ανάλογα με την περιοχή, την ηλικία κτλ. Έτσι, αντί να στέλνουν μια προσφορά σε όλους του υποψήφιους πελάτες, στέλνουν μόνο σε αυτούς οι οποίοι είναι πιο πιθανό να τη δεχτούν.

Μερικοί, επιπλέον τομείς, στους οποίους χρησιμοποιείται τα τελευταία χρόνια η διαδικασία του Data Mining είναι η βιοπληροφορική, η γενετική, η ιατρική και η εκπαίδευση. Για παράδειγμα, στον τομέα της ανθρώπινης γενετικής, ο σκοπός της χρήσης του Data Mining είναι να γίνει κατανοητή η σχέση της δομής του DNA με την ευαισθησία στις ασθένειες. Αυτό υλοποιείται τόσο με τη χρήση simulation προγραμμάτων όσο και με μοντέλα προβλέψεις, χρησιμοποιώντας τη τεχνολογία Data Mining πάνω στα πειράματα. Με τη παραπάνω διαδικασία βελτιώνεται η διάγνωση, η πρόσληψη και η θεραπεία των ασθενειών [15].

Συμπερασματικά, το Data Mining χρησιμοποιείται σε συστήματα Data Warehouse λόγω του αυξημένου όγκου πληροφορίας. Σε συνδυασμό με την τεχνολογία OLAP οι αναλυτές μπορούν να εξάγουν εύκολα δεδομένα και συμπεράσματα από τη βάση του Data Warehouse, Ο τρόπος εισαγωγής των δεδομένων στη βάση δεδομένων ενός Data Warehouse περιγράφεται στη παράγραφο 2.4.

#### **2.4. Τρόπος εισαγωγής δεδομένων σε DW**

Τα συστήματα Data Warehouse για την εισαγωγή δεδομένων χρησιμοποιούν τη διαδικασία ETL. Η διαδικασία ETL χρησιμοποιείται σε συστήματα Data Warehouse με σκοπό να εξάγει δεδομένα από εξωτερικές πηγές, να μετατρέψει τα εξαγόμενα δεδομένα ανάλογα με τις επιχειρησιακές ανάγκες και τελικά να αποθηκεύσει τα δεδομένα στο σύστημα, δηλαδή, στο Data Warehouse.

Κατά τη διαδικασία της εξαγωγής, τα επιθυμητά δεδομένα πρέπει στην αρχή να αναγνωριστούν και στη συνέχεια να εξαχθούν από πολλές διαφορετικές πηγές. Οι πηγές αυτές μπορεί να είναι είτε βάσεις δεδομένων άλλων συστημάτων είτε

εφαρμογές. Πολλές φορές, η αναγνώριση των δεδομένων δεν είναι εύκολη κι αυτό έχει σαν αποτέλεσμα να εξάγονται περισσότερα δεδομένα από όσα χρειάζονται και η αναγνώριση να γίνεται σε μετέπειτα στάδιο. Ανάλογα με τη πηγή βέβαια, κατά τη διαδικασία της εξαγωγής μπορεί να γίνει και η διαδικασία της μετατροπής [2].

Το επόμενο βήμα, μετά την εξαγωγή των δεδομένων, είναι η μεταφορά στο σύστημα που θέλουμε ή σε ένα ενδιάμεσο σύστημα ώστε να υποστούν την απαραίτητη επεξεργασία. Ανάλογα με ποιο τρόπο θα ακολουθηθεί, κάποιες μετατροπές μπορούν να εκτελεστούν κατά τη διαδικασία της εξαγωγής. Για παράδειγμα, ένα υποερώτημα SQL θα μπορούσε κατά τη διαδικασία του SELECT να κάνει και την ανάλογη μετατροπή [2]. Στη περίπτωση που η μετατροπή δεν είναι εύκολη να γίνει κατά τη διαδικασία της εξαγωγής, θα τρέξουν κανόνες σε σειρά ώστε να εκτελέσουν το συγκεκριμένο βήμα. Τα βασικά είδη κανόνων είναι τα εξής:

- **Κανόνας «μετάφρασης» των δεδομένων.** Χαρακτηριστικό παράδειγμα είναι στη πηγή να υπάρχει ένα πεδίο Φύλο το οποίο δέχεται τις τιμές 1 (Λεκτικό: Άντρας), 2 (Λεκτικό: Γυναίκα) και στη βάση δεδομένων του data warehouse να θέλουμε να περιέχονται τα λεκτικά κι όχι τα αναγνωριστικά.
- **Κανόνας υπολογισμός κάποιου νέου πεδίου.** Για παράδειγμα, το ποσοστό των ανθρώπων που έχουν αγοράσει ένα συγκεκριμένο προϊόν.
- **Ένωση δεδομένων** που προέρχονται από διαφορετικές πηγές.
- **Κανόνες ελέγχου** των δεδομένων. Κανόνες με τους οποίους ελέγχονται τα δεδομένα τόσο σε επίπεδο επιχείρησης, όσο και επίπεδο της βάσης δεδομένων. Για παράδειγμα, μια επιχείρηση μπορεί να μην δέχεται ως έγκυρη τιμή ενός αριθμού δεκαδικά ψηφία. Θα υπάρχει κανόνας ο οποίος θα ελέγχει τη μορφή του αριθμού. Ένα άλλο παράδειγμα είναι το πεδίο στη βάση δεδομένων να δέχεται μόνο αριθμούς κι όχι χαρακτήρες οπότε και θα υπάρχει αντίστοιχος κανόνας ώστε να ελέγχονται τα δεδομένα πριν εισαχθούν.
- **Σύγκριση των νέων δεδομένων** με τα ήδη υπάρχοντα με σκοπό να αποφασιστεί αν χρειάζεται να εισαχθούν στο data warehouse ή όχι.

Στη συνέχεια, αφού έχει ολοκληρωθεί και η διαδικασία της μετατροπής των δεδομένων, ακολουθεί η αποθήκευση τους στο σύστημα data warehouse. Η

διαδικασία αυτή, ανάλογα με τις ανάγκες της επιχείρησης εκτελείται συνήθως σε αραιά διαστήματα. Πολλές φορές όμως, αν υπάρχει ανάγκη για ιστορικότητα μπορεί να λαμβάνει μέρος και ανά ώρα.

Ο χρόνος που χρειάζεται για να προστεθούν ή να επαναγραφούν κάποια δεδομένα στο data warehouse είναι στρατηγικής σημασίας για το χρόνο εκτέλεσης όλης της διαδικασίας. Όπως επίσης, επηρεάζει και το πόσο πρόσφατα δεδομένα πρέπει να περιέχει το data warehouse. Όσο πιο πρόσφατα δεδομένα τόσο πιο συχνά τρέχει όλη η διαδικασία.

Η διαδικασία ETL είναι ιδιαίτερα σημαντική καθώς είναι αυτή που ενημερώνει το σύστημα data warehouse. Μια άλλη σημαντική χρήση της διαδικασίας αυτής είναι η ενσωμάτωση των δεδομένων με legacy συστήματα. Η παραπάνω χρήση δε θα μας απασχολήσει στη παρούσα εργασία.

Όπως έχει ήδη αναφερθεί, η διαδικασία ETL μπορεί να εκτελεστεί με δυο διαφορετικούς τρόπους. Η πρώτη είναι να χρησιμοποιηθεί μια ανεξάρτητη, αυτόματη εφαρμογή, εξωτερική της βάσης, η οποία θα κάνει όλη τη διαδικασία και η δεύτερη τα δεδομένα από διαφορετικές πηγές να συγκεντρωθούν στη βάση δεδομένων και να επεξεργαστούν εκεί.

Στη πρώτη περίπτωση, χρησιμοποιούνται ειδικά εργαλεία τα οποία συνεργάζονται με διάφορες πηγές δεδομένων και προσπαθούν να τις ενσωματώσουν ώστε να τρέξουν τα απαραίτητα βήματα του ETL. Στη περίπτωση που υπάρχουν δεδομένα στη βάση του warehouse τότε το ίδιο το warehouse αποτελεί μια πηγή δεδομένων. Συνήθως, οι πιο πολλές αλλαγές που θα πρέπει να γίνουν μπορούν να αντιμετωπιστούν με απλή SQL. Αφού, τα δεδομένα μετατραπούν στη επιθυμητή μορφή αποθηκεύονται στη βάση ή αν η βάση περιέχει ήδη δεδομένα ενημερώνονται κατάλληλα. Η παραπάνω διαδικασία εμπεριέχει διάφορους κινδύνους [2]:

- Η εξέλιξη των βημάτων της διαδικασίας ETL γίνεται όλη από εξωτερικό μηχανισμό. Στη περίπτωση, που ο μηχανισμός αυτός αποτύχει σε κάποιο από τα βήματα τότε όλη η διαδικασία θα «παγώσει» εκεί με αποτέλεσμα να υπάρχει δυσχέρεια ως προς την ενημέρωση του warehouse.

- Ανάλογα με την όλη αρχιτεκτονική που θα χρησιμοποιηθεί, το εξωτερικό αυτό εργαλείο θα έχει τον έλεγχο όλης της διαδικασίας ETL. Αυτό συνεπάγεται ότι θα παρέχει μηχανισμούς επανάκτησης και επανεργοποίησης της διαδικασίας.
- Ο τρόπος υλοποίησης δημιουργεί πρόβλημα καθώς είναι δύσκολο να διατηρηθεί και να επεκταθεί.
- Τέλος, με αυτό το τρόπο δεν εκμεταλλευόμαστε όλες τις δυνατότητες της βάσης δεδομένων.

Μια άλλη τεχνική είναι να μαζευτούν όλα τα δεδομένα από τις διάφορες πηγές σε ενδιάμεσους πίνακες στη βάση δεδομένων και στη συνέχεια αφού επεξεργαστούν κατάλληλα να αποθηκευτούν στο warehouse. Τα μειονεκτήματα της συγκεκριμένης λογικής είναι τα παρακάτω [2]. Θα πρέπει να αποθηκεύεται η αρχική πληροφορία ως έχει πριν γίνει το βήμα της επεξεργασίας στη βάση. Αυτό, συμβαίνει ιδιαίτερα όταν οι πηγές δεν είναι σχεσιακή βάση δεδομένων.

Παρόλο, που η γλώσσα SQL και PL/SQL μπορεί να καλύψει σε μεγάλο ποσοστό τις απαιτήσεις του βήματος της επεξεργασίας των δεδομένων, υπάρχει περίπτωση να μην είναι η βέλτιστη λύση η χρήση των συγκεκριμένων για την αντιμετώπιση των ETL προβλημάτων.

## 2.5. Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο δημιουργήθηκε το 1969 από τον Edgar Codd [20]. Το μοντέλο αυτό στηρίζεται στην σχεσιακή άλγεβρα και λογική. Η σχεσιακή άλγεβρα αποτελείται από μια συλλογή τελεστών, οι οποίοι δέχονται τελεστέους που είναι σχέσεις και επιστρέφουν αποτελέσματα που είναι επίσης σχέσεις. Τρεις από τους πιο βασικούς τελεστές είναι restrict (περιορισμός), project (προβολή) και join (σύζευξη). Ο σχεσιακός λογισμός είναι μια εναλλακτική προσέγγιση για ένα σχεσιακό μοντέλο. Η κυριότερη διαφορά μεταξύ τους είναι η εξής: ενώ η άλγεβρα παρέχει ένα σύνολο ρητών πράξεων που μπορούν να χρησιμοποιηθούν για να υποδείξουν στο σύστημα πως να κατασκευάσει κάποια ζητούμενη σχέση, ο λογισμός απλώς παρέχει ένα συμβολισμό για τη διατύπωση του ορισμού αυτής της ζητούμενης σχέσης. Ουσιαστικά, θα μπορούσαμε να πούμε ότι ο σχεσιακός λογισμός απλώς περιγράφει

ποιο είναι το πρόβλημα, ενώ η άλγεβρα καθορίζει μια διαδικασία για την επίλυση του προβλήματος. Οι διαφορές τους όμως είναι επιφανειακές και το γεγονός είναι ότι η άλγεβρα και ο λογισμός είναι προσεγγίσεις εντελώς ισοδύναμες μεταξύ τους. Για κάθε παράσταση της άλγεβρας υπάρχει μια ισοδύναμη παράσταση του λογισμού και το ανάποδο [21].

Το σχεσιακό μοντέλο είναι ένας τρόπος ο οποίος βοηθάει στην αναπαράσταση των δεδομένων. Ο σκοπός του μοντέλου είναι να μπορούν να γραφούν προγράμματα και διαδικασίες στους υπολογιστές τα οποία να εκφράζουν υποερωτήματα στις βάσεις δεδομένων χωρίς να υπάρχει απώλεια πληρότητας και με την ύπαρξη κανόνων ακεραιότητας. Οι κανόνες ακεραιότητας χρησιμοποιούνται για την ακρίβεια και την ορθότητα των δεδομένων της βάσης δεδομένων. Για την ακεραιότητα των δεδομένων το σύστημα χρειάζεται να είναι ενήμερο για ορισμένους κανόνες που δε θα πρέπει να παραβιάζουν οι χρήστες. Οι κανόνες αυτοί καθορίζεται από τον υπεύθυνο διαχείρισης της βάσης δεδομένων σε κάποια κατάλληλη γλώσσα. Τέτοιοι κανόνες δεν είναι ειδικά καθορισμένοι για το χρήστη (user-specific) [21].

Το σχεσιακό μοντέλο ασχολείται με τρεις πλευρές των δεδομένων. Τη δομή των δεδομένων (τα αντικείμενα), την ακεραιότητα των δεδομένων και το χειρισμό των δεδομένων (τους τελεστές). Ο χειρισμός των δεδομένων, όπως έχει αναφερθεί, μπορεί να υλοποιηθεί με δύο διαφορετικούς αλλά ισοδύναμους τρόπους, την σχεσιακή άλγεβρα και τον σχεσιακό λογισμό. Οι σημαντικότεροι από τους όρους που χρησιμοποιούνται σε σχέση με τη δομή των δεδομένων είναι: relation (σχέση), tuple (συστοιχία), cardinality (η πληθυκότητα), attribute (το γνώρισμα), degree (ο βαθμός), domain (το πεδίο ορισμού) και το primary key (πρωτεύον κλειδί). Εμπειρικά, relation θεωρούμε τον πίνακα στη βάση δεδομένων, ενώ tuple αντιστοιχεί σε μια γραμμή ενός τέτοιου πίνακα και attribute αντιστοιχεί σε μια στήλη. Το πλήθος των tuples ονομάζεται cardinality και το πλήθος των γνωρισμάτων degree. Το domain είναι ένα σύνολο τιμών από το οποίο τα συγκεκριμένα attributes των συγκεκριμένων relations αντλούν τις συγκεκριμένες τιμές τους. Στον παρακάτω πίνακα συνοψίζονται οι όροι.

Τυπικός σχεσιακός όρος	Άτυποι ισοδύναμοι όροι
Relation	Πίνακας

Tuple	Γραμμή πίνακα ή εγγραφή
Cardinality	Αριθμός γραμμών
Attribute	Στήλη ή πεδίο
Degree	Αριθμός στηλών
Primary key	Μοναδικό αναγνωριστικό
Domain	Σύνολο επιτρεπτών τιμών

**Πίνακας 1: Σχεσιακά αντικείμενα δεδομένων: Ορολογία**

Όσον αφορά το χειρισμό των δεδομένων, η σχεσιακή άλγεβρα, όπως ορίστηκε από τον Codd [22] αποτελείται από οκτώ τελεστές οι οποίοι χωρίζονται σε δυο βασικές ομάδες. Στη πρώτη ομάδα, ανήκουν οι παραδοσιακές πράξεις των συνόλων, δηλαδή, η ένωση, η τομή, η διαφορά και το καρτεσιανό γινόμενο. Ενώ στη δεύτερη ομάδα ανήκουν ειδικές σχεσιακές πράξεις όπως ο restrict (περιορισμός), project (προβολή), join (σύζευξη) και divide (διαίρεση). Η τελευταία ομάδα πράξεων ακολουθείται και στον σχεσιακό λογισμό.

Η τεχνολογία OLAP συνδέεται άμεσα με το σχεσιακό μοντέλο, αφού κατά κύριο λόγο όλα τα δεδομένα από τα οποία ενημερώνεται το σχήμα μια πολύ-διάστατης αρχιτεκτονικής προέρχονται από σχεσιακές βάσεις δεδομένων που ακολουθούν το πρότυπο του σχεσιακού μοντέλου.

## 2.6. Ερωτήματα σε πολυδιάστατη σχεδίαση

Μία από τις βασικές τεχνολογίες που ασχολείται η εργασία αυτή είναι η τεχνολογία OLAP. Η ορολογία OLAP πρωτοεμφανίστηκε το 1993 από τον Codd, ο οποίος είναι και ο πρώτος που δημιούργησε την έννοια των σχεσιακών βάσεων. Στην πραγματικότητα είναι η τεχνολογία που χρησιμοποιείται για να παρέχει τις απαντήσεις σε ερωτήματα πολυδιάστατης φύσης.. Η αρχική ορολογία ήταν OLTP (Online Transaction Processing) όπου από εκεί βγήκε και ο όρος OLAP [9]. Η τεχνολογία OLAP στηρίζεται πάνω στη σχεσιακή βάση δεδομένων και όλα τις τα στοιχεία αποθηκεύονται σε αυτή. Παρόλα αυτά όμως τα δεδομένα δεν αποθηκεύονται με βάση το σχεσιακό μοντέλο αλλά χρησιμοποιείται η Star (Αστέρι) ή Snowflake



(Χιονοστιβάδα) σχεδίαση. Τόσο η σχεδίαση Star όσο και η σχεδίαση Snowflake έχουν πάρει την ορολογία τους από το τρόπο με τον οποίο φαίνονται σε ένα γραφικό μοντέλο. Στη συνέχεια της παραγράφου, θα δοθούν παραδείγματα ώστε να γίνει κατανοητή η διαφορά των δύο αυτών σχεδιασμών.

Οι βάσεις δεδομένων που δημιουργούνται για χρήση OLAP χρησιμοποιούν ένα πολυδιάστατο μοντέλο δεδομένων που επιτρέπει αναλυτικά και ειδικά ερωτήματα με γρήγορο χρόνο εκτέλεσης. Στην ουσία, χρησιμοποιούν αρχές από τη μεθοδολογία των navigation database (βάσης δεδομένων πλοήγησης) και των ιεραρχικών βάσεων δεδομένων οι οποίες σε σχέση με τις σχεσιακές είναι ταχύτερες [10]. Στην ουσία είναι η κατάλληλη τεχνολογία για να μπορέσει να υποστηρίξει την λήψη αποφάσεων. Το κύριο τμήμα ενός OLAP συστήματος είναι το data warehouse, που είναι η μια decision-support βάση δεδομένων η οποία ενημερώνεται περιοδικά με τη βοήθεια της τεχνολογίας ETL (η οποία περιγράφεται στη παράγραφο 2.4).

Τα αποτελέσματα μιας ερώτησης OLAP παρουσιάζονται με τη μορφή πίνακα. Η OLAP τεχνολογία αποτελείται από dimensions (διαστάσεις), levels (επίπεδα), hierarchies (ιεραρχίες), attributes (γνωρίσματα), cubes (κύβοι), measures (μέτρα) και calculated measures (υπολογισμένα μέτρα). Τα dimensions παρέχουν το περιεχόμενο και τη δομή των πραγματικών δεδομένα και έχουν το ρόλο σειράς και στήλης του πίνακα. Τα dimensions σχηματίζουν τις πλευρές ενός cube και για να καθοριστούν πλήρως ο χρήστης πρέπει πρώτα να καθορίσει τα levels, τις ιεραρχίες και τα attributes. Τα levels αντιπροσωπεύουν τις θέσεις μέσα στην ιεραρχία και καθορίζουν τα σημεία στα οποία θα συνοψίζονται τα δεδομένα. Οι ιεραρχίες οργανώνουν τα δεδομένα σε διαφορετικά επίπεδα συνάθροισης. Η ιεραρχική δομή επιτρέπει στους αναλυτές να μπορούν να εντοπίσουν τις τάσεις στα υψηλότερα επίπεδα και κατεβαίνοντας σε level να αναγνωρίζουν τους παράγοντες που προκαλούν τη παραπάνω τάση. Τα attributes παρέχουν τις πληροφορίες για τα ξεχωριστά μέλη ενός dimension. Χρησιμοποιούνται για την συλλογή και την οργάνωση των μελών ενός dimension. Ο κύβος παρέχει τα μέτρα που έχουν την ίδια μορφή, αυτό σημαίνει ότι έχουν ακριβώς τα ίδια dimensions. Οι άκρες ενός cube περιέχουν τα μέλη των dimensions και το σώμα ενός cube περιέχει τις τιμές των δεδομένων. Τα measures χρησιμοποιούνται για να αποθηκεύσουν τα δεδομένα γεγονότος μέσα στο cube και

αποτελούν τις τιμές των κελιών του πίνακα. Κάθε OLAP ερώτημα στηρίζεται πάνω στο cube, ο οποίος ονομάζεται επίσης και πολυδιάστατος κύβος ή hypercube. Τα μεταδεδομένα του κύβου συνήθως δημιουργούνται είτε από star schema είτε από snowflake schema μιας σχεσιακής βάσης δεδομένων.

Υπάρχουν τρία διαφορετικά είδη OLAP.

- ROLAP – Relation OLAP
- MOLAP – Multidimensional OLAP
- HOLAP – Hybrid OLAP

### **ROLAP**

Το ROLAP (Relation OLAP) είναι μια άμεση αναλυτική επεξεργασία η οποία εκτελεί πολυδιάστατη ανάλυση των δεδομένων δυναμικά. Τα δεδομένα αυτά είναι αποθηκευμένα σε σχεσιακή βάση δεδομένων κι όχι σε πολυδιάστατη βάση δεδομένων (η οποία θεωρείται συνήθως πρότυπο OLAP).

Τα ερωτήματα ROLAP συνεργάζονται άμεσα με τις σχεσιακές βάσεις δεδομένων. Τα βασικά δεδομένα και οι διαστάσεις αποθηκεύονται σε σχεσιακοί πίνακες και νέοι πίνακες δημιουργούνται ώστε να αποθηκευτούν οι αθροισμένες πληροφορίες. Σε αντίθεση με το MOLAP, η χρήση του ROLAP δεν απαιτεί τον προ – υπολογισμό και την αποθήκευση των πληροφοριών. Πιο συγκεκριμένα, έχει απευθείας πρόσβαση στα δεδομένα μιας σχεσιακής βάσης δεδομένων και δημιουργεί κατάλληλα ερωτήματα SQL ώστε να υπολογίσει τη πληροφορία στο κατάλληλο επίπεδο που ο τελικός χρήστης τη ζητά. Είναι πιθανό να δημιουργηθούν επιπλέον πίνακες στη βάση στους οποίους θα αποθηκεύονται τα αθροίσματα των δεδομένων.

Τα πλεονεκτήματα της χρήσης ROLAP παρουσιάζονται παρακάτω:

- Η τεχνολογία ROLAP είναι πιο ευέλικτη με τον χειρισμό μεγάλου όγκου αποθηκευμένων δεδομένων, ιδιαίτερα με μοντέλο μεγάλων διαστάσεων.
- Καθώς υπάρχει ποικιλία στα εργαλεία με τα οποία φορτώνονται τα δεδομένα και ιδιαίτερα με τη χρήση της τεχνολογίας ETL, ο χρόνος εκτέλεσης της παραπάνω

διαδικασίας είναι σημαντικά μικρότερος σε σχέση με τις αυτόματες διαδικασίες του MOLAP.

- Τα δεδομένα εφόσον είναι αποθηκευμένα σε σχεσιακή βάση δεδομένων μπορούν να προσεγγιστούν με απλή SQL, χωρίς να χρειάζεται απαραίτητα κάποιο εργαλείο OLAP.
- Τα εργαλεία ROLAP μπορούν πιο εύκολα να διαχειριστούν μη αθροιστικά δεδομένα όπως λεπτομερείς περιγραφές. Σε αντίθεση με τα MOLAP, όπου η επεξεργασία των παραπάνω αργεί πάρα πολύ.
- Με τη χρήση ROLAP είναι πιο εύκολο να διατηρηθεί η ασφάλεια της βάσης δεδομένων καθώς μπορεί να γίνει έλεγχος και φιλτράρισμα πάνω στα αποτελέσματα ενός ερωτήματος SQL. (where by clause). Για παράδειγμα, συγκεκριμένη ομάδα χρηστών να μην έχει πρόσβαση σε συγκεκριμένους πίνακες.

Κάποια από τα αρνητικά του ROLAP είναι τα εξής:

- Καθώς η φόρτωση των αθροιστικών δεδομένων γίνεται με τη χρήση προσαρμοσμένης διαδικασίας ETL, χρειάζεται επιπλέον χρόνος ανάπτυξης και περισσότερος κώδικας για να υποστηριχθεί η παραπάνω διαδικασία.
- Στη περίπτωση που η δημιουργία των πινάκων που περιέχουν τα αθροιστικά δεδομένα παρακάμπτεται, ο χρόνος απόκρισης των αποτελεσμάτων αυξάνεται καθώς θα πρέπει να ερωτηθούν οι μεγαλύτεροι και λεπτομερείς πίνακες. Αυτό μπορεί να αντιμετωπιστεί ως ένα βαθμό με τη δημιουργία των πινάκων αυτών αλλά και πάλι δεν είναι πανάκεια αφού θα πρέπει να δημιουργηθούν πίνακες για κάθε συνδυασμό των διαστάσεων.
- Δεδομένου ότι η χρήση του ROLAP στηρίζεται στην SQL για όλους τους υπολογισμούς, στη περίπτωση που οι υπολογισμοί αυτοί είναι δύσκολοι η χρήση της SQL δε μπορεί να αντεπεξέλθει. Τέτοια παραδείγματα είναι η δημιουργία κατανομών, αναφορές οικονομικών στοιχείων κτλ.

## **MOLAP**

Το MOLAP (Multidimensional OLAP) είναι μια άμεση αναλυτική επεξεργασία η οποία χρησιμοποιεί πολυδιάστατη βάση δεδομένων. Το MOLAP είναι η κλασική

μορφή του OLAP και συνήθως αναφερόμαστε σε αυτό το είδος ακριβώς ως OLAP. Χρησιμοποιεί δομές βάσεων δεδομένων που στη γενικότερη μορφή είναι καλύτερες όσον αφορά το χρονικό διάστημα, το χρόνο, τη θέση και το προϊόν. Ο τρόπος που κάθε διάσταση θα αθροιστεί καθορίζεται εκ των προτέρων από μια ή περισσότερες ιεραρχίες. Η MOLAP τεχνολογία χρειάζεται προ – υπολογισμό των δεδομένων και την αποθήκευση τους στον κύβο. Τα δεδομένα αποθηκεύονται σε μια βελτιστοποιημένη πολυδιάστατη μορφή κι όχι σε μια σχεσιακή βάση δεδομένων.

Κάποια θετικά χαρακτηριστικά της τεχνολογίας αυτής είναι τα παρακάτω:

- Λόγω της βελτιστοποιημένης αποθήκευσης, των πολυδιάστατων ευρετηρίων και του caching τα ερωτήματα έχουν γρήγορη απόδοση.
- Σε αντίθεση με τη σχεσιακή βάση τα δεδομένα αποθηκεύονται στη βάση αφού πρώτα επεξεργαστούν με τεχνικές συμπίεσης. Αποτέλεσμα της παραπάνω διαδικασίας είναι να έχουν μικρότερο μέγεθος και κατ' επέκταση να χρειάζονται λιγότερο μέγεθος σκληρού δίσκου.
- Χάρη στη κατανομή των δεδομένων σε ιεραρχίες και διαστάσεις, υπάρχει αυτοματοποιημένος υπολογισμός αθροισμάτων των δεδομένων στα υψηλά επίπεδα της κάθε ιεραρχίας.

Μερικά αρνητικά της MOLAP τεχνολογίας είναι:

- Η διαδικασία επεξεργασίας των δεδομένων και η αποθήκευση τους στη βάση μπορεί να εκτελείται για αρκετά μεγάλο χρονικό διάστημα. Συνήθως, το μειονέκτημα αυτό αντιμετωπίζεται με το να επεξεργάζεται μόνο τα στοιχεία που έχουν αλλάξει ή προστέθηκαν κι όχι όλο το σύνολο των δεδομένων.
- Συνήθως, τα εργαλεία MOLAP δυσκολεύονται να ανακτήσουν δεδομένα από μοντέλα που χρησιμοποιούν διαστάσεις με πολύ μεγάλη λεπτομέρεια και μέγεθος.
- Έχει παρατηρηθεί σε ορισμένα εργαλεία MOLAP ότι δυσκολεύονται να ενημερώσουν ή να ανακτήσουν δεδομένα από μοντέλα που έχουν παραπάνω από 10 διαστάσεις.
- Τέλος, η προσέγγιση MOLAP δημιουργεί περιττή επανάληψη των δεδομένων.

## HOLAP

Το HOLAP (Hybrid OLAP) είναι ένας συνδυασμός του ROLAP και του MOLAP. Αυτή η ιδιαιτερότητα, δίνει τη δυνατότητα στους χρήστες να αποθηκεύουν κάποια μέρη από τα δεδομένα σε πολυδιάστατη βάση δεδομένων και άλλα σε σχεσιακή βάση δεδομένων.

Για το συγκεκριμένο τύπο OLAP δεν υπάρχει σαφής ορισμός για το τι αποτελεί «υβριδικό OLAP». Το βασικό χαρακτηριστικό είναι ότι η βάση δεδομένων θα χωρίσει τα δεδομένα μεταξύ σχεσιακής και εξειδικευμένης αποθήκευσης. Για παράδειγμα, για μερικούς προμηθευτές, μια βάση δεδομένων HOLAP θα χρησιμοποιεί σχεσιακούς πίνακες για να φυλάξει τη μεγαλύτερη ποσότητα λεπτομερών δεδομένων στους σχεσιακούς πίνακες και χρησιμοποιεί την εξειδικευμένη (πολυδιάστατη) αποθήκευση για αποθήκευση δεδομένων που είναι περισσότερο αθροιστικά ή λιγότερο λεπτομερή.

Στην πραγματικότητα, η επιλογή HOLAP συνδυάζει τη λειτουργικότητα του ROLAP και του MOLAP. Επιτρέπει την αποθήκευση ενός μέρους των δεδομένων σε μια βάση MOLAP και ενός άλλου μέρους σε μια βάση ROLAP. Για την υλοποίηση του HOLAP χρησιμοποιούνται δυο είδη σχεδίασης. Η πρώτη είναι ο «Κάθετος χωρισμός των δεδομένων». Με αυτό το τρόπο, τα αθροίσματα αποθηκεύονται σε MOLAP βάση για να υπάρχει γρήγορη ανταπόκριση στη περίπτωση ερωτήματος και τα πιο λεπτομερή δεδομένα αποθηκεύονται σε βάση ROLAP όπου με αυτό το τρόπο βελτιστοποιούμε το χρόνο επεξεργασίας των κύβων. Η δεύτερη σχεδίαση είναι ο «Οριζόντιος χωρισμός των δεδομένων». Στη συγκεκριμένη περίπτωση, ένα μέρος των δεδομένων, συνήθως τα πιο πρόσφατα, αποθηκεύονται σε βάση MOLAP ενώ τα παλαιότερα δεδομένα σε βάση ROLAP [11].

Στο κεφάλαιο αυτό έγινε μια εισαγωγή στις βασικές έννοιες της τεχνολογίας OLAP και στα πιο διαδεδομένα είδη που αυτή εμφανίζεται. Στο Κεφάλαιο 3 θα παρουσιαστεί η τεχνολογία OLAP συγκεκριμένα στη βάση δεδομένων Oracle 10G καθώς εκεί έγκειται και το θέμα της εργασίας αυτής.

## 2.7. Επίλογος

Συμπερασματικά, η χρήση ενός Data Warehouse καλύπτει την ανάγκη διαχείρισης αυξημένου όγκου δεδομένων. Με τη βοήθεια του Data Warehouse και της τεχνολογίας Data Mining οι χρήστες μπορούν εύκολα τόσο να εξορύξουν διάφορα βασικά δεδομένα όσο και να πάρουν βασικές αποφάσεις και στρατηγικές για τις επιχειρήσεις τους.

Η βάση δεδομένων που χρησιμοποιεί ένα σύστημα Data Warehouse δεν περιέχει πρωτογενή δεδομένα. Αντίθετα, γεμίζει από άλλα συστήματα αφού περάσουν από επεξεργασία, ανάλυση και μετατροπή. Η διαδικασία αυτή ονομάζεται ETL και είναι στην ουσία η τεχνολογία με την οποία φορτώνονται τα δεδομένα στη βάση του Data Warehouse. Η βάση δεδομένων δεν είναι σχεδιασμένη βασισμένη στο Σχεσιακό μοντέλο αλλά σχεδιάζεται ώστε να ακολουθεί τους κανόνες της τεχνολογίας OLAP.

Η σχεδίαση μπορεί να είναι είτε σε σχήμα Star είτε σε σχήμα Snowflake. Τα δεδομένα που βρίσκονται στη βάση είναι πλέον κατανοητά και εύκολα διαχειριζόμενα από χρήστες που δεν έχουν γνώσεις βάσεων δεδομένων καθώς περιγράφουν έννοιες του πραγματικού κόσμου.

Στα επόμενα κεφάλαια, περιγράφεται η συμμετοχή της Oracle 10G στις τεχνολογίες που παρουσιάστηκαν σε αυτό το κεφάλαιο.

## Κεφάλαιο 3

### 3. ORACLE

#### 3.1. Εισαγωγή

Η εργασία αυτή υλοποιήθηκε τόσο με τη χρήση της Oracle 10G όσο και με τη χρήση των εργαλείων που αυτή προσφέρει. Συγκεκριμένα, χρησιμοποιήθηκαν τα παρακάτω εργαλεία.

- Oracle Database 10G R2: σχεσιακή βάση δεδομένων της Oracle
- Oracle Client Tools 10G: client tools με τα οποία συνδεόμαστε στη παραπάνω βάση δεδομένων
- Olap Analytic Workspace Manager and Worksheet: εργαλείο με το οποίο σχεδιάζουμε και δημιουργούμε τις dimensions, levels, hierarchies και cubes.
- Oracle Warehouse builder 10G: όπου χρησιμοποιήθηκε για τη διαδικασία ETL.
- Oracle Business Intelligence Suite (BI): εργαλείο το οποίο χρησιμοποιήθηκε για τη παρουσίαση των αναφορών ανακτώντας τα δεδομένα από τη βάση δεδομένων του Data Warehouse.

Στο κεφάλαιο αυτό, θα παρουσιαστούν τα εργαλεία που χρησιμοποιήθηκαν για την εργασία αυτή. Πιο συγκεκριμένα, στη παράγραφο 3.2, θα παρουσιαστούν μερικά χαρακτηριστικά στοιχεία της Oracle σε συνεργασία με τη τεχνολογία OLAP.

Στη παράγραφο 3.3, θα παρουσιαστεί το εργαλείο Oracle Analytic Workspace, με το οποίο είναι δυνατή η δημιουργία των dimensions.

Στη παράγραφο 3.4, θα αναφερθούμε στο εργαλείο Oracle Warehouse Builder, το οποίο είναι αυτό που χρησιμοποιήθηκε για τη δημιουργία του Data Warehouse και την διαδικασία ETL, δηλαδή τη μεταφορά και τη μορφοποίηση των δεδομένων από τη σχεσιακή βάση «πηγή» στη σχεσιακή βάση «προορισμό».

Η Oracle, επίσης, παρέχει εργαλεία και για την εύκολη ανάκτηση των δεδομένων. Ένα από αυτά είναι το Oracle Business Intelligence Suite (BI) το οποίο συνδέεται με τη τεχνολογία Olap και δημιουργεί εύκολα και γρήγορα αναφορές χωρίς ο χρήστης

να χρειάζεται να έχει ειδικές γνώσεις βάσεων δεδομένων. Στη παράγραφο 3.5 το εργαλείο αυτό παρουσιάζεται αναλυτικά.

### 3.2. Oracle & OLAP

Η συγκεκριμένη εργασία ασχολείται με την Oracle 10G R2 και τη τεχνολογία OLAP. Στη παράγραφο αυτή θα αναφερθούν μερικά χαρακτηριστικά της βάσης δεδομένων Oracle 10G R2 και της δυνατότητά της να υποστηρίζει τη τεχνολογία OLAP. Η Oracle είναι η πρώτη εταιρεία που περιέχει ενσωματωμένο OLAP server (εξυπηρετητή). Η Oracle Olap παρέχει ένα φυσικό πολυδιάστατο χώρο αποθήκευσης και γρήγορο χρόνο ανταπόκρισης όταν αναλύονται δεδομένα σε πολλαπλές διαστάσεις. Η βάση δεδομένων Oracle παρέχει μια πλήρη υποστήριξη για την ανάλυση των δεδομένων. Στη συνέχεια της παραγράφου παρουσιάζονται μερικά από τα χαρακτηριστικά αυτά που την κάνουν ιδανική για την ανάλυση δεδομένων.

Η βάση δεδομένων της Oracle 10G παρουσιάζει τις νέες τεχνολογίες συμπίεσης των cubes (κύβων), παραλληλισμό και διαμερισμό των συνόλων πολυδιάστατων δεδομένων. Η τεχνολογία συμπίεσης των cubes βελτιστοποιεί τη συνάθροιση και την αποθήκευση των πολυδιάστατων δεδομένων. Ο διαμερισμός και ο παραλληλισμός επιτρέπουν την αποδοτικότερη διαχείριση των warehouses (αποθήκες δεδομένων) που περιέχουν μεγάλα πολυδιάστατα σύνολα δεδομένων.

Πιο συγκεκριμένα, η τεχνολογία συμπίεσης των cubes σκοπό έχει να βελτιστοποιήσει τη συνάθροιση και την αποθήκευση καθώς επίσης και το χρόνο εκτέλεσης υποερωτήσεων σε αραιά σύνολα δεδομένων [32]. Ένα αραιό σύνολο δεδομένων ορίζεται εκείνο το σύνολο το οποίο περιέχει πολλά δεδομένα με null (κενές) τιμές στο σύνολό του.

Γενικά, η επιλογή Oracle OLAP παρέχει μια πολύτιμη διορατικότητα στις επιχειρηματικές διαδικασίες και αγορές χρησιμοποιώντας τα χαρακτηριστικά γνωρίσματα που υπήρχαν σε εξειδικευμένες βάσεις δεδομένων OLAP. Επειδή η Oracle OLAP είναι πλήρως ενσωματωμένη στη σχεσιακή βάση δεδομένων, όλα τα δεδομένα και τα μεταδεδομένα αποθηκεύονται και διαχειρίζονται μέσα από το



διαχειριστικό περιβάλλον που παρέχει η ίδια η Oracle για τη σχεσιακή βάση δεδομένων.

Η Oracle OLAP επιτρέπει στις επιχειρήσεις να μπορούν εύκολα και ανά πάσα στιγμή να έχουν επίγνωση της επιχειρησιακής απόδοσης. Η Oracle OLAP καθιστά τη δυνατότητα αυτή παρέχοντας τα παρακάτω:

- Απλό πρότυπο χρηστών με σκοπό την επιχειρησιακή χρήση
- Εξαιρετική απόδοση υπό-ερωτημάτων, υπολογισμό και προετοιμασίας των δεδομένων
- Ανοικτή πρόσβαση στους χρήστες και στην SQL αλλά και στο αντίστοιχο εργαλείο που αφορά τα πολυδιάστατα αντικείμενα.

Η Oracle παρέχει ένα δωρεάν εργαλείο το οποίο ονομάζεται Analytic Workspace Manager (παράγραφος 3.3) για να βοηθήσει τους χρήστες να δημιουργούν και να διαχειρίζονται τα περιεχόμενα του Oracle OLAP. Αντίστοιχα, υπάρχουν και κάποια εργαλεία του Oracle Warehouse Builder (παράγραφος 3.4) τα οποία κι αυτά χρησιμοποιούνται για δημιουργία και τη διαχείριση του περιεχομένου Oracle OLAP [32].

Η αρχιτεκτονική της Oracle OLAP αποτελείται από τα δυο παρακάτω μέρη:

- OLAP Engine (μηχανή)
- Σχεδίαση ETL Εργαλεία

Το OLAP Engine περιέχει ένα δυνατό διαστατικό μοντέλο δεδομένων το οποίο απλοποιεί την εργασία που χρειάζεται για τον υπολογισμό των δεδομένων και τη δημιουργία υπό-ερωτήσεων, υποστηρίζει πολυδιάστατους υπολογισμούς και λειτουργίες προγραμματισμού ενώ παρέχει και ένα μοντέλο συναλλαγής, κατάλληλο για τη – «τι θα γίνει εάν» ανάλυση και μοντελοποίηση. Επίσης, έχει πολύ καλό χρόνο απόκρισης των υπο-ερωτήσεων και ο χρήστης μπορεί να έχει πρόσβαση σε όλα τα δεδομένα του Oracle OLAP. Η συγκεκριμένη μηχανή είναι βασισμένη στα παρακάτω χαρακτηριστικά [32]:

- OLAP Cubes (OLAP κύβοι)

➤ SQL Interface (SQL διεπαφή)

➤ OLAP API

Οι cubes είναι βασικό χαρακτηριστικό της επιλογής OLAP. Παρέχουν την αποθήκευση των δεδομένων, τη γρήγορη αναπροσαρμογή, τη γρήγορη εκτέλεση υποερωτήσεων, τους αναλυτικούς υπολογισμούς και τη δημιουργία μεταδεδομένων. Οι cubes OLAP υποστηρίζουν μια ευρεία ποικιλία υπολογισμών. Η διεπαφή SQL για τους πολυδιάστατους τύπους δεδομένων παρουσιάζει τους cubes και τις dimensions σαν σχεσιακές όψεις (views) και επιτρέπει να μπορούν να ερωτηθούν από εφαρμογές που είναι βασισμένες σε SQL. Τόσο τα αποθηκευμένα όσο και υπολογισμένα measures (μέτρα), στην λεπτομερή και συνοπτική τους μορφή, αντιπροσωπεύονται σαν columns στις όψεις των cubes. Συμπερασματικά το OLAP API έχει δύο σκοπούς: τον ορισμό και τη διατήρηση του cube και τις σχετικές dimensions συμπεριλαμβάνοντας την αντιστοίχιση των συγκεκριμένων αντικειμένων με τους σχεσιακούς πίνακες. Αφετέρου, το OLAP API μπορεί να χρησιμοποιηθεί για να ερωτηθούν τα cubes.

Η Oracle έχει δημιουργήσει δυο εργαλεία σχεδίασης και ETL. Το πρώτο είναι το Analytic Workspace Manager (AWM) και το δεύτερο Warehouse Builder (WB). Ο σκοπός του AWM είναι να βοηθήσει τους επιχειρηματικούς χρήστες να διαμορφώσουν γρήγορα έναν ή μια σειρά OLAP cubes. Ο κύριος περιορισμός του AWM είναι η έλλειψη μηχανής μετασχηματισμού των δεδομένων καθώς το εργαλείο υποθέτει ότι η πηγή περιέχει σωστά δεδομένα έτοιμα για χρήση. Το AWM δίνει τη πλήρη δύναμη της μηχανής OLAP παρέχοντας εύκολη πρόσβαση στις αναλυτικές λειτουργίες που είναι διαθέσιμες μέσα στη μηχανή. Το OWB είναι ένα εργαλείο ETL το οποίο υποστηρίζει και την σχεσιακή και την πολυδιάστατη σχεδίαση. Περιέχει πολλά χαρακτηριστικά γνωρίσματα του AWM αλλά στερείται την πρόσβαση σε μερικά από τα περιπλοκότερα χαρακτηριστικά γνωρίσματα υπολογισμού και διαμόρφωσης των δεδομένων.

Όπως παρουσιάστηκε στη συγκεκριμένη παράγραφο, η Oracle στην έκδοση 10G R2 έχει πλέον εισάγει τη τεχνολογία OLAP με τη δημιουργία νέων εργαλείων που σκοπό έχουν να διευκολύνουν τη δημιουργία και τη διαχείριση των πολυδιάστατων

αντικειμένων ενώ παράλληλα διατηρεί τα χαρακτηριστικά της όσον αφορά στα σχεσιακά αντικείμενα. Στη συνέχεια του κεφαλαίου παρουσιάζονται τα εργαλεία της Oracle που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας.

### 3.3. Oracle Analytic Workspace Manager

Σε αυτή τη παράγραφο θα παρουσιαστεί ένα από τα βασικά εργαλεία που χρησιμοποιήθηκαν για τη δημιουργία του Data Warehouse. Το εργαλείο της Oracle Analytic Workspace Manager εγκαθίσταται με την εγκατάσταση των Client Tools. Στη συγκεκριμένη εργασία έχουμε χρησιμοποιήσει τα Oracle Client Tools 10G. Το Oracle Workspace Manager χρησιμοποιείται για τη σχεδίαση και την υλοποίηση του διαστατικού μοντέλου.

Η σχεδίαση και η υλοποίηση του διαστατικού μοντέλου είναι μια από τις βασικές εργασίες ώστε να υλοποιηθεί μια εφαρμογή επιχειρηματικής ευφυΐας. Άλλες εργασίες που θα πρέπει να γίνουν είναι η απόκτηση των επιθυμητών δεδομένων και η δημιουργία αντίστοιχων εφαρμογών για τους τελικούς χρήστες [28]. Το Oracle Analytic Workspace αποτελεί ένα από τα διαχειριστικά εργαλεία για τη χρήση του Oracle OLAP και παρέχει στον χρήστη τη δυνατότητα να σχεδιάσει και να διαχειριστεί τα σύνολα των πολυδιάστατων τύπων δεδομένων της Oracle 10G.

Η Oracle 10G μπορεί να υποστηρίξει την υλοποίηση των διαστατικών μοντέλων. Τα μοντέλα αυτά υλοποιούνται σε analytic workspace (χώροι εργασίας ανάλυσης). Ένα analytic workspace μπορεί να θεωρηθεί το σύνολο από πολυδιάστατους τύπους δεδομένων και η φυσική υλοποίηση του λογικού διαστατικού μοντέλου.

Τα analytic workspaces παρέχουν την καλύτερη υποστήριξη για την απάντηση απρόβλεπτων ερωτημάτων. Σε σχέση με τους σχεσιακούς τύπους δεδομένων μπορεί να είναι πιο αποτελεσματικά για ερωτήματα που είναι από τη φύση τους τύπου «ad-hoc» (επί τόπου). Ως ερωτήματα «ad-hoc» ορίζονται εκείνα τα οποία μπορούν να ερωτούν κάθε περιοχή του μοντέλου δεδομένων κάθε στιγμή και χρησιμοποιώντας υπολογισμούς που έχουν οριστεί από τον χρήστη [28].

Τα δεδομένα παρουσιάζονται πλήρως στον τελικό χρήστη ανεξάρτητα εάν τα συγκεντρωτικά δεδομένα υπολογίζονται από την αρχή του υπολογισμού εν μέρη ή εξ' ολοκλήρου εκείνη την ώρα που ζητούνται. Επιπλέον, τα analytic workspace παρέχουν πιο σύνθετες μεθόδους συνάθροισης όπως μη-προσθετικές μέθοδοι, ιεραρχίες και σταθμισμένους υπολογισμούς.

Πιο συγκεκριμένα, το εργαλείο της Oracle Analytic Workspace Manager έχει τις παρακάτω δυνατότητες [28]:

➤ Απλοποίηση του διαστατικού μοντέλου

Οι χρήστες του Analytic Workspace μπορούν εύκολα και γρήγορα να καθορίσουν ποια θα είναι τα αντικείμενα τους. Με το πάτημα μόνο συγκεκριμένου κουμπιού μπορούν να δημιουργήσουν διαστατικά αντικείμενα ενώ παράλληλα το ίδιο το εργαλείο τα υλοποιεί και τα αρχικοποιεί στο analytic workspace. Αν ο χρήστης γνωρίζει ποιες είναι οι ανάγκες των αναφορών που θέλει να δημιουργήσει, μπορεί άμεσα να καθορίσει τη μορφή των δεδομένων του. Αυτό μπορεί να το κάνει χωρίς να είναι απαραίτητα διαθέσιμο το σχεσιακό σχήμα.

➤ Υλοποιεί τη φυσική αποθήκευση των δεδομένων

Μια επιπλέον λειτουργία που παρέχει το εργαλείο Analytic Workspace Manager είναι ότι μπορεί να διαχειριστεί την υλοποίηση των διαστατικών αντικειμένων στη βάση δεδομένων Oracle. Ο χρήστης περιγράφει το λογικό μοντέλο και παρέχει πληροφορίες για τη μορφή των δεδομένων και στην συνέχεια το Analytic Workspace Manager δημιουργεί τα αντίστοιχα αντικείμενα με βάση τη πληροφορία που έχει δώσει ο χρήστης. Τόσο οι διαχειριστές των βάσεων δεδομένων όσο και οι υπεύθυνοι ανάπτυξης εφαρμογών μπορούν παράλληλα να εκμεταλλευτούν το γραφιστικό περιβάλλον που παρέχει το Analytic Workspace Manager αλλά και όλες τις λειτουργίες που παρέχει η βάση δεδομένων Oracle όπως η συμπίεση (compression), ο κερματισμός (partitioning) και η συνάθροιση (aggregation).

Το Analytic Workspace Manager έχει τη δυνατότητα να διαχειρίζεται αποτελεσματικά τα αραιά σύνολα δεδομένων που είναι κοινά για εφαρμογές OLAP. Συνήθως, συμπιέζονται εκείνοι οι cubes (κύβοι) οι οποίοι αποτελούνται από πολλές dimensions (διαστάσεις) ή με dimensions με πολλά μέλη ή τέλος με dimensions που

έχουν βαθιές ιεραρχίες. Τα παραπάνω χαρακτηρίζουν τα αραιά σύνολα δεδομένων. Η τεχνολογία συμπίεσης των cubes της Oracle δίνει τη δυνατότητα στους cubes να υλοποιούνται σε λιγότερο χρόνο και με τη χρήση λιγότερου σκληρού δίσκου παρέχοντας όμως άριστη απόδοση στην εκτέλεση των υπό - ερωτημάτων.

Η απόδοση και η εξελισιμότητα μεγάλων measures (μέτρα) μπορεί να βελτιστοποιηθεί με τον κατακερματισμό της φυσικής αποθήκευσης των cubes. Τα πλεονεκτήματα που παρέχονται με την ενέργεια αυτή είναι η παράλληλη φόρτωση των δεδομένων και η συνάθροιση τους. Η διάσπαση των δεδομένων κατά τη φόρτωση και τη συνάθροιση και γενικά οι υπολογισμοί σε μικρότερες μονάδες εργασίας μπορούν να διαχειριστούν καλύτερα τη διαθέσιμη μνήμη ελαττώνοντας έτσι τις όποιες I/O (Input/Output) δυσχέρειες δημιουργούνται.

Τέλος, το Analytic Workspace Manager επιτρέπει στο χρήστη να φορτώνει τα δεδομένα σε φύλλο-επίπεδο και να μπορεί να τα συναθροίζει με βάση συγκεκριμένους κανόνες συνάθροισης. Δεδομένα επίσης μπορούν να φορτωθούν κι αν έχουν ήδη υπολογιστεί σε άλλο σύστημα. Όταν τα λεπτομερή δεδομένα φορτώνονται στο analytic workspace μπορούν να είναι είτε προϋπολογισμένα και αποθηκευμένα είτε να υπολογιστούν εκείνη την ώρα. Το Analytic Workspace Manager επιτρέπει στο χρήστη να επιλέξει ποια επίπεδα των ιεραρχιών θα είναι προϋπολογισμένα έτσι ώστε να υπολογίζονται αυτόματα μόνο τα επίπεδα τα οποία όντως χρειάζεται να υπολογιστούν. Συχνά, ένα μόνο μικρό σύνολο των δεδομένων είναι προϋπολογισμένο και τα υπόλοιπα υπολογίζονται κατά τη δημιουργία απάντησης μια υπό-ερώτησης.

Η Oracle για τη τεχνολογία Olap παρέχει έτοιμες μεθόδους συνάθροισης όπως το άθροισμα, μέσος όρος, ιεραρχικοί σταθμισμένοι μέσοι όροι και κλιμακωτά αθροίσματα. Σε αντίθεση με άλλες τεχνολογίες, η Oracle για να κάνει χρήση της τεχνολογίας OLAP υποστηρίζει διάφορες μεθόδους που εφαρμόζονται πάνω στις dimensions.

➤ Αντιστοιχεί τις σχεσιακές πηγές

Αφού έχει καθοριστεί το λογικό μοντέλο, το επόμενο βήμα είναι να φορτώσει ο χρήστης τα δεδομένα στο analytic workspace. Το Analytic Workspace Manager

προσφέρει ένα ευρύ φάσμα σχεσιακών πηγών μέσα στη βάση δεδομένων Oracle. Για την ακρίβεια μπορεί να αντιστοιχίσει οποιοδήποτε σύνολο πινάκων ή όψεων ώστε να παρέχει στις dimensions τα κατάλληλα στοιχεία. Συνήθως, τα δεδομένα αυτά είναι με τη μορφή star (άστρο) ή snowflake (χιονοστιβάδα). Το συγκεκριμένο εργαλείο δεν χρησιμοποιείται για τη διαδικασία ETL, γι' αυτό το λόγο προϋποθέτει ότι τα δεδομένα είναι αποτέλεσμα εξαγωγής και μετατροπής έτοιμα να χρησιμοποιηθούν ακριβώς όπως είναι. Αν χρειάζεται να γίνει η εκτέλεση της διαδικασίας ETL, τότε χρησιμοποιείται το εργαλείο Oracle Warehouse Builder το οποίο περιγράφεται στη παράγραφο 3.4.

- Διαχειρίζεται όλον τον κύκλο ζωής του analytic workspace

Το Analytic Workspace Manager μπορεί να ελέγχει τη φόρτωση, την ανανέωση και την συνάθροιση των δεδομένων κατά τη διάρκεια του κύκλου ζωής του analytic workspace. Οι χρήστες έχουν τη δυνατότητα να καθορίσουν αν ολόκληρα analytic workspace ή επιμέρους αντικείμενα θα διατηρηθούν κατά τον κύκλο ζωής.

Εργασίες συντήρησης μπορούν να υλοποιηθούν με σκοπό να τρέχουν άμεσα ή να αποθηκεύονται στο Oracle Job Queue (ουρά εργασιών της Oracle) ή ακόμα να σωθούν σε PL/SQL κώδικα. Συνηθίζεται να αποθηκεύονται στο Oracle Job Queue γιατί με αυτό τον τρόπο υπάρχει η δυνατότητα παράλληλης επεξεργασίας των εργασιών συντήρησης.

- Έχει τη δυνατότητα να σώζει τα μοντέλα σε μορφή xml αρχείου.

Το Analytic Workspace Manager έχει τη δυνατότητα να σώσει σε XML αρχείο το λογικό μοντέλο που δημιουργεί ο χρήστης. Μέσω αυτού οι χρήστες μπορούν να επαναχρησιμοποιούν τα ήδη σωσμένα αρχεία μοντέλων (template) και να δημιουργούν νέα analytic workspace βασισμένα σε αυτά. Τα μοντέλα δεν περιέχουν δεδομένα με αποτέλεσμα να είναι αρχεία μικρού μεγέθους και να μπορούν εύκολα να μεταφερθούν.

Συμπερασματικά, όταν ο χρήστης γνωρίζει ποιες είναι οι ανάγκες των αναφορών που θέλει να εξάγει και έχει πρόσβαση στα αντίστοιχα δεδομένα μπορεί εύκολα με τη χρήση του Analytic Workspace Manager να δημιουργήσει analytic workspace εκτελώντας έξι βασικά βήματα [28]:

- Αναγνώριση των απαιτήσεων των αναφορών
- Καθορισμός του λογικού διαστατικού μοντέλου
- Αναγνώριση των πηγών δεδομένων
- Αντιστοίχιση των λογικών αντικειμένων με τις πηγές δεδομένων
- Φόρτωση και ανανέωση των δεδομένων
- Ανάλυση των αποτελεσμάτων

Με τα παραπάνω βήματα οι χρήστες μπορούν εύκολα και γρήγορα να σχεδιάσουν τα λογικά διαστατικά μοντέλα του Data Warehouse και να τα μοιραστούν μεταξύ τους.

Στην επόμενη παράγραφο παρουσιάζεται το εργαλείο Oracle Warehouse Builder το οποίο συνεργάζεται με το Analytic Workspace Manager και χρησιμοποιείται κατά κόρον για την εκτέλεση της διαδικασίας ETL.

### **3.4. Oracle Warehouse Builder (OWB)**

Ένα από τα εργαλεία που χρησιμοποιήθηκαν για την ολοκλήρωση αυτής της εργασίας είναι το Oracle Warehouse Builder (OWB). Στη παράγραφο αυτή θα παρουσιαστούν μερικά από τα βασικά χαρακτηριστικά του.

Το OWB είναι ένα εργαλείο της Oracle το οποίο δίνει τη δυνατότητα εκτέλεσης της διαδικασίας ETL (extract, transform, load), το σχεδιασμό σχεσιακών και διαστατικών μοντέλων, τον έλεγχο ποιότητας των δεδομένων και γενικά τη πλήρη διαχείριση του κύκλου ζωής των δεδομένων και των μεταδεδομένων. Επίσης, συνεργάζεται με την βάση δεδομένων της Oracle με σκοπό να παρέχει ποιότητα στην διαδικασία ETL και την ενοποίηση των δεδομένων.

Όπως έχει αναφερθεί, η βασική χρήση του OWB είναι η διαδικασία ETL από ετερογενείς «πηγές» σε ετερογενείς «προορισμούς». Η αποθήκευση των δεδομένων μπορεί να γίνει σε σχεσιακή μορφή, σε πολυδιάστατη μορφή ή σε flat files (οριζόντια αρχεία). Οι υπεύθυνοι της ανάπτυξης του OWB μπορούν να ενσωματώσουν προηγμένες διαδικασίες όπως η ιστορικότητα των dimensions, η

επαναχρησιμοποίηση του κώδικα και ο καθορισμός προσαρμοσμένων τύπων αντικειμένων στο repository (αποθήκη).

Το OWB είναι ένα γραφικό περιβάλλον που σκοπό έχει να βοηθήσει στην υλοποίηση σύνθετων συστημάτων. Τα σχέδια που δημιουργεί ο χρήστης αποθηκεύονται ως metadata (μεταδεδομένα) σε ένα κεντρικό repository (αποθήκη). Το repository αυτό ονομάζεται Warehouse Builder Repository (αποθήκη κατασκευής Warehouse) και βρίσκεται σε βάση δεδομένων Oracle. Το Design Center (κέντρο σχεδίων) είναι η διεπαφή που παρέχει μια οπτική αντιπροσώπευση του Warehouse Repository και χρησιμοποιείται για να εισάγονται αντικείμενα από τη «πηγή» (tables, views), να σχεδιάζονται οι διαδικασίες ETL και να καθορίζεται και ο σχεδιασμός του «προορισμού» Data Warehouse.

Η αντιστοίχιση (mapping) είναι ένα αντικείμενο του OWB με το οποίο μπορείς να καθορίσεις τη ροή των δεδομένων από τη «πηγή» στον «προορισμό». Με βάση τον σχεδιασμό της αντιστοίχισης το OWB δημιουργεί αυτόματα τον κατάλληλο κώδικα για την υλοποίηση της ETL λογικής. Αφού έχει ολοκληρωθεί ο σχεδιασμός της αντιστοίχισης, σειρά έχει η ανάπτυξη (development). Η ανάπτυξη είναι η διαδικασία εκείνη με την αντιγράφονται πλέον τα σχετικά μεταδεδομένα που έχουν δημιουργηθεί και ο κώδικας που έχει παραχθεί με τη βοήθεια του Design Center στο σχήμα του προορισμού.

Το σχήμα προορισμού ορίζεται ως η βάση δεδομένων στην οποία θα εκτελεστεί η διαδικασία ETL. Συνήθως, όπως και σε αυτή την εργασία, το data warehouse με το σχήμα «προορισμού» ταυτίζονται.

Όπως έχουμε αναφέρει, το εργαλείο της Oracle OWB παρέχει μια ενδεδειγμένη λύση για την εκτέλεση της διαδικασίας ETL. Στον παρακάτω πίνακα παρουσιάζονται λειτουργίες που παρέχει σε σχέση με τη διαδικασία ETL ομαδοποιημένες με βάση την περιοχή εφαρμογής τους [27].

Περιοχή	Λειτουργία
Μοντελοποίηση Σχήματος	Μεταβαλλόμενες διαστάσεις
	Επιχειρηματική ευφυΐα (Business



	Intelligence)
Πηγή και Προορισμός	Χρησιμοποίηση XML αρχείων είτε με το ρόλο πηγή είτε με το ρόλο προορισμός
Σχεδίαση ETL	Προηγμένα χαρακτηριστικά ETL όπως μεταφερόμενες οντότητες, πολλαπλούς σχηματισμούς και αντιστοιχίες.
	Υποστήριξη RAC <sup>2</sup> . Υποστηρίζει την υλοποίηση ενός Warehouse Builder σε περιβάλλον RAC
	Χειρισμούς αντιστοίχισης
	Διαχείριση σειρά με την οποία θα φορτωθούν τα δεδομένα στους «προορισμούς». Χρησιμοποιείται ιδιαίτερα όταν υπάρχουν πολύ «προορισμοί».
	Μετασχηματισμοί.
	Ροή Διαδικασιών. Η συγκεκριμένα λειτουργία παρέχει τα παρακάτω: Φόρμες ενεργειών Υποστήριξη μεταβλητών Επαναλαμβανόμενες διαδικασίες όπως For Loop και While Loop Δραστηριότητες διευθύνσεων και ειδοποιήσεων
Διαχείριση Μεταδεδομένων	Ανάλυση καταγωγής και επίδρασης. (Lineage and impact analysis)
	Αλλαγή διάδοσης. Περιέχει αυτόματες διαδικασίες διάδοσης των αλλαγών των αντικειμένων.
	Επεκτασιμότητα. Περιέχει ορισμένα από το χρήστη αντικείμενα, σχέσεις ανάλογα με το πρόγραμμα.
Ανάπτυξη και Εκτέλεση	Προγραμματισμός. Παρέχει τη δυνατότητα ο χρήστης να προγραμματίσει να γίνουν αυτόματα η εκτέλεση των συσχετίσεων και η ροή διαδικασιών.
	Επιχειρηματική ευφυΐα και ανάπτυξη

<sup>2</sup> RAC: Real Applications Clusters

**Πίνακας 2: Oracle Warehouse Builder – Διαδικασία ETL**

Το εργαλείο της Oracle OWB εκτός από τη δυνατότητα εκτέλεσης της διαδικασίας ETL παρέχει επίσης και τη δυνατότητα εκτέλεσης διαδικασιών που αφορούν την ποιότητα των δεδομένων. Η παραπάνω επιλογή επιτρέπει τη μετατροπή των ακατέργαστων δεδομένων σε ποιοτική πληροφορία. Τόσο οι υπεύθυνοι της ανάπτυξης όσο και οι υπεύθυνοι των δεδομένων μπορούν να προσδιορίσουν εύκολα βασικά αρχικά προβλήματα των δεδομένων. Κατά συνέπεια, οι υπεύθυνοι της ανάπτυξης μπορούν να ορίσουν κανόνες και αντιστοιχίες όπου θα διορθώνουν τα δεδομένα αυτά. Με βάση τους κανόνες αυτούς, μπορούν επίσης να δημιουργήσουν και τρόπους ελέγχου των δεδομένων έτσι ώστε να εξασφαλίζεται η ποιότητα των εισερχόμενων δεδομένων κάθε φορά.

Αντίστοιχα, με το Πίνακα 2 στον επόμενο πίνακα παρουσιάζονται οι λειτουργίες που παρέχονται από το OWB για τη διασφάλιση της ποιότητας των δεδομένων [27].

Περιοχή	Λειτουργία
Περιγραφή δεδομένων	Σκιαγράφηση δεδομένων (data profiling). Περιέχει διαδικασίες για τη σκιαγράφηση των δεδομένων καθώς επίσης και μετάβαση από συγκεντρωτικά δεδομένα σε πιο αναλυτικά (drill-down).
	Κανόνες δεδομένων. Περιλαμβάνει λειτουργία για την παραγωγή κανόνων των δεδομένων και για τη σκιαγράφηση τους.
	Διόρθωση δεδομένων. Περιέχει τη παραγωγή των αντιστοιχιών που έχουν δημιουργηθεί με βάση τους κανόνες για τη διόρθωση των δεδομένων.
Σχεδίαση ETL	Ροή διαδικασιών

**Πίνακας 3: Oracle Warehouse Builder – Ποιότητα δεδομένων**

Στην επόμενη παράγραφο 3.5, θα παρουσιαστεί το εργαλείο της Oracle, Oracle Business Intelligence, το οποίο χρησιμοποιήθηκε για την παρουσίαση των δεδομένων και των αποτελεσμάτων.

### 3.5. Oracle Business Intelligence Suite

Στη παράγραφο αυτή θα γίνει λόγος για το εργαλείο Oracle Business Intelligence Suite. Θα παρουσιαστούν μερικά από τα χαρακτηριστικά του, που χρησιμοποιείται και τι αποτελέσματα προσφέρει στον τελικό χρήστη. Η Oracle παρέχει το συγκεκριμένο εργαλείο ως μια πλήρη πλατφόρμα και ένα σύνολο επιμέρους εργαλείων με σκοπό την υποστήριξη του Business Intelligence. (Επιχειρηματική ευφυΐα). Οι απαιτήσεις που καλύπτονται είναι οι παρακάτω [30]:

➤ Ενοποίηση ερώτησης και ανάλυσης

Ανεξάρτητα από τους τεχνικούς, όλοι οι χρήστες ενός οργανισμού ή μιας εταιρείας χρειάζεται να επεξεργάζονται συγκεκριμένα κομμάτια από δεδομένα τα οποία θα είναι σχετικά με τις αποφάσεις που θα θέλουν να λάβουν. Η Oracle παρέχει ειδικά εργαλεία ερώτησης και ανάλυσης το οποίο δίνει τη δυνατότητα στους χρήστες να προσαρμόζουν τα δεδομένα τους και να μπορούν να μεταβαίνουν από συγκεντρωτικά δεδομένα σε πιο αναλυτικά και το αντίστροφο. Επιπλέον, οι χρήστες μπορούν να προσαρμόζουν τις αναφορές τους ανάλογα με τις εκάστοτε απαιτήσεις.

➤ Συνεργασία

Σε έναν μεγάλο οργανισμό, έχει μεγάλη σημασία το πόσο εύκολο και εφικτό είναι να μοιράζονται οι συνεργάτες τις αναφορές. Με αυτό τον τρόπο, άνθρωποι που δουλεύουν παράλληλα μπορούν να μοιράζονται τα αποτελέσματα και τις γνώσεις τους για το ίδιο αντικείμενο. Ενώ παράλληλα, αποφεύγεται η επανάληψη προσπαθειών με αποτέλεσμα να μην υπάρχει περαιτέρω δαπάνη χρόνου και κόπου για το ίδιο πρόβλημα.

➤ Σχεδιασμός και διαχείριση του κύκλου ζωής

Με ένα γραφικό περιβάλλον, οι χρήστες μπορούν να σχεδιάσουν την αποθήκευση δεδομένων τους, να φορτώσουν τα δεδομένα και να τα διαχειριστούν σε όλο το κύκλο ζωής τους.

- Ασφάλεια, διαχειρισιμότητα και εξελξιμότητα

Όλα τα δεδομένα τα οποία χρησιμοποιούνται για business intelligence αποθηκεύονται σε βάση δεδομένων Oracle. Δεν διαμοιράζονται σε διαφορετικές βάσεις. Επιπροσθέτως, οι χρήστες χρησιμοποιούν τα ίδια εργαλεία για να εντοπίζουν τις πηγές των δεδομένων και να παρακολουθούν την απόδοση.

Η συγκεκριμένη πλατφόρμα παρέχει ανάλογα εργαλεία ανά ομάδα χρηστών. Οι ομάδες χρηστών όπου ξεχωρίζουν είναι οι παρακάτω [30]:

- Απλοί χρήστες πληροφορίας

Οι περισσότεροι χρήστες σκοπό έχουν να παρακολουθήσουν τα δεδομένα τους έτσι ώστε να λάβουν τις αντίστοιχες αποφάσεις. Την πληροφορία που τους ενδιαφέρει μπορούν να την παρακολουθήσουν με τη βοήθεια των αναφορών. Σκοπός τους είναι να μπορούν στο λιγότερο δυνατό χρόνο να καταλήξουν σε συμπεράσματα που αφορούν τη τάση και την επιχειρηματική στρατηγική. Οι συγκεκριμένοι χρήστες εκτελούν τις παρακάτω ενέργειες [31]:

- Εκτέλεση και παρακολούθηση των αναφορών που βρίσκονται στην αποθήκη (repository) του BI Publisher.
- Προγραμματισμός των αναφορών να τρέχουν σε συγκεκριμένα τμήματα ή ανθρώπους του οργανισμού.
- Δημιουργία επί τόπου ειδικής ανάλυσης των δεδομένων της αναφοράς.
- Άνοιγμα και διαχείριση των αναφορών που συνεργάζονται με το πρόγραμμα Microsoft Excel.
- Δημιουργοί αναφορών και αναλύσεων

Οι υπεύθυνοι ανάπτυξης των αναφορών και των αναλύσεων σκοπό έχουν να δημιουργεί ο καθένας τις δικές τους αναφορές βασισμένες στα αναλυτικά δεδομένα. Οι αναφορές αυτές μπορούν να δημοσιευτούν για γενική χρήση. Μια αναφορά

αποτελείται από ένα μοντέλο δεδομένων, ένα σχεδιάγραμμα και ένα σύνολο ιδιοτήτων. Συνεπώς, οι ενέργειες που εκτελούν οι συγκεκριμένοι χρήστες είναι οι παρακάτω [31]:

- Δημιουργούν το μοντέλο δεδομένων της αναφοράς συμπεριλαμβάνοντας όλες τις παραμέτρους, καθώς επίσης και ένα σύνολο έγκυρων τιμών.
- Δημιουργούν το πρότυπο σχεδιάγραμμα της αναφοράς.
- Καθορίζουν τις ιδιότητες του χρόνου εκτέλεσης της αναφοράς.
- Ορίζουν αρχεία ελέγχου για τις αναφορές εκείνες που πρόκειται να διαμοίραστούν
- Ενεργοποιούν τη δυνατότητα μετάφρασης μιας αναφοράς.
- Διαχειριστές βάσεων δεδομένων

Οι διαχειριστές των βάσεων δεδομένων πρέπει να παρέχουν μεταδεδομένα κατάλληλα για τα εργαλεία που θα χρησιμοποιήσουν οι υπόλοιπες ομάδες χρηστών. Τα εργαλεία που χρησιμοποιούν είναι Oracle Warehouse Builder και Analytic Workspace που έχουν παρουσιαστεί σε προηγούμενες παραγράφους [30].

- Υπεύθυνοι ανάπτυξης εφαρμογών

Είναι οι άνθρωποι εκείνοι οι οποίοι δημιουργούν νέες προσαρμοσμένες εφαρμογές με τη χρήση.

Η πλατφόρμα Oracle Business Intelligence αποτελείται από τα παρακάτω στοιχεία [32]:

- Oracle Business Intelligence Server

Η συγκεκριμένη πλατφόρμα είναι ένα εξυπηρετητής επιχειρηματικής ευφυΐας (BI Server) ο οποίος έχει σχεδιαστεί έτσι ώστε να είναι εξελικτικός, βελτιστοποιώντας τη συνεργασία και τις παράλληλες διεργασίες κάνοντας έτσι όλες τις εφαρμογές που αφορούν την επιχειρηματική ευφυΐα να είναι διαθέσιμες στο μεγαλύτερο δυνατό κοινό. Παρέχει συγκεντρωτική πρόσβαση των δεδομένων και των υπολογισμών τους από την οποία ο καθένας μπορεί να έχει πρόσβαση σε όποια πληροφορία θέλει με οποιαδήποτε μορφή μέσα στον οργανισμό. Ο BI Server είναι κεντρικός σε όλες τις επιχειρησιακές διαδικασίες που χρησιμοποιούν την αντίστοιχη πληροφορία όπως

dashboards (ταμπλό), ad hoc queries (επί τόπου ερωτήσεις), ευφυείς ικανότητες αλληλεπίδρασης, αναφορές που αφορούν την επιχείρηση και την παραγωγή, οικονομικής αναφορές, ανάλυση OLAP, ανάσυρση δεδομένων και άλλες δικτυακές υπηρεσίες.

Η παραπάνω πλατφόρμα υποστηρίζει την πρόσβαση, την ανάλυση και την παράδοση πληροφορίας, όλα σε ένα πλήρως ενσωματωμένο διαδικτυακό περιβάλλον. Κάθε ένα από αυτά τα χαρακτηριστικά εξυπηρετεί διαφορετικές ομάδες χρηστών μέσα στην εταιρεία οι οποίες όμως χρησιμοποιούν τα ίδια δεδομένα με διαφορετικό τρόπο. Σε αντίθεση με άλλα εργαλεία επιχειρηματικής ευφυΐας όλα τα στοιχεία είναι ενσωματωμένα σε μια κοινή αρχιτεκτονική.

#### ➤ Oracle Business Intelligence Dashboards

Η εφαρμογή Business Intelligence Dashboards – BI Dashboards (ταμπλό επιχειρηματικής ευφυΐας) παρέχει σε οποιοδήποτε εργαζόμενο μέσα στην επιχείρηση τη διαλογική πρόσβαση σε πληροφορία η οποία είναι δυναμικά προσαρμοσμένη ανάλογα με τον ρόλο του κάθε ατόμου. Στο περιβάλλον αυτό, ο τελικός χρήστης εργάζεται με τις αναφορές, τα ειδοποιητήρια (prompts), τα διαγράμματα, τους πίνακες, τους πίνακες αξόνων και τη γραφική παράσταση σε μια καθαρά Web αρχιτεκτονική. Ο χρήστης έχει τη δυνατότητα να πλοηγηθεί, να τροποποιήσει και να αλληλεπιδράσει με τα αποτελέσματα. Οι χρήστες της συγκεκριμένης εφαρμογής μπορούν επίσης να αθροίσουν τα περιεχόμενο από ένα ευρύ σύνολο πηγών συμπεριλαμβανομένου του διαδικτύου, τον κοινόχρηστων αρχείων και των αποθηκών εγγράφων (document repositories).

#### ➤ Oracle Business Intelligence Answers

Η συγκεκριμένη εφαρμογή παρέχει στους τελικούς χρήστες τη δυνατότητα εκτέλεσης επί – τόπου ενεργειών μέσα σε μια καθαρά Web αρχιτεκτονική. Οι χρήστες αλληλεπιδρούν με μια λογική άποψη της πληροφορίας, η οποία είναι εντελώς ανεξάρτητη από την πολυπλοκότητα των δομών δεδομένων, και μπορούν εύκολα να δημιουργήσουν διαγράμματα, πίνακες αξόνων και αναφορές. Τα παραπάνω μπορούν με την ίδια ευκολία να τα αποθηκεύσουν, να τα μοιραστούν, να τα τροποποιήσουν και να τα επεξεργαστούν.

➤ Oracle Business Intelligence Delivers

Η εφαρμογή Oracle BI Delivers είναι μια δυναμική λύση η οποία παρέχει τη δυνατότητα επιχειρηματικής δραστηριότητα για έλεγχο και προειδοποιήσεις. Οι προειδοποιήσεις μπορούν να φτάσουν στους χρήστες μέσω πολλών καναλιών όπως email, dashboards και κινητές συσκευές. Η Oracle BI Delivers περιλαμβάνει μια πλήρως βασισμένη σε web πύλη. Έχει τη δυνατότητα να ξεκινήσει και να περάσει τη σχετική πληροφορία σε άλλες προειδοποιήσεις με σκοπό την εκτέλεση πολλαπλών βημάτων, πολλών χρηστών, πολλών εφαρμογών αναλυτικής ροής εργασιών. Τέλος, μπορεί δυναμικά να αποφασίσει ποιοι θα είναι οι παράληπτες και να προσωποποιήσει το περιεχόμενο ώστε να φθάσει στους σωστούς χρήστες, τη σωστή στιγμή και με τη σωστή πληροφορία.

➤ Oracle Business Intelligence Disconnected Analytics

Το εργαλείο αυτό παρέχει τη δυνατότητα πλήρους αναλυτικής λειτουργίας για έναν κινητό επαγγελματία, επιτρέποντας την αλληλεπίδραση με τα dashboards και την ειδική ανάλυση ενώ ο χρήστης είναι εκτός του εσωτερικού δικτύου. Παρέχει την ίδια διεπαφή ανεξάρτητα αν ο χρήστης εργάζεται σε συνδεδεμένη ή αποσυνδεδεμένη μορφή. Τα δεδομένα είναι εξατομικευμένα για κάθε χρήστη ανάλογα με το ρόλο του διατηρώντας έτσι την ασφάλεια και τη διαφάνεια.

➤ Oracle Business Intelligence Publisher

Η εφαρμογή Oracle BI Publisher δίνει τη δυνατότητα η υποβολή εκθέσεων να είναι διαθέσιμη σε σύνθετα και διανεμημένα περιβάλλοντα. Παρέχει μια κεντρική αρχιτεκτονική για την παραγωγή και την παράδοση της πληροφορίας στους υπαλλήλους, τους πελάτες και τους συνεργάτες με ασφάλεια και με τη σωστή μορφή. Οι αναφορές του BI Publisher μπορούν να σχεδιαστούν με τη χρήση Microsoft Word ή του Adobe Acrobat, εργαλεία με τα οποία οι περισσότεροι χρήστες είναι εξοικειωμένοι. Επιπλέον, μπορεί να περιέχει πληροφορία η οποία λαμβάνεται από διαφορετικές πηγές δεδομένων σε ένα ενιαίο έγγραφο. Οι αναφορές μπορούν να παραδοθούν με ποικίλους τρόπους όπως με εκτύπωση, email, fax ή με τη δημοσίευση σε μια δικτυακή πύλη.

### 3.6. Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν τα βασικά εργαλεία της Oracle τα οποία χρησιμοποιήθηκαν για την υλοποίηση της εργασίας αυτής. Τα εργαλεία αυτά σκοπό έχουν να διευκολύνουν το χρήστη στην υλοποίηση και τη διαχείριση των διαστατικών αντικειμένων καθώς επίσης και των σχεσιακών. Επίσης, παρέχει αντίστοιχα εργαλεία με τα οποία ο χρήστης μπορεί να εκτελέσει και τη διαδικασία ETL (παράγραφος 2.4).

Αναφορικά τα εργαλεία αυτά είναι τα παρακάτω:

- Oracle Analytic Workspace Manager
- Oracle Warehouse Builder (OWB)
- Oracle Business Intelligence Suite

Το καθένα από αυτά χρησιμοποιήθηκε με διαφορετικό τρόπο. Το Oracle Analytic Workspace Manager χρησιμοποιήθηκε για τον σχεδιασμό και την υλοποίηση των διαστατικών αντικειμένων. Αντίστοιχα, το Oracle Warehouse Builder ενώ θα μπορούσε να ικανοποιήσει και την παραπάνω λειτουργία, χρησιμοποιήθηκε για την φόρτωση των σωστών δεδομένων μέσω της διαδικασίας ETL. Τέλος, το Oracle Business Intelligence Suite έδωσε τη δυνατότητα παρουσίασης των δεδομένων σε αναφορές.

Τα παραπάνω εργαλεία συνεργάζονται επιτυχώς ώστε να υλοποιηθεί στο τέλος ένα ολοκληρωμένο σύστημα Data Warehouse. Στο επόμενο κεφάλαιο, θα παρουσιαστεί η μεθοδολογία δημιουργίας ενός Data Warehouse.



## Κεφάλαιο 4

### 4. Μεθοδολογία ανάπτυξης ενός DW

#### 4.1. Εισαγωγή

Ο σκοπός δημιουργίας ενός DW, όπως έχει ήδη παρουσιαστεί στο Κεφάλαιο 2 (Συστήματα Data – Warehousing), είναι να καλύψει τις ανάγκες ενός οργανισμού που δε μπορεί να καλύψει μια απλή σχεσιακή βάση δεδομένων. Το DW απαντάει στα προβλήματα εξόρυξης δεδομένων, λήψης αποφάσεων καθώς επίσης και της ανάλυσης των δεδομένων. Στο κεφάλαιο αυτό θα παρουσιαστεί η γενική διαδικασία ανάπτυξης ενός DW και ιδιαίτερα με τη χρήση των εργαλείων της Oracle 10G. Τα βασικά βήματα που θα πρέπει να εκτελεστούν για τη δημιουργία ενός Data Warehouse παρουσιάζονται συνοπτικά παρακάτω.

- **Ανάλυση Απαιτήσεων:** είναι το πρώτο βήμα που πρέπει να εκτελεστεί πριν από όλα. Με το βήμα αυτό θα πρέπει να εντοπιστούν οι ανάγκες που θα καλύψει το Data Warehouse σύστημα.
- **Λογικός Σχεδιασμός:** στο βήμα αυτό, αφού θα έχουν εντοπιστεί οι ανάγκες, γίνεται ο λογικός σχεδιασμός του Data Warehouse. Πιο συγκεκριμένα, εντοπίζονται και σχεδιάζονται οι dimensions, οι cubes και ιεραρχίες.
- **Φυσικός Σχεδιασμός:** στη συνέχεια, σχεδιάζουμε και υλοποιούμε τη σχεσιακή βάση του Data Warehouse. Δηλαδή, tables και views τα οποία θα έχουν τη πληροφορία που επιθυμούμε.
- **Εισαγωγή και τροποποίηση των δεδομένων:** σε αυτό το βήμα εκτελούμε τη διαδικασία ETL. Εντοπίζουμε τη μορφή που θέλουμε να έχουν τα δεδομένα και με τη παραπάνω διαδικασία τα μεταφέρουμε από την «πηγή» στον προορισμό είτε απευθείας είτε μετά από επεξεργασία.
- **Ανάκτηση δεδομένων:** καθώς έχει ολοκληρωθεί η διαδικασία ETL πλέον έχουμε τη δυνατότητα να ανακτήσουμε τα δεδομένα μας. Αυτό μπορεί να γίνει είτε με τη χρήση SQL υπό-ερωτημάτων είτε με τη χρήση κατάλληλων εργαλείων που μπορούν και επεξεργάζονται cubes και εκτελούν διαδικασίες aggregation / rollup.

Στις παρακάτω παραγράφους, αναλύεται ο τρόπος με τον οποίο γίνονται τα παραπάνω βήματα. Συγκεκριμένα, θα παρουσιαστούν η εκτέλεση των παραπάνω βημάτων με τη χρήση των εργαλείων της Oracle 10G.

## 4.2. Ανάλυση Απαιτήσεων

Στη παράγραφο αυτή θα προταθεί ένας τρόπος σχετικά με το πώς πρέπει να γίνει η ανάλυση απαιτήσεων ενός Data Warehouse. Η δημιουργία ενός Data Warehouse διαφέρει από τη δημιουργία άλλων πληροφοριακών συστημάτων και γι' αυτό το λόγο χρειάζεται να ακολουθηθεί και άλλη μεθοδολογία. Μερικά χαρακτηριστικά των ιδιαιτεροτήτων των παραπάνω συστημάτων παρουσιάζονται στον παρακάτω πίνακα [25].

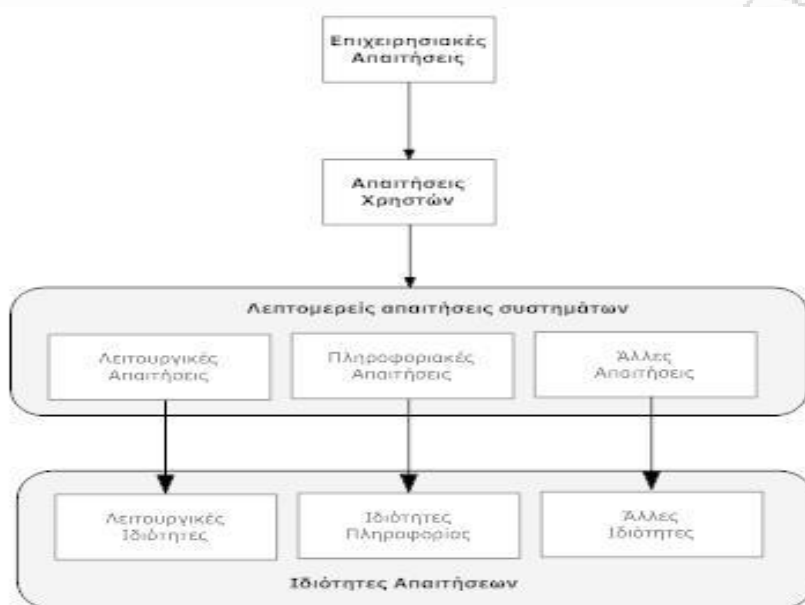
Data Warehouse	Πληροφοριακό Σύστημα
Η χρήση των δεδομένων είναι διερευνητική και λιγότερο προβλέψιμη	Οι αυτοματοποιημένες διαδικασίες επαναλαμβάνονται και είναι προβλέψιμες
Πολυδιάστατη μοντελοποίηση	Σχεσιακή μοντελοποίηση
Εστίαση στη φόρτωση και παρουσίαση των δεδομένων	Εστίαση στη γρήγορη και απευθείας ενημέρωση των δεδομένων

**Πίνακας 4: Διαφοροποίηση Data Warehouse με τα Πληροφοριακά Συστήματα**

Η ανάλυση απαιτήσεων είναι μια διαδικασία καίρια για τη σωστή λειτουργία και δημιουργία ενός Data Warehouse. Κι αυτό γιατί, ένα data warehouse μπορεί να επιδρά σε πολλές οργανωτικές μονάδες μιας επιχείρησης. Οι απαιτήσεις αυτές καθορίζουν τη λειτουργική συμπεριφορά και τις διαθέσιμες πληροφορίες (για παράδειγμα, ποια στοιχεία πρέπει να είναι προσπελάσιμα, πώς θα πρέπει να μετασχηματιστούν και να οργανωθούν, καθώς επίσης και πώς θα αθροιστούν ή θα υπολογιστούν). Οι υπεύθυνοι εκφράζουν τις προσδοκίες τους με σκοπό τη βελτίωση της επιχείρησης. Οι απόψεις αυτές είναι το θεμέλιο για την σωστή υλοποίηση του Data Warehouse.

Όμως, η ομάδα ανάπτυξης ενός Data Warehouse χρειάζεται μια πλήρη, σωστή και σαφή προδιαγραφή του συστήματος που θα πρέπει να δημιουργήσει, το οποίο σημαίνει περαιτέρω καθορισμός και έρευνα των επιχειρησιακών απαιτήσεων. Γι αυτό το λόγο, οι απαιτήσεις έχουν διαφορετικά επίπεδα αφαίρεσης. Συνοπτικά έχουμε [26]:

- Επιχειρησιακές απαιτήσεις
- Απαιτήσεις χρηστών
- Λεπτομερείς απαιτήσεις συστημάτων
- Ιδιότητες απαιτήσης



Εικόνα 1: Επίπεδα αφαίρεσης απαιτήσεων

Στη συνέχεια της παραγράφου αναφέρονται τα χαρακτηριστικά των παραπάνω απαιτήσεων.

### Επιχειρησιακές Απαιτήσεις

Οι επιχειρησιακές απαιτήσεις αποτελούν τους υψηλού επιπέδου στόχους της οργάνωσης που θα πρέπει να καλύψει το σύστημα Data Warehouse. Σε ένα έγγραφο καταγράφονται το όραμα και το πεδίο δράσης του έργου. Οι περαιτέρω υπηρεσίες του συστήματος προέρχονται από τις επιχειρησιακές απαιτήσεις, οι οποίες αναπαρίστανται σε διάγραμμα περιεχομένου. Στην ουσία προσδιορίζονται τα αρχικά οφέλη που θα παρέχει το Data Warehouse στην επιχείρηση καθώς επίσης και τους χρήστες. Οι επιχειρησιακές απαιτήσεις αποτελούν το πιο υψηλό επίπεδο αφαίρεσης στην αλυσίδα των απαιτήσεων, όπως παρουσιάζεται και στην Εικόνα 1. Εκφράζουν τις εμπορικές ευκαιρίες, τους επιχειρησιακούς στόχους και την παρεχόμενη αξία του συστήματος σε ένα υψηλό επίπεδο.

### **Απαιτήσεις Χρηστών**

Οι απαιτήσεις χρηστών περιγράφουν τις ενέργειες που θα πρέπει οι χρήστες του συστήματος να εκτελούν με τη βοήθεια του Data Warehouse. Οι απαιτήσεις χρηστών πρέπει να συλλεχθούν από τους ανθρώπους εκείνους που πραγματικά θα χρησιμοποιήσουν και θα εργαστούν με το Data Warehouse. Γι αυτό το λόγο, οι χρήστες μπορούν να περιγράψουν λεπτομερώς τόσο τις εργασίες που θα πρέπει να εκτελούν όσο και τα μη λειτουργικά χαρακτηριστικά που είναι σημαντικά ώστε να γίνει αποδεκτό το Data Warehouse. Οι απαιτήσεις χρηστών πρέπει να ευθυγραμμιστούν με το στόχους που έχουν περιγραφεί στο πρώτο επίπεδο αφαίρεσης ενώ παράλληλα να εστιάζουν και στις ανάγκες των χρηστών. Για τη κατανόηση τους περιγράφονται με σενάρια χρήσεων ή περιπτώσεων δοκιμών του Data Warehouse. Η παραπάνω προσέγγιση είναι σαφώς πιο αναλυτική και επικεντρώνεται στο πρόβλημα σε σχέση με τη παραδοσιακή προσέγγιση απλών ερωτήσεων των χρηστών για το τι θα ήθελαν να κάνει το σύστημα. Αυτό συμβαίνει γιατί με αυτό το τρόπο καθορίζονται τα είδη των χρηστών, οι στόχοι που έχουν να επιτύχουν και όλα αυτά σε συνδυασμό με τους επιχειρησιακούς κανόνες που έχουν ήδη καταγραφεί από το πιο πάνω επίπεδο.

### **Λεπτομερείς Απαιτήσεις συστήματος**

Οι λεπτομερείς απαιτήσεις αντιπροσωπεύουν τις απαιτήσεις του Data Warehouse σε ένα πολύ λεπτομερές επίπεδο. Σε αυτό το επίπεδο γίνεται μια λεπτομερής προδιαγραφή των απαιτήσεων η οποία θα χρησιμοποιηθεί από την ομάδα ανάπτυξης τους Data Warehouse. Η ομάδα θα πρέπει να λάβει υπόψη της τόσο τις απαιτήσεις των χρηστών όσο και της επιχείρησης και να δημιουργήσει κατάλληλα κριτήρια ώστε να είναι εύκολη η επαλήθευση της σωστής λειτουργία του συστήματος. Οι λεπτομερείς απαιτήσεις διαχωρίζονται σε δυο διαφορετικά είδη, τις λειτουργικές απαιτήσεις και τις απαιτήσεις πληροφορίας.

Οι λειτουργικές απαιτήσεις καθορίζουν τις λειτουργίες που θα πρέπει να αναπτύξει η ομάδα ανάπτυξης στο Data Warehouse ώστε να μπορούν να ολοκληρώνουν τους στόχους τους, με τρόπο ώστε να καλύπτονται έτσι και οι επιχειρησιακές απαιτήσεις. Στις λειτουργικές απαιτήσεις καταγράφεται η συμπεριφορά που προορίζεται να έχει

το σύστημα. Αυτή η συμπεριφορά μπορεί να αποτελείται από υπηρεσίες, στόχους ή λειτουργίες που θα πρέπει να εκτελεί το Data Warehouse.

Από την άλλη μεριά, οι απαιτήσεις πληροφορίας καθορίζουν τις ανάγκες της επιχείρησης ως προς τον τομέα της πληροφορίας. Περιγράφουν την πληροφορία και τα δεδομένα που θα πρέπει να παρουσιάζει και να παρέχει το Data Warehouse. Σε αυτό το επίπεδο, διευκρινίζονται τα ήδη υπάρχοντα δεδομένα, από πού θα προέρχονται, πώς θα επεξεργαστούν και ποια θα είναι η ποιότητα των νέων δεδομένων. Στη συνέχεια, καθορίζεται η διαδικασία ανάλυσης και ποιες μέθοδοι θα χρησιμοποιηθούν.

Άλλες απαιτήσεις εκτός από τις λειτουργικές και τις απαιτήσεις πληροφορίας μπορούν να δημιουργηθούν ώστε να περιγράψουν περαιτέρω σχετικές πτυχές του Data Warehouse. Σε αυτές τις απαιτήσεις συμπεριλαμβάνονται οι απαιτήσεις της διεπαφής ή περιβαλλοντικές (πολιτισμός, νομικές) απαιτήσεις.

### **Ιδιότητες Απαιτήσεων**

Οι ιδιότητες των απαιτήσεων αυξάνουν τη περιγραφή των λειτουργιών, της πληροφορίας ή άλλων απαιτήσεων με την λεπτομερή περιγραφή των χαρακτηριστικών στις διάφορες διαστάσεις που είναι σημαντικές είτε για τους χρήστες είτε για την ομάδα ανάπτυξης. Οι ιδιότητες απαιτήσεων είτε ιδιότητες είτε χαρακτηριστικά ποιότητας που πρέπει να έχει το σύστημα Data Warehouse. Περιλαμβάνουν πρότυπα, κανονισμούς και συνθήκες που τα δεδομένα πρέπει να ακολουθούν, περιγραφή της εξωτερικής διεπαφής, απαιτήσεις της απόδοσης του συστήματος, περιορισμούς και ποιοτικές ιδιότητες.

### **4.3. Λογικός Σχεδιασμός**

Στη παράγραφο αυτή, παρουσιάζεται η έννοια του λογικού σχεδιασμού ενός Data Warehouse. Ο λογικός σχεδιασμός είναι περισσότερος εννοιολογικός και αφηρημένος σε σχέση με τον φυσικό σχεδιασμό (παράγραφος 4.4). Στον λογικό σχεδιασμό αντικείμενο εξέτασης είναι οι λογικές σχέσεις μεταξύ των αντικειμένων ενώ στο φυσικό σχεδιασμό ο πιο αποτελεσματικός τρόπος για την καταχώρηση και την ανάρτηση των αντικειμένων όπως επίσης και τη μεταφορά. Στη πραγματικότητα, σε

αυτό το βήμα δεν εξετάζονται οι φυσικές λεπτομέρειες εκτέλεσης αλλά μας απασχολεί ο καθορισμός των ειδών της πληροφορίας που χρειαζόμαστε.

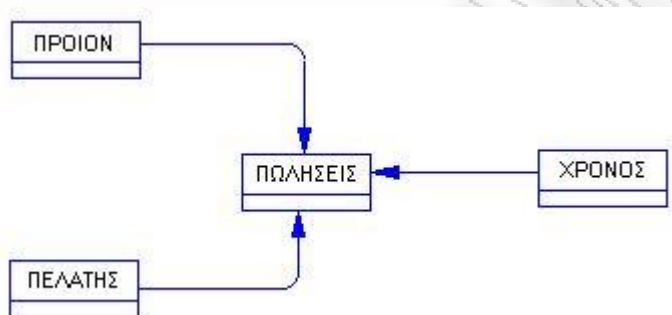
Μια τεχνική που μπορεί να χρησιμοποιηθεί για να μοντελοποιήσουμε τις λογικές απαιτήσεις της επιχείρησης είναι να χρησιμοποιηθεί το μοντέλο οντοτήτων – σχέσεων. Το μοντέλο αυτό καθαρίζει τα αντικείμενα τα οποία θεωρούνται σημαντικά για την επιχείρηση (οντότητες), τις ιδιότητες μεταξύ αυτών των αντικειμένων (ιδιότητες) και το τρόπο με τον οποίο συσχετίζονται μεταξύ τους (σχέσεις). Η συγκεκριμένη τεχνική διαδικασίας λογικού σχεδιασμού περιλαμβάνει την διευθέτηση των δεδομένων σε μια σειρά λογικών σχέσεων που αποκαλούνται οντότητες και ιδιότητες. Στις σχεσιακές βάσεις δεδομένων, μια οντότητα αντιστοιχεί συχνά σε έναν πίνακα (παράγραφο 2.5). Ενώ μια ιδιότητα είναι ένα αντικείμενο της οντότητας με σκοπό να καθορίσει τη μοναδικότητα της. Στις σχεσιακές βάσεις, η ιδιότητα αντιστοιχεί σε μια στήλη του πίνακα (παράγραφο 2.5).

Ενώ το μοντέλο οντοτήτων – σχέσεων έχει συνδεθεί με τα κανονικοποιημένα συστήματα, η τεχνική είναι χρήσιμη και για της σχεδίαση Data Warehouses υπό τη μορφή διαστατικής μοντελοποίησης. Με αυτή την τεχνική, αντί να εντοπιστούν οι μονάδες πληροφορίας (όπως οντότητες, ιδιότητες και σχέσεις), προσδιορίζουμε ποιες είναι αυτές οι πληροφορίες οι οποίες ανήκουν σε ένα κεντρικό πίνακα γεγονότος (fact table) (παράγραφο 2.6) και ποιες είναι αυτές που ανήκουν στους σχετικούς πίνακες διάστασης. Το μοντέλο αυτό θα δημιουργήσει ένα σύνολο οντοτήτων και ιδιοτήτων που αντιστοιχούν στους πίνακες γεγονότος (fact tables) και τους πίνακες διάστασης (dimension tables). Ο λογικός σχεδιασμός μπορεί να υλοποιηθεί είτε με χαρτί και μολύβι είτε χρησιμοποιώντας το εργαλείο της Oracle, Oracle Warehouse Builder το οποίο θα βοηθήσει και στη συνέχεια αφού μπορεί να υποστηρίξει τη διαδικασία ETL (παράγραφο 4.5) είτε ακόμα με κάποιο σχεδιαστικό εργαλείο μοντέλων όπως το Power Designer.

Στη συνέχεια, θα πρέπει να σχεδιαστεί το σχήμα της βάσης το οποίο θα έχει το Data Warehouse σύστημα. Το σχήμα θα αποτελείται από αντικείμενα της βάσης όπως tables (πίνακες), views (όψεις) και indexes (ευρετήρια). Τα περισσότερα συστήματα Data Warehouses χρησιμοποιούν διαστατικά μοντέλα. Υπάρχουν δυο βασικοί σχεδιασμοί όπου ονομάζονται star schema (σχήμα άστρου) και snowflake schema

(σχήμα χιονοστιβάδας). Και τα δυο σχήματα έχουν πάρει την ονομασία τους από τον τρόπο παρουσίασης τους.

Πιο συγκεκριμένα το star schema αποτελεί τη πιο απλή μορφή ενός Data Warehouse. Στο κέντρο του star βρίσκονται οι fact tables (πίνακες γεγονότων) και στις άκρες οι dimension tables (πίνακες διαστάσεων). Η σχεδίαση αυτή αφορά σχεσιακή βάση δεδομένων και αντιπροσωπεύει τα πολυδιάστατα δεδομένα. Το πλεονέκτημα της συγκεκριμένης σχεδίασης είναι ότι δίνει τη δυνατότητα να μπορείς να δεις τα αποτελέσματα αθροίσματος από ένα γενικό σύνολο σε ένα μικρότερο (slicing down), αυξάνεται η αποδοτικότητα αφού τα υπό – ερωτήματα στη βάση είναι απλής μορφής και υπάρχει καλύτερη κατανόηση των δεδομένων (Oracle, 2005). Στην Εικόνα 2, παρουσιάζεται ένα παράδειγμα Star σχεδίασης με τρεις dimensions και έναν fact table. Οι dimensions είναι το ΠΡΟΪΟΝ, ο ΠΕΛΑΤΗΣ και ο ΧΡΟΝΟΣ ενώ ο fact table αναφέρεται στις ΠΩΛΗΣΕΙΣ



Εικόνα 2: Παράδειγμα Star σχεδίασης

Από την άλλη μεριά, η σχεδίαση snowflake είναι στη πραγματικότητα μια star σχεδίαση όπου οι dimensions είναι κανονικοποιημένες σε δομή δέντρου. Οι ιεραρχίες των dimensions tables έχουν δημιουργηθεί σε πιο απλούς πίνακες σε αντίθεση με τη star σχεδίαση όπου βρίσκονται στο ίδιο row. Συνήθως, χρησιμοποιείται η star σχεδίαση κι αυτό γιατί η snowflake αυξάνει τον αριθμό των Joins (ενώσεις) μεταξύ των tables κι αυτό έχει σαν αποτέλεσμα να ελαχιστοποιείται η απόδοση κατά την ανάκτηση των δεδομένων [3].

#### 4.4. Φυσικός Σχεδιασμός

Στη παράγραφο αυτή, θα εξεταστεί ο φυσικός σχεδιασμός ενός Data Warehouse, τη σχέση έχει με τον λογικό σχεδιασμό και πως γίνεται. Ο φυσικός σχεδιασμός είναι η δημιουργία της βάσης δεδομένων με τη χρήση SQL δηλώσεων.

Κατά τη διάρκεια της φυσικής σχεδίασης, μετατρέπουμε τα δεδομένα που έχουν συλλεχθεί από τη φάση της λογικής σχεδίασης σε μια περιγραφή της φυσικής δομής της βάσης δεδομένων. Οι αποφάσεις που λαμβάνονται συνήθως κατά τη φυσική σχεδίαση επηρεάζονται κυρίως από την απόδοση ενός υπό-ερωτήματος για την ανάκτηση των δεδομένων και από τη συντήρηση της βάσης δεδομένων. Στο βήμα της λογικής σχεδίασης έχουμε ήδη σχεδιάσει ένα μοντέλο της βάσης δεδομένων το οποίο περιέχει entities (οντότητες), attributes (ιδιότητες) και relationships (σχέσεις). Οι entities συνδέονται μεταξύ τους με τη χρήση των relationships. Τα attributes χρησιμοποιούνται για τη περιγραφή των entities, ενώ το unique identifier (μοναδικό αναγνωριστικό) χρησιμοποιείται για τη διάκριση μιας πληροφορίας από μια άλλη μέσα σε μια entity. Έτσι κατά τη σχεδίαση του φυσικού σχεδιασμού αυτό που έχουμε να κάνουμε είναι να ερμηνεύσουμε το σχήμα που έχουμε δημιουργήσει σε πραγματική δομή της βάσης δεδομένων. Πιο συγκεκριμένα, τις entities σε tables (πίνακες), τις relationships σε foreign key constraints (σχέσεις σε περιορισμούς ξένων κλειδίων), τα attributes σε columns (στήλες) του table, τα primary unique identifiers (αρχικά μοναδικά αναγνωριστικά) σε περιορισμούς βασικών κλειδίων (primary key constrain) και τα unique identifiers (μοναδικά αναγνωριστικά) σε περιορισμούς μοναδικών κλειδίων (unique key constrain) [3].

Στη συνέχεια, καθώς έχουμε μετατρέψει τη λογική σχεδίαση σε φυσική, το επόμενο βήμα είναι να καθορίσουμε και να δημιουργήσουμε μερικά από τα παρακάτω αντικείμενα στη βάση δεδομένων [3].

##### ➤ Tablespaces (φυσικός χώρος)

Ένα tablespace αποτελείται από ένα ή περισσότερα datafiles (αρχεία δεδομένων), τα οποία είναι οι φυσικές δομές μέσα στο λειτουργικό σύστημα που χρησιμοποιούμε. Ένα datafile σχετίζεται μόνο με ένα tablespace. Οι κανόνες που χρησιμοποιούνται



συνήθως είναι οι tables να ανήκουν σε διαφορετικά tablespaces από τους indexes, όπως επίσης και οι μεγάλοι σε μέγεθος πίνακες να ανήκουν σε ξεχωριστό tablespace από τους μικρότερο σε μέγεθος πίνακες

➤ Tables and Partitioned Tables (πίνακες και καταναμημένοι πίνακες)

Οι tables είναι η βασική μονάδα αποθήκευσης δεδομένων. Η χρήση των partitioned tables σε σχέση με των απλών tables σκοπό έχει να δώσει τη δυνατότητα να υπάρχει υποστήριξη μεγάλου όγκου δεδομένων με τη διαίρεση αυτών σε μικρότερα και πιο εύχρηστα κομμάτια. Το βασικό κριτήριο για το αν θα χρησιμοποιηθούν απλοί tables ή materialized tables είναι η διαχείριση μέσα από την οποία παρατηρείται σαφής αποδοτικότητα καθώς μπορεί να υπάρξει ευφυή παράλληλη επεξεργασία. Έτσι, η κατανομή μεγάλων σε όγκο δεδομένων tables βελτιώνει την αποδοτικότητα αφού το κάθε καταναμημένο κομμάτι μπορεί πιο εύκολα να διαχειριστεί. Σε έναν Data Warehouse συνηθίζεται να χρησιμοποιείται κατανομή με βάση ημερομηνιών.

➤ Views (όψεις)

Τα views (όψεις) είναι στην ουσία μια παρουσίαση των δεδομένων. Τα δεδομένα αυτά μπορούν να βρίσκονται σε έναν ή περισσότερους tables ή ακόμα και σε άλλα views. Ένα view παίρνει το αποτέλεσμα ενός υπό-ερωτήματος και το διαχειρίζεται σαν να είναι table. Σε αντίθεση με έναν table, το view δε χρειάζεται κανένα φυσικό χώρο στη βάση δεδομένων.

➤ Integrity constraints (κανόνες ακεραιότητας)

Οι integrity constraints (κανόνες ακεραιότητας) χρησιμοποιούνται για να καθορίσουν τους επιχειρησιακούς κανόνες που είναι σχετικοί με τη βάση δεδομένων και για να αποτρέψουν την εισαγωγή μη έγκυρων δεδομένων στους πίνακες.

➤ Dimensions (διαστάσεις)

Μια dimension (διάσταση) είναι ένα αντικείμενο το οποίο καθορίζει τις ιεραρχικές σχέσεις μεταξύ των columns (στήλες). Μια ιεραρχική σχέση είναι μια λειτουργική εξάρτηση μεταξύ ενός επιπέδου μιας ιεραρχίας και της επόμενης. Μια dimension είναι ένα σύνολο από λογικές σχέσεις και δεν χρειάζεται κανένα φυσικό χώρο στη βάση δεδομένων.

➤ Indexes and partitioned indexes (ευρετήρια και κατανεμημένα ευρετήρια)

Οι indexes (ευρετήρια) είναι προαιρετικά και σχετίζονται με τους πίνακες. Υπάρχουν δύο είδη indexes, οι B-tree (b-δενδροειδή) indexes και οι bitmap.(δυναδικοί). Κατά τη δημιουργία ενός συστήματος Data Warehouse συνήθως χρησιμοποιούνται οι bitmap indexes γιατί είναι βελτιστοποιημένες δομές ευρετηρίων και γιατί είναι απαραίτητοι για τη βέλτιστη προσπέλαση των δεδομένων όταν γίνεται μετασχηματισμός σε star σχήμα. Οι indexes όπως και οι tables μπορούν να κατανεμηθούν. Οι κατανεμημένοι indexes διευκολύνουν την διαχείριση των δεδομένων του Data Warehouse κατά την ενημέρωση και βελτιώνουν την απόδοση των υπό-ερωτημάτων.

➤ Materialized views (υλοποιημένες όψεις)

Τα materialized views (υλοποιημένες όψεις) είναι τα αποτελέσματα ενός υπό-ερωτήματος το οποίο είναι αποθηκευμένο εκ των προτέρων ώστε να υπολογισμοί που καταλαμβάνουν μεγάλο χρονικό διάστημα να μην εκτελούνται όταν πραγματικά κάνουμε το υπό-ερώτημα. Τα materialized views μοιάζουν με τους tables ή τους κατανεμημένους tables και βελτιώνουν την απόδοση του συστήματός.

Μετά την ολοκλήρωση του φυσικού σχεδιασμού, δηλαδή την υλοποίηση της βάσης δεδομένων φυσικά. Το επόμενο βήμα είναι να μεταφέρουμε τα επιθυμητά δεδομένα μας από τη βάση δεδομένων «πηγή» στη βάση δεδομένων «προορισμός» όπου είναι αυτή που μόλις δημιουργήσαμε. Η διαδικασία αυτή περιγράφεται στη παράγραφο 4.5

#### **4.5. Διαδικασία ETL**

Στη παράγραφο αυτή θα παρουσιάσουμε το τρόπο με τον οποίο γεμίζουμε τη βάση δεδομένων του Data Warehouse. Η διαδικασία αυτή ονομάζεται ETL (Extract, Transform, Load) και έχει αρχικά περιγραφεί στην παράγραφο 2.4. Η διαδικασία αυτή δεν ορίζεται ακριβώς και μπορεί να είναι διαφορετική σε κάθε σύστημα. Το κοινό βασικό χαρακτηριστικό κάθε φορά είναι ότι τα δεδομένα μοιράζονται μεταξύ των εφαρμογών ή των συστημάτων, που προσπαθούν να τα ενσωματώσουν και σκοπό έχουν σε τουλάχιστον δυο εφαρμογές να υπάρχει η ίδια εικόνα.

### Εξαγωγή Δεδομένων

Κατά τη διαδικασία της εξαγωγής (extract), τα επιθυμητά δεδομένα αναγνωρίζονται και εξάγονται από πολλές διαφορετικές πηγές, οι οποίες μπορεί να είναι είτε άλλες βάσεις δεδομένων είτε εφαρμογές. Πολύ συχνά, δεν είναι εύκολο να αναγνωριστεί το επιθυμητό σύνολο των δεδομένων που χρειαζόμαστε. Έτσι εξάγονται περισσότερα δεδομένα με σκοπό να αναγνωριστούν τα επιθυμητά αργότερα. Ο σχεδιασμός και η υλοποίηση της εξαγωγής των δεδομένων είναι συνήθως ένας από τους πιο χρονοβόρους στόχους στη διαδικασία ETL. Αυτό μπορεί να συμβεί, είτε γιατί τα συστήματα «πηγή» να είναι πολύ σύνθετα είτε γιατί είναι κακώς τεκμηριωμένα, με αποτέλεσμα να είναι δύσκολος ο καθορισμός του στοιχείου που θα πρέπει να εξαχθεί. Τα δεδομένα εξάγονται όχι μόνο μια φορά αλλά ανά περιοδικά διαστήματα ώστε να παρέχονται κάθε φορά όλες οι αλλαγές και το Data Warehouse να είναι με ενημερωμένα δεδομένα. Ένα άλλο πρόβλημα που παρουσιάζεται είναι ότι το σύστημα «πηγή» δε μπορεί ούτε να τροποποιηθεί ούτε να αλλάξει η απόδοση και η διαθεσιμότητα του ανάλογα με τις ανάγκες ενημέρωσης του Data Warehouse.

Αφού έχουν εντοπιστεί ποια είναι τα δεδομένα που θα πρέπει να εξαχθούν, θα πρέπει να απαντηθούν τα παρακάτω δυο ερωτήματα ώστε να γίνει κι ο σωστός σχεδιασμός της διαδικασίας [3]:

➤ Ποια μέθοδο εξαγωγής θα χρησιμοποιηθεί;

Η απάντηση του παραπάνω ερωτήματος επηρεάζει το σύστημα «πηγή», τη διαδικασία μεταφοράς και το χρόνο που απαιτείται για την ενημέρωση του Data Warehouse.

➤ Πως θα παρέχονται τα εξαγόμενα δεδομένα για τη περαιτέρω επεξεργασία;

Η απάντηση του παραπάνω ερωτήματος επηρεάζει τη μέθοδο μεταφοράς και την ανάγκη καθαρισμού και επεξεργασίας των δεδομένων.

Η μέθοδος εξαγωγής των δεδομένων εξαρτάται άμεσα από το σύστημα «πηγή» και από τις επιχειρησιακές ανάγκες τους Data Warehouse περιβάλλοντος. Για την εξαγωγή, υπάρχουν λογικές μέθοδοι εξαγωγής και φυσικές μέθοδοι εξαγωγής δεδομένων. Στη περίπτωση της λογικής μέθοδος εξαγωγής δεδομένων, υπάρχουν δυο

διαφορετικοί τύποι: πλήρης εξαγωγή και αυξητική εξαγωγή [3]. Στον πρώτο τύπο, τα δεδομένα εξάγονται ολοκληρωτικά από το σύστημα «πηγή». Με αυτόν το τρόπο δεν χρειάζεται να ελέγχουμε τις αλλαγές σε σχέση με την προηγούμενη εξαγωγή. Αντίθετα, στον τύπο της αυξητικής εξαγωγής, εξάγονται μόνο τα δεδομένα που έχουν αλλάξει από τη τελευταία φορά που κάναμε εξαγωγή. Η πληροφορία αλλαγής των δεδομένων μπορεί να αποθηκεύεται είτε για κάθε row (δεδομένο) στον εκάστοτε πίνακα είτε να αποθηκεύονται συγκεντρωτικά σε έναν πίνακα αλλαγών.

Ανάλογα με ποιο τύπο λογικής μεθόδου θα χρησιμοποιήσουμε, τα εξαγόμενα δεδομένα μπορούν να εξαχθούν φυσικά με δυο διαφορετικούς μηχανισμούς. Ο πρώτος μηχανισμός είναι η εξαγωγή δεδομένων online από το σύστημα «πηγή» και ο δεύτερος από μια offline δομή. Μια τέτοια δομή μπορεί ήδη να υπάρχει ή να δημιουργείται κατά τη διαδικασία εξαγωγής [3]. Κατά την online εξαγωγή τα δεδομένα εξάγονται κατευθείαν από το σύστημα «πηγή». Η διαδικασία εξαγωγής συνδέεται άμεσα στο σύστημα «πηγής» για να προσπελάσει τους επιθυμητούς πίνακες ή αντίστοιχα στους ενδιάμεσους πίνακες που περιέχουν τα μη επεξεργασμένα δεδομένα. Αντίθετα, με την offline εξαγωγή, τα δεδομένα δεν εξάγονται κατευθείαν από το σύστημα «πηγή» αλλά δομούνται κάπου έξω από το αρχικό σύστημα.

### **Επεξεργασία Δεδομένων**

Αφού έχει ολοκληρωθεί η εξαγωγή των δεδομένων, το επόμενο βήμα που πρέπει να εκτελεστεί είναι η επεξεργασία τους. Η επεξεργασία των δεδομένων είναι η πιο χρονοβόρα διαδικασία στη συνολική διαδικασία ETL. Σχεδόν όλοι οι μετασχηματισμοί των δεδομένων μπορούν να γίνει σε μια βάση δεδομένων Oracle. Υπάρχουν δυο τεχνικές όπου ακολουθούνται για την επεξεργασία των δεδομένων [3].

- Multistage Data Transformation – MDT (Μετασχηματισμός δεδομένων σε πολλά επίπεδα)
- Pipelined Data Transformation – PDT (Μετασχηματισμός δεδομένων με χρήση αγωγού)

Στη περίπτωση της τεχνικής MDT<sup>3</sup>, η επεξεργασία και ο μετασχηματισμός των δεδομένων γίνεται σε παραπάνω από ένα βήμα. Για παράδειγμα, αν θέλουμε να εισάγουμε μια καινούρια εγγραφή σε έναν πίνακα μπορεί να υπάρξουν διαφορετικά λογικά βήματα που να ελέγχουν το κάθε κλειδί του αντίστοιχου πίνακα διάστασης. Η βάση δεδομένων Oracle υλοποιεί κάθε μετασχηματισμό σαν μια ξεχωριστή διαδικασία SQL με σκοπό να δημιουργήσει έναν προσωρινό πίνακα για να αποθηκεύει τα αποτελέσματα του κάθε βήματος. Αυτή η στρατηγική, παρέχει φυσικά σημεία ελέγχου καθ' όλη τη διαδικασία επεξεργασίας των δεδομένων, τα οποία δίνουν τη δυνατότητα να ελέγχεται και να επανεκκινείται εύκολα. Η τεχνική αυτή όμως καταλαμβάνει περισσότερο χώρο και χρόνο, αφού η επεξεργασία αποθηκεύεται σε ενδιάμεσους πίνακες και το κάθε βήμα γίνεται σειριακά κι όχι παράλληλα..

Σε αντίθεση με τη τεχνική MDT, η τεχνική PTD<sup>4</sup> καθιστά μερικά από τα προηγούμενα απαραίτητα βήματα της διαδικασίας απαρχαιωμένα ενώ μερικά αναδιαμορφώνονται για να ενισχυθεί η ροή δεδομένων και η επεξεργασία να είναι κλιμακωτή και μη διακοπτόμενη. Στην πραγματικότητα αντί να φορτώνονται τα δεδομένα και έπειτα να επεξεργάζονται (ή πρώτα να επεξεργάζονται και στη συνέχεια να φορτώνονται στη βάση δεδομένων προορισμός), η διαδικασία τα επεξεργάζεται καθώς φορτώνονται στη βάση δεδομένων. Η Oracle διαθέτει ένα σύνολο εργαλείων το οποίο καλύπτει όλα τα ζητήματα και τους στόχους που είναι σχετικοί με τη διαδικασία ETL. Η βάση δεδομένων Oracle δεν δίνει μια λύση για όλα τα σενάρια της διαδικασίας ETL. Απλά παρέχει τα εργαλεία ώστε να μπορεί να δημιουργηθεί η κατάλληλη ροή σύμφωνα με τις υφιστάμενες ανάγκες [3].

### **Φόρτωση Δεδομένων**

Αφού έχουμε ολοκληρώσει τα δυο παραπάνω βήματα (Εξαγωγή και Επεξεργασία δεδομένων), το επόμενο βήμα είναι να φορτώσουμε τα νέα μας δεδομένα στη βάση δεδομένων του Data Warehouse. Παρακάτω παρουσιάζονται οι τρόποι με τους οποίους μπορούμε να φορτώσουμε τα δεδομένα μας στη βάση δεδομένων.

➤ Φόρτωση δεδομένων με τη χρήση SQL Loader

<sup>3</sup> MDT: Multistage Data Transformation

<sup>4</sup> PDT: Pipelined Data Transformation

Ο SQL Loader χρησιμοποιείται για να φορτώσουμε δεδομένα στο σύστημα μας, με τη χρήση απλών αρχείων. Κατά τη διάρκεια φόρτωσης με χρήση `direct – path`<sup>5</sup> (άμεσο μονοπάτι), ο SQL Loader έχει τη δυνατότητα να εκτελεί και μερικούς απλούς μετασχηματισμούς των δεδομένων. Τέτοιοι μετασχηματισμοί μπορεί να είναι η μετατροπή του τύπου δεδομένων και ο χειρισμός των μηδενικών δεδομένων. Στη περίπτωση που γίνει χρήση του SQL Loader με `conventional – path`<sup>6</sup> (συμβατικό μονοπάτι), μπορεί να εφαρμοστεί στη διαδικασία περισσότερη επεξεργασία σε σχέση με την χρήση `direct – path` κι αυτό γιατί οι συναρτήσεις SQL μπορούν να χρησιμοποιηθούν στις τιμές κάθε στήλης που φορτώνονται τα δεδομένα. Η παραπάνω διαδικασία είναι σαφώς πιο αργή από τη φόρτωση δεδομένων με χρήση `direct – path` κι αυτό και χρησιμοποιείται σε μικρό πλήθος δεδομένων [3].

➤ Φόρτωση δεδομένων από εξωτερικούς πίνακες

Άλλος ένας τρόπος εισαγωγής των δεδομένων είναι με τη χρήση εξωτερικών πινάκων. Οι εξωτερικοί πίνακες της Oracle επιτρέπουν να χρησιμοποιούνται τα εξωτερικά δεδομένα ως εικονικός πίνακας. Έτσι, υπάρχει η δυνατότητα υπό-ερωτημάτων και ένωσης των δεδομένων ενώ παράλληλα να μη χρειάζεται αυτά να έχουν φορτωθεί πρώτα στη βάση δεδομένων. Με αυτή την τεχνική μπορούμε να χρησιμοποιήσουμε είτε SQL είτε PL/SQL είτε ακόμα και κώδικα JAVA για να έχουμε πρόσβαση στα εξωτερικά δεδομένα. Η βασική διαφορά των κανονικών πινάκων με τους εξωτερικούς είναι ότι οι δεύτεροι στη πραγματικότητα είναι μόνο για ανάγνωση και τα δεδομένα τους δε μπορούν να αλλαχθούν εκεί. Η τεχνική αυτή χρησιμοποιείται συνήθως όταν τα δεδομένα που θέλουμε να εισάγουμε θα πρέπει να επεξεργαστούν και να ενωθούν με τα δεδομένα που ήδη υπάρχουν στο σύστημα.

➤ Φόρτωση δεδομένων με τη χρήση Export / Import (Εξαγωγή / Εισαγωγή)

Ο τελευταίος τρόπος με τον οποίο μπορούμε να εισάγουμε τα δεδομένα μας στη βάση δεδομένων του Data Warehouse είναι να χρησιμοποιήσουμε τις εντολές της Oracle για `import` και `export`. Στη συγκεκριμένη τεχνική ενώ εισάγονται γρήγορα και εύκολα

<sup>5</sup> Η μέθοδος φόρτωσης με `direct – path` δε χρησιμοποιεί τους πόρους της βάσης δεδομένων Oracle κι αυτό το λόγο τα δεδομένα φορτώνονται με τη ταχύτητα που έχει ο σκληρός δίσκος.

<sup>6</sup> Η μέθοδος φόρτωσης με `conventional – path` εκτελεί SQL INSERT εντολές. Χρησιμοποιεί κατά κόρον τη βάση δεδομένων Oracle.

τα δεδομένα στη βάση δεδομένων, δε μπορούμε να έχουμε καμία περαιτέρω επεξεργασία αν επιθυμούμε.

Αφού έχουμε ολοκληρώσει και το βήμα φόρτωσης των δεδομένων στη βάση του Data Warehouse. Το επόμενο βήμα που έχουμε να κάνουμε είναι να ανακτήσουμε τα δεδομένα μας. Η διαδικασία αυτή περιγράφεται στη παρακάτω παράγραφο (4.6).

#### **4.6. Ανάκτηση δεδομένων**

Στη παράγραφο αυτή, θα παρουσιαστεί ο τρόπος με τον οποίο μπορούμε να κάνουμε ανάκτηση των δεδομένων από τη βάση δεδομένων του Data Warehouse που έχουμε δημιουργήσει. Η Oracle διαθέτει πολλά εργαλεία για την ανάκτηση των δεδομένων αλλά στη παρούσα εργασία θα παρουσιαστεί και θα χρησιμοποιηθεί Oracle Business Intelligence Suite(BI) για την παρουσίαση και χρήση των δεδομένων.

Η πλατφόρμα της Oracle, Oracle Business Intelligence, σκοπό έχει να παρέχει στους χρήστες του Data Warehouse ένα ενιαίο περιβάλλον υποβολής αναφορών. Στο περιβάλλον αυτό ο χρήστης μπορεί να δημιουργεί, να διαχειρίζεται τις αναφορές και τα επιχειρησιακά του έγγραφα. Χρησιμοποιώντας ένα σύνολο εργαλείων, οι χρήστες μπορούν να δημιουργήσουν και να διατηρήσουν τα σχέδια των αναφορών τους. Οι αναφορές που δημιουργούν οι χρήστες του εργαλείου μπορούν είτε να παρουσιάζονται on-line είτε να προγραμματίζονται για παράδοση σε ένα σύνολο παραληπτών (ομάδες χρηστών).

Το Oracle Business Intelligence παρέχει τη δυνατότητα να ανακτά ο χρήστης δεδομένα από διαφορετικές πηγές, να δημιουργεί templates (φόρμες) για τις αναφορές των δεδομένων και στη συνέχεια να μπορεί να τις δημοσιεύει. Η σύνδεση των χρηστών γίνεται μέσω web εφαρμογής, όπως φαίνεται και από την παρακάτω εικόνα (Εικόνα 4)



Εικόνα 3: Αρχική σελίδα σύνδεσης στη πλατφόρμα Oracle Business Intelligence

Ο κάθε χρήστης που συνδέεται έχει και τα αντίστοιχα δικαιώματα. Για παράδειγμα μπορεί να δει τις ήδη υπάρχουσες αναφορές (Εικόνα 4, Εικόνα 5) και να περιηγηθεί στα δεδομένα.



Εικόνα 4: Επιλογή Αναφοράς

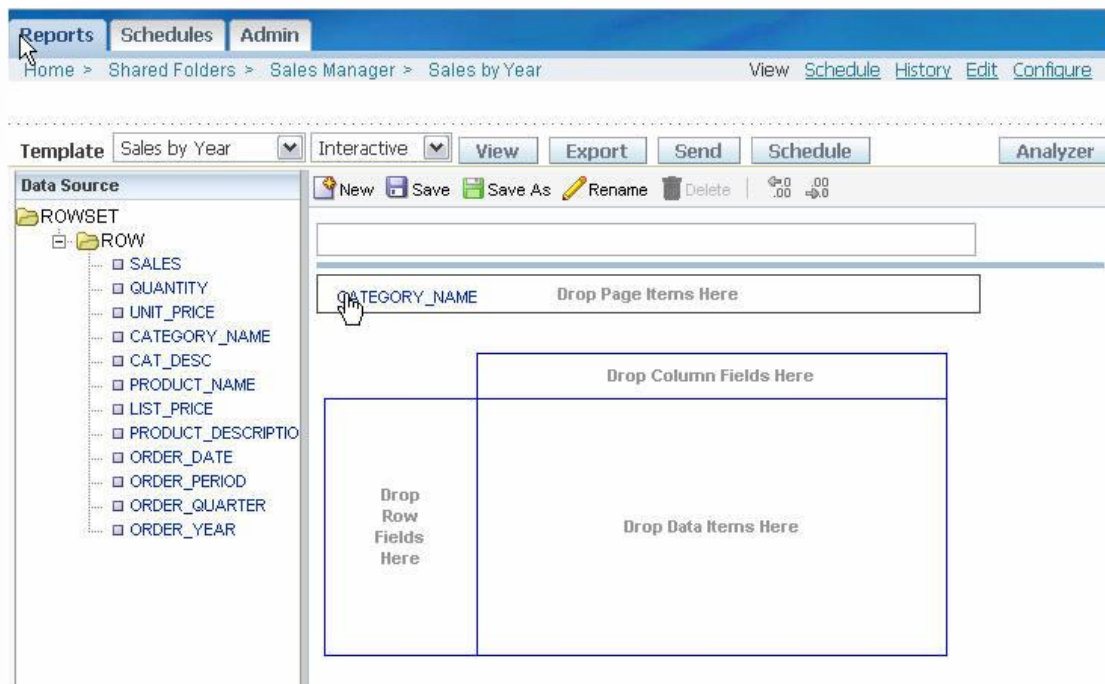


The screenshot shows the Oracle BI Publisher Enterprise interface. At the top, there is a navigation bar with 'ORACLE BI Publisher Enterprise' and user options like 'Welcome, operations', 'Preferences', 'Sign Out', and 'Help'. Below this, there are tabs for 'Reports' and 'Schedules'. The main content area shows a breadcrumb trail: 'Home > Shared Folders > Executive > Employee Salary Report'. There are filters for 'Department' (All), 'Manager' (All), and 'Customer'. Below the filters, there are buttons for 'View', 'Export', 'Send', and 'Schedule'. A 'Template' dropdown is set to 'Simple' with a format of 'html'. The report content is titled 'My Salary Report' and includes a table with the following data:

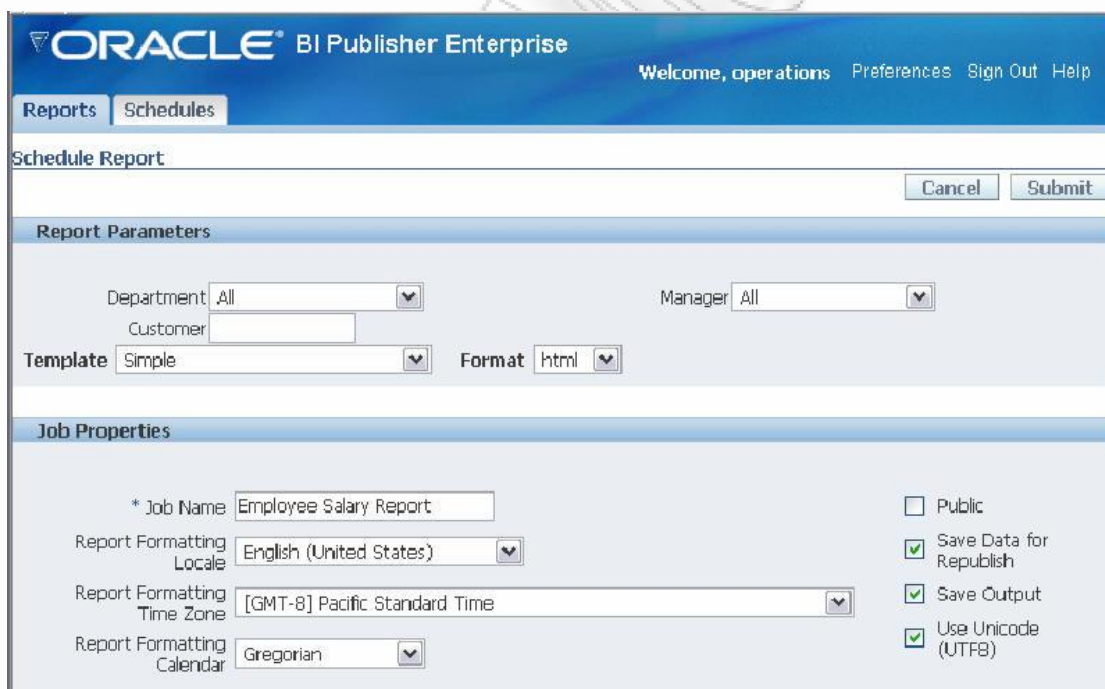
Name	Job Title	Manager	Department	Salary
Neena Kochhar	Administration Vice President	Steven King	Executive	17,000.00
Lex De Haan	Administration Vice President	Steven King	Executive	17,000.00
Alexander Hunold	Programmer	Lex De Haan	IT	9,000.00
Bruce Ernst	Programmer	Alexander Hunold	IT	6,000.00
David Austin	Programmer	Alexander Hunold	IT	4,800.00
Valli Pataballa	Programmer	Alexander Hunold	IT	4,800.00
Diana Lorentz	Programmer	Alexander Hunold	IT	4,200.00
Nancy Greenberg	Finance Manager	Neena Kochhar	Finance	12,000.00
Daniel Faviot	Accountant	Nancy Greenberg	Finance	9,000.00

Εικόνα 5: Προβολή αναφοράς

Ανάλογα με τα δικαιώματα που έχει ο χρήστης, έχει τη δυνατότητα να δημιουργήσει νέες αναφορές με σκοπό να καλύψει της επιχειρησιακές ανάγκες του οργανισμού / της εταιρεία (Εικόνα 6). Τέλος, μια επιπλέον δυνατότητα που έχει ο χρήστης είναι να προγραμματίσει μια αναφορά για δημοσίευση (Εικόνα 7, Εικόνα 8).



Εικόνα 6: Δημιουργία νέας αναφοράς



Εικόνα 7: Προγραμματισμός Αναφοράς (1/2)

Εικόνα 8: Προγραμματισμός Αναφοράς (2/2)

Οι πιο πάνω λειτουργίες είναι οι βασικές ενέργειες που μπορεί να εκτελέσει ο χρήστης όταν συνδεθεί στο σύστημα. Στο παρακάτω κεφάλαιο θα παρουσιαστεί ένα παράδειγμα χρήσης του συγκεκριμένου εργαλείου.

#### 4.7. Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε η μεθοδολογία με την οποία δημιουργείται ένα Data Warehouse. Η μεθοδολογία αυτή χρησιμοποιήθηκε και στο παράδειγμα χρήσης το οποίο έχει υλοποιηθεί (Κεφάλαιο 5). Τα βασικά βήματα τα οποία πρέπει να ακολουθηθούν για τη δημιουργία του Data Warehouse είναι τα παρακάτω:

Ανάλυση Απαιτήσεων

Λογικός Σχεδιασμός

Φυσικός Σχεδιασμός

Διαδικασία ETL

Ανάκτηση των δεδομένων.

Η ανάλυση απαιτήσεων είναι το πρώτο βήμα που πρέπει να εκτελεστεί πριν από όλα και σκοπό έχει να εντοπιστούν οι ανάγκες που θα καλύψει το Data Warehouse σύστημα. Οι απαιτήσεις μπορεί να είναι διαφορετικής υφής κάθε φορά και γι αυτό το λόγο χωρίζονται στις εξής ομάδες: Επιχειρησιακές Απαιτήσεις, Απαιτήσεις Χρηστών, Λεπτομερείς Απαιτήσεις συστήματος και Ιδιότητες Απαιτήσεων. Οι Επιχειρησιακές Απαιτήσεις αποτελούν το πιο υψηλό επίπεδο αφαίρεσης στην αλυσίδα των απαιτήσεων και στην ουσία είναι η καταγραφή του οράματος και το πεδίο δράσης του έργου. Οι Απαιτήσεις Χρηστών περιγράφουν τις ενέργειες που θα πρέπει να μπορεί να εκτελέσει ο εκάστοτε χρήστης. Οι Λεπτομερείς Απαιτήσεις συστήματος είναι η καταγραφή των κανόνων που θα πρέπει να δημιουργηθούν ώστε να καλυφθούν οι πιο πάνω απαιτήσεις. Τέλος, οι Ιδιότητες απαιτήσεων περιγράφουν είτε τις ιδιότητες είτε τα χαρακτηριστικά ποιότητας που θα πρέπει να έχει το Data Warehouse. Αποτελούνται από πρότυπα και κανονισμούς που θα πρέπει να ακολουθούν τα δεδομένα (Παράγραφος 4.2).

Ο λογικός σχεδιασμός είναι το αμέσως επόμενο βήμα που θα πρέπει να λάβει χώρα. Στον λογικό σχεδιασμό δεν εξετάζονται οι φυσικές λεπτομέρειες εκτέλεσης αλλά ο καθορισμός των ειδών της πληροφορίας που χρειαζόμαστε. Δηλαδή, γίνεται ο ορισμός του διαστατικού ή του σχεσιακού μοντέλου ανάλογα με τις ανάγκες που θα πρέπει να καλυφθούν (Παράγραφος 4.3).

Ο φυσικός σχεδιασμός είναι η δημιουργία της βάσης δεδομένων με τη χρήση SQL δηλώσεων. Σε αυτό το βήμα τα αντικείμενα που έχουν σχεδιαστεί από τον λογικό σχεδιασμό υλοποιούνται. Αυτά μπορεί να είναι είτε διαστάσεις είτε σχεσιακοί πίνακες είτε όψεις είτε επιμέρους αντικείμενα που θα αποτελούν τη βάση δεδομένων του Data Warehouse (Παράγραφος 4.4).

Καθώς έχει πλέον ολοκληρωθεί η υλοποίηση της βάσης δεδομένων του Data Warehouse το επόμενο βήμα είναι να γεμίσουν οι αντίστοιχοι πίνακες με τα επιθυμητά δεδομένα. Αυτό υλοποιείται με την εκτέλεση της διαδικασίας ETL (Παράγραφος 2.4). Η διαδικασία αυτή αποτελείται από τρία βασικά βήματα, την

εξαγωγή των δεδομένων, την επεξεργασία των δεδομένων και τελικά τη φόρτωση τους στην πηγή (Παράγραφος 4.5).

Τέλος, έχοντας εκτελέσει όλα τα παραπάνω βήματα μπορούμε με διάφορα εργαλεία και τρόπους να ανακτήσουμε τα δεδομένα μας. Στη παράγραφο 4.6, παρουσιάστηκε το Oracle Business Intelligence Suite(BI) το οποίο χρησιμοποιήθηκε στην εργασία αυτή. Με τη συγκεκριμένη πλατφόρμα της Oracle, ο χρήστης ανάλογα με τα δικαιώματα του μπορεί να δημιουργήσει, να δημοσιεύσει, να επεξεργαστεί και να προγραμματίσει για δημοσίευση αναφορές οι οποίες θα καλύπτουν τους τελικούς στόχους της εταιρείας/ του οργανισμού.

Στη συνέχεια, παρουσιάζεται ένα παράδειγμα χρήσης στο οποίο για τη δημιουργία του Data Warehouse έχει χρησιμοποιηθεί η μεθοδολογία που περιγράφεται στο κεφάλαιο αυτό όπως επίσης και τα αντίστοιχα εργαλεία.

## Κεφάλαιο 5

### 5. Παράδειγμα χρήσης

#### 5.1. Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιαστεί η διαδικασία υλοποίησης του σεναρίου. Θα παρουσιαστούν όλα τα βήματα που εκτελέστηκαν για τη δημιουργία ενός Data Warehouse και με ποιο τρόπο εκτελέστηκαν.

Πιο συγκεκριμένα στη παράγραφο 5.2, περιγράφεται το σενάριο που θέλουμε να εκτελεστεί. Περιγράφεται συνοπτικά η αρχική κατάσταση του συστήματος και στη συνέχεια ποιο είναι το επιθυμητό αποτέλεσμα.

Στη παράγραφο 5.3, παρουσιάζεται το αρχικό σύστημα. Δηλαδή, το σύστημα που θα αποτελέσει την πηγή των δεδομένων. Γίνεται η ανάλυση και η περιγραφή της αρχικής σχεσιακής βάσης για την καλύτερη κατανόηση της δομής των δεδομένων.

Στην επόμενη παράγραφο 5.4, γίνεται η ανάλυση απαιτήσεων του Data Warehouse. Η ανάλυση απαιτήσεων εστίασε στις επιχειρησιακές απαιτήσεις και στις απαιτήσεις των χρηστών (Ανάλυση Απαιτήσεων). Παρουσιάζονται ερωτήματα που πρέπει να καλυφθούν με την δημιουργία του Data Warehouse καθώς επίσης και οι απαιτήσεις των χρηστών όπου καθορίζονται ποια είναι τα είδη χρηστών και ποιες είναι οι απαιτήσεις τους όσον αφορά στην εύκολη διαχείριση και χρήση του Data Warehouse.

Στη συνέχεια του κεφαλαίου, στην παράγραφο 5.5, παρουσιάζεται ο λογικός σχεδιασμός του Data Warehouse. Με βάση την ανάλυση απαιτήσεων εντοπίζονται οι οντότητες που θα πρέπει να δημιουργηθούν και ορίζονται τα χαρακτηριστικά τους. Ορίζεται το διαστατικό μοντέλο που θα χρησιμοποιηθεί και ποιες dimensions, levels, attributes cube θα δημιουργηθούν ώστε να καλύψουν τις παραπάνω απαιτήσεις.

Ακολουθεί, στη παράγραφο 5.6, η διαδικασία υλοποίησης των παραπάνω αντικειμένων που περιγράφηκαν στην παράγραφο 5.5. Παρουσιάζονται τα SQL Scripts που εκτελέστηκαν τόσο για τη δημιουργία του χρήστη της βάσης δεδομένων όσο και ο τρόπος υλοποίησης των dimensions και του cube με τη χρήση του Olap Analytic Workspace Manager and Worksheet (παράγραφος 3.3)

Στη παράγραφο 5.7 παρουσιάζεται η διαδικασία ETL (Παράγραφος 2.4) με την οποία εισάγονται τα δεδομένα από το αρχικό σύστημα στη σχεσιακή βάση του Data Warehouse. Για την υλοποίηση, της διαδικασίας χρησιμοποιήθηκαν SQL Scripts και διαδικασίες της Oracle (Stored procedures) τα οποία παρουσιάζονται. Αναφέρεται επίσης, η δομή της κεντρικής διαδικασίας και τα βήματα που εκτελούνται.

Τέλος, στη παράγραφο 5.8, παρουσιάζεται ο τρόπος με τον οποίο μπορούν οι χρήστες να επεξεργαστούν τα δεδομένα. Με τη χρήση του Oracle Business Intelligence (παράγραφος 3.5) δημιουργούνται οι απαιτούμενες αναφορές. Παρουσιάζονται επίσης και παραδείγματα συνεργασίας του παραπάνω εργαλείου με το Microsoft Office και συγκεκριμένα με τη δημιουργία αναφορών τόσο σαν Word αρχείο όσο και σαν Excel.

## 5.2. Περιγραφή Σεναρίου

Στη παράγραφο αυτή θα παρουσιαστεί το σενάριο χρήσης. Το σενάριο αυτό αφορά μια μεγάλη εταιρεία η οποία προμηθεύει σε μικρότερες εταιρείες, στην Ελλάδα, προϊόντα που αφορούν τους ηλεκτρονικούς υπολογιστές. Τα προϊόντα αυτά μπορούν να αναφέρονται είτε σε λογισμικό, είτε σε υλικό είτε σε περιφερειακά. Τα στελέχη της εταιρείας (E) θέλουν να εξάγουν αναφορές και να απαντήσουν σε βασικά ερωτήματα με σκοπό να πάρουν αποφάσεις για την επιχειρηματική πορεία της εταιρείας.

Οι αποφάσεις που θέλει η εταιρεία να πάρει αφορούν τις πωλήσεις των προϊόντων. Η εταιρεία θέλει να ελέγξει ποια από όλα τα προϊόντα έχουν περισσότερες πωλήσεις, σε ποιες περιοχές και πότε. Επίσης, επιθυμεί να μπορεί να βγάλει συγκριτικές αναφορές προηγούμενων χρόνων ώστε να εντοπίσει τυχόν λάθη στη προώθηση των προϊόντων. Τέλος, επιπλέον ερωτήματα που πρέπει να απαντηθούν για τη καλύτερη διαχείριση της εταιρείας αφορούν το τομέα ζήτησης των προϊόντων. Η εταιρεία επιθυμεί να μπορεί εύκολα να βλέπει τις πωλήσεις ενός προϊόντων και να βγάζει υποθετικά μελλοντικά συμπεράσματα για το αν το προϊόν αυτό θα έχει εξίσου την ίδια ή παραπάνω ζήτηση από τους πελάτες της.

Η εταιρεία του παραδείγματος μας χρησιμοποιεί για τη διαχείριση των πελατών της ένα σύστημα CRM (Customer Relationship Management) το οποίο εμπεριέχει τόσο την πληροφορία που αφορά τους πελάτες όσο και τα προϊόντα τα οποία έχουν προμηθευτεί. Σε συνεργασία με το ERP (Enterprise Resource Planning) τους μπορούν να διαχειριστούν τόσο τα προϊόντα τους όσο και τις παραγγελίες τους. Για την υλοποίηση του παραδείγματος έχει γίνει η παραδοχή ότι και τα δυο συστήματα χρησιμοποιούν το ίδιο σχήμα της βάσης δεδομένων.

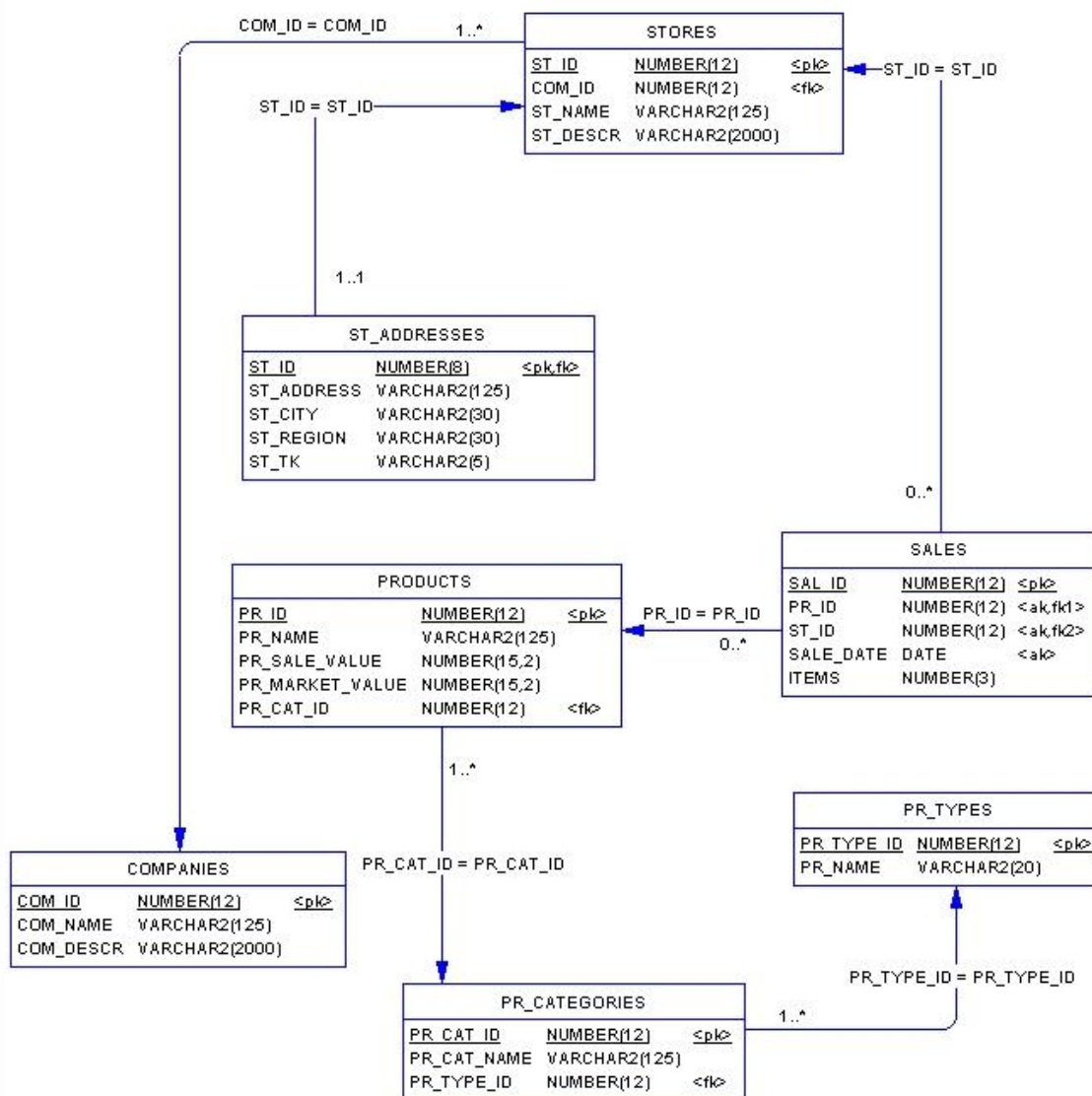
Κάποια στοιχεία θα μπορούσαν να εξαχθούν με τη χρήση υπό – ερωτημάτων στην σχεσιακή βάση δεδομένων, κάποια αλλά όμως είναι δύσκολο καθώς ο χρόνος εκτέλεσης μπορεί να είναι μεγάλος ή για να τη δημιουργία του υπό-ερωτήματος θα πρέπει ο απλός χρήστης να έχει καλές γνώσεις συστήματος βάσης δεδομένων και δημιουργίας SQL υπό – ερωτημάτων. Γι' αυτό το λόγο, η εταιρεία αποφάσισε να δώσει τη λύση του προβλήματος με τη τεχνολογία Data Warehouse και τη δημιουργία διεπαφής διάφανη προς τον τελικό χρήστη.

Στις επόμενες παραγράφους, παρουσιάζεται βήμα – βήμα ο τρόπος με τον οποίο δημιουργήθηκε το Data Warehouse και πως τελικά μπόρεσαν οι τελικοί χρήστες να το χρησιμοποιήσουν.

### **5.3. Ανάλυση αρχικής βάσης δεδομένων - ER**

Στη παράγραφο αυτή θα παρουσιαστεί αναλυτικά το σχήμα της βάσης του συστήματος CRM της εταιρείας. Το σχήμα αυτό είναι η πηγή δεδομένων της βάσης δεδομένων του Data Warehouse. Για λόγους ευκολίας περιγράφονται και αναλύονται μόνο οι πίνακες που αφορούν τη μεταφορά δεδομένων από το CRM στο Data Warehouse. Το γενικό σχήμα της βάσης παρουσιάζεται στη παρακάτω εικόνα.





Εικόνα 9: Αρχικό Σχήμα Βάσης δεδομένων

Στην Εικόνα 9 παρουσιάζονται οι πίνακες από τους οποίους θα παραχθούν τα δεδομένα. Οι πίνακες αυτοί είναι οι COMPANIES, PRODUCTS, STORES, SALES, ST\_ADDRESSES, PR\_CATEGORIES, PR\_TYPES. Ο πίνακας COMPANIES περιέχει τα βασικά στοιχεία των εταιρειών που είναι πελάτες στη δική στην εταιρία του σεναρίου. Ο πίνακας PRODUCTS περιέχει τα βασικά δεδομένα των προϊόντων που πουλάει η εταιρεία (E) στις εταιρείες πελάτες. Ο πίνακας SALES περιέχει τα στοιχεία των πωλήσεων. Ο πίνακας STORES περιέχει δεδομένα των υποκαταστημάτων των εταιρειών πελατών ενώ αντίστοιχα στο πίνακα ST\_ADDRESSES αποθηκεύονται οι διευθύνσεις των καταστημάτων αυτών. Τέλος, οι δυο πίνακες PR\_CATEGORIES και PR\_TYPES συνδέονται με τον πίνακα PRODUCTS καθώς συγκρατούν για το κάθε προϊόν σε ποια κατηγορία ανήκει και

αντίστοιχα τι τύπου είναι κάθε κατηγορία. Στο σενάριο αυτό έχουμε περιοριστεί καθαρά στην πληροφορία που μας ενδιαφέρει χωρίς αυτό να αποκλείει ότι οι πίνακες θα μπορούσαν να συγκρατούν και επιπλέον πληροφορία ανάλογα με τις ανάγκες της κάθε εταιρείας.



Εικόνα 10: Πίνακας COMPANIES

Πιο αναλυτικά, ο πίνακας COMPANIES (Εικόνα 10) αποτελείται από τα παρακάτω πεδία.

- COM\_ID: το πρωτεύον κλειδί του πίνακα.
- COM\_NAME: το όνομα της εταιρείας (πελάτης)
- COM\_DESCR: πεδίο στο οποίο μπορεί να υπάρχει περιγραφή της εταιρείας ή κάποιο σχόλιο.

Ο πίνακας STORES (Εικόνα 11) περιέχει τα υποκαταστήματα των εταιρειών και αποτελείται από τα παρακάτω πεδία

- ST\_ID: Το πρωτεύον κλειδί του πίνακα
- COM\_ID: Το ξένο κλειδί του πίνακα COMPANIES και χρησιμοποιείται για να γνωρίζουμε ποιο υποκατάστημα ανήκει σε ποια εταιρεία.
- ST\_NAME: Στο πεδίο αυτό αποθηκεύεται η επωνυμία του υποκαταστήματος
- ST\_DESCR: Το πεδίο αυτό όπως και στον πίνακα COMPANIES περιέχει κάποια περιγραφή.



Εικόνα 11: Πίνακας STORES

Ο πίνακας ο οποίος περιέχει τη πληροφορία των πωλήσεων που κάνει το κάθε κατάστημα είναι ο πίνακας SALES (Εικόνα 12). Ο πίνακας αυτός περιέχει τα παρακάτω πεδία.

SALES		
<u>SAL_ID</u>	NUMBER(12)	<pk>
PR_ID	NUMBER(12)	<ak, fk1>
ST_ID	NUMBER(12)	<ak, fk2>
SALE_DATE	DATE	<ak>
ITEMS	NUMBER(3)	

Εικόνα 12: Πίνακας SALES

- SAL\_ID: Το πρωτεύον κλειδί του πίνακα
- PR\_ID: Ξένο κλειδί από το πίνακα PRODUCTS. Αναγνωρίζουμε έτσι η πώληση για ποιο προϊόν αναφέρεται.
- ST\_ID: Ξένο κλειδί από το πίνακα STORES. Με το ξένο αυτό κλειδί γνωρίζουμε τη συγκεκριμένη πώληση ποιο υποκατάστημα την έκανε.
- SALE\_DATE: Η ημερομηνία όπου έγινε η πώληση του προϊόντος.
- ITEMS: Αριθμός ο οποίος δηλώνει πόσα τεμάχια του συγκεκριμένου προϊόντος πουλήθηκαν.

PRODUCTS		
<u>PR_ID</u>	NUMBER(12)	<pk>
PR_NAME	VARCHAR2(125)	
PR_SALE_VALUE	NUMBER(15,2)	
PR_MARKET_VALUE	NUMBER(15,2)	
PR_CAT_ID	NUMBER(12)	<fk>

Εικόνα 13: Πίνακας PRODUCTS

Στην Εικόνα 13 παρουσιάζεται ο πίνακας στον οποίο είναι αποθηκευμένα τα προϊόντα. Ο πίνακας αυτός αποτελείται από τα παρακάτω πεδία.

- PR\_ID: Το πρωτεύον κλειδί του πίνακα. Το κλειδί αυτό είναι και ξένο κλειδί στον πίνακα SALES
- PR\_NAME: Το όνομα του προϊόντος.
- PR\_SALE\_VALUE: Η τιμή πώλησης του προϊόντος στις εταιρείες.
- PR\_MARKET\_VALUE: Η τιμή αγοράς του προϊόντος από τους προμηθευτές.
- PR\_CAT\_ID: Ξένο κλειδί από τον πίνακα PR\_CATEGORIES. Το ξένο αυτό κλειδί δηλώνει σε ποια κατηγορία βρίσκεται το προϊόν.

Στον πίνακα PR\_CATEGORIES (Εικόνα 14) αποθηκεύονται οι κατηγορίες των προϊόντων. Ο πίνακας αποτελείται από τα παρακάτω πεδία.

- PR\_CAT\_ID: Το πρωτεύον κλειδί του πίνακα, το οποίο είναι και ξένο κλειδί στον πίνακα PRODUCTS.
- PR\_CAT\_NAME: Το όνομα της κατηγορίας του προϊόντος
- PR\_TYPE\_ID: Ξένο κλειδί από τον πίνακα PR\_TYPES. Το ξένο αυτό κλειδί δείχνει τι τύπος είναι η κατηγορία.

PR_CATEGORIES		
<u>PR_CAT_ID</u>	NUMBER(12)	<pk>
PR_CAT_NAME	VARCHAR2(125)	
PR_TYPE_ID	NUMBER(12)	<fk>

**Εικόνα 14: Πίνακας PR\_CATEGORIES**

Ο τελευταίος πίνακας (Εικόνα 15) που χρησιμοποιείται στο συγκεκριμένο παράδειγμα είναι ο πίνακας PR\_TYPES. Στον πίνακα αυτό αποθηκεύονται οι τύποι των κατηγοριών. Τα πεδία του πίνακα είναι τα παρακάτω.

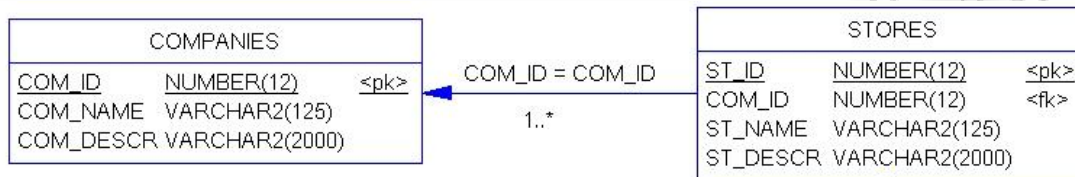
PR_TYPES		
<u>PR_TYPE_ID</u>	NUMBER(12)	<pk>
PR_NAME	VARCHAR2(20)	

**Εικόνα 15: Πίνακας PR\_TYPES**

- PR\_TYPE\_ID: Το πρωτεύον κλειδί του πίνακα, το οποίο είναι και ξένο κλειδί στον πίνακα PR\_CATEGORIES.

➤ PR\_NAME: Το όνομα του τύπου της κατηγορίας του προϊόντος.

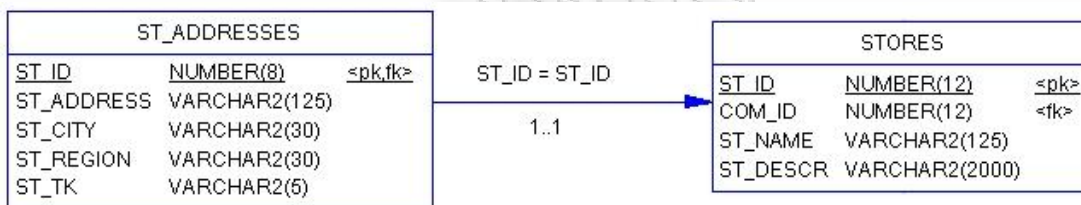
Στη συνέχεια της παραγράφου θα αναλυθούν οι σχέσεις μεταξύ των πινάκων.



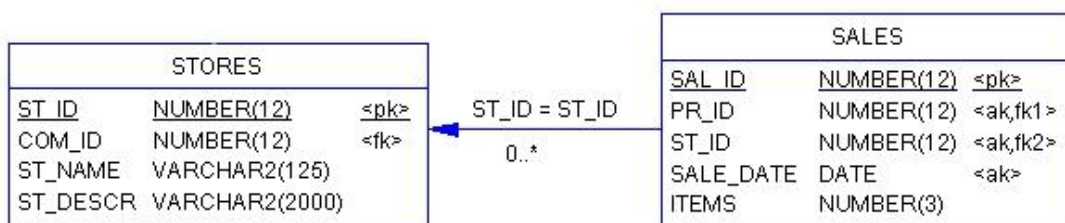
Εικόνα 16: Σχέση COMPANIES – STORES

Η σχέση μεταξύ της εταιρείας και του υποκαταστήματος (Εικόνα 16) είναι 1 προς N. Μια εταιρεία θα έχει τουλάχιστον ένα υποκατάστημα ενώ αντίστοιχα ένα υποκατάστημα ανήκει αποκλειστικά σε μια εταιρεία και μόνο. Οι δύο πίνακες συνδέονται με το κλειδί του πίνακα COMPANIES.

Ένα υποκατάστημα έχει μια μοναδική διεύθυνση και αντίστοιχα μια διεύθυνση ανήκει σε ένα μόνο κατάστημα. Η παραπάνω σχέση παρουσιάζεται στην Εικόνα 17.

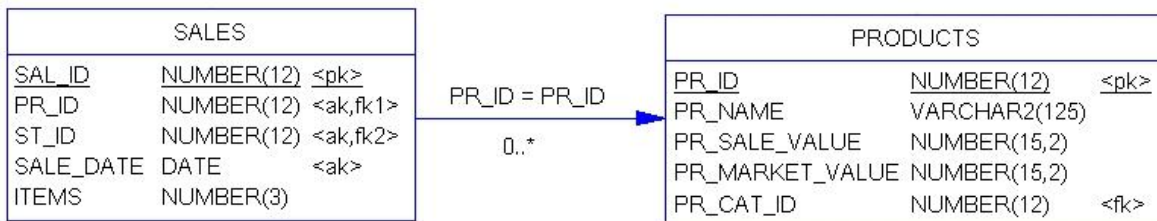


Εικόνα 17: Σχέση ST\_ADDRESSES - STORES



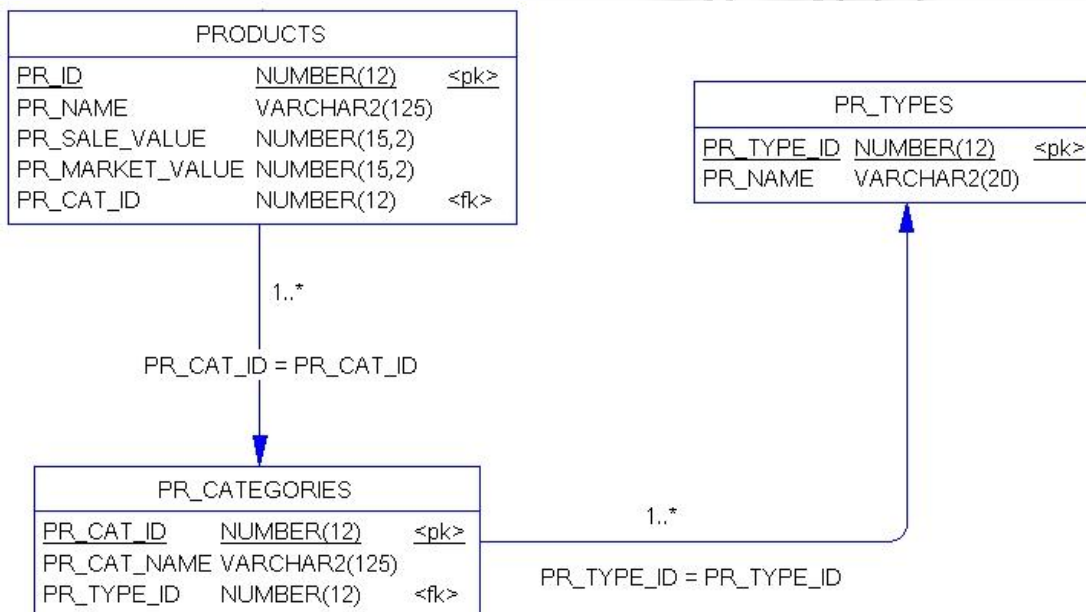
Εικόνα 18: Σχέση STORES - SALES

Στη παραπάνω εικόνα (Εικόνα 18) παρουσιάζεται η σχέση μεταξύ του πίνακα STORES και του πίνακα SALES. Ένα υποκατάστημα θα έχει τουλάχιστον μια πώληση ενώ μια πώληση αντιστοιχεί είναι μόνο για ένα υποκατάστημα. Στην Εικόνα 19 παρουσιάζεται η σχέση μεταξύ του πίνακα PRODUCTS και SALES. Ένα προϊόν μπορεί να έχει πολλές ή καμία πώληση ενώ μια πώληση θα αντιστοιχεί μοναδικά σε ένα και μόνο προϊόν.



Εικόνα 19: Σχέση PRODUCTS – SALES

Οι επόμενες δυο σχέσεις αφορούν την κατηγοριοποίηση των προϊόντων. Ένα προϊόν μπορεί ανήκει μόνο σε μια κατηγορία ενώ μια κατηγορία μπορεί να έχει ένα ή πολλά προϊόντα. Αντίστοιχα, μια κατηγορία προϊόντος είναι ενός συγκεκριμένου τύπου, ενώ σε έναν τύπο μπορεί να ανήκουν πολλές κατηγορίες.



Εικόνα 20: Σχέσεις PRODUCTS - PR\_CATEGORIES & PR\_CATEGORIES - PR\_TYPES

Στην επόμενη παράγραφο παρουσιάζονται τα βήματα, ο τρόπος και η σχεδίαση της βάσης δεδομένων του Data Warehouse.

#### 5.4. Ανάλυση Απαιτήσεων

Όπως, παρουσιάστηκε στο Κεφάλαιο 4, το πρώτο βήμα που θα πρέπει να γίνει πριν ξεκινήσει η υλοποίηση του Data Warehouse είναι η ανάλυση των απαιτήσεων που καλείται να καλύψει. Στο κεφάλαιο αυτό, θα παρουσιαστούν οι απαιτήσεις που θέλουμε να καλύψει το Data Warehouse στο παράδειγμα χρήσης.

Αρχικά, θα αναλυθούν οι επιχειρησιακές απαιτήσεις. Ο τομέας των επιχειρησιακών απαιτήσεων θα καλύψει τους επιχειρησιακούς στόχους, το όραμα, το σκοπό ορισμού και το επιχειρησιακό πλαίσιο της εταιρείας. Με βάση τα παραπάνω πεδία η εταιρεία κατέληξε στις παρακάτω επιχειρησιακές απαιτήσεις που θα πρέπει να καλύψει η δημιουργία του Data Warehouse.

- Να αυξηθεί η παραγωγικότητα της εταιρείας
- Να εντοπιστούν προϊόντα τα οποία δεν έχουν αύξηση πωλήσεων κατά τη διάρκεια ενός χρονικού διαστήματος.
- Να μπορούν να εντοπίσουν οι περιοχές εκείνες στις οποίες οι πωλήσεις είναι μειωμένες ώστε να αλλάξουν το τρόπο προώθησης των προϊόντων.

Μερικά παραδείγματα ερωτήσεων που θέλει η εταιρεία να μπορεί να απάντησει εύκολα και γρήγορα είναι της παρακάτω μορφής.

- Για τα οικονομικά έτη 2003 και 2004, να παρουσιαστεί η αλλαγή επί τοις εκατό στις πωλήσεις για τα πρώτα 10 προϊόντα για κάθε έναν από τους πρώτους πελάτες βασισμένους στις αγορές των προϊόντων.
- Σύγκριση των πωλήσεων στη Θεσσαλονίκη αυτού του τριμήνου σε σχέση με ένα έτος πριν.
- Τι μπορούμε να προβλέψουμε για τις πωλήσεις στο επόμενο τρίμηνο για μια συγκεκριμένη περιοχή;

Γενικά, η εταιρεία χρειάζεται να μπορεί να εξάγει εύκολα και γρήγορα συμπεράσματα για τις πωλήσεις των προϊόντων της, το κέρδος της και τις αποφάσεις που θα πρέπει να λάβει για την καλύτερη επιχειρησιακή εξέλιξη. Όλα αυτά τα χρειάζεται με την παρουσίαση αναφορών, οι οποίες θα μπορούν εύκολα να δημιουργηθούν από τους χρήστες του Data Warehouse και να είναι εύκολη η πρόσβαση σε αυτές τις αναφορές για κάθε ενδιαφερόμενο χρήστη.

Σε αυτό το σημείο, αφού πλέον έχουν καθοριστεί οι επιχειρησιακές απαιτήσεις, θα πρέπει να παρουσιαστούν οι απαιτήσεις των χρηστών. Οι χρήστες καθορίζουν και καταγράφουν τις εργασίες που θα πρέπει να εκτελούν. Έτσι, για τη καλύτερη κατανόηση δημιουργούνται σενάρια χρήσης και οριστικοποιούνται τα είδη χρηστών.

Καθώς έχει ολοκληρωθεί η αναζήτηση της πληροφορίας και σε συνδυασμό πάντα με τις επιχειρησιακές απαιτήσεις, καταλήγουμε στις παρακάτω απαιτήσεις χρηστών.

- Χρειάζεται ένα εύκολο και ευχάριστο περιβάλλον λειτουργίας.
- Δημιουργούνται οι παρακάτω ομάδες χρηστών.
- Απλοί θεατές των αναφορών
- Δημιουργοί αναφορών
- Διαχειριστές της εφαρμογής

Στη συνέχεια, συνδυάζοντας τη παραπάνω πληροφορία δημιουργούνται πιο λεπτομερείς απαιτήσεις. Οι απαιτήσεις αυτές έγκειται στη βασική λειτουργία του Data Warehouse καθώς επίσης και στη πληροφορία που θα πρέπει να παρουσιάζει αυτό. Οι λειτουργικές απαιτήσεις σκοπό έχουν να καλύψουν τις επιχειρησιακές απαιτήσεις της εταιρείας. Συνεπώς, οι λειτουργίες που θα πρέπει να υποστηρίζει το Data Warehouse είναι οι παρακάτω:

- Δυνατότητα εύκολης δημιουργίας αναφορών
- Δυνατότητα διαμερισμού των αναφορών στο πλήθος των χρηστών
- Διασφάλιση της ασφάλειας των δεδομένων
- Δυνατότητα επεξεργασίας των αναφορών

Όσον αφορά στο περιεχόμενο της πληροφορίας, αυτό καθορίζεται τόσο από τις επιχειρησιακές απαιτήσεις όσο και από τις απαιτήσεις χρηστών. Η πληροφορία που θα πρέπει να παρουσιάζεται από το χρήστη θα αφορά τους πελάτες (τα υποκαταστήματα), τις περιοχές (η έδρα του υποκαταστήματος), τον χρόνο και τις πωλήσεις. Συνεπώς, όλη η παραπάνω πληροφορία θα πρέπει να αποθηκεύεται στη βάση δεδομένων του Data Warehouse και να είναι προσβάσιμη από τους δημιουργούς των αναφορών. Επιπλέον, θα πρέπει να υπάρχει η δυνατότητα οι χρήστες αυτοί να χρησιμοποιούν έτοιμες διαδικασίες υπολογισμού ποσοστών, επί τοις εκατό και αθροίσματα για να μπορούν να τα χρησιμοποιήσουν στις αναφορές.



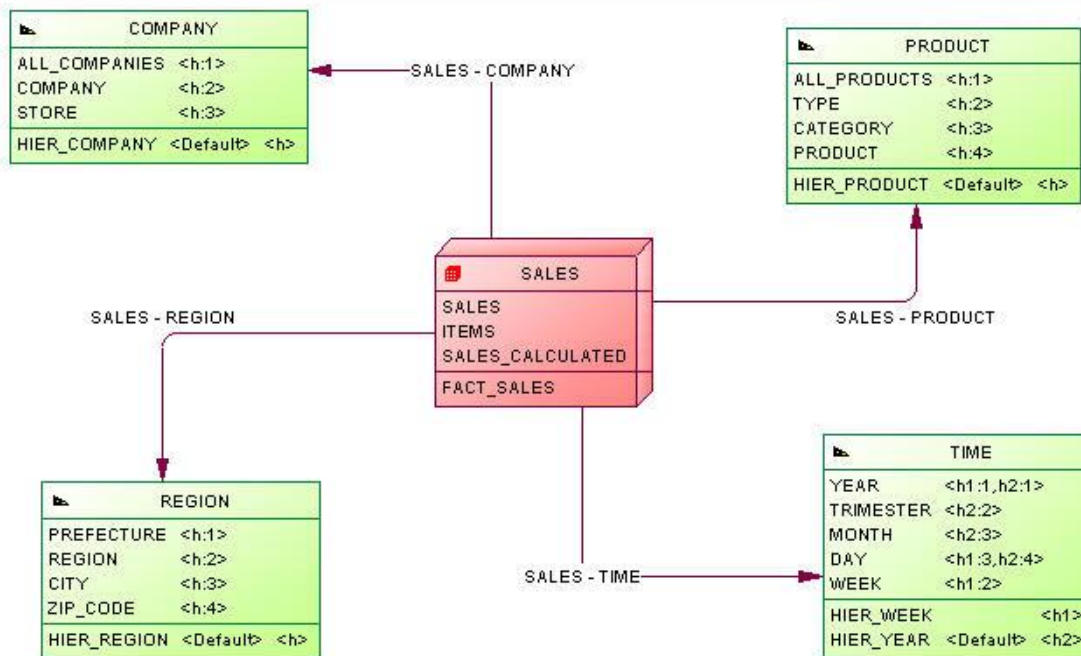
Στην επόμενη παράγραφο, θα παρουσιαστεί η λογική και φυσική σχεδίαση του Data Warehouse η οποία προκύπτει από τις απαιτήσεις συστήματος που παρουσιάστηκαν σε αυτή τη παράγραφο.

## 5.5. Λογικός Σχεδιασμός

Στο παράγραφο αυτή θα παρουσιαστεί η λογική σχεδίαση του Data Warehouse. Όλη η διαδικασία και τα αντικείμενα που θα δημιουργηθούν στηρίζονται στην ανάλυση απαιτήσεων που παρουσιάστηκε στη προηγούμενη παράγραφο 5.4. Με βάση λοιπόν τα παραπάνω στοιχεία που έχουμε συλλέξει μπορούμε να ορίσουμε τις επιμέρους εννοιολογικές οντότητες που θα ικανοποιούν τις ανάγκες μας:

- Χρόνος
- Εταιρεία
- Περιοχή
- Προϊόν
- Πωλήσεις

Αφού έχουν εντοπιστεί οι επιχειρησιακές ανάγκες στη συνέχεια θα σχεδιαστεί το λογικό διαστατικό μοντέλο του DW το οποίο θα αποτελείται από dimensions, levels, hierarchies, attributes, measures και calculated measures. Για τη σχεδίαση της βάσης δεδομένων θα χρησιμοποιηθεί η τεχνολογία MOLAP (Multidimensional OLAP) η οποία καλύπτει τις ανάγκες του σεναρίου. Οι οντότητες Χρόνος, Εταιρεία, Περιοχή και Προϊόν θα αποτυπωθούν σαν dimensions στο μοντέλο ενώ οι Πωλήσεις σαν Cube. Ο λόγος που θα χρησιμοποιηθεί η συγκεκριμένη αντιστοιχία είναι αποτέλεσμα της ανάλυσης απαιτήσεων αφού η πληροφορία των πωλήσεων θα πρέπει να μπορεί να αναλυθεί στις έννοιες που αναφέρθηκαν παραπάνω.



Εικόνα 21: Σχεδίαση Star

Η μοντελοποίηση της βάσης θα έχει τη σχεδίαση “star”. Ο αρχικός σχεδιασμός της βάσης δεδομένων μπορεί να υλοποιηθεί είτε σε χαρτί είτε σε κάποιο ειδικό πρόγραμμα του εμπορίου. Στην Εικόνα 21, παρουσιάζεται η μοντελοποίηση η σχεδίαση “star” η οποία έγινε με το εργαλείο Power Designer της Sybaseδεδομένων.

Όπως παρουσιάζεται στην Εικόνα 21, οι dimensions που δημιουργούνται είναι COMPANY, PRODUCT, REGION, TIME και ο κύβος SALES. Αντίστοιχα, για κάθε dimension έχουν δημιουργηθεί τα εξής levels στα οποία θέλουμε να υπολογίζεται το άθροισμα των πωλήσεων. Για τη dimension COMPANY έχουν δημιουργηθεί τα levels ALL\_COMPANIES, COMPANY, STORE όπου αναφέρονται αντίστοιχα στο σύνολο των εταιρειών, στη κάθε εταιρεία και στα υποκαταστήματα της κάθε εταιρείας.

Η dimension PRODUCT έχει τα levels ALL\_PRODUCTS, TYPE, CATEGORY, PRODUCT όπου αναφέρονται στο σύνολο των προϊόντων, στο τύπο κάθε προϊόντος, στην κατηγορία του κάθε προϊόντος και τέλος στο επιμέρους προϊόν αντίστοιχα. Η dimension REGION αποτελείται από τα levels PREFECTURE, REGION, CITY και ZIP\_CODE όπου αναφέρονται στους νομούς, στις περιοχές, στις πόλεις και για κάθε ταχυδρομικό κώδικα. Η dimension TIME έχει τα levels YEAR, SEASON,

TRIMESTER, MONTH όπου αναφέρονται στον χρόνο, σε εποχή, στο τρίμηνο και στον μήνα αντίστοιχα.

Για την κάθε dimension έχουν δημιουργηθεί αντίστοιχες ιεραρχίες με βάση τις οποίες θα υπολογίζονται τα αθροίσματα. Η dimension COMPANY έχει την ιεραρχία HIER\_COMPANY στην οποία συμμετέχουν τα levels ALL\_COMPANIES, COMPANY, STORE με σειρά από το πιο μεγάλο σύνολο ως το πιο μικρό. Αντίστοιχα, η dimension PRODUCT έχει την ιεραρχία HIER\_PRODUCT στην οποία συμμετέχουν τα levels ALL\_PRODUCTS, TYPE, CATEGORY, PRODUCT ενώ στη διάσταση REGION υπάρχει η ιεραρχία HIER\_REGION στην οποία συμμετέχουν τα levels PREFECTURE, REGION, CITY και ZIP\_CODE. Ιδιαίτερο ενδιαφέρον έχει η dimension TIME όπου με βάση την ανάλυση απαιτήσεων έχουν δημιουργηθεί δυο διαφορετικές ιεραρχίες. Η πρώτη είναι η HIER\_YEAR αναφέρεται σε έναν ημερολογιακό καταμερισμό και αποτελείται από τα levels YEAR, TRIMESTER, MONTH. Ενώ η δεύτερη, η HIER\_SEASON δημιουργήθηκε για να καλύψει συγκεκριμένες απαιτήσεις χρηστών και αποτελείται από τα levels YEAR, SEASON, MONTH όπου στο level SEASON καταμερίζονται οι τιμές σε διάφορες χρονικές περιόδους όπου ορίζονται από τους χρήστες. Στη συγκεκριμένη περίπτωση αναφέρεται στη περίοδο των Χριστουγέννων (Ιανουάριος, Φεβρουάριος, Δεκέμβριος), στη περίοδο του Πάσχα (Μάρτιος, Απρίλιος, Μάιος), στη καλοκαιρινή περίοδο (Ιούνιος, Ιούλιος, Αύγουστος, Σεπτέμβριος) και στην ομαδοποίηση των υπόλοιπων μηνών (Οκτώβριος, Νοέμβριος).

Τέλος ο cube SALES συνδέεται με τις παραπάνω dimensions ώστε να δημιουργούνται σωστά τα επιθυμητά αθροίσματα. Για τον cube Sales, δημιουργήθηκαν δυο βασικά measures τα οποία αναφέρονται στα αθροίσματα των πωλήσεων και των τεμαχίων που πουλήθηκαν και ένα calculated measure το οποίο θα υπολογίζει τη ποσοστιαία διαφορά από μια προγενέστερη περίοδο.

Στον παρακάτω πίνακα παρουσιάζονται συγκεντρωτικά οι Dimensions, τα Levels, οι Hierarchies, τα Attributes και τα στοιχεία του cube που δημιουργήθηκαν κατά τον λογικό σχεδιασμό.

Dimensions	Levels	Attributes	Hierarchies
COMPANY	ALL_COMPANIES COMPANY STORE	LONG_DESCRIPTION SHORT_DESCRIPTION	HIER_COMPANY
PRODUCT	ALL_PRODUCTS TYPE CATEGORY PRODUCT	LONG_DESCRIPTION SHORT_DESCRIPTION	HIER_PRODUCT
REGION	PREFECTURE REGION CITY ZIP_CODE	LONG_DESCRIPTION SHORT_DESCRIPTION	HIER_REGION
TIME	YEAR TRIMESTER MONTH WEEK DAY	LONG_DESCRIPTION SHORT_DESCRIPTION TIME_SPAN END_DATE	HIER_YEAR HIER_WEEK
<b>Cube</b>			
Measures	SALES ITEMS		
Calculated Measures	SUM_SALES_CALCULATED		

Πίνακας 5: Συγκεντρωτικός πίνακας μοντελοποίησης

Στην επόμενη παράγραφο, θα παρουσιαστεί η υλοποίηση της παραπάνω σχεδίασης και τα βήματα τα οποία εκτελέστηκαν για τη δημιουργία της. Για τον φυσικό σχεδιασμό της παραπάνω σχεδίασης θα χρησιμοποιήσουμε το εργαλείο της Oracle, Oracle Warehouse Builder με το οποίο θα εκτελέσουμε και τη διαδικασία ETL.

## 5.6. Φυσικός Σχεδιασμός

Στη παράγραφο αυτή θα παρουσιαστεί ο τρόπος υλοποίησης του λογικού σχεδιασμού που παρουσιάστηκε στην παράγραφο 5.5. Δηλαδή, θα περιγραφεί ο φυσικός σχεδιασμός. Για τον φυσικό σχεδιασμό θα χρησιμοποιηθεί το εργαλείο της Oracle,

Olap Analytic Workspace and Manager σε συνδυασμό με την εκτέλεση SQL Scripts (παράγραφο 3.3). Πριν γίνει όμως χρήση του παραπάνω εργαλείου θα πρέπει να γίνουν κάποια βασικά βήματα που αφορούν τη δημιουργία του σχήματος της σχεσιακής βάσης.

Το πρώτο βήμα που θα πρέπει να εκτελεστεί είναι η δημιουργία του σχήματος και των αντίστοιχων tablespaces που θα χρησιμοποιηθούν για τη σχεσιακή βάση του Data Warehouse. Η δημιουργία των tablespaces γίνεται με τη χρήση των παρακάτω SQL Scripts.

```
CREATE SMALLFILE TABLESPACE "TS_SCENARIO_OLAP" DATAFILE
'F:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\ts_scenario_olap.dbf' SIZE 200704K REUSE
AUTOEXTEND ON NEXT 32768K MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT
LOCAL SEGMENT SPACE MANAGEMENT AUTO

CREATE SMALLFILE TEMPORARY TABLESPACE "TS_SCENARIO_OLAP_TMP" TEMPFILE
'F:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\ts_scenario_olap_tmp.dbf' SIZE 102400K
AUTOEXTEND ON NEXT 32K MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
UNIFORM SIZE 1M

CREATE SMALLFILE UNDO TABLESPACE "TS_SCENARIO_OLAP_UNDO" DATAFILE
'F:\ORACLE\PRODUCT\10.2.0\ORADATA\ORCL\ts_scenario_olap_undo.dbf' SIZE 102400K
AUTOEXTEND ON NEXT 32768K MAXSIZE UNLIMITED RETENTION NOGUARANTEE
```

Στη συνέχεια, αυτό που θα πρέπει να κάνουμε είναι να δημιουργήσουμε τον χρήστη της σχεσιακής βάσης. Τα παρακάτω SQL Scripts δημιουργούν τον χρήστη SCENARIO\_OLAP που θα χρησιμοποιηθεί για τη σχεσιακή βάση δεδομένων του Data Warehouse. Στον χρήστη αυτό δόθηκαν τα κατάλληλα δικαιώματα ώστε να μπορεί να έχει ολοκληρωτική πρόσβαση στο σχήμα, όπως και κάποια επιπλέον δικαιώματα τα οποία είχαν σχέση με στατιστικά και χρόνους.

```
CREATE USER "SCENARIO_OLAP" PROFILE "DEFAULT" IDENTIFIED BY "*****"
DEFAULT TABLESPACE "TS_SCENARIO_OLAP" TEMPORARY TABLESPACE
"TS_SCENARIO_OLAP_TMP" ACCOUNT UNLOCK

GRANT "CONNECT" TO "SCENARIO_OLAP"

/** System privileges **/

GRANT COMMENT ANY TABLE TO "SCENARIO_OLAP"
GRANT CREATE ANY DIMENSION TO "SCENARIO_OLAP"
GRANT CREATE ANY INDEX TO "SCENARIO_OLAP"
GRANT CREATE ANY PROCEDURE TO "SCENARIO_OLAP"
GRANT CREATE ANY SEQUENCE TO "SCENARIO_OLAP"
GRANT CREATE ANY TABLE TO "SCENARIO_OLAP"
GRANT CREATE ANY VIEW TO "SCENARIO_OLAP"
GRANT CREATE TABLE TO "SCENARIO_OLAP"
```

```
GRANT ALTER TABLESPACE TO "SCENARIO_OLAP"
GRANT ALTER ANY PROCEDURE TO "SCENARIO_OLAP"
GRANT CREATE PROCEDURE TO "SCENARIO_OLAP"
GRANT DROP ANY PROCEDURE TO "SCENARIO_OLAP"
GRANT EXECUTE ANY PROCEDURE TO "SCENARIO_OLAP"
GRANT DEBUG ANY PROCEDURE TO "SCENARIO_OLAP"
GRANT DEBUG CONNECT SESSION TO "SCENARIO_OLAP"
/** QUOTAS **/
ALTER USER "SCENARIO_OLAP" QUOTA UNLIMITED ON "TS_SCENARIO_OLAP"
/* Object privileges */
GRANT SELECT ON "SYS"."V_$SESSION" TO "SCENARIO_OLAP"
GRANT SELECT ON "SYS"."V_$SESSTAT" TO "SCENARIO_OLAP"
GRANT SELECT ON "SYS"."V_$STATNAME" TO "SCENARIO_OLAP"
```

Πλέον έχουμε δημιουργήσει το σχήμα και το χρήστη και μπορούμε να εκτελέσουμε την δημιουργία των οντοτήτων που έχουν περιγραφεί κατά τον λογικό σχεδιασμό. Η υλοποίηση των διαστατικών οντοτήτων θα γίνει με τη χρήση του Olap Analytic Workspace and Manager ενώ η υλοποίηση των σχεσιακών οντοτήτων θα γίνει με την εκτέλεση SQL Queries.

Αρχικά, δημιουργούμε ένα workspace (χώρος εργασίας, Εικόνα 22). Σε αυτό το workspace θα εκτελεστούν οι εργασίες. Αφού έχει δημιουργηθεί το workspace πλέον μπορούμε να δημιουργηθούν οι dimensions που περιγράφηκαν στην προηγούμενη παράγραφο και η μοντελοποίηση που φαίνεται στην Εικόνα 21. Θα πρέπει να σημειωθεί ότι η υλοποίηση των dimensions δεν δημιουργεί αυτόματα και τους αντίστοιχους πίνακες, γι αυτό και η δημιουργία τους εκτελείται σε ξεχωριστό βήμα.

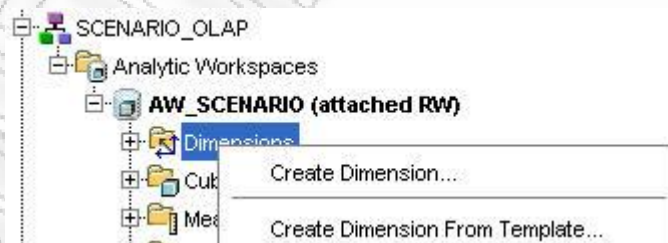
Αφού συνδεθούμε στο Analytic Workspace, στο «δέντρο» που εμφανίζεται κάνουμε δεξί κλικ στην επιλογή Dimensions και επιλέγουμε «Create Dimension» (Εικόνα 23). Με την επιλογή αυτή εμφανίζεται ένα νέο παράθυρο στο οποίο αναθέτουμε κάποια βασικά χαρακτηριστικά της dimension όπως για παράδειγμα Name (όνομα), Short Label (μικρή ετικέτα) Ιδιαίτερη σημασία έχει η επιλογή Dimension Type (Εικόνα 24). Υπάρχουν δυο διαφορετικά είδη dimension: το πρώτο είναι το User Dimension και το δεύτερο το Time Dimension. Οι dimension COMPANY, PRODUCT, REGION θα δημιουργηθούν και θα είναι τύπου User Dimension ενώ η dimension Time θα είναι τύπου Time Dimension. Αυτό συμβαίνει γιατί με βάση το σενάριο επιθυμούμε

ανάλυση των δεδομένων με βάση το χρόνο. Αφού συμπληρώσουμε κατάλληλα όλα τα απαιτούμενα πεδία επιλέγουμε Create ώστε να δημιουργηθεί η dimension.

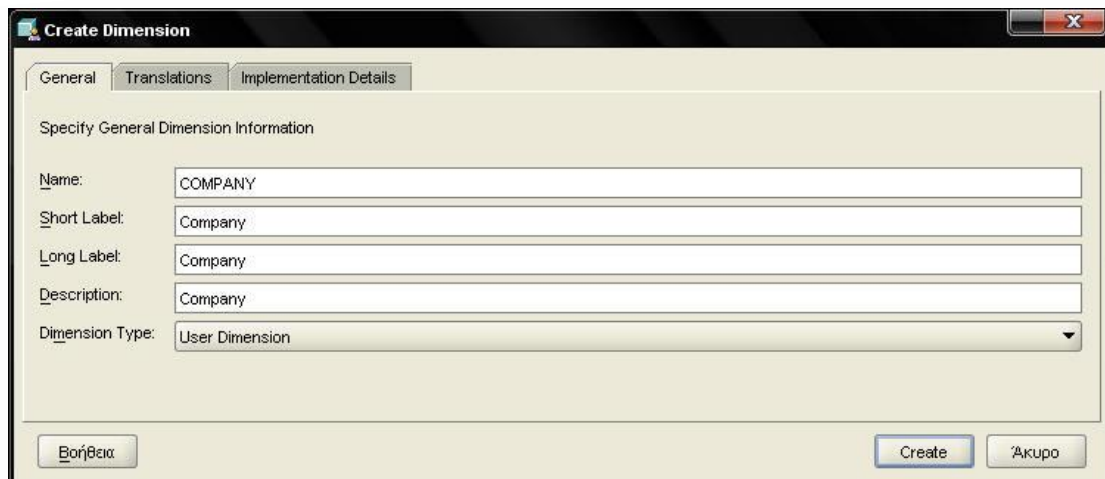
Με αυτό τον τρόπο δημιουργούμε όλες τις dimensions. Το επόμενο βήμα είναι για κάθε dimension να οριστούν τα level της (επίπεδα) τα οποία θα χρησιμοποιηθούν αργότερα στη δημιουργία της hierarchy (ιεραρχίας). Στη συγκεκριμένη dimension COMPANY, τα levels τα οποία θα δημιουργηθούν είναι το σύνολο των εταιρειών (ALL\_COMPANIES), η εταιρεία σε σύνολο (COMPANY) και το κάθε υποκατάστημα ξεχωριστό (STORE). Τα levels αυτά δημιουργήθηκαν με βάση τον λογικό σχεδιασμό και την ανάλυση απαιτήσεων καθώς επιθυμούμε την εύκολη αναζήτηση αθροισμάτων τόσο στο σύνολο όλων των εταιρειών όσο και για κάθε εταιρεία, καθώς επίσης και για κάθε υποκατάστημα.



Εικόνα 22: Δημιουργία Analytic Workspace

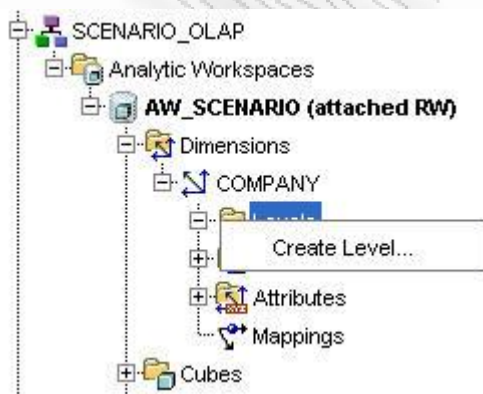


Εικόνα 23: Δημιουργία Dimension



Εικόνα 24: Δημιουργία Dimension – Ετικέτα General

Για τη δημιουργία του level επιλέγουμε την dimension στην οποία θα αντιστοιχεί και στη συνέχεια «Create Level» (Εικόνα 25). Όμοια με τη διαδικασία δημιουργίας dimension, εμφανίζεται παράθυρο στο οποίο αναθέτουμε τα βασικά χαρακτηριστικά, για παράδειγμα Name, Short Label (Εικόνα 26). Με τον ίδιο τρόπο δημιουργούνται και τα υπόλοιπα levels της dimension.



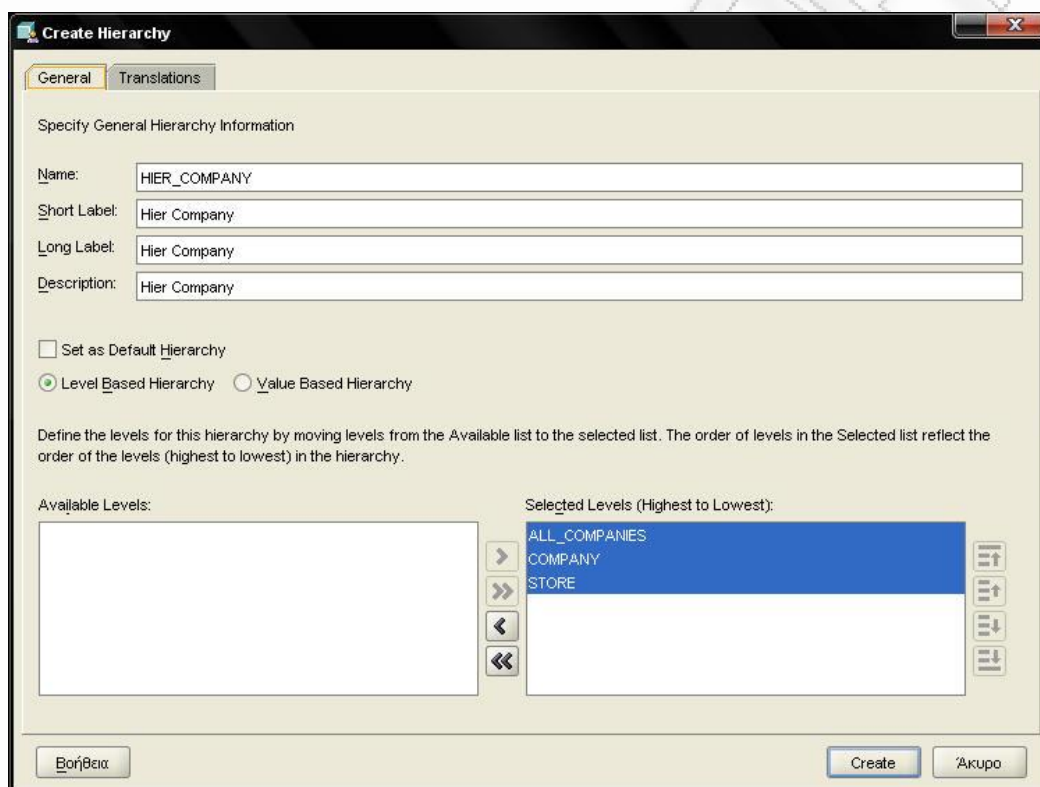
Εικόνα 25: Δημιουργία Level



Εικόνα 26: Δημιουργία Level – Ετικέτα General



Το επόμενο βήμα, αφού έχει ολοκληρωθεί η υλοποίηση των levels, είναι ο καθορισμός της hierarchy της συγκεκριμένης dimension. Η ιεραρχία μας θα αποτελείται από τα παραπάνω level και ορίζεται σαν προεπιλεγμένη ιεραρχία αφού μια διάσταση μπορεί να έχει πολλές ιεραρχίες ανάλογα με τις ανάγκες που έχει να καλύψει. Στην Εικόνα 27 φαίνεται ο τρόπος δημιουργίας μια ιεραρχίας στην οποία με τη βοήθεια των levels ορίζουμε τα επίπεδα της συνάθροισης για τη συγκεκριμένη dimension.



Εικόνα 27: Δημιουργία Hierarchy

Αφού έχουμε φτιάξει τις ιεραρχίες της dimension, που στο συγκεκριμένο παράδειγμα μας είναι μόνο μια, το επόμενο βήμα που έχουμε να κάνουμε είναι να δημιουργήσουμε τα attributes της. Τα attributes είναι τα χαρακτηριστικά που θέλουμε να έχει η dimension στη συγκεκριμένη περίπτωση είναι οι περιγραφές του Συνόλου των Εταιρειών, η περιγραφή της κάθε Εταιρείας και η περιγραφή του κάθε υποκαταστήματος. Το εργαλείο Analytic Workspace κατά τη δημιουργία της dimension δημιουργεί αυτόματα δυο attributes το Long Description (Μεγάλη περιγραφή) και το Short Description (Μικρή περιγραφή). Τα attributes αυτά καλύπτουν τις ανάγκες μας συνεπώς δε θα χρειαστεί να δημιουργήσουμε επιπλέον.

Τα παραπάνω attributes που δημιουργούνται αυτόματα αφορούν τις dimension τύπου User Dimension, ενώ αντίστοιχα για τη dimension τύπου Time Dimension δημιουργούνται επιπλέον τα attributes END\_DATE, που δηλώνει τη τελευταία ημέρα κάθε χρονικής περιόδου και TIME\_SPAN, που δηλώνει τον σύνολο των ημερών μιας χρονικής περιόδου.

Θα πρέπει να σημειωθεί ότι για τη dimension TIME είναι η μοναδική που επιθυμούμε να έχει δυο διαφορετικές hierarchies όπως παρουσιάστηκε κι από τον λογικό σχεδιασμό. Η βασική hierarchy συναθροίζει τα δεδομένα στα levels YEAR, TRIMESTER και MONTH, DAY ενώ η δεύτερη ιεραρχία συναθροίζει τα δεδομένα στα levels YEAR, WEEK και DAY.

Αφού έχουμε τελειώσει τη δημιουργία της dimension και έχουμε καθορίσει όλα τα απαιτούμενα στοιχεία, στη συνέχεια θα πρέπει να συνδέσουμε τη dimension με τον αντίστοιχο πίνακα της σχεσιακής βάσης.

Η δημιουργία των πινάκων αναφέρεται στον φυσικό σχεδιασμό ο οποίος θα περιγραφεί στη συνέχεια του κεφαλαίου.

Η φυσική δημιουργία των tables, views και γενικά όλων των αντικειμένων της σχεσιακής βάσης δεδομένων του Data Warehouse υλοποιείται κατά τον φυσικό σχεδιασμό. Για τον φυσικό σχεδιασμό μπορούμε είτε να χρησιμοποιήσουμε τον εργαλείο της Oracle Oracle Warehouse Builder είτε να τα δημιουργήσουμε με τη χρήση SQL Scripts. Στο σημείο αυτό, δημιουργούμε τα αντίστοιχα αντικείμενα με τη χρήση απλών SQL Scripts.

Ακολουθούν τα SQL Scripts που χρησιμοποιήθηκαν για τη δημιουργία των πινάκων της βάσης δεδομένων .

#### Δημιουργία table COMPANYY

```
create table COMPANYY_
(
  STORE_ID      NUMBER(12) not null,
  STORE        VARCHAR2(50) not null,
  COMPANYY_ID  NUMBER(12) not null,
  COMPANYY     VARCHAR2(40) not null,
  ALL_COMPANIES_ID NUMBER(5) not null,
  ALL_COMPANIES VARCHAR2(40) not null
)
tablespace TS_SCENARIO_OLAP
```

```

pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

-- Create/Recreate primary, unique and foreign key constraints
alter table COMPANY_DIM
add constraint PK_COMPANY primary key (STORE_ID)
using index
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

#### Δημιουργία table PRODUCT

```

create table PRODUCT
(
  PRODUCT_ID    NUMBER(5) not null,
  PRODUCT       VARCHAR2(70) not null,
  CATEGORY_ID   NUMBER(5) not null,
  CATEGORY      VARCHAR2(25) not null,
  TYPE_ID       NUMBER(5) not null,
  TYPE          VARCHAR2(15) not null,
  ALL_PRODUCTS_ID NUMBER(5) not null,
  ALL_PRODUCTS VARCHAR2(15) not null
)
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

-- Create/Recreate primary, unique and foreign key constraints
alter table PRODUCT
add constraint PK_PRODUCT primary key (PRODUCT_ID)
using index
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

**Δημιουργία table REGION**

```

create table REGION
(
  ZIP_CODE   VARCHAR2(5) not null,
  REGION_ID  NUMBER(3) not null,
  REGION     VARCHAR2(50) not null,
  PREFECTURE_ID NUMBER(3) not null,
  PREFECTURE VARCHAR2(40) not null,
  CITY_ID    NUMBER(3) not null,
  CITY       VARCHAR2(40) not null
)
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table REGION
add constraint PK_REGION primary key (ZIP_CODE)
using index
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

**Δημιουργία table TIME**

```

create table TIME
(
  MONTH_ID      NUMBER(5) not null,
  MONTH         VARCHAR2(20) not null,
  MONTH_DAYS    NUMBER(2) not null,
  MONTH_END_DATE DATE not null,
  TRIMESTER_ID  NUMBER(5) not null,
  TRIMESTER     VARCHAR2(50) not null,
  TRIMESTER_DAYS NUMBER(3) not null,
  TRIMESTER_END_DATE DATE not null,
  YEAR_ID       NUMBER(5) not null,
  YEAR          VARCHAR2(5) not null,
  YEAR_DAYS     NUMBER(3) not null,
  YEAR_END_DATE DATE not null,
  WEEK_ID       NUMBER(5) not null,
  WEEK          VARCHAR2(50) not null,
  WEEK_DAYS     NUMBER(3) not null,
  WEEK_END_DATE DATE not null
)
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 1

```

```

maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table TIME_DIM
add constraint PK_TIME primary key (MONTH_ID)
using index
tablespace TS_SCENARIO_OLAP
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

Για τη διαδικασία ETL που θα παρουσιαστεί στο παρακάτω κεφάλαιο (παράγραφος 5.7) χρειάζεται να δημιουργούνται αυτόματα μοναδικά κλειδιά για τους πίνακες. Η αυτοματοποίηση αυτή γίνεται με τη χρήση sequence. Το αντικείμενο sequence στην Oracle δημιουργεί αυτόματα αριθμούς. Τα παρακάτω SQL Scripts ορίζουν τα sequence που θα χρησιμοποιηθούν.

```

-- Create sequence
create sequence TRIMESTER_SEQ
minvalue 1
maxvalue 1000000
start with 1
increment by 1
cache 20;

-- Create sequence
create sequence YEAR_SEQ
minvalue 1
maxvalue 1000000
start with 1
increment by 1
cache 20;

-- Create sequence
create sequence WEEK_SEQ
minvalue 1
maxvalue 1000000
start with 1
increment by 1
cache 20;

-- Create sequence
create sequence DAY_SEQ
minvalue 1
maxvalue 1000000000

```

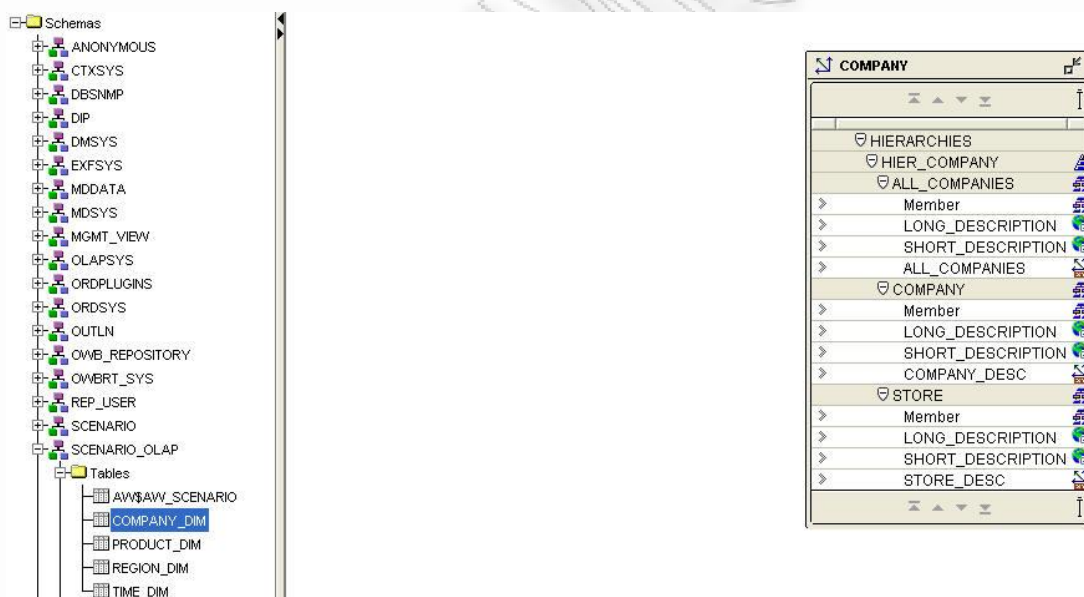
```

start with 1
increment by 1
cache 20;

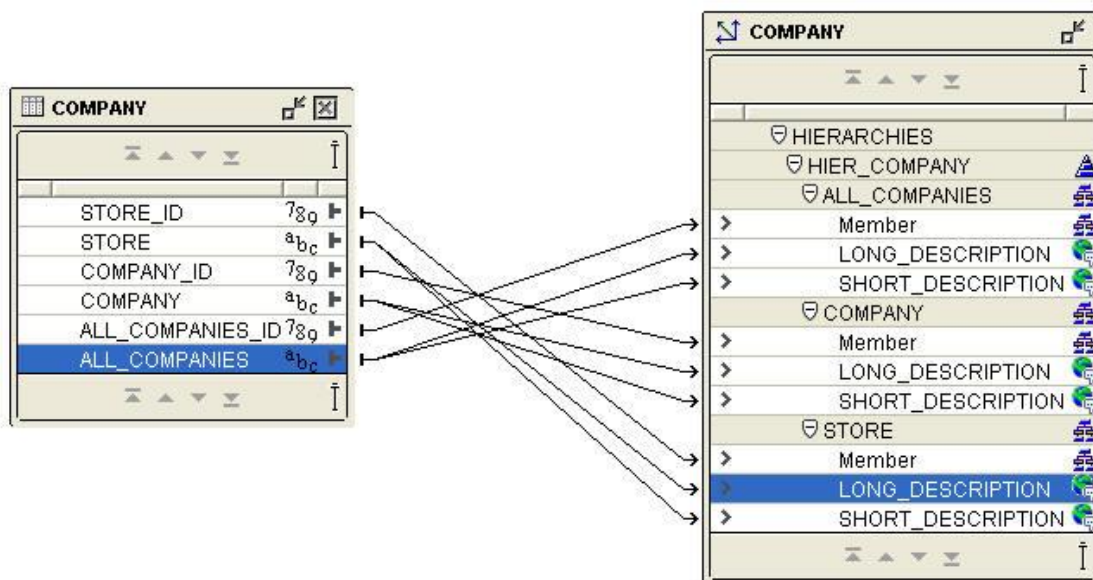
-- Create sequence
create sequence MONTH_SEQ
minvalue 1
maxvalue 1000000
start with 1
increment by 1
cache 20;
    
```

Το επόμενο βήμα που θα πρέπει να εκτελεστεί είναι η αντιστοίχιση των διαστάσεων στους αντίστοιχους σχεσιακούς πίνακες που μόλις δημιουργήθηκαν.

Αφού πρώτα επιλέξουμε το σχήμα και τον πίνακα που επιθυμούμε, τον μεταφέρουμε στο διπλανό παράθυρο και στη συνέχεια συνδέουμε τα στοιχεία της dimension με το πεδία του πίνακα (Εικόνα 29). Κάθε attribute ενός level αντιστοιχεί και σε ένα πεδίο του πίνακα.



Εικόνα 28: Mapping



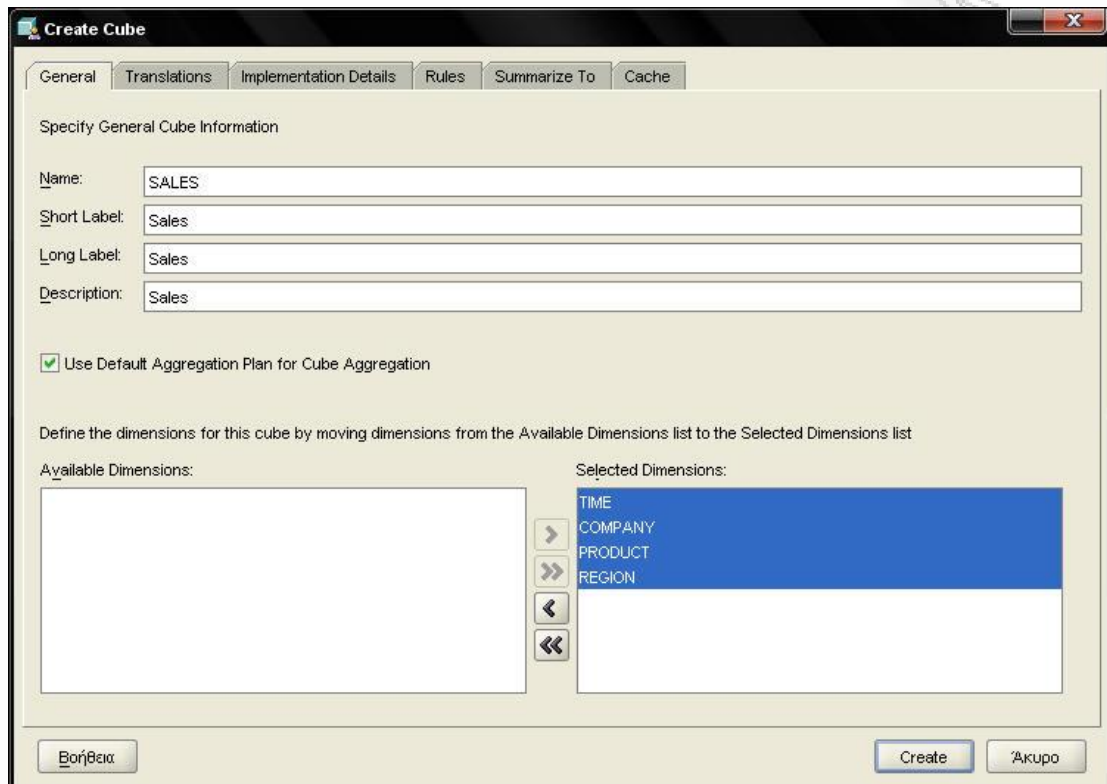
Εικόνα 29: Mapping dimension Company με τον σχεσιακό πίνακα της βάσης

Αφού δημιουργήσουμε όλες τις απαιτούμενες dimensions στη συνέχεια αυτό που θα πρέπει να κάνουμε είναι να δημιουργήσουμε τον cube στο οποίο θα βρίσκονται τα αποτελέσματά μας. Σύμφωνα με την ανάλυση απαιτήσεων που έχει παρουσιαστεί στη παράγραφο 5.4 τα αποτελέσματα που θέλουμε να έχουμε αφορούν τις πωλήσεις των προϊόντων και τα σύνολα αυτών. Γι αυτό το λόγο και ο cube θα περιέχει την απαραίτητη δομή ώστε να παρέχει την επιθυμητή πληροφορία. Δημιουργούμε τον cube στο εργαλείο της Oracle Analytic Workspace.

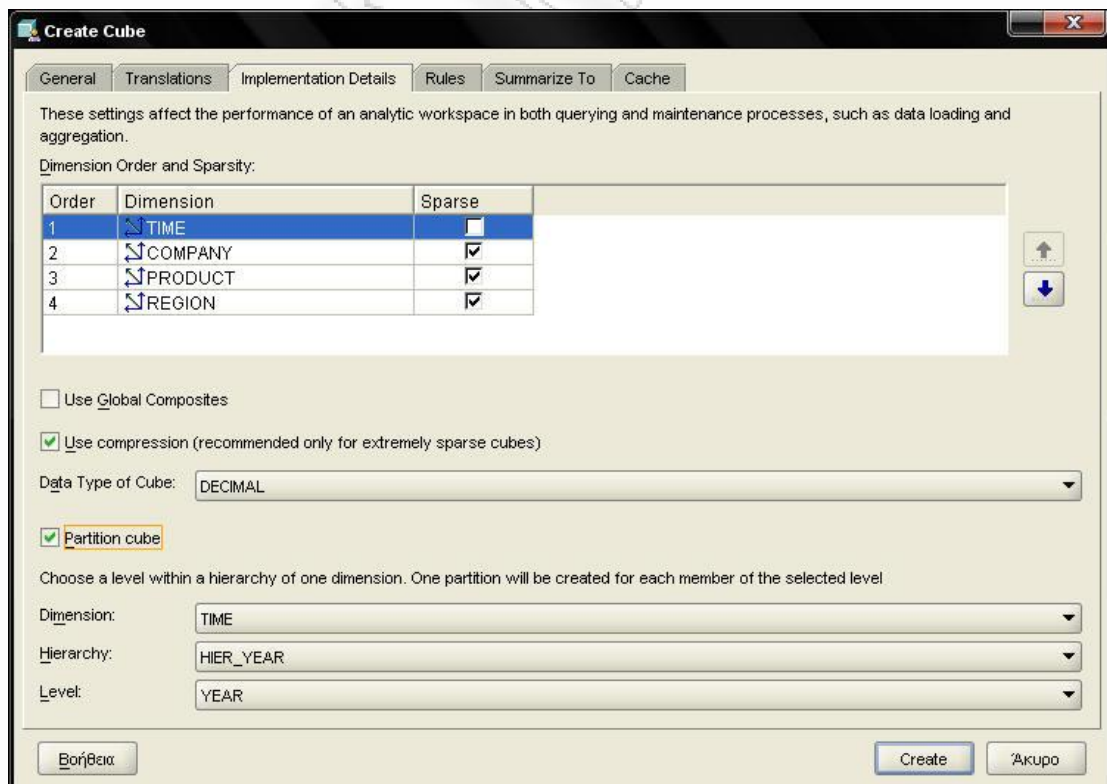
Όπως φαίνεται στην Εικόνα 30, το πρώτο πράγμα που θα πρέπει να ορίσουμε κατά τη δημιουργία του cube είναι οι dimensions από τις οποίες θα αποτελείται. Στο συγκεκριμένο σενάριο ο cube SALES θα αποτελείται από τις τέσσερις dimensions που περιγράφηκαν πιο πάνω. Αυτές είναι η dimension TIME, η COMPANY, η PRODUCT και η REGION. Στο επόμενο Tab Translations (Ετικέτα) καταχωρούμε την ελληνική μετάφραση για τον cube SALES.

Ιδιαίτερο ενδιαφέρον έχει το επόμενο Tab Implementation Details καθώς σε αυτό ορίζουμε ποιες dimensions θα είναι πιο αραιές και ποιες όχι. Με αυτόν το τρόπο βελτιστοποιείται ο χρόνος απάντησης των υποερωτημάτων και της φόρτωσης των δεδομένων στον κύβο. Στον κύβο SALES θεωρούμε ότι όλες οι dimensions θα είναι αραιές εκτός από τον χρόνο. Το αποτέλεσμα θα είναι σε δεκαδική μορφή. Τέλος,

δημιουργούμε partition (χώρισμα) με βάση το χρόνο για καλύτερη αποδοτικότητα (Εικόνα 31).



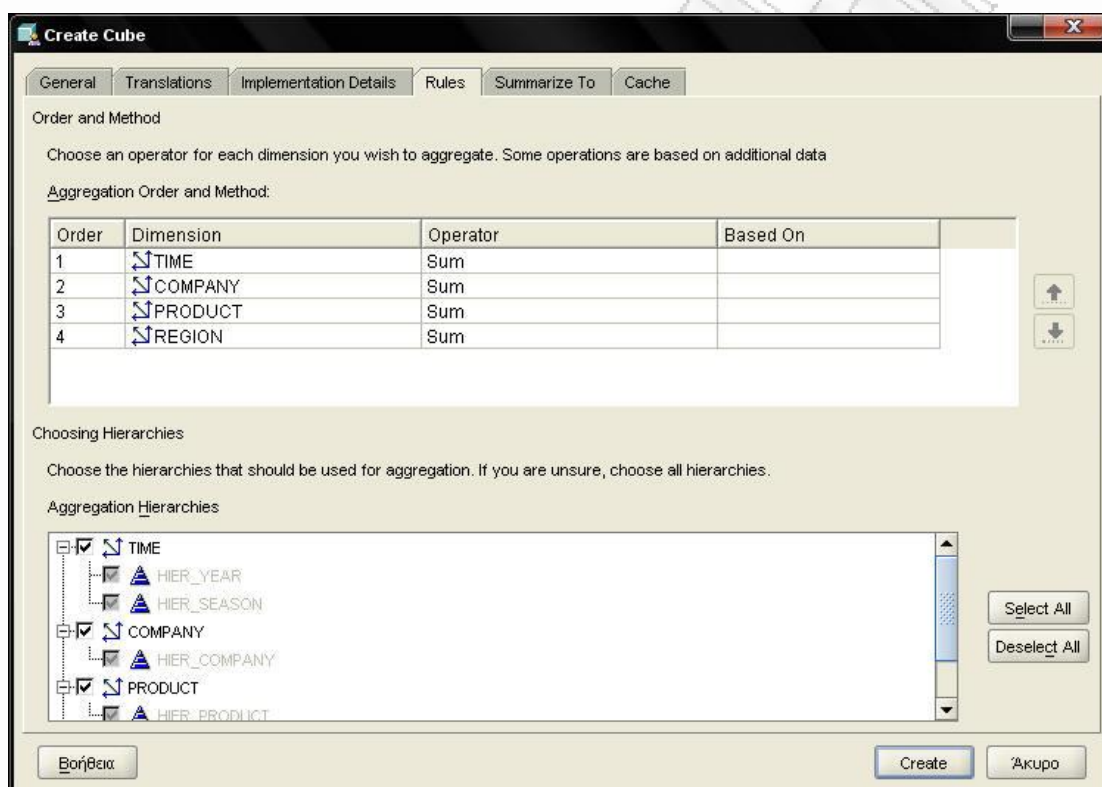
Εικόνα 30: Δημιουργία Cube Sales – Ετικέτα General





**Εικόνα 31: Δημιουργία Cube Sales – Ετικέτα Implementation Details**

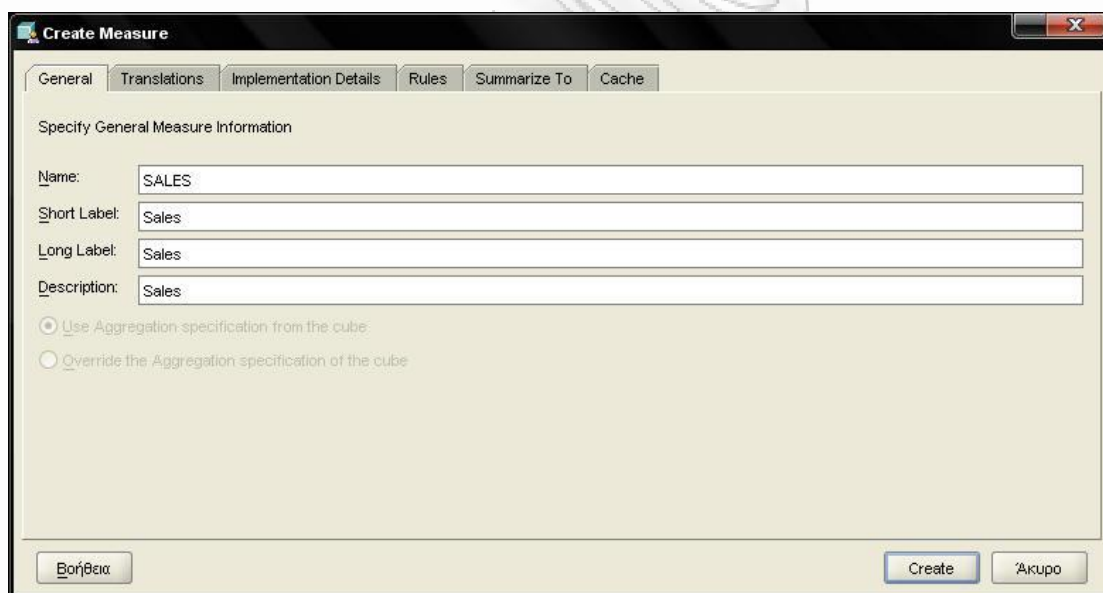
Στο Tab Rules ( Ετικέτα Κανόνες) ορίζουμε ποιοι θα είναι οι βασικοί κανόνες που θα τρέχουν στον cube. Στο δικό μας σενάριο αρχικά ορίζουμε τον απλό κανόνα της πρόσθεσης. Συγκεκριμένα, ο κανόνας τρέχει για κάθε ιεραρχία και υπολογίζει τα αθροίσματα ανά level (Εικόνα 32). Στο επόμενο Tab (Summarize To) ορίζουμε για ποιες περιοχές του cube θα υπάρχουν προ-υπολογισμένα τα αθροίσματα. Τέλος, στο Tab Cache, ορίζουμε αν τα δεδομένα θα αποθηκεύονται ανά σύνοδο (session) και δε θα ξανά - υπολογίζονται.



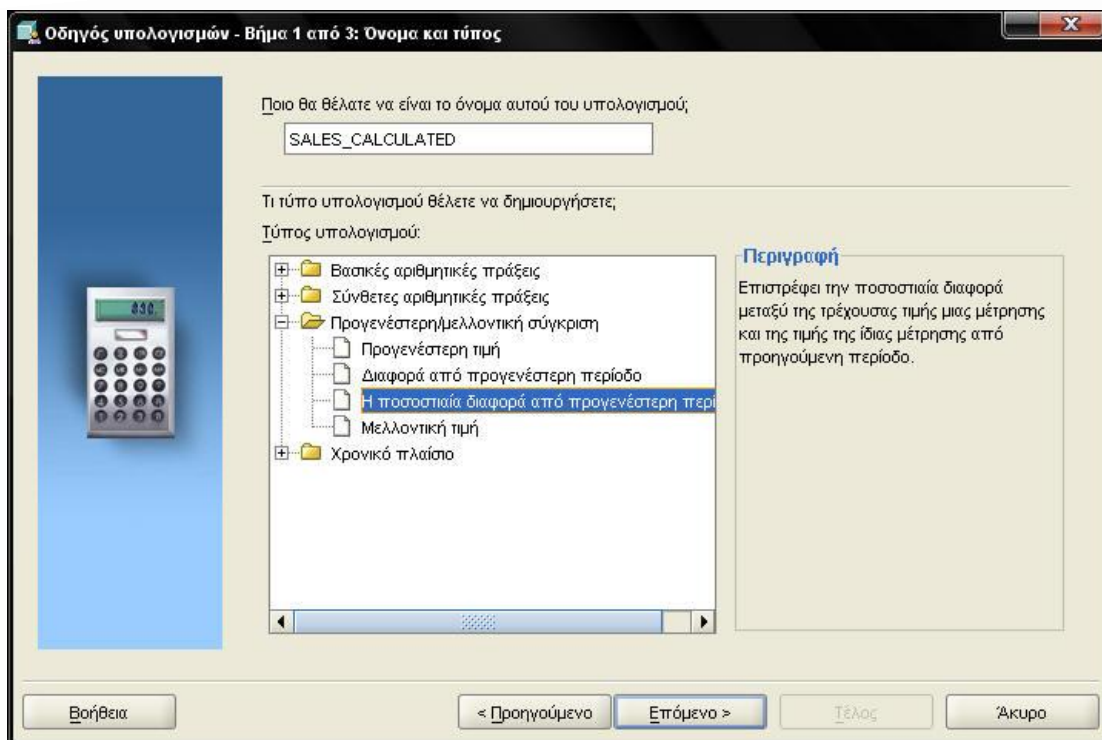
**Εικόνα 32: Δημιουργία Cube – Ετικέτα Rules**

Επόμενη κίνηση είναι η δημιουργία των measures και των calculated measures. Τα measures που θα δημιουργηθούν αντιστοιχούν στις πωλήσεις και στα τεμάχια για τα οποία θέλουμε να αθροίζονται σε κάθε level ιεραρχίας. Αντίστοιχα, θα φτιαχτεί και ένα επιπλέον calculated measure το οποίο θα υπολογίζει το ποσοστό διαφοράς τιμών μεταξύ της τρέχουσας και της προηγούμενης περιόδου. Για παράδειγμα, η αναφορά παρουσιάζει τα δεδομένα για το μήνα Ιούνιο του 2008, το calculated measure θα υπολογίσει σε ποσοστό τη διαφορά των βασικών measures μεταξύ του μήνα Μάιο του 2008 και Ιούνιο του 2008.

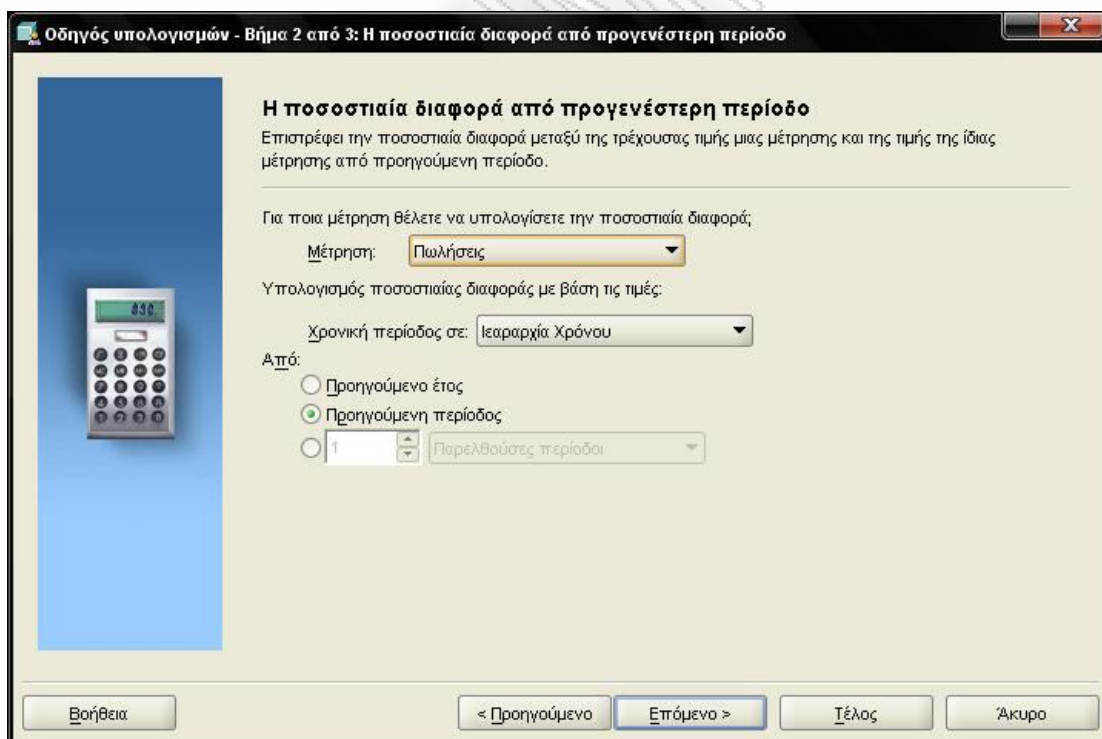
Η δημιουργία των measures γίνεται κάνοντας δεξί κλικ στον κόμβο Measures και στη συνέχεια επιλέγουμε Create Measure (Εικόνα 33). Στην οθόνη που εμφανίζεται ορίζουμε το όνομα του measure και αφήνουμε τα υπόλοιπα στοιχεία όπως είναι. Και τα δυο measures δημιουργούνται με τον ίδιο ακριβώς τρόπο. Ενώ αντίστοιχα, για να μπορούμε να έχουμε συγκριτικά αποτελέσματα μεταξύ δυο περιόδων (χρόνος, μήνας κτλ) δημιουργήθηκε ένα calculated measure το οποίο καλύπτει αυτή την ανάγκη. Η δημιουργία του calculated measure εκτελείται σε τρία βήματα. Στο πρώτο βήμα, ορίζουμε το όνομα και τη πράξη που θέλουμε να κάνει (Εικόνα 34). Στο δεύτερο βήμα, ορίζουμε ποιο πεδίο θα χρησιμοποιεί. Συγκεκριμένα, έχουμε τη δυνατότητα να χρησιμοποιήσουμε είτε measures είτε άλλα calculated measures. Επιπρόσθετα, ορίζουμε ποια ιεραρχία θα χρησιμοποιεί και το διάστημα σύγκρισης των δεδομένων (Εικόνα 35). Στο τελευταίο βήμα, ορίζεται με ένα κείμενο η περιγραφή του calculated measure.



Εικόνα 33: Δημιουργία measure SALES



Εικόνα 34: Δημιουργία Calculated measure – Βήμα 1



Εικόνα 35: Δημιουργία Calculated measure – Βήμα 2

Εφόσον, έχουν εκτελεστεί όλα τα παραπάνω βήματα για όλες τις dimensions μπορούμε να προχωρήσουμε στην επόμενη διαδικασία που είναι η εισαγωγή των

δεδομένων από την αρχική σχεσιακή βάση δεδομένων στη σχεσιακή βάση δεδομένων του Data Warehouse.

Στην ακόλουθη παράγραφο παρουσιάζεται η διαδικασία αυτή και ο τρόπος με τον οποίο εκτελέστηκε για το συγκεκριμένο σενάριο.

### 5.7. Διαδικασία ETL

Στην παράγραφο αυτή θα περιγραφεί ο τρόπος με τον οποίο εκτελέστηκε η διαδικασία ETL (Extract, Transform, Transaction). Δηλαδή, ο τρόπος με τον οποίο μεταφέρθηκαν τα δεδομένα από το αρχικό σχήμα της βάσης δεδομένων στη βάση δεδομένων του Data Warehouse. Για τη συγκεκριμένη εργασία δημιουργήθηκαν procedures (διαδικασίες) στην Oracle και προγραμματισμένα jobs (ενέργειες).

Στο συγκεκριμένο σενάριο έχουμε μόνο μια πηγή δεδομένων που συγκεκριμένα είναι κι αυτή σχεσιακή βάση δεδομένων σε Oracle. Η πληροφορία που μας ενδιαφέρει έχει ήδη καθοριστεί και αναγνωριστεί από το βήμα του λογικού σχεδιασμού. Συγκεκριμένα, τα δεδομένα που ενδιαφέρουν το σενάριο αυτό βρίσκονται στους παρακάτω πίνακες:

- COMPANIES
- PR\_COMPANIES
- PRODUCTS
- PR\_TYPES
- SALES
- ST\_ADDRESSES
- STORES

του αρχικού σχήματος SCENARIO.

Ένα πλεονέκτημα της συγκεκριμένης υλοποίησης είναι ότι και τα δυο σχήματα (πηγή/ προορισμός) βρίσκονται σε βάση Oracle. Συνεπώς, δεν χρειάζονται αλλαγές στους τύπους δεδομένων (Data Type) της εκάστοτε πληροφορίας. Για να μπορέσουν

οι εργασίες της διαδικασίας ETL να εκτελεστούν θα πρέπει να δοθούν τα κατάλληλα δικαιώματα στη βάση δεδομένων του Data Warehouse ώστε να μπορεί να εξάγει στοιχεία από τη βάση δεδομένων της πηγής. Επίσης, θα πρέπει να δημιουργηθεί μια σύνδεση των σχημάτων (dblink) εάν αυτά βρίσκονται σε διαφορετικές βάσεις δεδομένων. Στο σενάριο αυτό δεν είναι απαραίτητο καθώς τα δυο σχήματα βρίσκονται στην ίδια βάση δεδομένων, αλλά θα πρέπει να εκτελεστούν συγκεκριμένα SQL Scripts τα οποία θα δίνουν δικαίωμα στο χρήστη SCENARIO\_OLAP να διαβάσει (SELECT) δεδομένα από τους πίνακες του χρήστη SCENARIO. Τα SQL Scripts παρατίθενται πιο κάτω.

```
GRANT SELECT ON COMPANIES TO SCENARIO_OLAP;  
GRANT SELECT ON STORES TO SCENARIO_OLAP;  
GRANT SELECT ON PRODUCTS TO SCENARIO_OLAP  
GRANT SELECT ON PR_CATEGORIES TO SCENARIO_OLAP  
GRANT SELECT ON PR_TYPES TO SCENARIO_OLAP  
GRANT SELECT ON SALES TO SCENARIO_OLAP;  
GRANT SELECT ON ST_ADDRESSES TO SCENARIO_OLAP;  
GRANT CREATE JOB TO SCENARIO_OLAP;
```

Ο τρόπος εξαγωγής των δεδομένων που χρησιμοποιήθηκε είναι η πλήρης εξαγωγή (παράγραφος 4.5). Με τον τρόπο αυτό εξάγουμε ολοκληρωτικά δεδομένα και δεν ελέγχουμε τις αλλαγές σε σχέση με τη προηγούμενη εξαγωγή. Η επεξεργασία των δεδομένων γίνεται χρησιμοποιώντας τη τεχνική MDT (παράγραφος 4.5). Δηλαδή, ο μετασχηματισμός των δεδομένων γίνεται σε παραπάνω από ένα βήμα και σταδιακά. Η παραπάνω τεχνική γίνεται αντιληπτή κατά τη παρουσίαση των SQL Scripts των διαδικασιών. Η φόρτωση των δεδομένων γίνεται αμέσως μετά από την επεξεργασία τους πάλι με τη χρήση SQL Scripts.

Η δομή των εργασιών που τρέχουν και εκτελούν τη διαδικασία ETL, είναι ένα βασικό job που έχει προγραμματιστεί να τρέχει κάθε βράδυ και καλεί επιμέρους procedures. Η κάθε procedure τρέχει μια υπό-διαδικασία ETL για κάθε πίνακα που έχει σκοπό να φορτώσει με δεδομένα. Παρακάτω παρουσιάζονται τα SQL Scripts τόσο του job όσο και των procedures.

Η procedure IMPORT\_COMPANY\_DIM είναι αυτή που εξάγει τα δεδομένα από τα χρήστη SCENARIO τα επεξεργάζεται και στη συνέχεια τα φορτώνει στο σχήμα

SCENARIO\_OLAP στον table COMPANY που είναι κι αυτός που θα περιέχει τα δεδομένα της dimension COMPANY.

```

CREATE OR REPLACE PROCEDURE IMPORT_COMPANY_DIM is

STORE_ID NUMBER(12);
STORE VARCHAR2(50);
COMPANY_ID NUMBER(12);
COMPANY VARCHAR2(50);

CURSOR CUR_COMPANY IS
SELECT ST.ST_ID, ST_NAME, CM.COM_ID, CM.COM_NAME FROM SCENARIO.STORES ST,
SCENARIO.COMPANIES CM WHERE
CM.COM_ID = ST.COM_ID AND ST_ID NOT IN (SELECT STORE_ID FROM COMPANY);

begin

OPEN CUR_COMPANY;
LOOP
FETCH CUR_COMPANY INTO STORE_ID,STORE,COMPANY_ID, COMPANY;
EXIT WHEN CUR_COMPANY%NOTFOUND;

INSERT INTO
COMPANY(STORE_ID,STORE,COMPANY_ID,COMPANY,ALL_COMPANIES_ID,ALL_COMPANIES)
VALUES(STORE_ID, STORE, COMPANY_ID, COMPANY,1, 'Σύνολο Εταιρειών');
END LOOP;
CLOSE CUR_COMPANY;

COMMIT;
END IMPORT_COMPANY_DIM;
    
```

Η procedure IMPORT\_PRODUCT\_DIM εκτελεί τη διαδικασία ETL συγκεκριμένα για τη dimension PRODUCT. Διαβάζει και φορτώνει δεδομένα από τους πίνακες PRODUCTS, PR\_CATEGORIES και PR\_TYPES, τα συνδέει και στη συνέχεια τα εισάγει στον πίνακα PRODUCT του σχήματος SCENARIO\_OLAP για να έχει δεδομένα η dimension PRODUCT.

```

CREATE OR REPLACE PROCEDURE IMPORT_PRODUCT_DIM IS

PRODUCT_ID NUMBER(5);
PRODUCT VARCHAR2(70);
CATEGORY_ID NUMBER(5);
CATEGORY VARCHAR2(25);
TYPE_ID NUMBER(5);
TYPE_DESC VARCHAR2(15);

CURSOR PRODUCTS IS
SELECT PR.PR_ID, PR.PR_NAME, PRC.PR_CAT_ID, PRC.PR_CAT_NAME, PRT.PR_TYPE_ID,
PRT.PR_NAME
FROM SCENARIO.PRODUCTS PR, SCENARIO.PR_CATEGORIES PRC, SCENARIO.PR_TYPES
PRT
WHERE
PRC.PR_CAT_ID = PR.PR_CAT_ID
AND PRC.PR_TYPE_ID = PRT.PR_TYPE_ID
    
```

```
AND PR.PR_ID NOT IN (SELECT E.PRODUCT_ID FROM PRODUCT E);
```

```
begin
  OPEN PRODUCTS;
  LOOP
    FETCH PRODUCTS INTO PRODUCT_ID, PRODUCT, CATEGORY_ID,
    CATEGORY,TYPE_ID,TYPE_DESC ;
    EXIT WHEN PRODUCTS%NOTFOUND;

    INSERT INTO PRODUCT(PRODUCT_ID, PRODUCT_DESC, CATEGORY_ID,
    CATEGORY_DESC, TYPE_ID, TYPE_DESC, ALL_PRODUCTS_ID, ALL_PRODUCTS_DESC)
    VALUES(PRODUCT_ID, PRODUCT, CATEGORY_ID, CATEGORY, TYPE_ID,TYPE_DESC,
    1, 'Όλα τα προϊόντα');
    END LOOP;
  CLOSE PRODUCTS;

  COMMIT;
END IMPORT_PRODUCT_DIM;
```

Η procedure IMPORT\_TIME\_DIM έτρεξε αρχικά μια φορά για τα έτη 2002 μέχρι και 2009 και στη συνέχεια καλείται από τη διαδικασία που γεμίζει τον table SALES που αντιστοιχεί στον κύβο, όταν οι πωλήσεις αναφέρονται σε ημερομηνία που δεν έχει περαστεί. Για παράδειγμα, την 01/01/2010 η διαδικασία που φορτώνει τα δεδομένα του κύβου θα εντοπίσει ότι υπάρχουν πωλήσεις στη συγκεκριμένη ημερομηνία και θα γεμίσει το πίνακα TIME με τα σωστά δεδομένα του έτους 2010.

```
create or replace procedure IMPORT_TIME_DIM(currentYEAR in number) is
```

```
month VARCHAR2(20);
month_days number;
month_last_day Date;
month_id number;
trimester_desc VARCHAR2(50);
trimester_days number;
trimester_last_date DATE;
trimesterID NUMBER;
counterMonth number;
counterDayWeek number;
counterWeek NUMBER;
yearID NUMBER;
year_days number;
year_last_date DATE;
currentDate DATE;
weekID NUMBER;
week VARCHAR2(50);
week_last_date DATE;
dayID NUMBER;
dayStr VARCHAR2(10);
trimesterID1 NUMBER;
trimesterID2 NUMBER;
trimesterID3 NUMBER;
trimesterID4 NUMBER;
```

```
begin
```

```
counterMonth :=1;
```

```

-- Days of current year
SELECT LAST_DAY(TO_DATE('01/12' || currentYEAR, 'dd/mm/yyyy')) - TO_DATE('01/01' ||
currentYEAR, 'dd/mm/yyyy') + 1 into year_days from dual;
-- Current year is input parameter
--Last date of the year
SELECT LAST_DAY(TO_DATE('01/12' || currentYear,'dd/mm/yyyy')) into year_last_date from
dual;
-- YEAR ID
SELECT YEAR_SEQ.NEXTVAL INTO yearID FROM DUAL;
--initialize 4 different ids for trimesters
SELECT TRIMESTER_SEQ.NEXTVAL INTO trimesterID1 FROM DUAL;
SELECT TRIMESTER_SEQ.NEXTVAL INTO trimesterID2 FROM DUAL;
SELECT TRIMESTER_SEQ.NEXTVAL INTO trimesterID3 FROM DUAL;
SELECT TRIMESTER_SEQ.NEXTVAL INTO trimesterID4 FROM DUAL;

-- For each month
FOR counter IN 1..12 LOOP
  -- CurrentDate
  currentDate := TO_DATE('01/' || counter || '/' || currentYEAR, 'dd/mm/YYYY');
  -- text of month in greek, format ex. Ιανουάριος, 08
  month := GET_MONTH(currentDate);
  -- Month ID
  SELECT MONTH_SEQ.NEXTVAL INTO month_id FROM DUAL;
  -- Days of current month
  select to_number(to_char(last_day(currentDate), 'dd')) into month_days from dual;
  --Last day of current month
  month_last_day := last_day(currentDate);
  -- text timester, format ex Τρίμηνο 1, Τρίμηνο 2 κτλ
  trimester_desc := GET_TRIMESTER(currentDate);
  --Days of current trimester
  trimester_days := GET_DAYS_TRIMESTER(currentDate);
  --last day of current trimester
  trimester_last_date := GET_LASTDAY_TRIMESTER(currentDate);

  counterDayWeek := 1;
  counterWeek := 1;
  -- First Week ID
  SELECT WEEK_SEQ.NEXTVAL INTO weekID FROM DUAL;

  FOR counterMonth IN 1..month_days LOOP
    --Day ID
    SELECT DAY_SEQ.NEXTVAL INTO dayID FROM DUAL;
    -- Current date as a type DATE
    currentDate:= TO_DATE( counterMonth || '/' || counter || '/' || currentYEAR,
'dd/mm/YYYY');
    -- Current date as a string Δευτέρα, Τρίτη etc
    dayStr := GET_DAY_NAME(currentDate);
    IF (counterDayWeek = 7) THEN
      SELECT WEEK_SEQ.NEXTVAL INTO weekID FROM DUAL;
      counterDayWeek := 1;
      counterWeek := counterWeek + 1;
    END IF;

    -- Name of the week
    week := 'Εβδομάδα' || counterWeek;
    --Last Date of the week, according to the current date, Monday THROUGH Sunday
    -- Select NEXT_DAY(TRUNC(currentDate, 'IW'), 'SUNDAY') INTO week_last_date FROM
DUAL;

    Select NEXT_DAY(TRUNC(currentDate, 'IW'), 7) INTO week_last_date FROM DUAL;

```



```

    if (counter = 1 or counter = 2 or counter = 3 ) then
        trimesterID := trimesterID1;
    ELSIF (counter = 4 or counter = 5 or counter = 6) then
        trimesterID := trimesterID2;
    ELSIF (counter = 7 or counter = 8 or counter = 9) then
        trimesterID := trimesterID3;
    ELSE
        trimesterID := trimesterID4;
    END IF;

    INSERT INTO
    TIME(MONTH_ID,MONTH,MONTH_DAYS,MONTH_END_DATE,TRIMESTER_ID,TRIMESTER
    ,TRIMESTER_DAYS,TRIMESTER_END_DATE,
    YEAR_ID,YEAR,YEAR_DAYS,YEAR_END_DATE,WEEK_ID,WEEK,WEEK_DAYS,WEEK_END_
    DATE,DAY_ID,DAY,DAY_NAME,DAY_OF_WEEK,
    DAY_OF_YEAR,DAY_OF_QUARTER,DAY_OF_MONTH)VALUES (month_id, month,
    month_days, month_last_day, trimesterID, trimester_desc, trimester_days,
    trimester_last_date, yearID, currentYEAR, year_days, year_last_date, weekID, week, 7,
    week_last_date,
    DAY_SEQ.NEXTVAL, currentDate, dayStr,0 ,0 ,0 ,0 );
    -- Last Action
    counterDayWeek := counterDayWeek + 1;
    END LOOP;

    END LOOP;

    commit;
end IMPORT_TIME_DIM;

```

Η παραγωγή των δεδομένων για τον πίνακα REGION που αντιστοιχεί στη dimension REGION γίνεται μια μόνο φορά κατά την αρχικοποίηση του συστήματος. Εφόσον, το σενάριο αναφέρεται μόνο για την Ελλάδα, βρέθηκε υλικό από το ταχυδρομείο και περάστηκαν όλοι οι ταχυδρομικοί κώδικες και σε ποιες περιοχές αναφέρονται.

```

create or replace procedure IMPORT_SALE_CUBE is

PRODUCT_ID NUMBER(12);
STORE_ID NUMBER(12);
SALE_DATE DATE;
ITEMS NUMBER(3);
ZIP_CODE VARCHAR2(5);
YEAR VARCHAR2(5);
SALE_DAY DATE;
SALE_VALUE NUMBER(17,2);
BUY_VALUE NUMBER(17,2);
SALE_ID NUMBER(12);
errorNumber number;
errorMsg varchar2(200);
CURRENT_YEAR NUMBER;
DAY_ID NUMBER(5);

CURSOR SALES IS
SELECT S.SAL_ID, ADR.ST_TK, ST.ST_ID, PR.PR_ID,PR.PR_SALE_VALUE,
PR.PR_MARKET_VALUE, S.ITEMS, S.SALE_DATE
FROM SCENARIO.SALES S, SCENARIO.PRODUCTS PR,
SCENARIO.STORES ST, SCENARIO.ST_ADDRESSES ADR

```

```

WHERE S.PR_ID = PR.PR_ID
AND S.ST_ID = ST.ST_ID
AND ADR.ST_ID = ST.ST_ID
AND S.SAL_ID NOT IN (SELECT A.TEMP_SALE_ID FROM SALES_CUBE A)
ORDER BY SALE_DATE DESC;

begin

    IMPORT_PRODUCT_DIM;
    IMPORT_COMPANY_DIM;

    OPEN SALES;
    LOOP
        FETCH SALES INTO SALE_ID, ZIP_CODE,STORE_ID, PRODUCT_ID, SALE_VALUE,
        BUY_VALUE, ITEMS, SALE_DAY;
        EXIT WHEN SALES%NOTFOUND;

        CURRENT_YEAR := extract(YEAR from SALE_DAY);
        if (check_if_dayyear_exists(SALE_DAY) = 0) THEN
            IMPORT_TIME_DIM (CURRENT_YEAR);
        END IF;

        SELECT T.DAY_ID INTO DAY_ID FROM TIME T WHERE
        TO_DATE(T.DAY,'DD/MM/YYYY') = TO_DATE(SALE_DAY,'DD/MM/YYYY');

        INSERT INTO SALES_CUBE (ZIP_CODE, STORE_ID, PRODUCT_ID, DAY_ID,
        SALE_VALUE, ITEMS, TEMP_SALE_ID, BUY_VALUE)
        VALUES (ZIP_CODE, STORE_ID, PRODUCT_ID, DAY_ID, SALE_VALUE, ITEMS,
        SALE_ID, BUY_VALUE);

    END LOOP;

    CLOSE SALES;

    COMMIT;

    EXCEPTION WHEN OTHERS THEN
    ROLLBACK;
    errorMsg := SQLERRM;
    errorNumber := SQLCODE;
    INSERT INTO ERROR VALUES(errorNumber || ' - '
    || errorMsg
    || ' PWL=ID ='
    || SALE_ID
    || ' YPOK ='
    || STORE_ID
    || ' - '
    || ' PR='
    || PRODUCT_ID
    || ' DAY='
    || TO_CHAR(SALE_DAY,'DD/MM/YYYY'),SYSDATE);
    commit;

END;

```

Το τελευταίο SQL Script παρουσιάζει το τρόπο με τον οποίο γεμίζει με δεδομένα ο σχεσιακός πίνακας που θα περιέχει τα δεδομένα για τον cube SALES. Η δομή της διαδικασίας είναι να τρέξει πρώτα τις υπό – διαδικασίες που γεμίζουν τους πίνακες

PRODUCT και COMPANY και στη συνέχεια ελέγχει αν υπάρχει η ημερομηνία πώλησης του προϊόντος στον πίνακα TIME και τρέχει αν χρειάζεται τη διαδικασία IMPORT\_TIME\_DIM.

Όπως παρουσιάστηκε στη παράγραφο αυτή, η διαδικασία ETL για το συγκεκριμένο σενάριο εκτελείται σε απλά βήματα. Από τους βασικούς πίνακες οι δύο πίνακες REGION και TIME δε φορτώνονται κάθε φορά με δεδομένα. Συγκεκριμένα, ο πρώτος (REGION) αρχικοποιείται μια φορά ενώ αντίστοιχα ο πίνακας TIME ανανεώνεται με δεδομένα μόνο όταν χρειάζεται. Το τελευταίο βήμα που θα πρέπει να αναφερθεί είναι η δημιουργία αυτόματης διαδικασίας η οποία θα τρέχει την διαδικασία IMPORT\_SALES\_CUBE κάθε 30 μέρες.

```
begin
  sys.dbms_job.submit(job => :job,
    what => 'BEGIN
import_sale_cube;
end;',
    next_date => to_date('30-09-2009', 'dd-mm-yyyy'),
    interval => 'SYSDATE + 30');
  commit;
end;
```

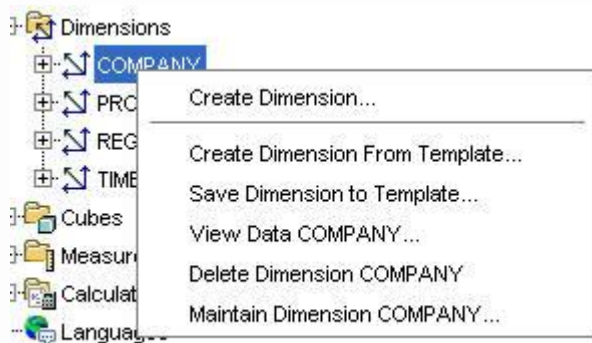
Με το παραπάνω SQL Script ολοκληρώνεται η παρουσίαση της διαδικασίας ETL. Η κεντρική διαδικασία εισαγωγής των δεδομένων εκτελέστηκε σε λιγότερο από μια ώρα και εισήγαγε 1000320 γραμμές στον βασικό πίνακα του cube Sales. Στην επόμενη παράγραφο θα παρουσιαστεί ο τρόπος με τον οποίο παράγονται οι αναφορές και πως μπορεί ο απλός χρήστης να τις χρησιμοποιήσει.

## 5.8. Παραγωγή αναφορών και αναλύσεων

Στη παράγραφο αυτή θα παρουσιαστεί με ποιον τρόπο οι χρήστες μπορούν να δουν και να επεξεργαστούν τα δεδομένα. Καθώς έχει ολοκληρωθεί ο φυσικός σχεδιασμός και η διαδικασία ETL, τα δεδομένα πλέον μπορούν να παρουσιαστούν με ποικίλους τρόπους. Οι τρόποι παρουσίασης των δεδομένων που θα χρησιμοποιηθούν στο σενάριο αυτό είναι οι παρακάτω:

- Με τη χρήση του Olap Analytic Workspace Manager and Worksheet
- Με τη χρήση του Oracle BI Publisher
- Και τέλος με τη χρήση του Oracle BI Publisher Desktop

Πριν δούμε τα δεδομένα στο πρώτο εργαλείο θα πρέπει να εκτελεστεί πρώτα ένα βήμα. Αφού συνδεθούμε στο Analytic Workspace Manager για κάθε dimension και cube που έχουμε δημιουργήσει θα πρέπει να κάνουμε δεξί κλικ και να επιλέξουμε Maintain Dimension (Εικόνα 36). Με αυτό το τρόπο φορτώνονται τα δεδομένα στο Analytic Workspace και μπορούμε στη συνέχεια να κάνουμε View Data. Αντίστοιχα, εκτελούμε αυτή τη διαδικασία και για τον cube SALES.



Εικόνα 36: Επιλογή Maintain Dimension

Μόλις ολοκληρωθεί η διαδικασία Maintain για τον cube SALES και επιλέξουμε να δούμε τα δεδομένα, έχουμε την επιλογή από το ίδιο το εργαλείο να περιηγηθούμε στα δεδομένα από το μεγαλύτερο σύνολο έως το μικρότερο υποσύνολο. Ένα χαρακτηριστικό παράδειγμα είναι η Εικόνα 37, όπου για το έτος 2007 φαίνεται ότι το κατάστημα της εταιρείας «Πλαίσιο» στην Αθήνα και συγκεκριμένα αυτό που βρίσκεται στην οδό Στουρνάρη (1) έχει κάνει σύνολο πωλήσεων 218.899€. Το ποσό αυτό αναλύεται μέσα στο έτος στο δεύτερο, τρίτο και τέταρτο τρίμηνο μετά αντίστοιχα ποσά 31.049€, 97.416€ και 90.434€. Μπορούμε να αναλύσουμε περαιτέρω τα αποτελέσματα. Για παράδειγμα, υπάρχει η δυνατότητα να δούμε μέσα στο δεύτερο τρίμηνο του 2007 ποιος μήνας είχε τις περισσότερες πωλήσεις και με πόσα τεμάχια και ποια κατηγορία προϊόντων για το κατάστημα που αναφέρθηκε στο προηγούμενο παράδειγμα (Εικόνα 38).

Measure Data Viewer

File

Στοιχεία σελίδας: Περιοχή: Ν. ΑΤΤΙΚΗΣ Προϊόν: Όλα τα προϊόντα

	Sum Sales				Sum Items	
	2007	Τρίμηνο 2	Τρίμηνο 3	Τρίμηνο 1	Τρίμηνο 4	2007
Πλαίσιο	2.348.616	386.291	994.977		967.348	729.845
Πλαίσιο - Αθήνα - Κέντρο - Στουρνάρη 1	218.899	31.049	97.416		90.434	67.200
Πλαίσιο - Αθήνα - Κέντρο - Στουρνάρη 2	196.398	35.435	79.877		81.086	63.507
Πλαίσιο - Αθήνα - Κέντρο - Σύνταγμα	208.526	30.752	88.861		88.914	63.414
Πλαίσιο - Αθήνα - ΒΠ - Ψυχικό	213.909	28.783	92.919		92.206	66.146
Πλαίσιο - Αθήνα - ΒΠ - Μεταμόρφωση	210.280	36.101	97.145		77.034	66.924
Πλαίσιο - Αθήνα - ΒΠ - Κηφισιά	226.048	40.159	95.068		90.820	65.930
Πλαίσιο - Αθήνα - ΒΠ - Μαρούσι	205.606	34.460	88.611		82.534	66.833
Πλαίσιο - Αθήνα - ΑΠ - Αγ. Παρασκευή	214.014	36.002	88.638		89.375	69.057
Πλαίσιο - Αθήνα - ΔΠ - Περιστερί	215.894	38.236	88.871		88.788	66.185
Πλαίσιο - Θεσσαλονίκη - Κέντρο						
Πλαίσιο - Θεσσαλονίκη - Καλαμαριά						
Πλαίσιο - Κρήτη - Ηράκλειο						
Πλαίσιο - Πάτρα						
Πλαίσιο - Αθήνα - ΝΠ - Καλλιθέα	216.486	34.061	88.402		94.023	65.742
Πλαίσιο - Αθήνα - ΝΠ - Πειραιάς	222.557	41.253	89.170		92.134	68.907
Κωισόβολος	2.936.619	447.730	1.280.291		1.208.597	924.207
Γερμανός	2.017.621	328.852	873.965		814.804	648.788
Mediamarkt	857.352	130.081	370.239		357.032	258.745
Expert	2.166.340	339.006	901.859		925.475	682.899
Expert						

Εικόνα 37: Αποτελέσματα με τη χρήση του Analytic Workspace and Manager για το έτος 2007

Measure Data Viewer

File

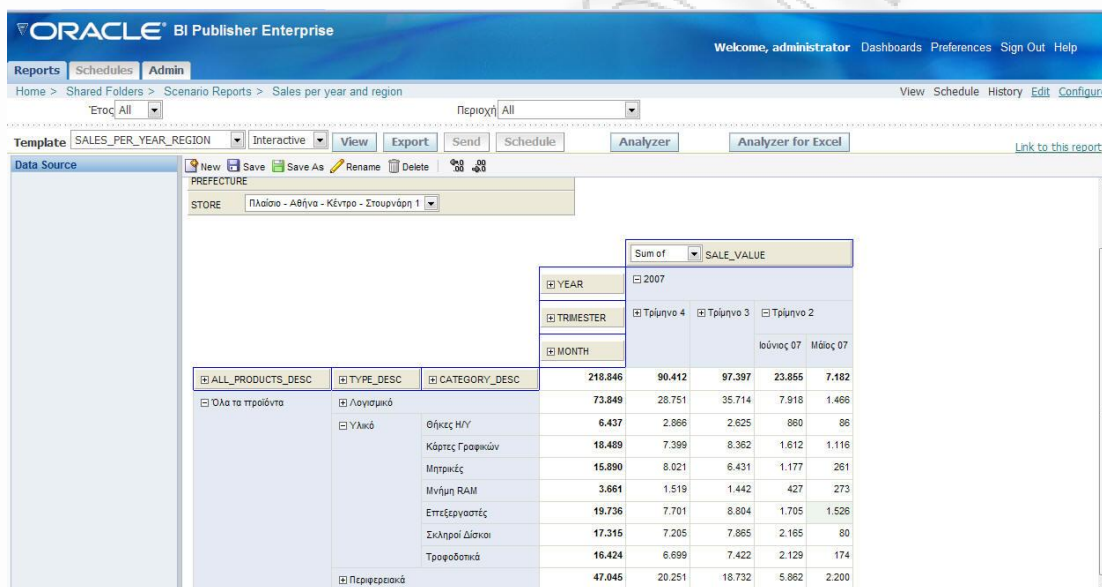
Στοιχεία σελίδας: Περιοχή: Ν. ΑΤΤΙΚΗΣ Εταιρεία: Πλαίσιο - Αθήνα - Κέντρο - Στουρνάρη 1

	Sum Sales			Sum Items	
	Τρίμηνο 2	Απρίλιος 07	Μάιος 07	Ιούνιος 07	Τρίμηνο 2
Όλα τα προϊόντα	31.049		7.187	23.862	10.971
Λογισμικό	9.388		1.468	7.920	2.140
Περιφερειακά	8.065		2.202	5.864	3.072
Υλικό	13.596		3.517	10.079	5.759
Τροφοδοτικά	2.304		174	2.130	1.188
Επεξεργαστές	3.231		1.526	1.705	1.195
Μητρικές	1.439		262	1.177	909
Μνήμη RAM	700		273	427	689
Κάρτες Γραφικών	2.729		1.116	1.613	510
Σκληροί Δίσκοι	2.246		80	2.166	760
Θήκες Η/Υ	946		86	860	508

Εικόνα 38: Αποτελέσματα για το δεύτερο τρίμηνο του έτους 2007 και το κατάστημα Πλαίσιο – Αθήνα – Κέντρο – Στουρνάρη 1 (Olap Analytical Workspace Manager)

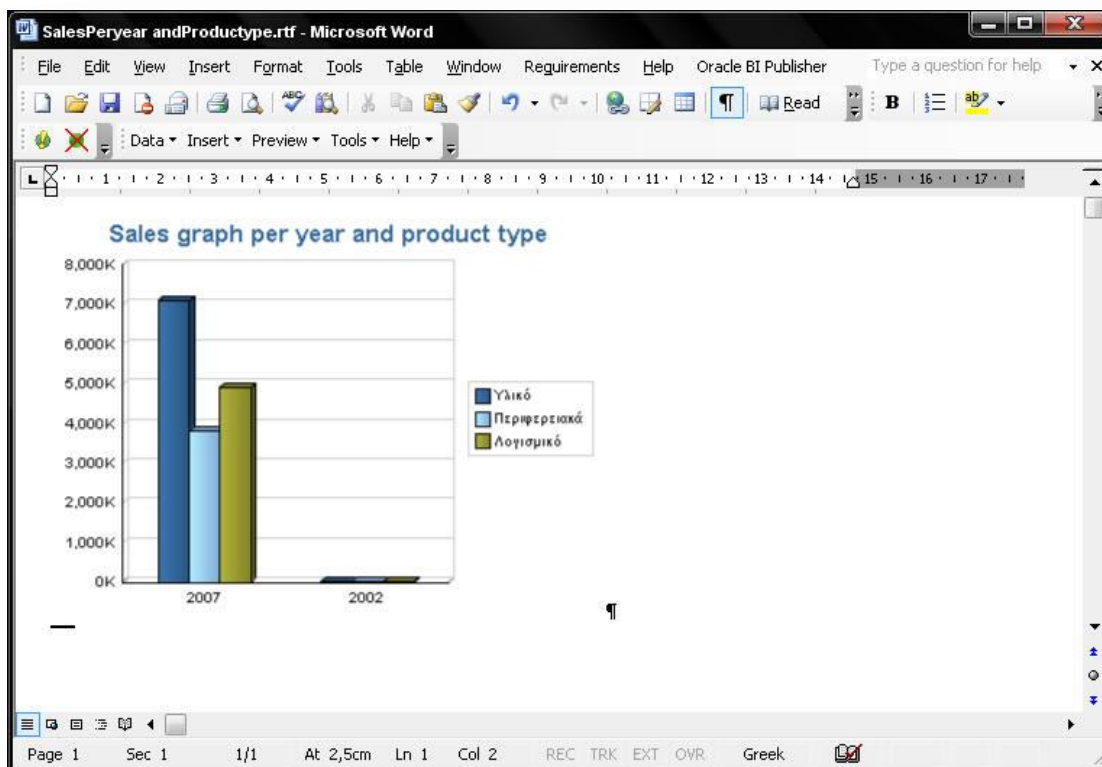
Όπως φαίνεται στην παραπάνω εικόνα ο Ιούνιος ήταν ο μήνας με τις περισσότερες πωλήσεις. Συγκεκριμένα το μεγαλύτερο ποσοστό κέρδους το είχε από τις πωλήσεις σκληρών δίσκων ενώ τα περισσότερα τεμάχια που πουλήθηκαν βρίσκονταν στη κατηγορία των επεξεργαστών (1195 τεμάχια).

Πιο φιλικά στο χρήστη αποτελέσματα παρουσιάζονται με τη χρήση του Oracle BI Publisher. Το συγκεκριμένο εργαλείο έχει τη δυνατότητα δημοσιοποίησης ή όχι μιας αναφοράς. Στην παρακάτω εικόνα παρουσιάζονται αντίστοιχα τα δεδομένα που φαίνονται στην Εικόνα 38 με τη χρήση του Oracle BI Publisher. Ο χρήστης έχει τη δυνατότητα να περιηγηθεί στα δεδομένα από τη πιο συνολική στη πιο εξειδικευμένη (drill – down) και να βγάλει τα απαραίτητα συμπεράσματα.



**Εικόνα 39: Αποτέλεσμα για το δεύτερο τρίμηνο του έτους 2007 και το κατάστημα Πλαίσιο – Αθήνα – Κέντρο – Στουρνάρη 1 (Oracle BI Publisher)**

Τέλος, η Oracle δίνει τη δυνατότητα να ενσωματώσει κάποιος τη δομή των δεδομένων σε έγγραφο MS Word. Με τη εγκατάσταση του Oracle BI Publisher εγκαθίσταται παράλληλα και ένα plug-in στο MS Word με το οποίο ο χρήστης μέσω του εγγράφου μπορεί να συνδεθεί στο Oracle BI Publisher και να δημιουργήσει αναφορές. Ένα παράδειγμα είναι η Εικόνα 40, στο οποίο έχει δημιουργηθεί ένα διάγραμμα (chart) που απεικονίζει τις πωλήσεις για τα είδη των προϊόντων μέσα στο χρόνο.



Εικόνα 40: Παράδειγμα αναφοράς με τη χρήση του Oracle BI Publisher και του MS Office

## 5.9. Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε η ο τρόπος με τον οποίο δημιουργήθηκε το Data Warehouse του σεναρίου. Η δημιουργία του Data Warehouse έγινε σε πέντε διαφορετικά βήματα. Τα βήματα αυτά αποτελούνται από την ανάλυση απαιτήσεων, τον λογικό σχεδιασμό, τον φυσικό σχεδιασμό, τη διαδικασία ETL και τέλος την παρουσίαση και τη δημιουργία των αναφορών.

Στην αρχή του κεφαλαίου, στη παράγραφο 5.2, περιγράφεται το σενάριο και τι ακριβώς θέλουμε να υλοποιήσουμε. Συγκεκριμένα, το σενάριο αναφέρεται σε μια εταιρεία η οποία προμηθεύει διάφορες ελληνικές εταιρείες με προϊόντα που αφορούν τους ηλεκτρονικούς υπολογιστές. Τα προϊόντα αυτά μπορούν να αναφέρονται είτε σε λογισμικό είτε σε υλικό είτε σε περιφερειακά. Τα στελέχη της εταιρείας (E) θέλουν να εξάγουν αναφορές και να απαντήσουν σε βασικά ερωτήματα με σκοπό να πάρουν αποφάσεις για την επιχειρηματική πορεία της εταιρείας και να δημιουργήσουν νέες στρατηγικές ανάπτυξης..

Το κεφάλαιο συνεχίζει με την ανάλυση της βάσης δεδομένων του αρχικού συστήματος της εταιρείας. Με αυτό το τρόπο ο αναγνώστης εισάγεται στις έννοιες και στις οντότητες που θα παρουσιαστούν στη συνέχεια. Στη παράγραφο 5.3, γίνεται η ανάλυση και η περιγραφή της σχεσιακής βάσης δεδομένων για την καλύτερη κατανοητή της δομής των δεδομένων.

Στην επόμενη παράγραφο 5.4, γίνεται η ανάλυση απαιτήσεων του Data Warehouse. Η ανάλυση απαιτήσεων εστίασε στις επιχειρησιακές απαιτήσεις και στις απαιτήσεις των χρηστών (Ανάλυση Απαιτήσεων). Παρουσιάζονται ερωτήματα που πρέπει να καλυφθούν με την δημιουργία του Data Warehouse καθώς επίσης και οι απαιτήσεις των χρηστών όπου καθορίζονται ποια είναι τα είδη χρηστών και ποιες είναι οι απαιτήσεις τους όσον αφορά το την εύκολη διαχείριση και χρήση του Data Warehouse.

Στη συνέχεια του κεφαλαίου, στην παράγραφο 5.5, παρουσιάζεται ο λογικός σχεδιασμός του Data Warehouse. Με βάση την ανάλυση απαιτήσεων εντοπίζονται οι οντότητες που θα πρέπει να δημιουργηθούν και ορίζονται τα χαρακτηριστικά τους. Ορίζεται το διαστατικό μοντέλο που θα χρησιμοποιηθεί, ποιες dimensions, levels, attributes cube θα δημιουργηθούν ώστε να καλύψουν τις παραπάνω απαιτήσεις.

Ακολουθεί, στη παράγραφο 5.6, η διαδικασία υλοποίησης των παραπάνω αντικειμένων που περιγράφηκαν στην παράγραφο 5.5. Παρουσιάζονται τα SQL Scripts που εκτελέστηκαν τόσο για τη δημιουργία του χρήστη της βάσης δεδομένων όσο και ο τρόπος υλοποίησης των dimensions και του cube με τη χρήση του Olap Analytic Workspace Manager and Worksheet (παράγραφος 3.3)

Στη παράγραφο 5.7 παρουσιάζεται η διαδικασία ETL (Παράγραφος 2.4) με την οποία εισάγονται τα δεδομένα από το αρχικό σύστημα στη σχεσιακή βάση του Data Warehouse. Για την υλοποίηση, της διαδικασίας χρησιμοποιήθηκαν SQL Scripts και διαδικασίες της Oracle (Stored procedures) τα οποία παρουσιάζονται. Αναφέρεται επίσης, η δομή της κεντρικής διαδικασίας και τα βήματα που εκτελούνται.

Τέλος, στη παράγραφο 5.8, παρουσιάζεται ο τρόπος με τον οποίο μπορούν οι χρήστες να επεξεργαστούν τα δεδομένα. Με τη χρήση του Oracle Business Intelligence (παράγραφος 3.5) δημιουργούνται οι απαιτούμενες αναφορές. Παρουσιάζεται επίσης



και παραδείγματα συνεργασία του παραπάνω εργαλείου με το Microsoft Office και συγκεκριμένα με τη δημιουργία αναφορών τόσο σαν Word αρχείο όσο και σαν Excel

Στο επόμενο κεφάλαιο παρουσιάζονται συγκεντρωτικά τα συμπεράσματα από τη παραπάνω υλοποίηση.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

## Κεφάλαιο 6

### 6. Συμπεράσματα

Στην εργασία αυτή παρουσιάσαμε τεχνολογίες Data Warehouse οι οποίες καλύπτουν την ανάγκη διαχείρισης δεδομένων σε μεγάλα πληροφοριακά συστήματα με πολύπλοκα και μεγάλα σε όγκο δεδομένα. Είδαμε πως μπορεί να σχεδιαστεί και να υλοποιηθεί ένα σύστημα Data Warehouse, ποια βήματα πρέπει να εκτελούνται και με ποιο τρόπο μπορεί ο χρήστης να εκμεταλλευτεί τα δεδομένα που δημιουργούνται. Αναφερθήκαμε στους νέους όρους που παρουσιάζονται όπως η τεχνολογία OLAP - Online Analytical Processing και η διαδικασία ETL - Extract, Transformation Loading. Είδαμε με πιο διεξοδικό τρόπο ένα παράδειγμα βασισμένο στην Oracle 10G μια από τις πιο ισχυρές και ευρέως γνωστές βάσεις δεδομένων.

Η επιλογή της συγκεκριμένης βάσης δεδομένων έγινε καθώς η Oracle είναι μια από τις πιο διαδεδομένες βάσεις δεδομένων για μεγάλα πληροφοριακά συστήματα. Επίσης, σε αντίθεση με τον MS SQL Server έχει τη δυνατότητα να εγκατασταθεί και σε λειτουργικό σύστημα Microsoft Windows αλλά και σε λειτουργικό σύστημα της οικογένειας Unix. Η δυνατότητα αυτή είναι ένα θετικό χαρακτηριστικό και αναφέρεται γιατί τα Unix / Linux έχουν καλύτερη διαχείριση μνήμης από τα Windows και σε ένα σύστημα Data Warehouse αυτό το στοιχείο απαιτείται.

Επίσης, η Oracle παρέχει πλήθος εργαλείων που χρησιμοποιούνται για την υλοποίηση ενός Data Warehouse. Ένα από τα βασικότερα είναι το Oracle Warehouse Builder με το οποίο μπορεί να γίνει ολόκληρη η διαχείριση του κύκλου ζωής του Data Warehouse και το Oracle Business Intelligence Suite το οποίο είναι μια πλήρης πλατφόρμα που σκοπό έχει να υποστηρίξει την επιχειρηματική ευφυΐα. Το μειονέκτημα με τα παραπάνω εργαλεία είναι ότι δεν είναι εύχρηστα για απλούς διαχειριστές βάσεων δεδομένων. Με αποτέλεσμα ένας οργανισμός μικρού / μεσαίου μεγέθους να μην τον συμφέρει να χρησιμοποιήσει την Oracle για τη δημιουργία του Data Warehouse καθώς θα πρέπει να επενδύσει σε χρόνο και κόστος για την εκπαίδευση του προσωπικού και για τα ίδια τα εργαλεία. Αυτό έρχεται βέβαια σε αντιπαράθεση με άλλες βάσεις δεδομένων, όπως η MySQL, όπου δυναμικά αναπτύσσεται στο τομέα αυτό και υποστηρίζει ότι μπορεί να καλύψει τη δημιουργία ενός Data Warehouse και με πολύ μικρό κόστος [33].

Γενικά, αντιμετωπίστηκαν πολλά προβλήματα για τη δημιουργία του Data Warehouse καθώς τόσο η βάση δεδομένων όσο και τα εργαλεία είχαν εγκατασταθεί σε ένα κοινό μηχάνημα με αποτέλεσμα τόσο η μνήμη όσο και ο επεξεργαστής να μην επαρκούν για τη παράλληλη εκτέλεση των διαφόρων προγραμμάτων. Επίσης, η Oracle θα έπρεπε να παρέχει ένα εργαλείο μόνο για τη δημιουργία ολόκληρου του Data Warehouse, όπως το Oracle Warehouse Builder. Γιατί, ο διαχειριστής μπορεί να καταναλώσει πολύτιμο χρόνο στο να συνδυάσει όλα τα εργαλεία που παρέχει η Oracle.

Συμπερασματικά, με τη βοήθεια του Data Warehouse και της τεχνολογίας Data Mining οι χρήστες μπορούν εύκολα τόσο να εξορύξουν διάφορα βασικά δεδομένα όσο και να πάρουν βασικές αποφάσεις και στρατηγικές για τις επιχειρήσεις τους. Δυνατότητα που με αρκετά δύσκολο τρόπο παρέχεται στα συνήθη συστήματα βάσεων δεδομένων. Στις παραδοσιακές υλοποιήσεις συστημάτων υποστήριξης αποφάσεων ο αναλυτής – προγραμματιστής θα πρέπει να γνωρίζει εκ των προτέρων τις σημαντικές διαστάσεις στα δεδομένα τους και να σχεδιάσει κατάλληλα την βάση δεδομένων όσο και επιπλέον προγράμματα για να ανακτή τα συγκεκριμένα στοιχεία. Με το Data Warehouse είναι δυνατή η ανακάλυψη νέων διαστάσεων στα δεδομένα στην πορεία χωρίς να απαιτείται εκ νέου ανακατασκευή και επαναπρογραμματισμός των συστημάτων δεδομένων.

Η βάση δεδομένων που χρησιμοποιεί ένα σύστημα Data Warehouse δεν περιέχει ωστόσο πρωτογενή δεδομένα. Αντίθετα, γεμίζει από άλλα συστήματα αφού περάσουν από επεξεργασία, ανάλυση και μετατροπή. Η διαδικασία αυτή ονομάζεται ETL και είναι στην ουσία η τεχνολογία με την οποία φορτώνονται τα δεδομένα στη βάση του Data Warehouse. Η βάση δεδομένων δεν είναι σχεδιασμένη βασισμένη στο Σχεσιακό μοντέλο αλλά σχεδιάζεται ώστε να ακολουθεί τους κανόνες της τεχνολογίας OLAP. Αυτό δημιουργεί μια οικειότητα στα δύο επίπεδα τεχνολογίας που κάνει εύκολη την συντήρηση και διαχείριση. Επίσης δεν απαιτεί υφιστάμενα συστήματα να καταργηθούν για να λειτουργήσει το νέο Data Warehouse σύστημα.

Σε πολλές περιπτώσεις τα δεδομένα που βρίσκονται στη βάση του Data Warehouse είναι πλέον κατανοητά και εύκολα διαχειριζόμενα από χρήστες που δεν έχουν γνώσεις βάσεων δεδομένων καθώς περιγράφουν έννοιες του πραγματικού κόσμου.

Στα υπάρχοντα παραδοσιακά συστήματα μόνο πολύ προχωρημένοι χρήστες έχουν την δυνατότητα να πραγματοποιούν αναζητήσεις (queries) σε μια βάση και συνήθως απαιτείται development από προγραμματιστές και διαχειριστές βάσεων δεδομένων για να αποκτήσουν πρόσβαση στα δεδομένα που θέλουν.

Παρουσιάσαμε επίσης την μεθοδολογία με την οποία δημιουργείται ένα Data Warehouse και την χρησιμοποιήσαμε στο παράδειγμα χρήσης (Κεφάλαιο 5). Δεν διαφέρει ιδιαίτερα με τις υπόλοιπες μεθοδολογίες υλοποίησης συστημάτων με ειδοποιό διαφορά την εκτέλεση της διαδικασίας ETL (Παράγραφος 2.4). Με την χρήση διαφόρων εργαλείων διευκολύνεται η υλοποίηση. Στη παράγραφο 4.6, παρουσιάστηκε το Oracle Business Intelligence Suite(BI) το οποίο χρησιμοποιήθηκε στην εργασία αυτή. Με τη συγκεκριμένη πλατφόρμα της Oracle, ο χρήστης ανάλογα με τα δικαιώματα του μπορεί να δημιουργήσει, να δημοσιεύσει, να επεξεργαστεί και να προγραμματίσει για δημοσίευση αναφορές οι οποίες θα καλύπτουν τους τελικούς στόχους της εταιρείας/ του οργανισμού. Αυτή η δυνατότητα λείπει από παραδοσιακά συστήματα λήψης αποφάσεων όπου οι χρήστες έχουν ένα διακριτό ρόλο από τους προγραμματιστές-αναλυτές.

Παρά τις δυνατότητες του συστήματος Data Warehouse και των εργαλείων του ακόμα και από τους τελικούς χρήστες, είδαμε κατά την διαδικασία υλοποίησης των αντικειμένων και των SQL Scripts ότι χρειάζεται η συνεργασία προγραμματιστών και χρηστών.

Κλείνοντας την εργασία αυτή πιστεύουμε ότι τα συστήματα Data Warehouse θα παίξουν ένα πολύ σημαντικότερο ρόλο στο μέλλον και θα αυξηθεί το πεδίο των εφαρμογών τους στις Επιχειρήσεις και αλλού. Καθώς το οικονομικό περιβάλλον τόσο για Εταιρίες όσο και για τα ίδια τα κράτη επιβάλλουν την εξεύρεση καλύτερων τρόπων κατανόησης των δεδομένων, σε όποια συστήματα και να βρίσκονται, για την επίτευξη καλύτερου στρατηγικού πλεονεκτήματος, αύξησης της αποτελεσματικότητας και βελτίωση των παρεχομένων υπηρεσιών και προϊόντων. Τα παραδοσιακά συστήματα βάσεων δεδομένων δεν πρόκειται να καταργηθούν και αναμένεται να συνεχίσουν να είναι η ραχοκοκαλιά των πληροφοριακών συστημάτων.

Είναι η προσωπική μας άποψη ότι τα δύο συστημάτων θα αναπτύξουν κοντινότερους ακόμα δεσμούς και interfaces με αποτέλεσμα την διαφανή (transparent) διαχείριση από τον τελικό χρήστη. Στο μέλλον ένας τελικός χρήστης θα μπορεί όχι μόνο να φτιάξει τις αναφορές που θέλει αλλά και να αποκτήσει πρόσβαση και να αναζητήσει τα δεδομένα που χρειάζεται όπου και να βρίσκονται αυτά.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

## 7. Αναφορές

- [1] Surajit Chaudhuri, Umeshwar Dayal, “An Overview of Data Warehousing and OLAP Technology”, SIGMOD Record 26(1): 65-74, 1997
- [2] Oracle White Paper, “ETL Processing within Oracle9i”, Oracle, June 2001
- [3] Oracle, “Data Warehousing Guide 10g Release 2 (10.2), December 2005
- [4] <http://www.wikipedia.com>
- [5] <http://www.informationbuilders.com>
- [6] <http://www.oracle.com>
- [7] Inmon, W.H. Tech Topic: What is a Data Warehouse? Prism Solutions. Volume 1. 1995.
- [8] Deepak Pareek (2007). Business Intelligence for Telecommunications (in English). CRC Press, 294 pp. Retrieved on 2008-03-18.
- [9] OLAP Council White Paper (English). OLAP Council (1997). Retrieved on 2008-03-18.
- [10] Hari Mailvaganam (2007). Introduction to OLAP - Slice, Dice and Drill! (English). Data Warehousing Review. Retrieved on 2008-03-18.
- [11] Owen Kaser and Daniel Lemire, Attribute Value Reordering for Efficient Hybrid OLAP, Information Sciences, Volume 176, Issue 16, pages 2279-2438, 2006.
- [12] Χάλαρης Γεώργιος. “Data warehouse για χωρικά δεδομένα”
- [13] Ericsson, R. "Building Business Intelligence Applications with .NET". 1st Ed. Charles River Media. February 2004. pp. 28-29.
- [14] Kimball, R. and Ross, M. "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling". pp. 310. Wiley. 2nd Ed. 2002. ISBN 0-471-20024-7.

- [15] Xingquan Zhu, Ian Davidson (2007). *Knowledge Discovery and Data Mining: Challenges and Realities*. Hershey, New Your, pp 18. ISBN 978-159904252-7.
- [16] Joseph M. Firestone (1997). Data Mining and KDD: A Shifting Mosaic, March 12 1997
- [17] Maria S. Perez\_, Alberto S´anchez, V´ictor Robles, Pilar Herrero, Jos´e M. Pena. Design and implementation of a data mining grid-aware architecture, 15 June 2006
- [18] Panos Vassiliadis, Alkis Simitsis, Panos Georgantas, Manolis Terrovitis Spiros Skiadopoulos, A generic and customizable framework for the design of ETL scenarios, 2004 Elsevier
- [19] Thomas Thalhammer, Michael Scherfl, Mukesh Mohania, Active data warehouses: complemeting OLAP with analysis rules, 24 July 2001, Elsevier
- [20] E.F. Codd. “A Relational Model of Data for Large Shared Data Banks” CACM13, No. 6 (June 1970)
- [21] C.J. Date, “An introduction to database systems”, Addison-Wesley Publishing Company, 1995
- [22] E.F Codd. “Relational Completeness of Data Base Sublanguages”, Englewood Cliffs, N.L: Prentice-Hall (1972)
- [23] A.L Huyet, “Optimization and analysis aid via data-mining for simulated production systems”, 21 July 2005
- [24] Joseph M. Firestone (1997), “Data Warehouse and Data Marts: A Dynamic View”, Executive Information Systems, March 27 1997
- [25] <http://www.infogoal.com/>
- [26] Robert M. Bruckner, Beate List, Josef Sciefer, “Developing Requirements for data Warehouse Systems with use case”, Seventh Americas Conference on Information Systems, 2001

[27] Oracle, “Oracle Warehouse Builder, User’s Guide Volume 1, 10G Release (10.2.0.2), November 2006

[28] Oracle, “Analytic Workspace Manager and Oracle OLAP 10g”, An Oracle White Paper, November 2004

[29] <http://www.dwinfocenter.org>

[30] Oracle, “Oracle, Business Intelligence, Concepts Guide 10g Release 2 (10.1.2.0.0)”, November 2004

[31] Oracle, “Oracle Business Intelligence Publisher, Report Designer's Guide Release 10.1.3.4”, August 2008

[32] <http://wiki.oracle.com>

[33] MySQL, “Enterprise Data Warehousing with MySQL”, Business White Paper, 2007



## **8. Appendix**

### **Συντομογραφίες – Λατινικές**

OLAP	On – line Analytical Processing
MOLAP	Multidimensional server – based OLAP
ROLAP	Relational OLAP
VTOLAP	Vertical Technology OLAP
OWB	Oracle Warehouse Builder
BI	Business Intelligence
RAC	Real Applications Clusters
CRM	Customer Relationship Management
ERP	Enterprise Resource Planing