Πανεπιστήμιο Πειραιώς - Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Ανάπτυξη Εφαρμογών για Φορητές Συσκευές στο Υπολογιστικό Νέφος HANA της SAP** **Mobile Application Development on SAP HANA Cloud Platform**
Ονοματεπώνυμο Φοιτητή	**Νικόλαος Κινόπουλος**
Πατρώνυμο	**Ηλίας**
Αριθμός Μητρώου	**ΜΠΠΛ / 12020**
Επιβλέπων	**Χρήστος Δουληγέρης, Καθηγητής**

Ημερομηνία Παράδοσης **Ιούλιος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Χρήστος Δουληγέρης Καθηγητής	Ιωάννης Θεοδωρίδης Καθηγητής	Κων/νος Πατσάκης Λέκτορας

Περίληψη

Η μεγάλη αύξηση του όγκου των δεδομένων των σύγχρονων πληροφοριακών συστημάτων, σε συνδυασμό με την απαίτηση για άμεση πρόσβαση σε αυτά από όλο και περισσότερους χρήστες και διαφορετικές συσκευές, δημιουργούν μεγάλες καθυστερήσεις που οφείλονται στη συμφόρηση που παρατηρείται στο επίπεδο της Βάσης Δεδομένων. Το SAP HANA απαντά σε αυτά τα προβλήματα, αλλάζοντας εντελώς την αρχιτεκτονική του συστήματος βάζοντας τη βάση δεδομένων στη μνήμη (In-Memory), αυξάνοντας κατακόρυφα την απόδοση των συστημάτων.

Αντικείμενο της παρούσας διατριβής είναι η ανάπτυξη εφαρμογών για κινητές συσκευές με βάση το SAP HANA Cloud Platform, την έκδοση του SAP HANA στο σύννεφο. Η διατριβή διακρίνεται σε 5 κεφάλαια:

- **Κεφάλαιο 1:** Εισαγωγή στο SAP HANA, και εξέταση των προβλημάτων στην απόδοση των πληροφοριακών συστημάτων, τα οποία αυτό επιλύει. Επίσης στο κεφάλαιο αυτό εξετάζεται η αρχιτεκτονική του SAP HANA.
- **Κεφάλαιο 2:** Γίνεται λεπτομερής αναφορά στις καινοτομίες του SAP HANA ως βάσης δεδομένων, και εξετάζονται αναλυτικά ο τρόπος αποθήκευσης των δεδομένων στη μνήμη, οι αλγόριθμοι συμπίεσης που χρησιμοποίει, οι δυνατότητες παράλληλης επεξεργασίας και τα αντικείμενα των RDBMS που πλέον καταργούνται.
- **Κεφάλαιο 3:** Εξετάζεται ο τρόπος ανάπτυξης λογισμικού στο SAP HANA. Εξετάζονται τα διάφορα μοντέλα ανάπτυξης, οι γλώσσες προγραμματισμού που χρησιμοποιούνται, και οι σύνδεση εφαρμογών για φορητές συσκευές στο SAP HANA.
- **Κεφάλαιο 4:** Αναλύονται οι λειτουργίες που προσφέρει το SAP HANA Cloud Platform, τα σενάρια ανάπτυξης εφαρμογών σε αυτό, και εξετάζονται οι δυνατότητες ενσωμάτωσης της υφιστάμενης μηχανογραφικής υποδομής σε αυτό.
- **Κεφάλαιο 5:** Περιγράφεται η υλοποίησης εφαρμογής πωλήσεων για Android συσκευές, με βάση το SAP HANA Cloud Platform. Αναλύεται η υλοποίηση του Backend στο SAP HANA Cloud Platform, του Frontend στο Android και η μεταξύ τους επικοινωνία.

Πίνακας περιεχομένων

Περίληψη 3

Κεφάλαιο 1 - Εισαγωγή 8

1.1 Τι είναι το SAP HANA 8

1.2 Το Πρόβλημα - Βάσεις Δεδομένων και Big Data 8

1.3 Η λύση – In Memory Database 10

1.4 SAP HANA Platform 11

1.5 Αρχιτεκτονική του SAP HANA 13

SAP HANA – Βάση Δεδομένων στη Μνήμη (In-Memory Database) 13

Αρχιτεκτονική της βάσης δεδομένων 14

Κεφάλαιο 2 - Καινοτομίες του SAP HANA 15

2.1 Αποθήκευση πίνακα ανά γραμμή ή στήλη (Row or Column Store) 15

Αποθήκευση ανά γραμμή 16

Αποθήκευση ανά στήλη 16

Αποθήκευση πίνακα στο SAP Hana 17

Επιλογή αποθήκευσης ανά γραμμή ή ανά στήλη 18

2.2 Συμπίεση δεδομένων 19

Dictionary Encoding 20

Prefix Encoding 23

Sparse Encoding 26

Run-length Encoding (RLE) 27

Cluster Encoding 29

Indirect Encoding 30

2.3 Κατάργηση Ευρετηρίων (Indexes) 33

2.4 Παράλληλη Επεξεργασία 33

2.5 Κατάργηση Συγκεντρωτικών Πινάκων (Aggregate Tables) 35

2.6 Βελτιστοποίηση Απόδοσης κατά την Εγγραφή 35

Main και Delta Storage 35

Delta Merge 36

Τύποι Delta Merge 37

Προσέγγιση Insert Only on Delta 37

2.7 Table Partitioning 39

Κεφάλαιο 3 - Ανάπτυξη Λογισμικού στο SAP HANA 41

3.1 Δημιουργία Persistence Model στο SAP HANA 41

Δημιουργία Persistence Model με χρήση σύνταξης Core Data Services (CDS) 42

Δημιουργία Persistence Model με χρήση σύνταξης HDBTable 44

3.2 Βελτιστοποίηση των εφαρμογών για το SAP HANA 46

SAP HANA XS 47

Βελτιστοποιήσεις στα ερωτήματα SQL 48

Μοντέλο Ανάπτυξης Code to Data 50

3.3 Δημιουργία Analytic Model στο SAP HANA 52

SAP HANA Information Views 52

3.4 Ανάπτυξη Διαδικασιών (Procedures) στο SAP HANA 56

SQLScript 56

3.5 Server Side JavaScript 58

Χρήση Server Side JavaScript στο SAP HANA XS 59

Παράδειγμα ανάπτυξης Server Side JavaScript 59

Θέματα Ασφαλείας Server Side JavaScript 62

3.6 Σύνδεση εφαρμογών με το SAP HANA 70

OData 70

Σύνδεση μέσω οδηγών JDBC / ODBC 77

3.7 Mobile Clients 79

Native Apps 80

Hybrid Apps – SAPUI5 80

Κεφάλαιο 4 - SAP HANA Cloud Platform 84

4.1 Περιγραφή 84

4.2 Σενάρια Ανάπτυξης Εφαρμογών 84

4.3 Μοντέλα Ανάπτυξης Εφαρμογών 85

4.4 Εργαλεία του SAP HANA Cloud Platform 86

SAP HANA Cloud Cockpit 86

SAP HANA Cloud Connector 87

SAP Web IDE 90

Git Repository 92

4.5 Ασφάλεια και Ταυτοποίηση Χρηστών 93

Identity Federation με χρήση Εταιρικού Identity Provider 94

Ταυτοποίηση χρηστών εφαρμογών SAP HANA Cloud με χρήση SAML 2.0 95

Ταυτοποίηση χρηστών τρίτων εφαρμογών με χρήση OAuth 2.0 97

Κεφάλαιο 5 - Υλοποίηση Εφαρμογής Android 99

5.1 Περιγραφή Εφαρμογής 99

5.2 Τεχνολογίες 100

SAP HANA Cloud 100

Android L 100

5.3 Υλοποίηση 103

Υλοποίηση στο SAP HANA Cloud _____ 103
Υλοποίηση στο Android _____ 114

Συμπεράσματα_____ 126

Βιβλιογραφία_____ 128

Κεφάλαιο 1 - Εισαγωγή

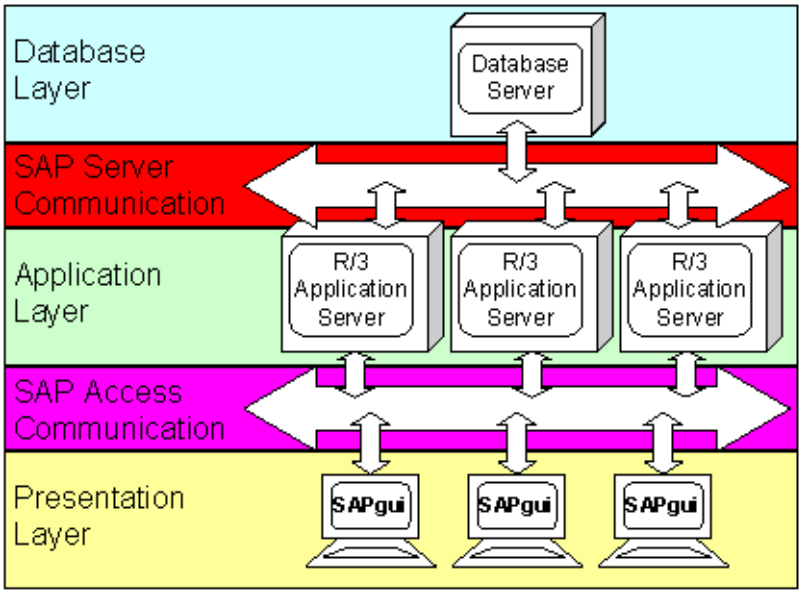
1.1 Τι είναι το SAP HANA

Το SAP HANA, είναι ένα ολοκληρωμένο σύστημα διαχείρισης δεδομένων, χαρακτηριστικό του οποίου είναι οι πολύ υψηλές επιδόσεις. Τεχνικά είναι ένα πλήρως λειτουργικό RDBMS το οποίο βρίσκεται εξ ολοκλήρου στην Κύρια Μνήμη του Υπολογιστικού Συστήματος (In-Memory Database). Εξαιτίας των πολύ υψηλών επιδόσεων είναι η ιδανική πλατφόρμα, για την ενοποίηση των OLTP και OLAP συστημάτων των οργανισμών, απλοποιώντας τη μηχανογραφική υποδομή και επιτρέποντας τη λήψη αναφορών από το σύστημα σε πραγματικό χρόνο.

Παρακάτω αναλύονται ποια είναι τα προβλήματα που υπάρχουν σήμερα λόγω του μεγάλου όγκου δεδομένων, και πως αυτά επιλύονται από το SAP HANA.

1.2 Το Πρόβλημα - Βάσεις Δεδομένων και Big Data

Όταν κυκλοφόρησε η έκδοση R/3 του SAP ERP το 1992, ήταν σχεδιασμένο ώστε να κάνει χρήση της τότε νέας αρχιτεκτονικής πελάτη – διακομιστή, όπου η εφαρμογή είχε 3 επίπεδα και μπορούσε να τρέχει σε πολλούς σχετικά φθηνούς Διακομιστές Εφαρμογών (Application Servers) οι οποίοι συνδέονταν σε ένα μεγάλο διακομιστή βάσης δεδομένων (Database Server).



Εικόνα 1. Αρχιτεκτονική του SAP R/3 [9]

Το μεγάλο πλεονέκτημα αυτής της αρχιτεκτονικής είναι ότι η αύξηση των χρηστών του συστήματος μπορεί να αντιμετωπιστεί απλά με προσθήκη επιπλέον διακομιστών εφαρμογών (Application Servers) που λειτουργούν παράλληλα. Ο διακομιστής της βάσης δεδομένων όμως παραμένει ένας και αυτό οδηγεί σταδιακά σε μεγάλη συμφόρηση. Ακόμα και όταν χρησιμοποιούνται τεχνικές clustering για την αύξηση των επιδόσεων της βάσης δεδομένων, η συμφόρηση παραμένει στην μεταφορά δεδομένων ανάμεσα στο επίπεδο των εφαρμογής και στο επίπεδο της βάσης δεδομένων. Αυτό έχει ως αποτέλεσμα τη μεγάλη πτώση της απόδοσης του συστήματος.

Οι συνεχώς αυξανόμενες αιτήσεις για δεδομένα ξεπερνούν τα όρια και των πλέον ισχυρών συστημάτων βάσεων δεδομένων. Το πρόβλημα δεν εντοπίζεται τόσο στην επεξεργαστική ισχύ των συστημάτων όσο στις ουρές εισόδου και εξόδου (I/O), καθώς τα συστήματα των βάσεων δεδομένων αγωνίζονται για να μεταφέρουν τα δεδομένα από και προς το δίσκο. Παρ' όλες τις τεχνικές βελτιστοποίησης που εφαρμόζονται, τελικά κάθε επιπλέον αίτηση προς τη βάση δεδομένων επιβραδύνει συνολικά ολόκληρο το σύστημα.

Το πρόβλημα γίνεται ακόμα πιο έντονο όταν ζητούνται αναφορές (Reports) από το σύστημα. Τα δεδομένα συναλλαγών (Transactional Data) στα συστήματα OLTP, αποτελούνται από μικρές σε μέγεθος εγγραφές που είναι αποθηκευμένες σε διάφορα σημεία της βάσης δεδομένων. Παράδειγμα συναλλαγής αποτελεί η πώληση εμπορευμάτων από κατάστημα λιανικής πώλησης, όπως μια αλυσίδα Super Market. Η ίδια η συναλλαγή πώλησης είναι σχετικά μικρή, καθώς αποτελείται από τα βασικά δεδομένα της συναλλαγής και τις αναλυτικές γραμμές πώλησης ανά προϊόν. Όταν όμως ζητηθεί από το σύστημα αναφορά η οποία για να εξαχθεί πρέπει να συλλέξει τις συναλλαγές πώλησης από όλα τα καταστήματα για ένα διάστημα αρκετών μηνών, τότε έχουμε έναν τεράστιο όγκο δεδομένων ο οποίος πρέπει να περάσει από τη βάση δεδομένων στους διακομιστές εφαρμογών, δημιουργώντας μεγάλη συμφόρηση και αντίστοιχα μεγάλες καθυστερήσεις.

Για να αντιμετωπιστεί το πρόβλημα των αναφορών, δημιουργήθηκαν τα συστήματα OLAP ένα από τα οποία είναι το SAP Business Warehouse, τα οποία δημιουργούν αντίγραφα των δεδομένων του συστήματος σε ξεχωριστό διακομιστή βάσης δεδομένων που χρησιμοποιείται μόνο για αναφορές, με σκοπό την απελευθέρωση πόρων του βασικού OLTP συστήματος.

Δυστυχώς, ακόμα και εάν το υλικό των διακομιστών γίνεται συνεχώς ταχύτερο και οικονομικότερο, η συμφόρηση όσον αφορά την πρόσβαση στα δεδομένα όχι μόνο δε βελτιώνεται, αλλά στην πραγματικότητα χειροτερεύει. Όσο περισσότερες διαδικασίες μηχανογραφούνται, τόσο μεγαλύτερη είναι η παραγωγή δεδομένων. Επειδή το σύστημα περιέχει επιπλέον πληροφορίες περισσότεροι χρήστες θέλουν πρόσβαση στα δεδομένα του, δημιουργώντας την ανάγκη νέων αναφορών. Προφανώς αφού αυξάνονται οι αναφορές που πρέπει να δημιουργεί το σύστημα, τα δεδομένα που πρέπει να μεταφέρονται από τη βάση δεδομένων αυξάνουν αντίστοιχα. Υπάρχει ένας φαύλος κύκλος ο οποίος μπορεί εύκολα να ξεφύγει από τον έλεγχο.

1.3 Η λύση – In Memory Database

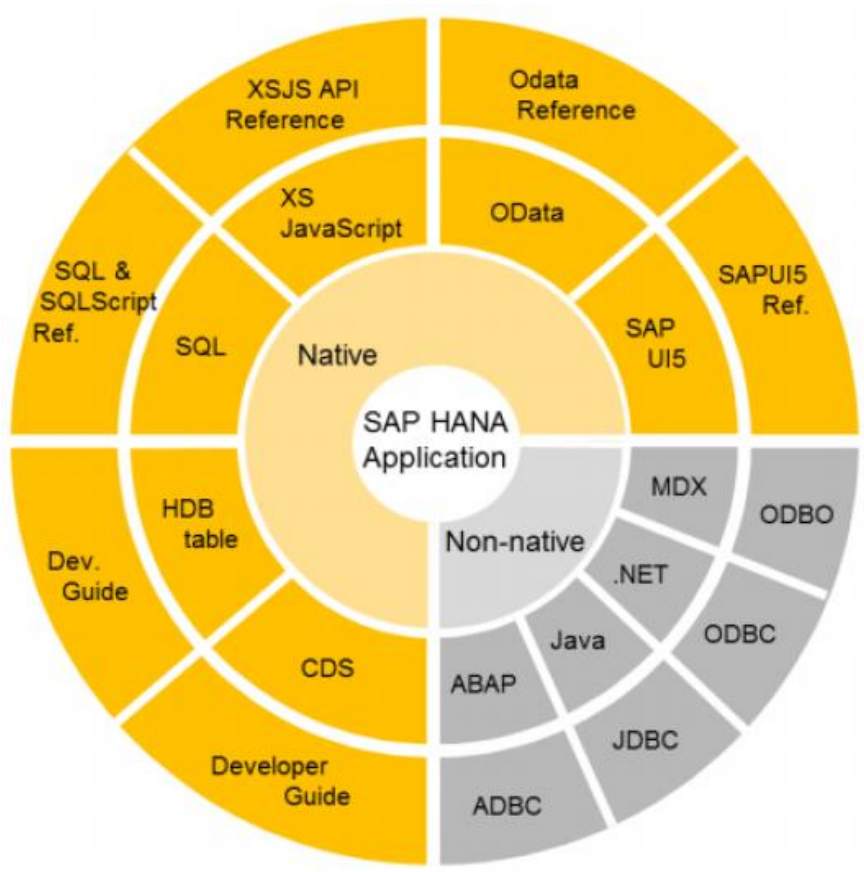
Αυτή που περιγράφηκε ήταν η πραγματικότητα την οποία αντιμετώπιζε η SAP με τα συστήματα των πελατών της, στις αρχές της δεκαετίας του 2000. Με την εξαγορά από την SAP των εταιρειών Sybase και Business Objects, αυξήθηκε κατακόρυφα η ανάγκη πρόσβασης στα δεδομένα τόσο σε transactional όσο και σε analytical από χρήστες εφαρμογών analytics και mobile. Παράλληλα αυξήθηκε η πρόσβαση

στα εταιρικά δεδομένα και από χρήστες εφαρμογών web. Ξαφνικά υπήρξε τεράστια αύξηση των αιτήσεων για άμεση πρόσβαση στα εταιρικά δεδομένα, πολύ μεγαλύτερη από αυτήν που άντεχαν τα εταιρικά συστήματα βάσεων δεδομένων.

Το 2004 η SAP ξεκίνησε διάφορα έργα τα οποία αφορούσαν την αλλαγή της αρχιτεκτονικής του πυρήνα των εταιρικών εφαρμογών, ώστε να αντιμετωπίσουν το πρόβλημα της συμφόρησης στην ανάκτηση των δεδομένων. Το αποτέλεσμα ήταν η δημιουργία του SAP HANA το 2011, ως το πρώτο και βασικότερο βήμα για τη μετάβαση των εταιρικών εφαρμογών και δεδομένων στη νέα In-Memory Αρχιτεκτονική. Το SAP HANA αλλάζει το μοντέλο εφαρμογών αφού η βάση δεδομένων πλέον μετατράπηκε από «βραδυκίνητο καράβι» που αποτελούσε πηγή καθυστερήσεων για την εφαρμογή, σε «αεριωθούμενο αεροσκάφος» το οποίο επιταχύνει κάθε πλευρά των επιχειρηματικών λειτουργιών [20].

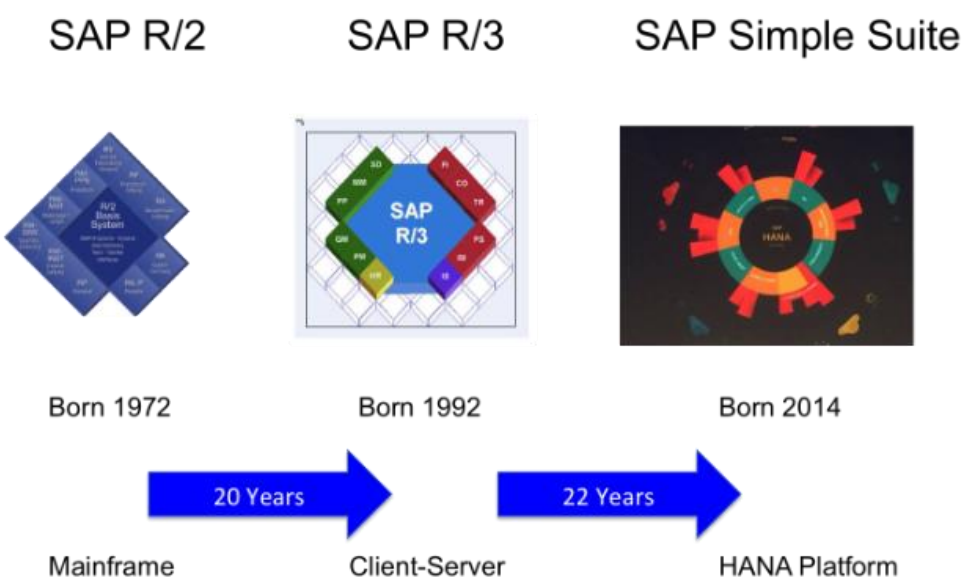
1.4 SAP HANA Platform

Το SAP HANA Platform αποτελεί την πλατφόρμα που συνδυάζει τη SAP HANA in Memory Database με μια πληθώρα τεχνολογιών και γλωσσών προγραμματισμού, προκειμένου να είναι δυνατή η μεταφορά των εφαρμογών μέσα στο επίπεδο της βάσης δεδομένων ώστε να είναι εφικτή η πλήρης εκμετάλλευση της απόδοσης της In-Memory βάσης δεδομένων.



Εικόνα 2. Οι γλώσσες προγραμματισμού του SAP HANA [36]

Το SAP HANA Platform προσφέρεται από την SAP ως αυτόνομο πακέτο, προκειμένου να χρησιμοποιηθεί ως βάση δεδομένων για τη δημιουργία νέων εφαρμογών και τη μεταφορά υπαρχόντων σε αυτό. Η SAP προσφέρει και την πλατφόρμα εφαρμογών της SAP R/3, η οποία πλέον ονομάζεται SAP Simple Suite και είναι βελτιστοποιημένη για τη λειτουργία με το SAP HANA Platform.



Εικόνα 3. Η εξέλιξη του SAP Suite [24]

Το SAP HANA Platform προσφέρεται ως πακέτο μαζί με πιστοποιημένο Hardware. Δηλαδή για να χρησιμοποιήσει κάποια εταιρεία το SAP HANA, πρέπει να αγοράσει πιστοποιημένη συσκευή από τους προτεινόμενους κατασκευαστές (Dell,HP,IBM κλπ). Εναλλακτικά, προσφέρεται το SAP HANA Cloud Platform για γρήγορη και άμεση εκκίνηση με το SAP HANA, με μηδενική υποδομή από την πλευρά της εταιρείας.

1.5 Αρχιτεκτονική του SAP HANA

Το SAP HANA είναι μια πλατφόρμα διαχείρισης δεδομένων στη μνήμη RAM (in-memory), η οποία μπορεί να αναπτυχθεί είτε στις εγκαταστάσεις του πελάτη (on-premise) είτε στο σύννεφο (on-cloud). Στον πυρήνα του είναι ένα καινοτόμο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων στην μνήμη (In-Memory RDBMS).

SAP HANA – Βάση Δεδομένων στη Μνήμη (In-Memory Database)

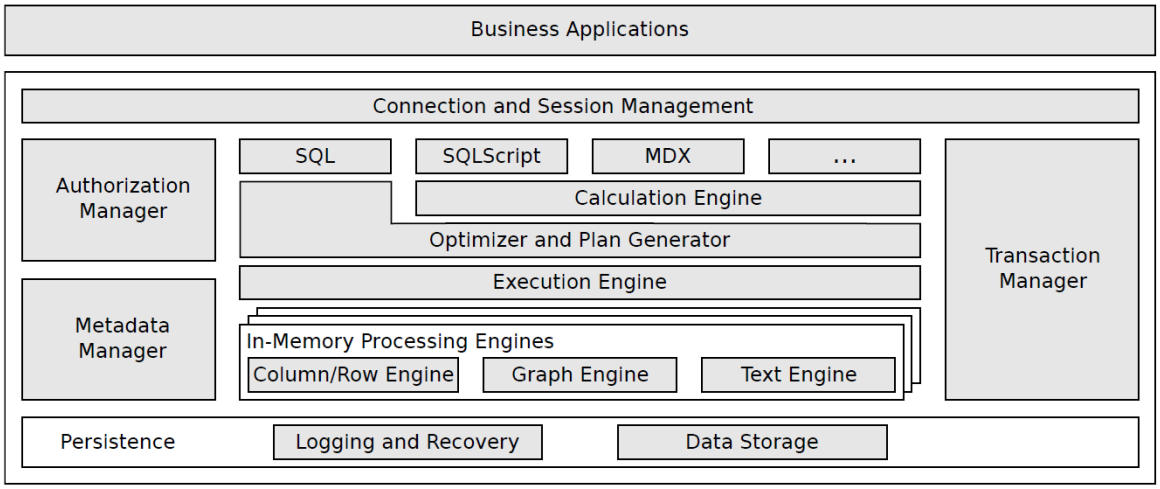
Τα υπολογιστικά συστήματα τα οποία εκτελούν το SAP HANA διαθέτουν πολλούς πολυπύρηνους επεξεργαστές με γρήγορη επικοινωνία μεταξύ των πυρήνων, και διαθέτουν πολλά Gigabytes ή

Terabytes μνήμης RAM. Τα δεδομένα είναι όλα φορτωμένα και διαθέσιμα στη μνήμη RAM, αποφεύγοντας έτσι την καθυστέρηση από την εγγραφή και ανάγνωση σε σκληρούς δίσκους.

Για την μόνιμη αποθήκευση των δεδομένων στην περίπτωση που το σύστημα τερματιστεί είτε προγραμματισμένα είτε λόγω διακοπής ρεύματος ή κάποιας άλλης καταστροφής, είναι απαραίτητη η χρήση σκληρών δίσκων τεχνολογίας Solid State. Η αντιγραφή των δεδομένων σε αυτούς τους δίσκους δεν επιβαρύνει την απόδοση του συστήματος καθώς εκτελείται ασύγχρονα ως μια διαδικασία παρασκηνίου (background task).

Αρχιτεκτονική της βάσης δεδομένων

Ένα SAP HANA σύστημα αποτελείται από πολλές διεργασίες που επικοινωνούν μεταξύ τους. Στο παρακάτω σχεδιάγραμμα φαίνονται οι υπηρεσίες του SAP HANA ως βάση δεδομένων σε ένα κλασικό περιβάλλον εφαρμογής.



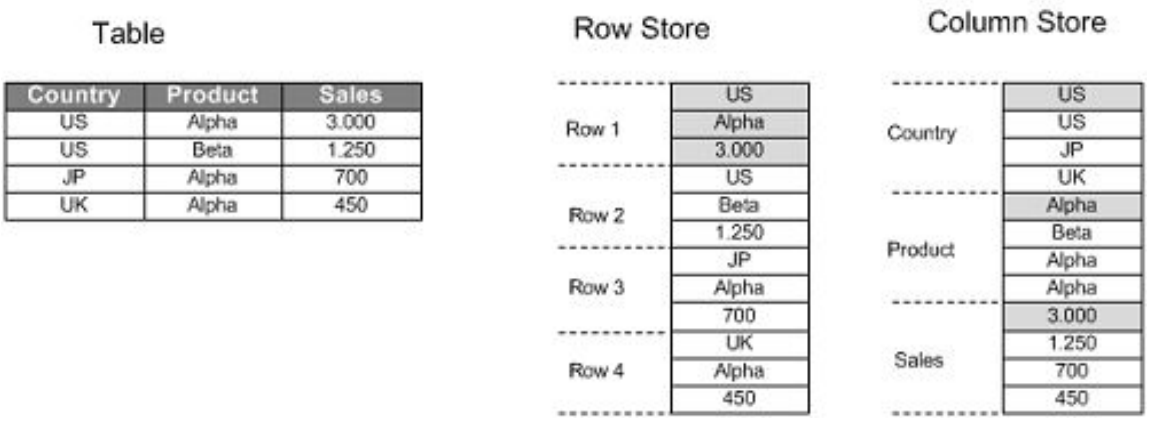
Εικόνα 4. Αρχιτεκτονική SAP HANA Βάσης Δεδομένων [11]

Κεφάλαιο 2 - Καινοτομίες του SAP HANA

Το SAP HANA ως βάση δεδομένων πέρα από την βασική του καινοτομία που είναι η βάση δεδομένων στη μνήμη, χρησιμοποιεί και μια σειρά από άλλες καινοτόμες τεχνικές. Αυτές είναι απαραίτητες γιατί πρέπει ταυτόχρονα να περιοριστεί το μέγεθος της βάσης δεδομένων ώστε να χωράει στον σχετικά περιορισμένο χώρο της μνήμης RAM, και ταυτόχρονα η ταχύτητα να μην επηρεάζεται από αυτό. Κάθε μια από τις τεχνικές έχει πλεονεκτήματα και μειονεκτήματα όμως ο συνδυασμός τους επιτρέπει την πλέον αποδοτική λειτουργία του συστήματος. Αυτές οι καινοτομίες εξετάζονται παρακάτω.

2.1 Αποθήκευση πίνακα ανά γραμμή ή στήλη (Row or Column Store)

Η βασική λογική μονάδα αποθήκευσης σε μια σχεσιακή βάση δεδομένων είναι ο δισδιάστατος πίνακας. Ο πίνακας αποτελείται από εγγραφές (γραμμές), οι οποίες αποτελούνται από πεδία (στήλες). Ο φυσικός τρόπος αποθήκευσης του πίνακα στη μνήμη γίνεται αποθηκεύοντας τις πληροφορίες είτε ανά γραμμή, είτε ανά στήλη.

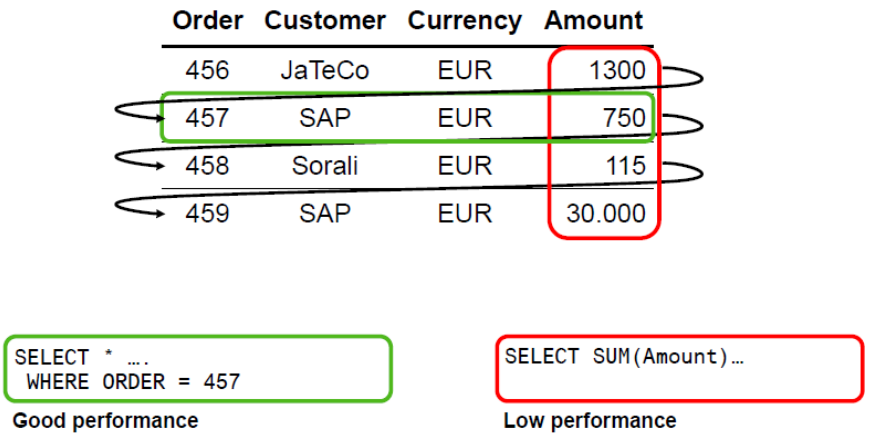


Εικόνα 5. Αποθήκευση ανά γραμμή ή στήλη [8]

Αποθήκευση ανά γραμμή

Ο κλασικός τρόπος της φυσικής αποθήκευσης στη μνήμη της πληροφορίας των πινάκων σε ένα παραδοσιακό σύστημα βάσης δεδομένων είναι ανά γραμμή. Αυτό συνεπάγεται ότι και το διάβασμα των δεδομένων του πίνακα γίνεται ανά γραμμή.

Στην παρακάτω εικόνα απεικονίζεται το διάβασμα ενός πίνακα που τα στοιχεία του είναι αποθηκευμένα ανά γραμμή. Πρώτα διαβάζεται η πληροφορία της γραμμής με κλειδί 456, έπειτα της γραμμής με κλειδί 457 κ.ο.κ. Αυτό δίνει μεγάλη ταχύτητα όταν εκτελείται αναζήτηση μιας γραμμής με βάση το κλειδί και ανακτούνται όλα τα πεδία της. Όμως η εκτέλεση συγκεντρωτικών συναρτήσεων (Aggregate Functions) στα δεδομένα μιας στήλης καθυστερεί καθώς πρέπει να ανακτηθούν όλες οι εγγραφές για να συγκεντρωθούν οι τιμές μιας στήλης.

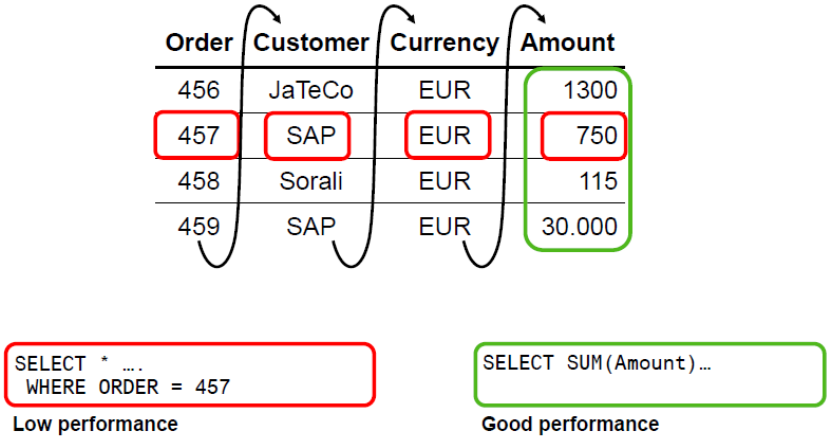


Εικόνα 6 Διάβασμα πίνακα ανά γραμμή [4]

Αποθήκευση ανά στήλη

Ένας εναλλακτικός τρόπος αποθήκευσης της πληροφορίας του πίνακα, είναι η αποθήκευση ανά στήλη. Αυτό στο παραπάνω παράδειγμα σημαίνει ότι πρώτα είναι αποθηκευμένη η πληροφορία της στήλης Order, έπειτα της στήλης Customer κ.ο.κ. Η ανάκτηση της πληροφορίας αντίστοιχα γίνεται ανά στήλη.

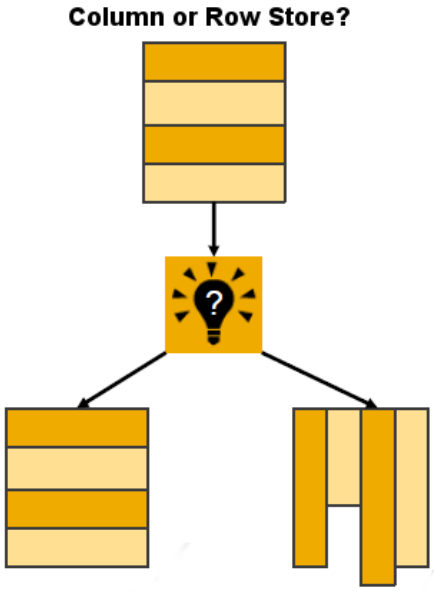
Αυτό δίνει πολύ μεγάλη ταχύτητα σε ανάκτηση πληροφορίας που βρίσκεται σε αποθηκευμένη σε στήλες όπως η εκτέλεση συγκεντρωτικών συναρτήσεων (Aggregate Functions). Ανάλογα όμως καθυστερεί ανάκτηση όλων των πεδίων μιας εγγραφής αφού πρέπει να γίνει η ανασύνθεσή της μέσα από το διάβασμα όλων των στηλών.



Εικόνα 7. Διάβασμα πίνακα ανά στήλη [4]

Αποθήκευση πίνακα στο SAP Hana

Το SAP Hana υποστηρίζει τον κλασικό τρόπο αποθήκευσης ανά γραμμή καθώς και τον νεώτερο ανά στήλη. Ο πίνακας σε λογικό επίπεδο διατηρεί την κλασσική του μορφή και λειτουργικότητα, ανεξάρτητα από τον φυσικό τρόπο αποθήκευσης ο οποίος είναι σε χαμηλότερο επίπεδο. Ο σχεδιαστής της βάσης πρέπει να αποφασίσει ποιος είναι ο καταλληλότερος τρόπος αποθήκευσης με βάση συγκεκριμένα κριτήρια.



Εικόνα 8. Ο προγραμματιστής πρέπει να επιλέξει αποθήκευση του πίνακα ανά στήλη ή ανά γραμμή [4]

Επιλογή αποθήκευσης ανά γραμμή ή ανά στήλη

Η αποθήκευση ανά στήλη είναι καλύτερη για πίνακες οι οποίοι πληρούν τα παρακάτω κριτήρια:

- Γίνονται συχνά λειτουργίες που αφορούν πληροφορία συγκεκριμένων στηλών σε μεγάλο πλήθος εγγραφών.
- Έχουν μεγάλο πλήθος στηλών οι οποίες δε χρησιμοποιούνται συχνά.
- Εκτελούνται συχνά σε αυτούς συγκεντρωτικές συναρτήσεις (Aggregate Functions).
- Εκτελούνται συχνά σε αυτούς λειτουργίες εντατικής αναζήτησης (Intensive Search).

Αντίστοιχα η αποθήκευση ανά γραμμή είναι καλύτερη για πίνακες οι οποίοι πληρούν τα παρακάτω κριτήρια:

- Οι στήλες έχουν κυρίως διακριτές τιμές, γεγονός το οποίο θα οδηγήσει σε χαμηλό βαθμό συμπίεσης του πίνακα.
- Σχετίζονται νοηματικά οι στήλες μεταξύ τους και αναμένεται σε κάθε ερώτημα η επιστροφή των περισσότερων ή όλων των πεδίων.

- Ο αριθμός των εγγραφών είναι μικρός.
- Δεν αναμένεται η συχνή εκτέλεση συγκεντρωτικών συναρτήσεων (Aggregate Functions).
- Δεν αναμένεται η συχνή εκτέλεση λειτουργιών εντατικής αναζήτησης (Intensive Search).

Πρέπει να σημειωθεί ότι η φυσική αποθήκευση του πίνακα ανά γραμμή ή ανά στήλη, στο SAP HANA αναφέρεται στην αναπαράσταση του πίνακα στη μνήμη RAM και όχι στη μόνιμη αποθήκευσή του στο σκληρό δίσκο.

Γενικά αν και το SAP HANA υποστηρίζει και τους δύο τρόπους αποθήκευσης, προτιμάται η αποθήκευση των πινάκων ανά στήλη, διότι το σύστημα είναι ειδικά βελτιστοποιημένο για αυτή. Η αποθήκευση ανά γραμμή γίνεται μόνο σε ειδικές περιπτώσεις όταν πληρούνται τα περισσότερα από τα παραπάνω κριτήρια.

2.2 Συμπίεση δεδομένων

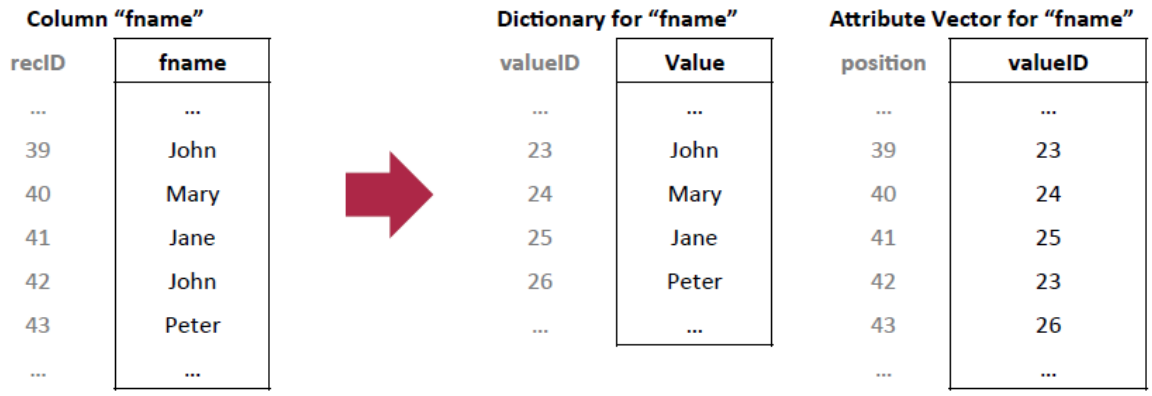
Η αποθήκευση των πινάκων ανά στήλη επιτρέπει την πολύ υψηλή συμπίεσή τους. Αυτό είναι απαραίτητο για να περιοριστεί το συνολικό κόστος της λύσης καθότι η βάση δεδομένων πρέπει να βρίσκεται ολόκληρη στην μνήμη RAM. Βέβαια η συμπίεση των δεδομένων δεν είναι αυτοσκοπός, για αυτό το SAP HANA χρησιμοποιεί τεχνικές συμπίεσης χαμηλής υπολογιστικής πολυπλοκότητας, ώστε να μην επηρεάζεται αρνητικά η συνολική επίδοση του συστήματος. Πολλές μάλιστα από αυτές, αυξάνουν τις επιδόσεις αφού επιταχύνουν τη λειτουργία της αναζήτησης καθώς και τους υπολογισμούς στους πίνακες.

Η προεπιλεγμένη τεχνική συμπίεσης που εφαρμόζεται σε όλες τις στήλες είναι η Dictionary Encoding. Κάθε στήλη μπορεί να συμπιεστεί περεταίρω χρησιμοποιώντας την λεγόμενη Προχωρημένη Συμπίεση (Advanced Compression). Αυτή περιλαμβάνει τη συμπίεση χρησιμοποιώντας τους αλγορίθμους prefix encoding, run length encoding (RLE), cluster encoding, sparse encoding, και indirect encoding. Το σύστημα αποφασίζει ποιος αλγόριθμος συμπίεσης είναι καλύτερος για τη συγκεκριμένη στήλη. Στα δεδομένα που είναι στο Delta Storage τα οποία είναι βελτιστοποιημένα για εγγραφή, εφαρμόζεται μόνο η Κωδικοποίηση Λεξικού.

Dictionary Encoding

Η τεχνική Dictionary Encoding είναι σχετικά απλή. Αυτό δε σημαίνει μόνο ότι είναι εύκολη στην κατανόηση, αλλά και απλή στην υλοποίηση η οποία δε χρειάζεται σύνθετες πολυεπίπεδες διαδικασίες οι οποίες θα επιβάρυναν την απόδοση.

Με το Dictionary Encoding δημιουργούνται πίνακες με μια στήλη που ονομάζονται λεξικά, και κάθε λεξικό περιέχει μοναδικά τις τιμές της στήλης του πίνακα που κωδικοποιεί. Έπειτα σε κάθε πεδίο της στήλης του λογικού πίνακα που κωδικοποιήθηκε τοποθετείται η τιμή του δείκτη στη θέση του λεξικού. Η τεχνική αυτή επαναλαμβάνεται για όλες τις στήλες του πίνακα. Επειδή για την τιμή της θέσης χρειάζονται ακέραιοι αριθμοί, υπάρχει μεγάλη εξοικονόμηση χώρου, ειδικά όταν οι αποθηκευμένες τιμές είναι τύπου String.



Εικόνα 9. Συμπίεση με χρήση Dictionary Encoding [1]

Έστω ότι υπάρχει ο παρακάτω πίνακας με μια εγγραφή για κάθε άτομο του παγκόσμιου πληθυσμού.

Attribute	# of Distinct Values	Size
first name	5 million	49 Byte
last name	8 million	50 Byte
gender	2	1 Byte
country	200	49 Byte
city	1 million	49 Byte
birthday	40 000	2 Byte
	Sum	200 Byte

Εικόνα 10 Ο πίνακας με τα στοιχεία του Παγκόσμιου Πληθυσμού [1]

Κάθε εγγραφή έχει μέγεθος 200 Bytes έχουμε 8x10^9 εγγραφές. Το συνολικό μέγεθος είναι περίπου 1.6 TB.

Με τα χρήση Dictionary Encoding θα δημιουργηθούν λεξικά για κάθε στήλη όπου θα εμφανίζουν μια φορά τις τιμές της στήλης και στη θέση των τιμών θα μπουν οι θέσεις στο λεξικό ως ακέραιοι αριθμοί.

Ξεκινώντας την κωδικοποίηση για τη στήλη first name θα πρέπει να υπολογίσουμε πόσα bits χρειάζονται για να απεικονιστεί το πλήθος των διαφορετικών τιμών.

$$Μέγεθος\ Κλειδιού = \lceil \log_2(5.000.000) \rceil\ bit = 23\ bit$$

Επομένως οι τιμές της πρώτης στήλης θα αντικατασταθούν από κλειδιά μήκους 23 bits. Άρα αντί για μέγεθος:

$$Αρχικό\ Μέγεθος\ Στήλης = Πλήθος\ Στοιχείων \cdot Μέγεθος\ Στοιχείου = 8 \cdot 10^9 \cdot 49\ Byte \approx 365{,}1\ GB$$

Το νέο μέγεθος θα είναι:

$$Τελικό\ Μέγεθος\ Στήλης = Πλήθος\ Στοιχείων \cdot Μέγεθος\ Κλειδιού = 8 \cdot 10^9 \cdot 23\ bit \approx 21{,}4\ GB$$

Σε αυτό πρέπει να προστεθεί και το μέγεθος του λεξικού το οποίο είναι:

$$Μέγεθος\ Λεξικού = Πλήθος\ Στοιχείων \cdot Μέγεθος\ Στοιχείου = 49 \cdot 5 \cdot 10^6\ Byte \approx 0{,}23\ GB$$

Επομένως το τελικό μέγεθος της συμπιεσμένης δομής είναι 21,63 GB και το ποσοστό συμπίεσης είναι 94%. Στα δεδομένα της στήλης υπάρχει απευθείας πρόσβαση.

Το τελικό μέγεθος του πίνακα εάν επαναληφθεί η διαδικασία για όλες τις στήλες είναι περίπου 92 GB, επομένως το ποσοστό συμπίεσης είναι 94%.

Column	Cardi-nality	Bits Needed	Item Size	Plain Size	Size with Dictionary (Dictionary + Column)	Compression Factor
First names	5 millions	23 bit	50 Byte	400GB	250MB + 23GB	≈ 17
Last names	8 millions	23 bit	50 Byte	400GB	400MB + 23GB	≈ 17
Gender	2	1 bit	1 Byte	8GB	2b + 1GB	≈ 8
City	1 million	20 bit	50 Byte	400GB	50MB + 20GB	≈ 20
Country	200	8 bit	47 Byte	376GB	9.4kB + 8GB	≈ 47
Birthday	40000	16 bit	2 Byte	16GB	80kB + 16GB	≈ 1
Totals			**200 Byte**	**≈ 1.6TB**	**≈ 92GB**	**≈ 17**

Εικόνα 11 Η τελική συμπίεση σε όλες τις στήλες του πίνακα [1]

Ο βαθμός συμπίεσης της στήλης εξαρτάται από τον τύπο των δεδομένων της. Επίσης είναι αντιστρόφως ανάλογος ενός μεγέθους που ονομάζεται εντροπία της στήλης. Η εντροπία εκφράζει το πόση πληροφορία περιέχεται σε μια στήλη και υπολογίζεται από τον παρακάτω τύπο

$$Entropy = \frac{column\ cardinality}{table\ cardinality}$$

όπου

- *Column cardinality* είναι το πλήθος των διακριτών τιμών μιας στήλης

- *Table cardinality* είναι το πλήθος των εγγραφών του πίνακα ή της στήλης

Το Dictionary Encoding λειτουργεί ευεργετικά και στην αύξηση των επιδόσεων. Το μέγεθος της βάσης δεδομένων μικραίνει και επομένως μικραίνουν και οι χρόνοι για την μεταφορά των δεδομένων από τη μνήμη στον επεξεργαστή.

Εφόσον τα λεξικά είναι ταξινομημένα, η αναζήτηση μιας τιμής επιταχύνεται από O(n) που είναι η πλήρης σάρωση του λεξικού, σε O(log(n)), γιατί πλέον μπορεί να εκτελεστεί δυαδική αναζήτηση. Βέβαια το να διατηρούνται ταξινομημένα τα ευρετήρια έχει ένα επιπλέον κόστος κατά την εισαγωγή ή μεταβολή των στοιχείων του λεξικού. Σε επόμενα κεφάλαια θα εξεταστεί λεπτομερώς πως το SAP HANA αντιμετωπίζει αυτό το θέμα.

Οι ενέργειες στα δεδομένα γίνονται πλέον ανάμεσα σε ακεραίους. Ειδικά σε πεδία τύπου String αυτό οδηγεί σε μεγάλη αύξηση των επιδόσεων. Ενέργειες όπως η COUNT και η NOT NULL μπορούν να εκτελεστούν απευθείας πάνω στη στήλη με τους ακεραίους, χωρίς να ανακτηθούν οι πραγματικές τιμές από το Λεξικό.

Βέβαια η αναπαράσταση του πίνακα από τη συμπιεσμένη με Dictionary Encoding μορφή έχει κάποια επιβάρυνση για τον επεξεργαστή, όσον αφορά την αναπαράσταση όλων των στηλών του πίνακα. Αυτό όμως δε φαίνεται να δημιουργεί ιδιαίτερο πρόβλημα γιατί η συμφόρηση πλέον παρατηρείται στην επικοινωνία του επεξεργαστή με την μνήμη και όχι στους υπολογισμούς. Το μεγαλύτερο πρόβλημα παρατηρείται στα ερωτήματα που ζητούν πολλές ή όλες τις στήλες του πίνακα, για αυτό προτείνεται στους προγραμματιστές ως βέλτιστη πρακτική να ζητούν μόνο τις απαραίτητες στήλες στα ερωτήματά τους.

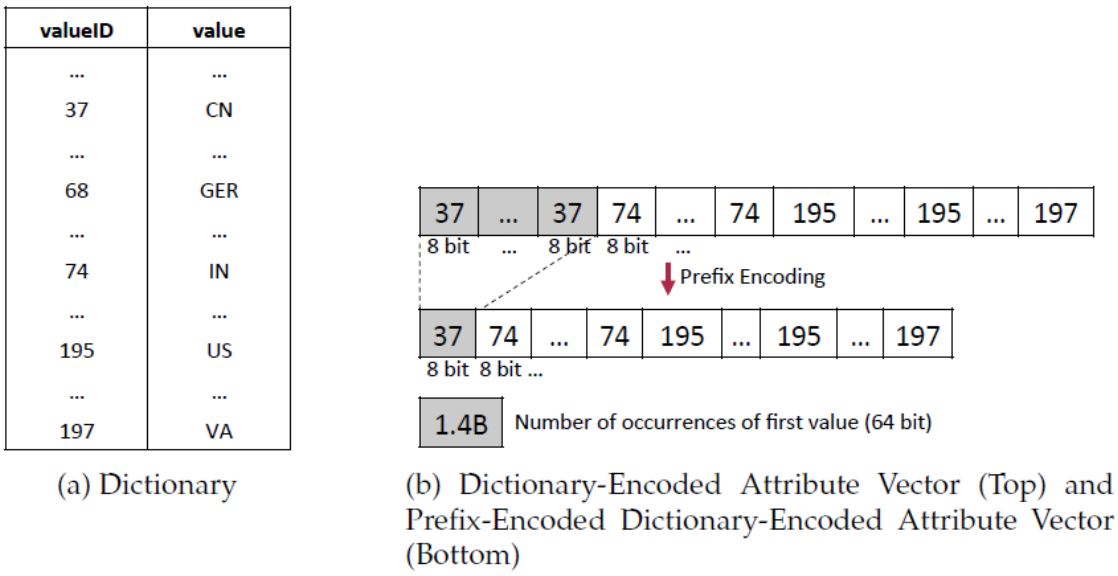
Το Dictionary Encoding είναι η βάση πάνω στην οποία το SAP HANA θα εκτελέσει κάποια από τις υπόλοιπες τεχνικές συμπίεσης όπως θα εξεταστεί παρακάτω.

Prefix Encoding

Σε πραγματικά παραδείγματα βάσεων δεδομένων, συχνά υπάρχει μια στήλη στην οποία κυριαρχεί μια τιμή. Στην συμπίεση Prefix Encoding, η τιμή που κυριαρχεί δεν πρέπει να αποθηκεύεται πολλές φορές. Για να επιτευχθεί αυτό η στήλη ταξινομείται ώστε η τιμή που κυριαρχεί να βρεθεί στην αρχή της λίστας.

Έπειτα η τιμή αποθηκεύεται μια φορά και αντίστοιχα αποθηκεύεται και το πλήθος των επαναλήψεων. Η συμπιεσμένη μορφή περιέχει τα εξής:

- Το πλήθος των επαναλήψεων της κυρίαρχης τιμής
- Το κλειδί της τιμής από το λεξικό
- Τα κλειδιά των τιμών των υπόλοιπων τιμών



(a) Dictionary

(b) Dictionary-Encoded Attribute Vector (Top) and Prefix-Encoded Dictionary-Encoded Attribute Vector (Bottom)

Εικόνα 12. Prefix Encoding [1]

Παράδειγμα είναι μια στήλη στην οποία καταχωρείται η χώρα σε ένα πίνακα που αποθηκεύει τον παγκόσμιο πληθυσμό ταξινομημένο με φθίνουσα ταξινόμηση, ως προς το πληθυσμό κάθε χώρας. Έτσι οι 1,4 δις Κινέζοι πολίτες εμφανίζονται πρώτοι, έπειτα οι Ινδοί κ.ο.κ. Οπότε το κλειδί 37 για την τιμή CN στο λεξικό, εμφανίζεται 1,4 δις φορές στην αρχή της ασυμπίεστης στήλης.

Στη συμπιεσμένη μορφή θα αποθηκευθεί η τιμή 37 που είναι ο κωδικός, η τιμή 1,4 δις που είναι το πλήθος των επαναλήψεων και οι υπόλοιπες τιμές της στήλης θα παραμείνουν ως έχουν και στην ασυμπίεστη μορφή. Ο βαθμός συμπίεσης μπορεί να υπολογιστεί ως εξής:

Το μέγεθος του κλειδιού που απαιτείται για να κωδικοποιηθούν οι τιμές 200 χωρών:

$$Μέγεθος\ Κλειδιού = \log_2(200)\ bit = 8\ bit$$

Εφόσον ο πίνακας έχει 8 x 10^9 εγγραφές, χρειαζόμαστε 8bit για το κάθε κλειδί άρα:

$$\begin{aligned} &Μέγεθος\ Ασυμπίεστης\ Στήλης \\ &= Πλήθος\ εγγραφών \cdot Μέγεθος\ Κλειδιού \\ &= 8 \cdot 10^9 \cdot 8\ bit \\ &= 7{,}45\ GB \end{aligned}$$

Με τη συμπίεση Prefix Encoding η τιμή για την Κίνα που επαναλαμβάνεται 1.4 x 10^9 φορές στην αρχή της στήλης θα αντικατασταθεί από μια εμφάνιση με μέγεθος 8bit. Ένας αριθμός 64 bit χρειάζεται για την αποθήκευση του πλήθους των επαναλήψεων. Επομένως το μέγεθος της συμπιεσμένης στήλης είναι:

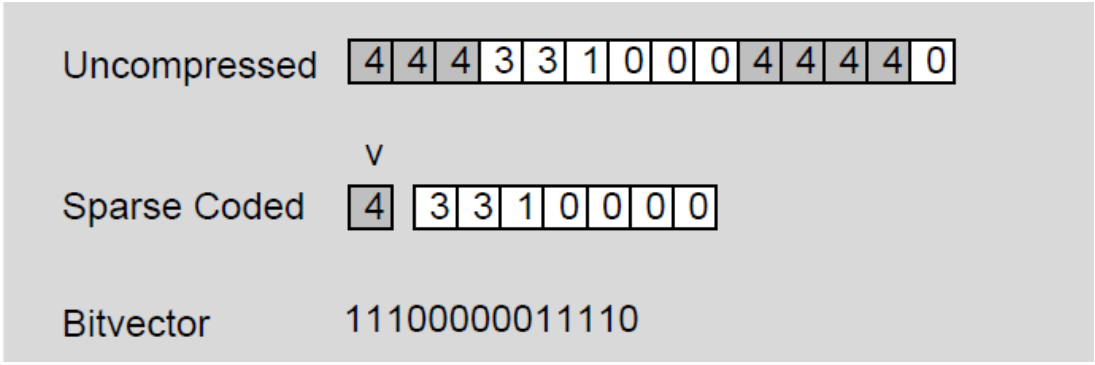
$$\begin{aligned} &Μέγεθος\ Συμπιεσμενης\ Δομής \\ &= Πλήθος\ Υπόλοιπων\ Εγγραφών \cdot Μέγεθος\ Κλειδιού + \\ &\quad Μέγεθος\ Αριθμού\ Πλήθους\ Επαναλήψεων + Μέγεθος\ Κλειδιού \\ &= (8 \cdot 10^9 - 1{,}4 \cdot 10^9) \cdot 8\ bit + 64\ bit + 8\ bit \\ &\approx 6{,}15\ GB \end{aligned}$$

Επομένως έχουμε μια συμπίεση της τάξης του 17,5%. Επιπλέον στα ερωτήματα που αφορούν τη κυρίαρχη τιμή, δίνεται άμεση πρόσβαση στη θέση των στοιχείων. Παράδειγμα ερωτήματος είναι το να βρεθούν όλοι οι πολίτες της Κίνας γένους αρσενικού. Σε αυτή την περίπτωση το φίλτρο του ερωτήματος σχετικά με το γένος θα εκτελεστεί στις πρώτες 1,4 x 10^9 εγγραφές.

Το μειονέκτημα της μεθόδου είναι ότι οι υπόλοιπες τιμές συνεχίζουν να επαναλαμβάνονται οπότε η είναι κατάλληλη μόνο εάν υπάρχει μόνο μια τιμή που επαναλαμβάνεται και αυτή βρίσκεται στην αρχή της στήλης.

Sparse Encoding

Για το στοιχείο της αρχικής στήλης που έχει τις περισσότερες επαναλήψεις κρατούνται η τιμή του πεδίου και ένα διάνυσμα από bits όσες και οι θέσεις της λίστας, στο οποίο υπάρχει η τιμή ένα εάν υπάρχει εμφάνιση τιμής και μηδέν εάν όχι. Και αυτή η τεχνική αφορά το κυρίαρχο στοιχείο αλλά σε αντίθεση με την Prefix Encoding δεν απαιτεί ο πίνακας να είναι ταξινομημένος.



Εικόνα 13. Sparse Encoding [17]

Στο προηγούμενο παράδειγμα με τη στήλη της χώρας, η οποία δεν είναι απαραίτητο να είναι ταξινομημένη. Ο βαθμός συμπίεσης μπορεί να υπολογιστεί ως εξής:

Το μέγεθος του κλειδιού που απαιτείται για να κωδικοποιηθούν οι τιμές 200 χωρών:

$$Μέγεθος\ Κλειδιού = \log_2(200) bit = 8\ bit$$

Εφόσον ο πίνακας έχει 8 x 10^9 εγγραφές, χρειαζόμαστε 8 bit για το κάθε κλειδί άρα:

$$Αρχικό\ Μέγεθος\ \ Στήλης$$
$$= Πλήθος\ εγγραφών \cdot Μέγεθος\ Κλειδιού$$
$$= 8 \cdot 10^9 \cdot 8\ bit = 7{,}45\ GB$$

Με τη συμπίεση Sparse Encoding, η τιμή για την Κίνα που επαναλαμβάνεται 1,4 x 10^9 φορές στην στήλη θα αντικατασταθεί από μια εμφάνιση με μέγεθος 8bit.

Το μέγεθος του Διανύσματος bit είναι:

$$\begin{aligned} &Μέγεθος\ Διανύσματος\ bit \\ &= Πλήθος\ Στοιχείων\ Αρχικής\ Στήλης \cdot 1\ bit \\ &= 8 \cdot 10^9 \cdot 1\ bit \approx 0{,}93\ GB \end{aligned}$$

Το μέγεθος της συμπιεσμένης δομής είναι:

$$\begin{aligned} &Μέγεθος\ Συμπιεσμένης\ Δομής \\ &\quad = Μέγεθος\ Διανύσματος\ Bit \\ &\quad + (Πλήθος\ Στοιχείων\ Αρχικής\ Στήλης \\ &\quad - Πλήθος\ Εμφανίσεων\ Κυρίαρχου\ Στοιχειου) \cdot Μέγεθος\ Κλειδιού \\ &\quad = 0{,}93\ GB + (8 \cdot 10^9 - 1{,}4 \cdot 10^9) \cdot 8\ bit + 8\ bit \approx 7{,}08\ GB \end{aligned}$$

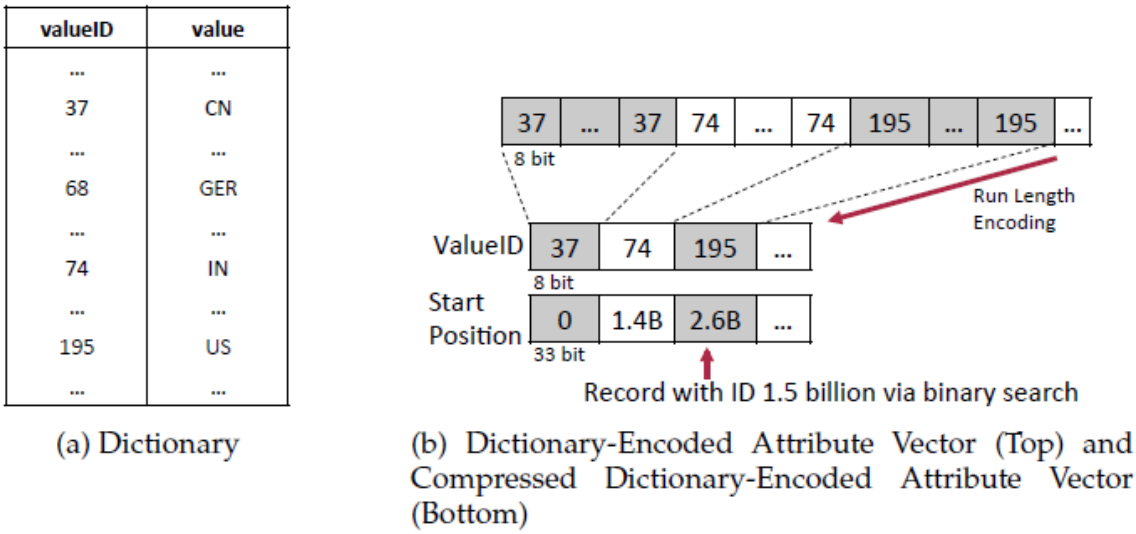
Άρα ο βαθμός συμπίεσης που επιτεύχθηκε είναι 5,0%. Δεν υπάρχει απευθείας πρόσβαση στα δεδομένα, αφού για την εύρεση της θέσης χρειάζονται υπολογισμοί με τη χρήση του Διανύσματος bit.

Το μειονέκτημα και αυτής μεθόδου είναι ότι οι υπόλοιπες τιμές συνεχίζουν να επαναλαμβάνονται οπότε η είναι κατάλληλη μόνο εάν υπάρχει μόνο μια τιμή που επαναλαμβάνεται και η στήλη δεν είναι ταξινομημένη.

Run-length Encoding (RLE)

Η τεχνική συμπίεσης Run-length Encoding λειτουργεί καλύτερα εάν η στήλη περιέχει λίγες διαφορετικές τιμές με μεγάλο πλήθος εμφανίσεων. Για βέλτιστο αποτέλεσμα η στήλη πρέπει να είναι ταξινομημένη, ώστε οι επαναλαμβανόμενες τιμές να είναι σε συνεχόμενες θέσεις.

Η συμπιεσμένη δομή αποτελείται από λίστα δύο στηλών. Στην πρώτη στήλη περιέχεται η τιμή και στη δεύτερη η θέση που βρίσκεται η πρώτη εμφάνιση, παραλείποντας όλες τις επαναλαμβανόμενες θέσεις.



Εικόνα 14. Run Length Encoding (RLE) [1]

Στο παράδειγμα με ταξινομημένη τη στήλη της χώρας χρειαζόμαστε τα εξής:

Μια στήλη για την καταχώρηση από μια φορά των τιμών των κλειδιών των 200 χωρών.

$$Μέγεθος\ Κλειδιού = \log_2(200) bit = 8\ bit$$

Το μέγεθος του λεξικού όπου:

$$Μέγεθος\ Στήλης\ A = Πλήθος\ Κλειδιών \cdot Μέγεθος\ Κλειδιού = 200 \cdot 8\ bit = 1600\ bit$$

Μια δεύτερη στήλη με τις αρχικές θέσεις κάθε τιμής. Το μέγεθος του αριθμού για να κωδικοποιήσουμε 8 x 10^9 θέσεις είναι:

$$Μέγεθος\ Αριθμού\ Offset = \lceil \log_2(8 \cdot 10^9) \rceil\ bit = 33\ bit$$

Άρα το μέγεθος της δεύτερης στήλης είναι 200 φορές τα 33 bit του αριθμού offset και επιπλέον 33 bits για τον αριθμό των επαναλήψεων του τελευταίου στοιχείου. Άρα:

$$Μέγεθος\ Στήλης\ B = 200 \cdot 33\ bit + 33\ bit = 6633\ bit$$

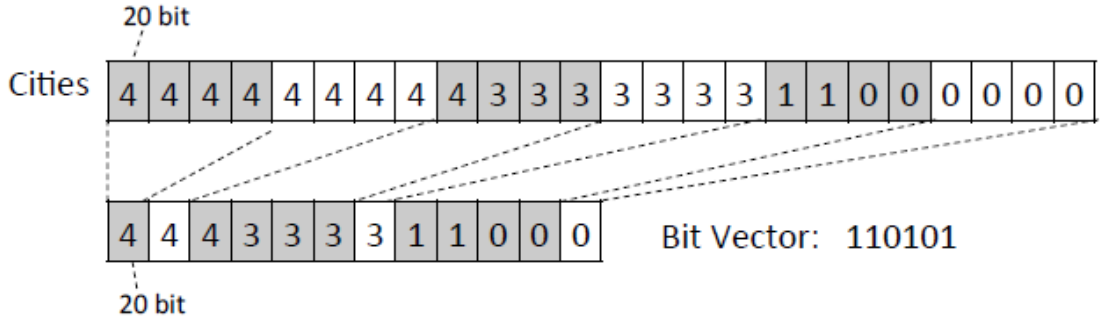
Επομένως το συνολικό μέγεθος της συμπιεσμένης μορφής είναι:

$$Μέγεθος\ Συμπιεσμένης\ Δομής$$
$$= Μέγεθος\ Στήλης\ A + Μέγεθος\ Στήλης\ B$$
$$= 1600\ bit + 6633bit \approx 1KB$$

Ο βαθμός συμπίεσης στην συγκεκριμένη μορφή ξεπερνά το 99,99% αφού από 7,45 GB που χρειαζόταν η αρχική στήλη φθάνουμε στο 1KB που χρειάζεται η συμπιεσμένη.

Cluster Encoding

Αυτή η τεχνική λειτουργεί σε blocks ίσου μεγέθους μιας στήλης. Η ασυμπίεστη στήλη χωρίζεται σε N blocks ίσου πλήθους στοιχείων (συνήθως 1024 στοιχεία) που ονομάζονται clusters. Στη συμπιεσμένη μορφή δημιουργείται μια λίστα όπου εάν τα στοιχεία ενός cluster έχουν την ίδια τιμή, τότε εκείνα αντικαθίστανται από ένα στοιχείο με την τιμή αυτή, διαφορετικά το cluster παραμένει ασυμπίεστο. Κρατείται και ένα επιπλέον διάνυσμα από bit ίσου πλήθους με τις θέσεις τις λίστας, με κάθε bit να έχει την τιμή 1 εάν αντιστοιχεί σε συμπιεσμένο cluster ή 0 εάν όχι.



Εικόνα 15. Cluster Encoding με μέγεθος block 4 [1]

Για παράδειγμα μπορεί να εξεταστεί η στήλη πόλη από τον πίνακα παγκόσμιου πληθυσμού. Ο πίνακας είναι ταξινομημένος ανά χώρα και πόλη. Έτσι οι πόλεις της ίδιας χώρας είναι τοποθετημένες μαζί. Επομένως η ίδια πόλη επαναλαμβάνεται συνεχόμενα αρκετές φορές. Η στήλη περιέχει 1 x 10^6 κλειδιά για τις πόλεις άρα το απαιτούμενο μέγεθος του κλειδιού είναι:

$$Μέγεθος\ Κλειδιού = \lceil \log_2(10^6) \rceil\ bit = 20\ bit$$

Το μέγεθος της ασυμπίεστης στήλης είναι:

$$Μέγεθος\ Αρχικής\ Στήλης = Πλήθος\ Στοιχείων \cdot 20\ bit = 8 \cdot 10^9 \cdot 20\ bit \approx 18{,}6\ GB$$

Έστω ότι το μέγεθος του Cluster είναι 1024 στοιχεία. Επομένως το πλήθος των Cluster είναι:

$$Πλήθος\ Cluster = \frac{Πλήθος\ Στοιχείων}{Μέγεθος\ Cluster} = \frac{8 \cdot 10^9}{1024} = 7{,}8 \cdot 10^6$$

Επομένως το μέγεθος του bit διανύσματος είναι:

$$Μέγεθος\ Διανύσματος\ Bit = Πλήθος\ Cluster \cdot 1\ bit = 7{,}8\ Mbit$$

Στη χειρότερη περίπτωση κάθε πόλη δημιουργεί ένα ασυμπίεστο cluster. Το μέγεθος της συμπιεσμένης δομής είναι:

$$\begin{aligned} &Μέγεθος\ Συμπιεσμένης\ Δομής \\ &= Πλήθος\ Ασυμπίεστων\ Cluster \cdot Μέγεθος\ Cluster \cdot Μέγεθος\ Κλειδιού \\ &+ Πλήθος\ Συμπιεσμένων\ Cluster \cdot Μέγεθος\ Κλειδιού + Μέγεθος\ Δυανίσματος\ Bit \\ &= 1 \cdot 10^6 \cdot 1024 \cdot 20\ bit + 6{,}8 \cdot 10^6 \cdot 20\ bit + 7{,}8 \cdot 10^6\ bit \approx 2{,}4 GB \end{aligned}$$

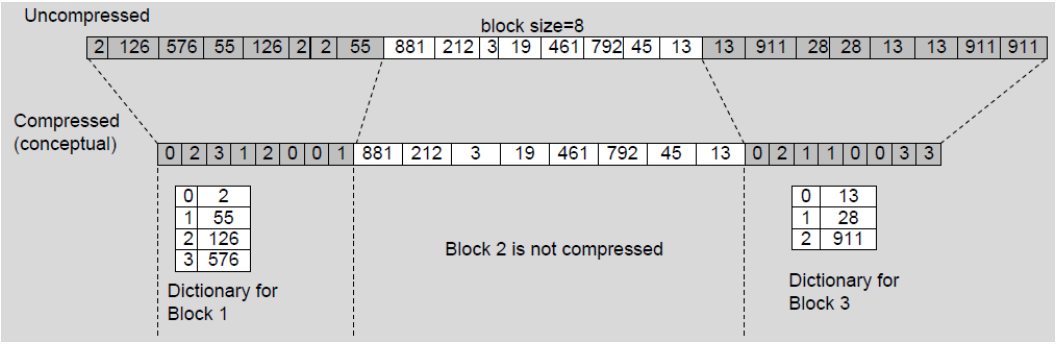
Άρα ο βαθμός συμπίεσης που επιτεύχθηκε είναι 87,1%.

Η τεχνική αυτή έχει το μειονέκτημα ότι δεν δίνει απευθείας πρόσβαση στις εγγραφές και σε κάθε ερώτημα πρέπει να υπολογίζονται οι θέσεις των στοιχείων με τη χρήση του διανύσματος bit.

Indirect Encoding

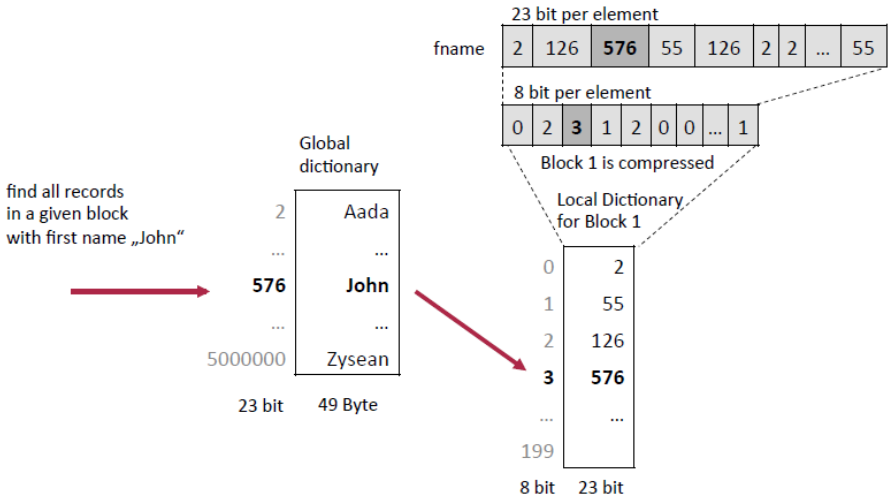
Παρόμοια με το Cluster Encoding, το Indirect Encoding εκτελείται σε block δεδομένων συγκριμένου μεγέθους (συνήθως 1024). Η συμπίεση αυτή είναι αποδοτική εάν κάθε block διαθέτει λίγα διαφορετικά στοιχεία. Συχνά χρησιμοποιείται όταν ο πίνακας είναι ταξινομημένος με βάση άλλη στήλη η οποία έχει συσχέτιση με τη στήλη που ταξινομούμε όπως η στήλη ΟΝΟΜΑ σε έναν πίνακα που είναι ταξινομημένος με βάση τη στήλη ΧΩΡΑ.

Κατά την κωδικοποίηση αυτή, ορίζουμε το μέγεθος του block και σε κάθε block της αρχικής στήλης που έχει επαναλαμβανόμενες τιμές και μπορεί να συμπιεστεί, εκτελούμε Dictionary Encoding και δημιουργούμε ένα μίνι λεξικό για αυτό το block. Η συμπιεσμένη μορφή είναι μια λίστα όπου στην περίπτωση του συμπιεσμένου block οι τιμές του αρχικού block έχουν αντικατασταθεί από τις θέσεις στο λεξικό καθώς και το λεξικό για αυτό το block.



Εικόνα 16. Indirect Encoding [17]

Έστω ότι δίνεται η στήλη FIRST NAME του πίνακα με τον παγκόσμιο πληθυσμό ο οποίος είναι ταξινομημένος με βάση τη στήλη COUNTRY. Η στήλη FIRST NAME έχει 5 x 10^6 διαφορετικές τιμές και έχει ήδη εφαρμοστεί σε αυτήν Dictionary Encoding.



Εικόνα 17. Παράδειγμα Indirect Encoding [1]

Το απαιτούμενο μέγεθος του κλειδιού της αρχικής στήλης είναι:

$$Μέγεθος\ Κλειδιού\ Αρχικής\ Στήλης = \lceil \log_2(5 \cdot 10^6) \rceil\ bit = 23\ bit$$

Το αρχικό μέγεθος της στήλης είναι:

$$\begin{aligned} &Αρχικό\ Μέγεθος\ Στήλης \\ &= Πλήθος\ Εγγραφών \cdot Μέγεθος\ Κλειδιού\ Αρχικής\ Στήλης \\ &= 8 \cdot 10^6\ \cdot 23\ bit \approx 21{,}4\ GB \end{aligned}$$

Αν τεθεί ως μέγεθος του block 1024 στοιχεία, τότε το πλήθος των block είναι:

$$Πλήθος\ Block = \frac{Πλήθος\ Εγγραφών}{Μέγεθος\ Block} = \frac{8 \cdot 10^9}{1024} \approx 7{,}8 \cdot 10^6$$

Για απλοποίηση ας υποτεθεί ότι σε κάθε block που υπάρχουν 1024 άνθρωποι από την ίδια χώρα, περιέχονται κατά μέσο όρο 200 διαφορετικά ονόματα και ότι όλα τα blocks θα συμπιεστούν. Το μέγεθος του κλειδιού block είναι:

$$Μέγεθος\ Κλειδιού\ Block = \lceil \log_2(200) \rceil\ bit = 8\ bit$$

Επομένως κάθε block για τη συμπίεσή του χρειάζεται μέγεθος κλειδιού 8 bit αντί για 23 bit. Το κάθε block χρειάζεται και ένα δείκτη μεγέθους 64 bit προς το λεξικό του. Το μέγεθος κάθε λεξικού block είναι:

$$\begin{aligned} &Μέγεθος\ Λεξικού\ Block \\ &= Πλήθος\ Διακριτών\ Ονομάτων\ \cdot Μέγεθος\ Κλειδιού + Δείκτης\ στο\ Λεξικό \\ &= (200 \cdot 23\ bit + 64\ bit) = 4664\ bit \end{aligned}$$

Το τελικό μέγεθος της συμπιεσμένης δομής είναι:

$$\begin{aligned} &Μέγεθος\ Συμπιεσμένης\ Δομής \\ &= Μέγεθος\ Λεξικού\ Block \cdot Πλήθος\ Block + Πλήθος\ Εγγραφών\ Στήλης \cdot Μέγεθος\ Κλειδιού\ Block \\ &= 4664\ bit \cdot 7{,}8 \cdot 10^6 + 8 \cdot 10^9 \cdot 8\ bit \approx 11{,}8\ GB \end{aligned}$$

Άρα ο βαθμός συμπίεσης που επιτεύχθηκε είναι 55,6%. Με την τεχνική αυτή υπάρχει άμεση πρόσβαση στα δεδομένα του πίνακα.

2.3 Κατάργηση Ευρετηρίων (Indexes)

Σε πολλές περιπτώσεις, η αποθήκευση ανά στήλη καταργεί την ανάγκη για επιπλέον δομές ευρετηρίων, αφού η αποθήκευση ανά στήλη με τις παραπάνω τεχνικές είναι παρόμοια σε λειτουργία με τη δημιουργία ευρετηρίου για κάθε στήλη. Η ταχύτητα σάρωσης της κάθε στήλης στη μνήμη και οι μηχανισμοί συμπίεσης, ειδικά η κωδικοποίηση λεξικού, επιτρέπουν λειτουργίες ανάγνωσης και αναζήτησης με πολύ υψηλές επιδόσεις. Έτσι στις περισσότερες περιπτώσεις δεν είναι απαραίτητες επιπλέον δομές ευρετηρίων.

Καταργώντας τα επιπλέον ευρετήρια, μειώνεται ακόμα περισσότερο ο χώρος που απαιτεί η βάση δεδομένων στην μνήμη και μειώνεται η πολυπλοκότητα της εφαρμογής. Επίσης αυξάνονται οι επιδόσεις κατά τις ενέργειες της εγγραφής, ενημέρωσης και διαγραφής στον πίνακα αφού πλέον δεν συντηρούνται κατά τις ενέργειες αυτές οι δομές των ευρετηρίων.

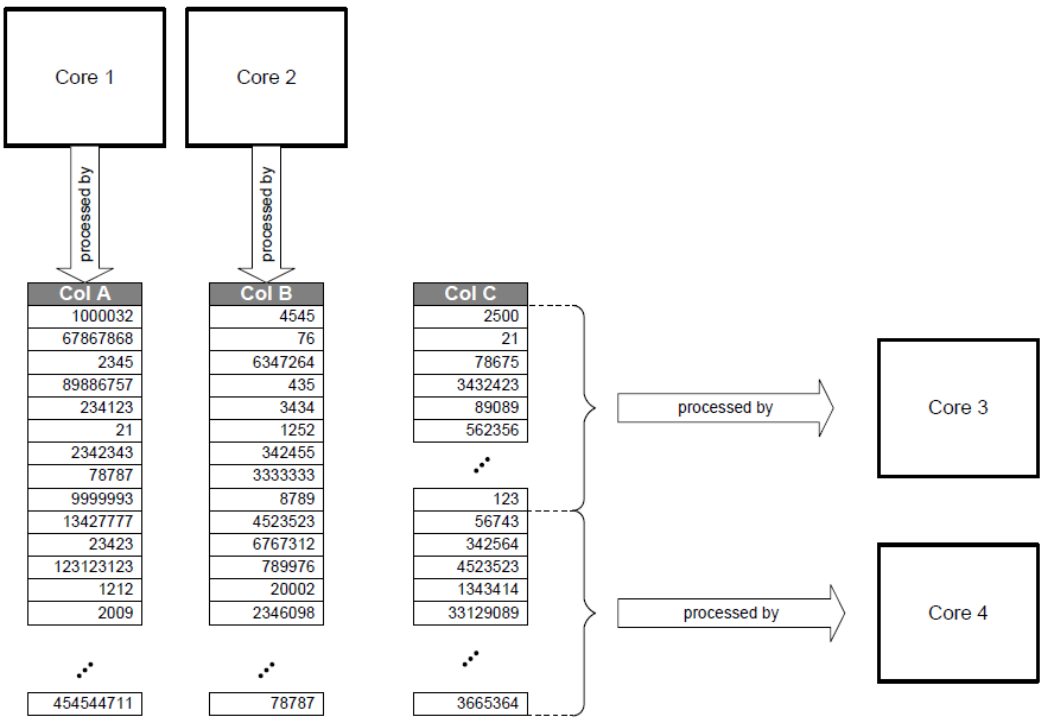
2.4 Παράλληλη Επεξεργασία

Το SAP HANA είναι σχεδιασμένο ώστε να εκτελεί τις εργασίες του παράλληλα. Συχνά χρησιμοποιεί εκατοντάδες πυρήνες ταυτόχρονα ώστε να εκμεταλλεύεται πλήρως την υπολογιστική ισχύ των κατανεμημένων συστημάτων.

Με την αποθήκευση των δεδομένων ανά στήλη, λειτουργίες που αφορούν μια στήλη όπως είναι η αναζήτηση ή οι συγκεντρωτικές συναρτήσεις (Aggregate Functions), υλοποιούνται ως επαναλήψεις σε έναν μονοδιάστατο πίνακα σε συνεχόμενες θέσεις της μνήμης. Μια τέτοια λειτουργία έχει υψηλή χωρική τοπικότητα (high spatial locality) και μπορεί να εκτελεστεί αποδοτικά στη μνήμη cache του επεξεργαστή. Αντίθετα εάν η αποθήκευση είναι ανά γραμμή, η ίδια λειτουργία θα είναι πιο αργή αφού η ίδια στήλη βρίσκεται διασκορπισμένη στη μνήμη και ο επεξεργαστής θα καθυστερεί λόγω των πολλών αστοχιών της cache.

Τα συμπιεσμένα δεδομένα μπορούν να φορτωθούν στην cache μνήμη του επεξεργαστή πιο γρήγορα, αφού έχουν μικρότερο μέγεθος. Τελικά η βελτίωση των επιδόσεων που παρατηρείται υπερκαλύπτει τις όποιες καθυστερήσεις οφείλονται στις επιπλέον εργασίες του επεξεργαστή για την αποσυμπίεση των δεδομένων.

Η αποθήκευση ανά στήλη επιτρέπει την παράλληλη εκτέλεση των λειτουργιών χρησιμοποιώντας πολλούς επεξεργαστικούς πυρήνες. Στην αποθήκευση ανά στήλη τα δεδομένα είναι ήδη κάθετα χωρισμένα, και έτσι οι λειτουργίες σε διαφορετικές στήλες μπορούν εύκολα να εκτελεστούν παράλληλα. Η λειτουργία για κάθε στήλη μπορεί να εκτελεστεί σε διαφορετικό επεξεργαστικό πυρήνα. Επιπρόσθετα, λειτουργίες σε μια στήλη μπορούν να εκτελεστούν παράλληλα διαχωρίζοντας τη στήλη σε τομείς οι οποίοι μπορούν να επεξεργασθούν από διαφορετικούς επεξεργαστικούς πυρήνες.



Εικόνα 18. Παράλληλη εκτέλεση ερωτήματος στις στήλες του πίνακα [31]

2.5 Κατάργηση Συγκεντρωτικών Πινάκων (Aggregate Tables)

Οι επιχειρηματικές εφαρμογές χρησιμοποιούν συχνά συγκεντρωτικούς πίνακες (Aggregate Tables), με βάση κάποιον αναλυτικό πίνακα, για να αυξήσουν την απόδοσή τους στην εκτέλεση συγκεκριμένων συγκεντρωτικών ερωτημάτων. Η ενημέρωση αυτών των πινάκων γίνεται είτε μετά από κάθε εγγραφή στα δεδομένα του κυρίως πίνακα, είτε σε προγραμματισμένες χρονικές στιγμές. Κατά την εκτέλεση του ερωτήματος αντί να υπολογίζονται τα συγκεντρωτικά στοιχεία από τον αναλυτικό πίνακα απλά διαβάζονται οι συγκεντρωτικοί πίνακες.

Το SAP HANA διαθέτει ταχύτητα ανάγνωσης αρκετών Gigabytes ανά χιλιοστό του δευτερολέπτου. Λόγω αυτής της υψηλής επίδοσης, συγκεντρωτικά ερωτήματα σε μεγάλο όγκο δεδομένων μπορούν να υπολογιστούν σε πραγματικό χρόνο από το σύστημα. Επομένως, σε πολλές περιπτώσεις δεν υπάρχει ανάγκη για συγκεντρωτικούς πίνακες, και η κατάργηση αυτών οδηγεί σε απλούστερα μοντέλα δεδομένων και αντίστοιχα σε απλοποίηση της λογικής της εφαρμογής. Ο υπολογισμός των ερωτημάτων σε πραγματικό χρόνο εγγυάται ενημερωμένα δεδομένα, σε αντίθεση με τους συγκεντρωτικούς πινάκες που ενημερώνονται με χρονοπρογραμματιζόμενες εργασίες. Επιπρόσθετα η κατάργηση των συγκεντρωτικών πινάκων έχει ως αποτέλεσμα το μικρότερο μέγεθος της βάσης δεδομένων.

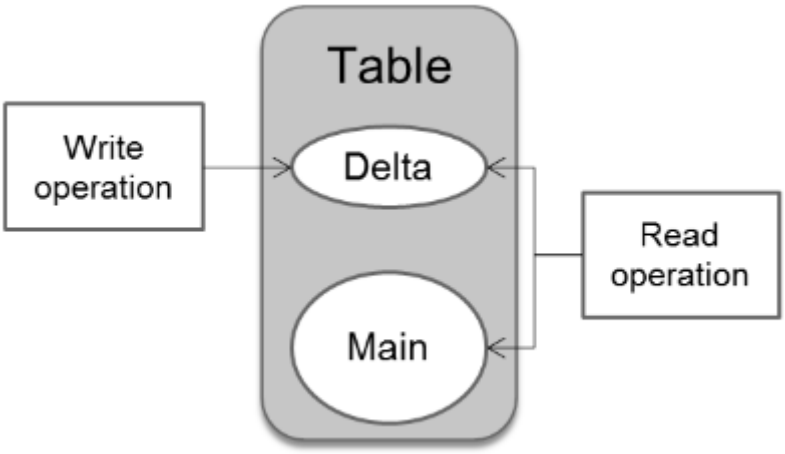
2.6 Βελτιστοποίηση Απόδοσης κατά την Εγγραφή

Η συμπίεση των δεδομένων που πραγματοποιεί το SAP HANA, έχει επιπτώσεις στην απόδοση του συστήματος όταν πρόκειται για εισαγωγή ή μεταβολή των δεδομένων. Το SAP HANA λύνει αυτό το πρόβλημα με τη χρήση του Delta Storage και της λογικής Insert Only on Delta Storage.

Main και Delta Storage

Κάθε πίνακας, που είναι αποθηκευμένος ανά στήλη, αποτελείται από δύο κομμάτια, το Delta και Main Storage. Το Delta Storage είναι βελτιστοποιημένο ώστε να γίνεται σε αυτό η εγγραφή των δεδομένων, ενώ το Main Storage είναι βελτιστοποιημένο για γρήγορη ανάγνωση και έχει υψηλή συμπίεση.

Η λειτουργία της εγγραφής γίνεται αποκλειστικά στο Delta Storage, ενώ η ανάγνωση γίνεται συνδυαστικά στο Main καθώς και στο Delta Storage. Έτσι το σύστημα μπορεί και έχει υψηλή απόδοση χωρίς να επιβαρύνεται κατά τη διαδικασία της εγγραφής των δεδομένων.

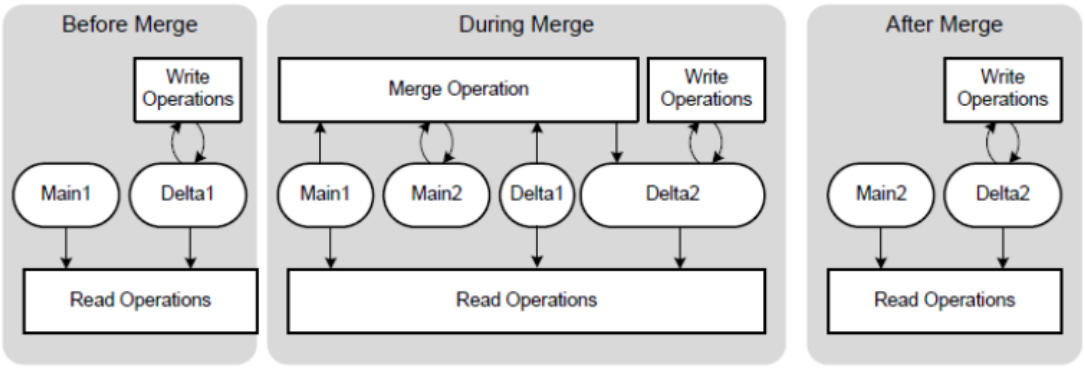


Εικόνα 19. Λειτουργίες ανάγνωσης και εγγραφής στο SAP HANA [23]

Delta Merge

Βέβαια σε τακτά χρονικά διαστήματα τα δεδομένα του Delta Storage πρέπει να μεταφέρονται στο Main Storage, και αυτή η διαδικασία ονομάζεται Delta Merge. Ο πίνακας κατά τη διαδικασία αυτή παραμένει online και είναι διαθέσιμος για ανάγνωση και εγγραφή.

Στην παρακάτω εικόνα υπάρχει ένα παράδειγμα Delta Merge. Σε αυτό πριν ξεκινήσει η διαδικασία όλες οι διαδικασίες εγγραφής πηγαίνουν στο Delta1 και η ανάγνωση γίνεται από το Delta1 και το Main1. Κατά την εκτέλεση της διαδικασίας δημιουργείται νέο Delta Storage το Delta2, στο οποίο κατευθύνονται όλες οι διαδικασίες εγγραφής. Η ανάγνωση γίνεται και από το Main1 και τα Delta1 και Delta2. Οι εγγραφές του Delta1 που δεν έχουν γίνει commit, αντιγράφονται στο Delta2. Οι υπόλοιπες εγγραφές του Delta1 συγχωνεύονται με το Main1 και δημιουργείται τελικά το Main2. Μόλις ολοκληρωθεί η διαδικασία τα Main1 και Delta1 διαγράφονται.



Εικόνα 20. Η διαδικασία Delta Merge [23]

Έπειτα υπολογίζεται ποιος αλγόριθμος συμπίεσης είναι ο καταλληλότερος για κάθε στήλη στο νέο Main Storage. Εάν η συμπίεση κάποιας στήλης αλλάξει τότε εκείνη φορτώνεται απευθείας ξανά στη μνήμη.

Τύποι Delta Merge

Για τη διαδικασία του Delta Merge ο προεπιλεγμένος τύπος είναι το Auto Merge, κατά τον οποίο μια Διαδικασία Συστήματος που ονομάζεται Mergedog ελέγχει σε τακτά χρονικά διαστήματα εάν πρέπει να εκτελεστεί Delta Merge, βάση προεπιλεγμένων κριτηρίων.

Εναλλακτικά μπορεί να απενεργοποιηθεί το Auto Merge ενός πίνακα και μια εφαρμογή μπορεί να αιτηθεί την εκτέλεση Delta Merge μέσω εντολής SQL, κάτι που ονομάζεται Smart Merge. Αυτό μπορεί να συμβεί εάν πρόκειται να φορτώσουμε σε έναν πίνακα μεγάλη ποσότητα δεδομένων, και για λόγους απόδοσης θέλουμε το Delta Merge να εκτελεστεί στο τέλος της διαδικασίας.

Προσέγγιση Insert Only on Delta

Στο SAP HANA οι λειτουργίες της εισαγωγής, ενημέρωσης και διαγραφής γίνονται με την προσέγγιση Insert Only, ώστε να μην επηρεάζουν αρνητικά την απόδοση του συστήματος. Αυτό σημαίνει ότι κάθε μια από αυτές δημιουργεί νέα εγγραφή στο Delta Storage, δηλαδή:

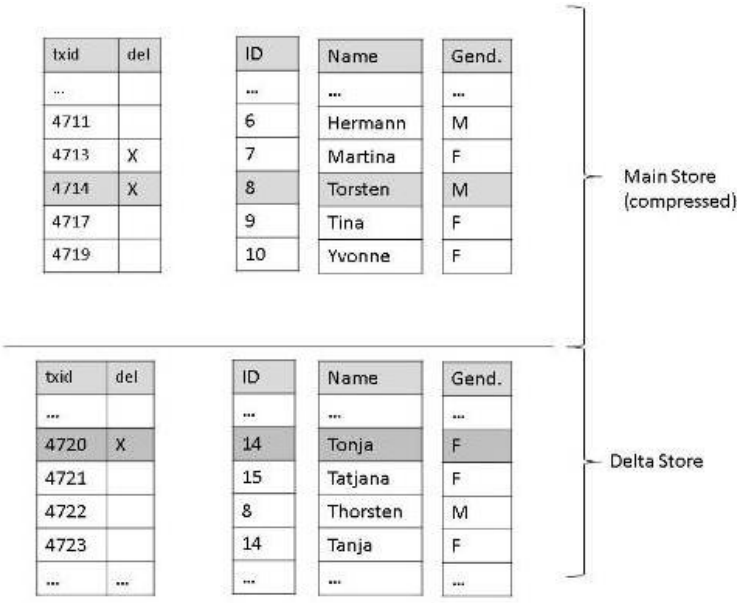
- Η Εισαγωγή (INSERT) προσθέτει μια νέα εγγραφή στο Delta Storage. Κατά τη διαδικασία Delta Merge, η εγγραφή θα μεταφερθεί στο Main Storage.

- Η Ενημέρωση (UPDATE) προσθέτει μια νεότερη έκδοση της είδη υπάρχουσας εγγραφής στο Delta Storage. Κατά την Επιλογή (SELECT) επιστρέφεται μόνο η τελευταία έκδοση της εγγραφής. Κατά τη διαδικασία Delta Merge η τελευταία έγκυρη εγγραφή θα μεταφερθεί στο Main Storage και όλες οι παλαιότερες εκδοχές της εγγραφής θα διαγραφούν.

- Η Διαγραφή (DELETE) θα σημειώσει την εγγραφή με flag διαγραφής είτε αυτή βρίσκεται στο Delta Storage είτε στο Main Storage. Κατά τη διαδικασία Delta Merge η εγγραφή αυτή θα διαγραφεί.

Οι διαδικασίες αυτές είναι κρυμμένες από τον προγραμματιστή, ο οποίος συνεχίζει να εκτελεί τις ενέργειες INSERT, UPDATE και DELETE όπως αυτές ορίζονται στην κλασσική SQL.

Στο παρακάτω παράδειγμα φαίνεται σχηματικά η διαδικασία. Η εγγραφή με id=7 διαγράφηκε από το Main Storage, και οι δύο παλαιές εκδόσεις των εγγραφών με id=8 και id=14 στις οποίες άλλαξε τιμή το όνομα, κάτι που οδήγησε στην προσθήκη νέων εκδόσεων των εγγραφών.



Εικόνα 21. Παράδειγμα της προσέγγισης Insert Only [24]

2.7 Table Partitioning

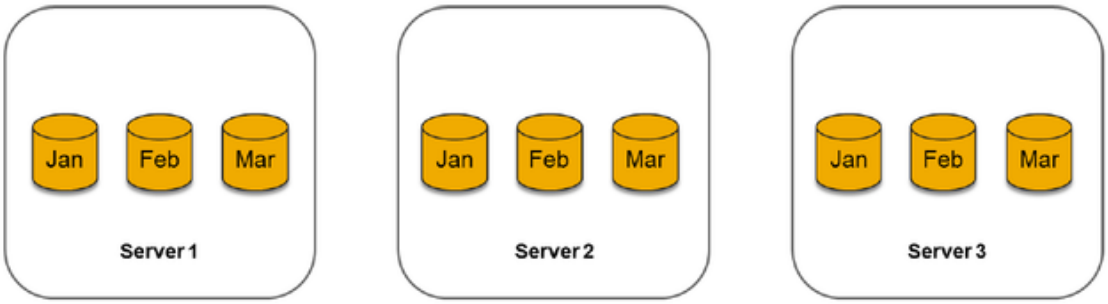
Το SAP HANA έχει τη δυνατότητα οριζόντιου διαχωρισμού του πίνακα σε υποπίνακες που ονομάζονται partitions. Με αυτήν την τεχνική υποστηρίζεται η δημιουργία πολύ μεγάλων πινάκων με την αποσύνθεσή τους σε μικρότερα και πιο διαχειρίσιμα αντικείμενα. Το Table Partitioning δεν γίνεται αντιληπτό από τον προγραμματιστή και δεν επηρεάζει την εκτέλεση των εντολών SQL.

Οι τυπικές περιπτώσεις Table Partitioning είναι οι:

- Load Balancing: Τα partitions διαμοιράζονται σε ξεχωριστούς servers. Αυτό σημαίνει ότι ένα ερώτημα εκτελείται από όλους τους servers που έχουν κάποιο partition του πίνακα.
- Parallelization: Οι ενέργειες σε έναν πίνακα εκτελούνται παράλληλα λόγω του διαμοιρασμού.

- Partition Pruning: Τα ερωτήματα αναλύονται για να βρεθεί σε ποια partitions αναφέρονται. Με αυτήν την μέθοδο τα ερωτήματα εκτελούνται μόνο στους servers που φιλοξενούν τα αντίστοιχα partitions, μειώνοντας συνολικά τον φόρτο εργασίας του συστήματος.

- Explicit Partition Handling: Οι εφαρμογές μπορούν να ελέγχουν ενεργά τα partitions. Για παράδειγμα, μπορούν να προσθέσουν partitions τα οποία θα περιέχουν τα δεδομένα του επόμενου μήνα.



Εικόνα 22. Partitioning Απλού Επιπέδου σε 3 Servers [25]

Το μέγιστο μέγεθος ενός μη διαχωρισμένου πίνακα στο SAP HANA είναι $2x10^9$ εγγραφές. Αυτό μπορεί να ξεπεραστεί έμμεσα, χωρίζοντας τον πίνακα σε partitions όπου το κάθε partition έχει μέγιστο μέγεθος $2x10^9$ εγγραφές.

Το Table Partitioning βοηθά και στην απόδοση του Delta Merge, αφού αυτό πλέον γίνεται σε επίπεδο partition. Έτσι Delta Merge γίνεται μόνο στα partitions που έχουν μεταβολές και όχι σε ολόκληρο τον πίνακα. Για αυτό το Table Partitioning προτείνεται και για συστήματα SAP HANA που εκτελούνται σε έναν Server.

Κεφάλαιο 3 - Ανάπτυξη Λογισμικού στο SAP HANA

3.1 Δημιουργία Persistence Model στο SAP HANA

Το Persistence Model ορίζει τα database schema, tables, sequences και views τα οποία προσδιορίζουν ποια δεδομένα θα είναι διαθέσιμα στις εφαρμογές και πώς.

Με τη χρήση του SAP HANA Extended Application Services (SAP HANA XS), το Persistence Model συνδέεται και αντιστοιχίζεται με το Consumption Model το οποίο είναι διαθέσιμο στις Εφαρμογές – Πελάτες, ώστε τελικά τα δεδομένα να προβληθούν στον χρήστη με την κατάλληλη μορφή μέσα από τη γραφική διεπαφή της τελικής εφαρμογής.

Με το SAP HANA XS μπορούν να δημιουργηθούν όλα τα αντικείμενα της βάσης δεδομένων ως αρχεία μέσα στο Repository. Για τη δημιουργία του Persistence Model μπορεί να χρησιμοποιηθούν είτε η σύνταξη Core Data Services (CDS), είτε η σύνταξη HDBTable ή και οι δύο μαζί. Στον παρακάτω πίνακα υπάρχουν τα αντικείμενα που μπορούν να δημιουργηθούν ανά Σύνταξη καθώς και η κατάληξη του αντίστοιχου αρχείου.

Artifact Type	CDS	HDBTable
Schema	.hdbschema	.hdbschema
Table	.hdbdd	.hdbtable
Table Type	.hdbdd	.hdbstructure
View	.hdbdd	.hdbview
Association	.hdbdd	-
Sequence	.hdbsequence	.hdbsequence
Structured Types	.hdbdd	-
Data import	.hdbti	.hdbti

Πίνακας 1. Data Persistence Artifacts ανά Γλώσσα και κατάληξη αρχείου [31]

Δημιουργία Persistence Model με χρήση σύνταξης Core Data Services (CDS)

Όπως αναφέρθηκε τα Core Data Services αποτελούν μια γλώσσα - σύνταξη που χρησιμοποιείται για τον ορισμό και την κατανάλωση μοντέλων δεδομένων στο SAP HANA.

Με τη χρήση της γλώσσας CDS μπορούν να δημιουργηθούν τα αντίστοιχα CDS Documents που προσδιορίζουν τα παρακάτω αντικείμενα [31]:

- Entities (tables)
- Views
- User-defined data types (including structured types)
- Contexts
- Associations
- Annotations

Στο παράδειγμα της παρακάτω εικόνας στο CDS Document EmployeeModel.hdbdd ορίζονται οι τύποι FullName και Street, καθώς και 2 οντότητες (Πίνακες), οι Address και Employee, που χρησιμοποιούν τους τύπους αυτούς.

EmployeeModel.hdbdd

```
namespace demo.cds.CoreDataServicesDemo;

@Schema: 'SAPUIA'
context EmployeeModel {

    type Name : String(80);

    type FullName {
        firstName  : Name;
        middleName : Name;
        lastName   : Name;
    };

    type Street {
        street : String(80);
        number : String(8);
    };

    entity Address {
        key id : Integer;
        street : EmployeeModel.Street;
        city   : String(40);
        zip    : String(16);
    };

    entity Employee {
        key id : Integer;
        name   : FullName;
        salary : Decimal(15,2);
    };

};
```

Εικόνα 23. Παράδειγμα αρχείου .hdbdd στο CDS Editor [08]

Με τη χρήση CDS μπορούμε να δημιουργήσουμε και views. Για παράδειγμα μπορούμε να δημιουργήσουμε ένα view για τον Πίνακα employee με τις παρακάτω εντολές.

```
view EmployeeView as select from Employee
{
   id,
   name
};
```

Όταν επιλέξουμε ενεργοποίηση (activation) του CDS Document, τότε στο SAP HANA δημιουργούνται τα αντικείμενα που περιγράφονται σε αυτό.

Δημιουργία Persistence Model με χρήση σύνταξης HDBTable

Η σύνταξη HDBTable αποτελεί εναλλακτική επιλογή αντί της σύνταξης CDS για τη δημιουργία των αντικειμένων του Persistence Model.

Ακολουθούν παραδείγματα δημιουργίας αντικειμένων με τη χρήση HDBTable [31]:

- **Δημιουργία DB Schema με όνομα MYSCHEMA:**

  Όνομα Αρχείου: `MYSCHEMA.hdbschema`

  Το αρχείο θα έχει την παρακάτω εντολή:

  ```
  schema_name=”MYSCHEMA”;
  ```

  Με την ενεργοποίηση (activation) του αρχείου θα δημιουργηθεί το αντίστοιχο DB Schema

- **Δημιουργία πίνακα με όνομα MYTABLE στο DB Schema MYSCHEMA με αποθήκευση ανά στήλη, με 4 στήλες και 2 ευρετήρια:**

  Όνομα Αρχείου: `MYTABLE.hdbtable`

  Το αρχείο θα έχει τις παρακάτω εντολές:

  ```
  table.schemaName = "MYSCHEMA";
  ```

```
table.tableType = COLUMNSTORE;

table.columns = [

 {name = "Col1"; sqlType = VARCHAR; nullable = false; length =
20; comment = "dummy comment";},

 {name = "Col2"; sqlType = INTEGER; nullable = false;},

 {name = "Col3"; sqlType = NVARCHAR; nullable = true; length =
20; defaultValue = "Defaultvalue";},

 {name  =  "Col4";  sqlType  =  DECIMAL;  nullable  =  false;
precision = 2; scale = 3;}];

table.indexes = [

 {name = "MYINDEX1"; unique = true; indexColumns = ["Col2"];},

 {name  =  "MYINDEX2";  unique  =  true;  indexColumns  =  ["Col1",
"Col4"];}];

table.primaryKey.pkcolumns = ["Col1", "Col2"];
```

Με την ενεργοποίηση (activation) του αρχείου θα δημιουργηθεί ο πίνακας στο MYSCHEMA

- **Δημιουργία View, με χρήση SQL, με όνομα MYVIEW στο DB Schema MYSCHEMA:**

  Όνομα Αρχείου: `MYVIEW.hdbview`

  Το αρχείο θα έχει τις παρακάτω εντολές:

```
schema="MYSCHEMA";

query="SELECT T1.\"Column2\" FROM \"MYSCHEMA\".

\"acme.com.test.views::MY_VIEW1\" AS T1 LEFT JOIN \"MYSCHEMA\".
```

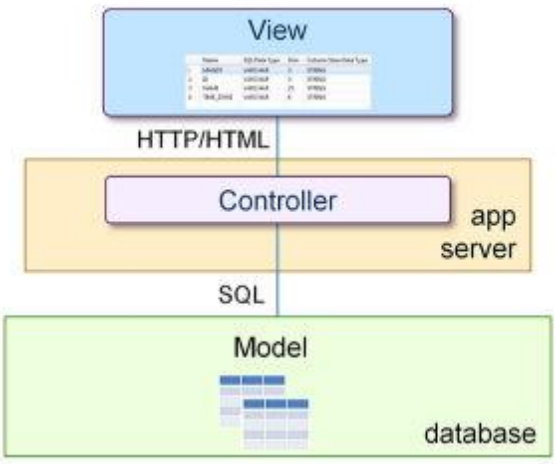
```
\"acme.com.test.views::MY_VIEW2\" AS T2 ON T1.\"Column1\" =
T2.\"Column1\"";

depends_on=["acme.com.test.views::MY_VIEW1",
"acme.com.test.views::MY_VIEW2"];
```

Με την ενεργοποίηση (activation) του αρχείου θα δημιουργηθεί η View στο MYSCHEMA

3.2 Βελτιστοποίηση των εφαρμογών για το SAP HANA

Όπως αναφέρθηκε, το SAP HANA μπορεί να αντικαταστήσει το ΣΔΒΔ που χρησιμοποιεί μια εφαρμογή η οποία θα συνδεθεί σε αυτό μέσω οδηγών JDBC / ODBC. Η εφαρμογή θα συνεχίσει μετά την αλλαγή να λειτουργεί κανονικά. Όμως για να αξιοποιηθεί πλήρως ένα σύστημα SAP HANA, τόσο για τις επιδόσεις όσο και για τις δυνατότητες που δίνει, χρειάζονται αλλαγές στο μοντέλο ανάπτυξης των εφαρμογών.

Στην παρακάτω εικόνα περιγράφεται το μοντέλο μιας κλασσικής εφαρμογής η οποία χρησιμοποιεί την αρχιτεκτονική Model-View-Controller (MVC). Σε αυτή την εφαρμογή το SAP HANA μπορεί να χρησιμοποιηθεί για το επίπεδο Model, ως σχεσιακή βάση δεδομένων. Παρόλο που η διαφορά στην απόδοση θα είναι εμφανής, χρειάζονται βελτιστοποιήσεις σε διάφορα επίπεδα για να μεγιστοποιηθεί η απόδοση του συστήματος.

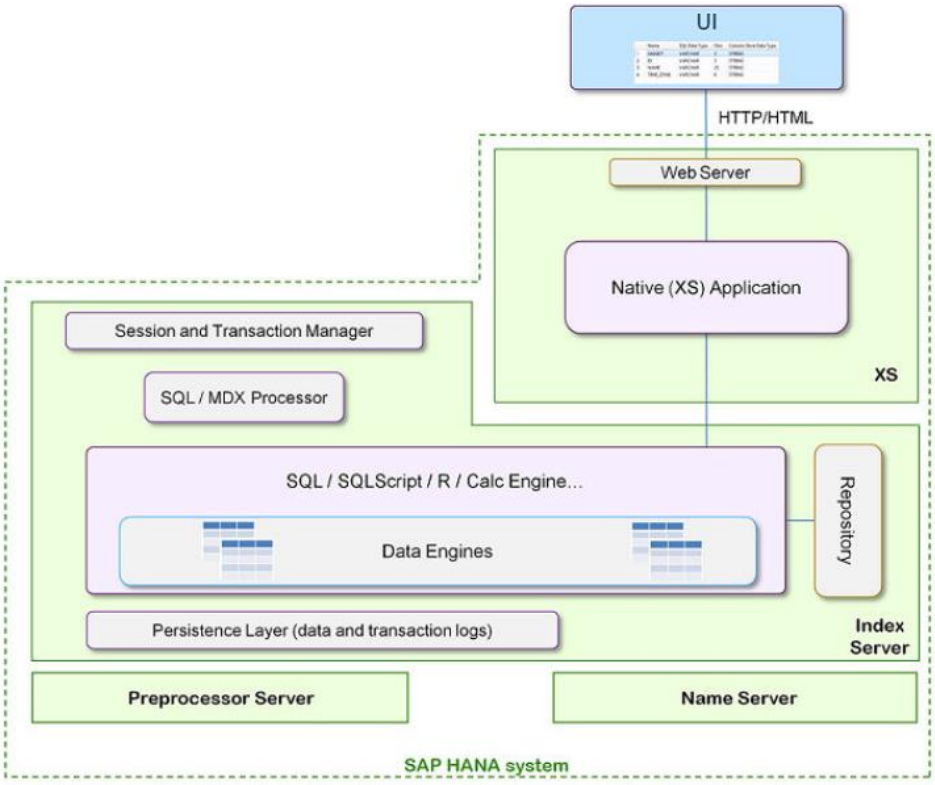


Εικόνα 24. Κλασσική εφαρμογή με Αρχιτεκτονική MVC [31]

SAP HANA XS

Το SAP HANA επεκτείνει το ρόλο του από Σύστημα Βάσεων Δεδομένων, σε ολοκληρωμένη πλατφόρμα ανάπτυξης. Για να υποστηρίξει το data-integrated μοντέλο ανάπτυξης διαθέτει τα SAP HANA Extended Application Services ή εν συντομία SAP HANA XS. Με τα SAP HANA XS η πλατφόρμα δίνει ένα πλήρες πακέτο εργαλείων για την ανάπτυξη web-based εφαρμογών. Αυτά περιλαμβάνουν έναν ελαφρύ (lightweight) Web-Server, υποστήριξη OData, εκτέλεση server-side JavaScript και πλήρη πρόσβαση στις λειτουργίες της SQL και SQL Script.

Οι υπηρεσίες SAP HANA XS, παρέχονται από τον ενσωματωμένο SAP HANA XS Server, ο οποίος επιτρέπει σε εφαρμογές-πελάτες να αποκτήσουν πρόσβαση στο σύστημα SAP HANA μέσω σύνδεσης HTTP. Έτσι το επίπεδο της εφαρμογής μπορεί να τρέξει εγγενώς στο SAP HANA χωρίς την ανάγκη εξωτερικού Application Server.



Εικόνα 25. SAP HANA XS Server στο Σύστημα SAP HANA [31]

Βελτιστοποιήσεις στα ερωτήματα SQL

Το πρώτο επίπεδο βελτιστοποίησης της εφαρμογής είναι τα ερωτήματα SQL που περιλαμβάνει. Αυτά μπορούν να βελτιστοποιηθούν εφαρμόζοντας βέλτιστες πρακτικές και λαμβάνοντας υπ' όψιν τις ιδιαιτερότητες του SAP HANA. Οι σημαντικότεροι κανόνες [28], είναι οι παρακάτω:

- Αποφυγή υπολογισμών στα κριτήρια των ερωτημάτων. Εάν αυτό δεν μπορεί να αποφευχθεί προτείνεται η χρήση παραγόμενων στηλών (generated columns).

Slow query	SELECT * FROM T WHERE b * c = 10;
Fast query	SELECT * FROM T WHERE bc = 10;
DDL for faster query	ALTER TABLE T ADD (bc INTEGER GENERATED ALWAYS AS b * c);

Εικόνα 26. Αποφυγή υπολογισμών στα κριτήρια των ερωτημάτων [28]

- Εάν δύο στήλες συγκρίνονται στα κριτήρια ερωτημάτων, τότε πρέπει οι στήλες αυτές να είναι ίδιου τύπου. Για στήλες διαφορετικού τύπου το SAP HANA κάνει αυτόματη μετατροπή τύπου (type casting) για να είναι δυνατή η σύγκριση. Η αυτόματη μετατροπή είναι όμως χρονοβόρα. Για να αυξηθεί η απόδοση πρέπει είτε να αλλάξει ο τύπος της τιμής ρητά, είτε να χρησιμοποιηθεί παραγόμενη στήλη.

Slow query	SELECT * FROM T WHERE date_string < CURRENT_DATE;
Fast query	SELECT * FROM T WHERE date_string < TO_CHAR(CURRENT_DATE, 'YYYYMMDD');

Εικόνα 27. Ρητή αλλαγή τύπου τιμής [28]

- Αποφυγή της χρήσης του κατηγορήματος JOIN με συνθήκη διαφορετική της ισότητας. Κατηγόρημα JOIN με χρήση OR, καρτεσιανό γινόμενο ή χωρίς συνθήκη δεν υποστηρίζεται εγγενώς από το SAP HANA, με αποτέλεσμα την πτώση της απόδοσης.

Slow query	SELECT M.year, M.month, SUM(T.ship_amount) FROM T JOIN M ON T.ship_date BETWEEN M.first_date AND M.last_date GROUP BY M.year, M.month;
Fast query	SELECT M.year, M.month, SUM(T.ship_amount) FROM T JOIN M ON EXTRACT(YEAR FROM T.ship_date) = M.year AND EXTRACT(MONTH FROM T.ship_date) = M.month GROUP BY M.year, M.month;

Εικόνα 28. Αποφυγή χρήσης κατηγορήματος JOIN με συνθήκη διαφορετική της ισότητας [28]

- Αποφυγή των φίλτρων στο κατηγόρημα JOIN, καθώς αυτό δεν υποστηρίζεται εγγενώς. Εάν υπάρχει ανάγκη, αυτό προτείνεται να υλοποιηθεί με τη δημιουργία παραγόμενης στήλης.

Slow query	SELECT * FROM T JOIN S ON T.a = S.a AND T.b = 1;
Fast query	SELECT * FROM T JOIN S ON T.a = S.a AND T.b = S.one;
DDL for faster query	ALTER TABLE S ADD (one INTEGER GENERATED ALWAYS AS 1);

Εικόνα 29. Αποφυγή χρήσης κατηγορήματος JOIN με φίλτρα [28]

- Η κυκλική συνένωση πινάκων (CYCLIC JOIN) δεν υποστηρίζεται εγγενώς και πρέπει να αποφεύγεται ή να αντικαθίσταται από ακυκλική συνένωση πινάκων (ACYCLIC JOIN). Στην παρακάτω εικόνα η κυκλική συνένωση έχει αντικατασταθεί από ακυκλική, με τη μετακίνηση της στήλης NATION του πίνακα SUPPLIER στον πίνακα LINEITEM.

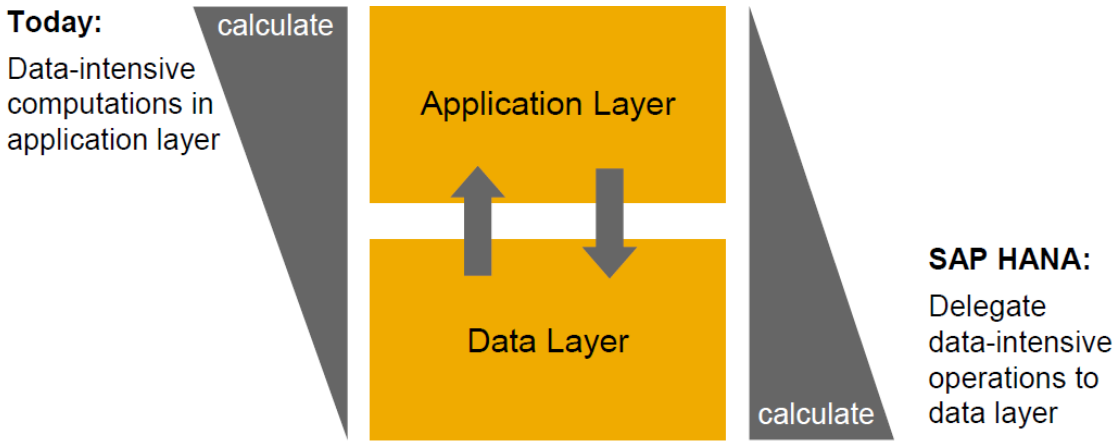
Cyclic join	SELECT * FROM supplier S, customer C, lineitem L WHERE L.supp_key = S.key AND L.cust_key = C.key AND S.nation = C.nation;
Acyclic join	SELECT * FROM supplier S, customer C, lineitem L WHERE L.supp_key = S.key AND L.cust_key = C.key AND L.supp_nation = C.nation;

Εικόνα 30. Μετατροπή κυκλικής συνένωσης πινάκων σε ακυκλική [9]

- Πρέπει να αποφεύγεται η χρήση του OR για την σύνδεση των κατηγορημάτων EXISTS or NOT EXISTS με άλλα κατηγορήματα.
- Όπου είναι δυνατό, πρέπει να χρησιμοποιείται το κατηγόρημα NOT EXISTS αντί για το NOT IN. Το κατηγόρημα NOT IN είναι πιο αργό.
- Πρέπει να αποφεύγονται τα κατηγορήματα UNION ALL, UNION, INTERSECT και EXCEPT, αφού δεν υποστηρίζονται εγγενώς.
- Τέλος, πρέπει να περιορίζεται ο αριθμός των στηλών που επιστρέφει ένα SQL ερώτημα στις απαραίτητες εάν ο πίνακας είναι αποθηκευμένος ανά στήλη. Εάν υπάρχει ανάγκη ανάκτησης πολλών στηλών με λίγες εγγραφές θα πρέπει να εξεταστεί το ενδεχόμενο της μετατροπής του πίνακα σε αποθήκευση ανά γραμμή.

Μοντέλο Ανάπτυξης Code to Data

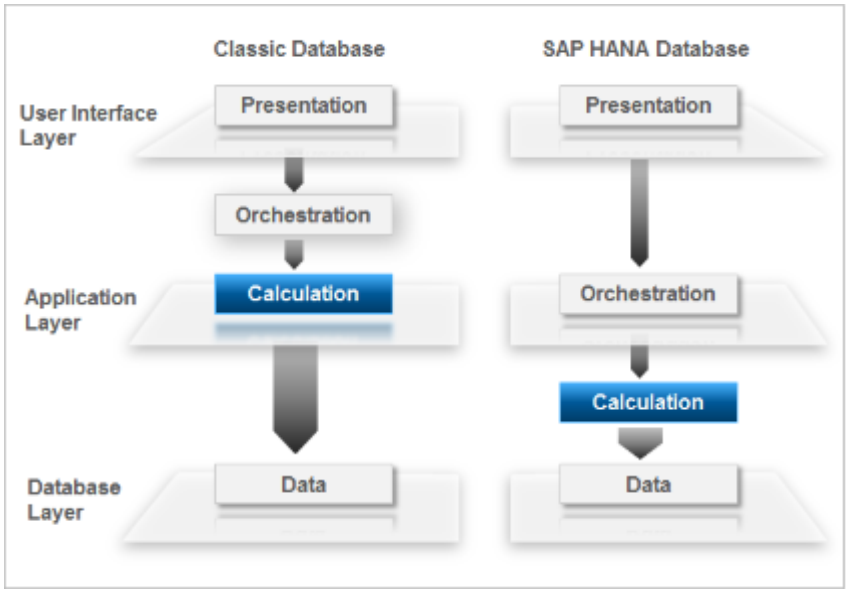
Για να εκμεταλλευτεί η εφαρμογή τη μέγιστη απόδοση του SAP HANA, πρέπει να ακολουθηθεί το προτεινόμενο μοντέλο ανάπτυξης, που ονομάζεται Code to Data Paradigm. Εδώ μέρος των υπολογισμών της εφαρμογής μεταφέρεται πλέον από τον Application Server στο επίπεδο της Βάσης Δεδομένων μια διαδικασία που ονομάζεται Code Pushdown.



Εικόνα 31. Code to Data Paradigm [8]

Η αλλαγή αυτή σημαίνει ότι η λογική της εφαρμογής θα μεταφερθεί με τη δημιουργία συναρτήσεων μέσα στο περιβάλλον του SAP HANA. Έτσι τα δεδομένα που βρίσκονται στη μνήμη του SAP HANA χρησιμοποιούνται εκεί και ελαχιστοποιείται η περιττή μεταφορά των δεδομένων στον Application Server.

Η μεταφορά των υπολογισμών στην βάση δεδομένων, αποτελεί σημαντική αλλαγή στο κλασσικό μοντέλο ανάπτυξης εφαρμογών. Για υφιστάμενες εφαρμογές που μεταφέρονται στο SAP HANA απαιτείται σημαντική προσπάθεια, όμως είναι μεγάλη η αύξηση των επιδόσεων που επιτυγχάνεται.



Εικόνα 32. Μετασχηματισμός μοντέλου κλασσικής εφαρμογής σε μοντέλο Code to Data [24]

Για την υλοποίηση του Code to Data Paradigm η SAP έχει αναπτύξει μια σειρά από τεχνολογίες, όπως τα HANA Information Views και η SQLScript, οι οποίες αποτελούν εργαλεία για την υλοποίηση της λογικής της εφαρμογής μέσα στο επίπεδο της βάσης δεδομένων. Οι τεχνολογίες αυτές θα εξεταστούν αναλυτικά παρακάτω.

3.3 Δημιουργία Analytic Model στο SAP HANA

Η βελτίωση της απόδοσης και το Code Push Down αφορά κυρίως την ανάκτηση δεδομένων από το SAP HANA πάνω στα οποία έχουν γίνει υπολογισμοί, πριν την μεταφορά τους στον Application Server. Για την ανάπτυξη του application logic και την ανάκτηση των δεδομένων στο SAP HANA υπάρχουν τα HANA Information Views και Stored Procedures οι οποίες μπορούν να δημιουργηθούν με SQL, SQLScript και τη γλώσσα R.

SAP HANA Information Views

Τα SAP HANA Information Views χρησιμοποιούν διάφορους συνδυασμούς των δεδομένων που υπάρχουν στους πίνακες, ώστε να μοντελοποιήσουν ένα επιχειρηματικό σενάριο και να εξομοιώσουν τις απαραίτητες οντότητες όπως πχ ΠΕΛΑΤΗΣ, ΠΡΟΪΟΝ και ΠΩΛΗΣΕΙΣ καθώς και τις σχέσεις μεταξύ αυτών. Τα Information Views μπορούν στη συνέχεια να χρησιμοποιηθούν από την εφαρμογή για την ανάκτηση δεδομένων από το SAP HANA.

Τα δεδομένα που χρησιμοποιούνται στα HANA Views ταξινομούνται σε:

- *Attributes*

  Περιγραφικά δεδομένα, π.χ. Κωδικός Πελάτη, Πόλη, Χώρα.

- *Measures*

  Μετρήσιμα δεδομένα, π.χ. Αξία, Ποσότητα Ειδών.

Τα Attributes διακρίνονται σε:

- *Simple Attributes*

  Είναι ανεξάρτητα μη μετρήσιμα στοιχεία τα οποία υπάρχουν στα δεδομένα, π.χ. Κωδικός Προϊόντος, Όνομα Προϊόντος της ενότητας Προϊόν.

- *Calculated Attributes*

Τα Calculated Attributes δημιουργούνται από συνδυασμό ενός ή περισσότερων χαρακτηριστικών ή σταθερών, π.χ. το Πλήρες Όνομα Πελάτη (Όνομα και Επώνυμο).

- *Local Attributes*

  Με τη χρήση Local Attribute μπορούμε να αλλάξουμε την συμπεριφορά ενός χαρακτηριστικού, μόνο για το συγκεκριμένο View.

Τα Measures διακρίνονται σε:

- *Simple Measures*

  Ένα Simple Measure είναι ένα μετρήσιμο χαρακτηριστικό, το οποίο προέρχεται από τα δεδομένα, πχ Κέρδος

- *Calculated Measures*

  Προέρχονται από ένα συνδυασμό δεδομένων από κύβους OLAP, Αριθμητικούς Τελεστές, Σταθερές και Συναρτήσεις. Παράδειγμα αποτελεί ο υπολογισμός του συνόλου πωλήσεων ενός προϊόντος ανά περιοχή.

- *Restricted Measures*

  Χρησιμοποιούνται για το φιλτράρισμα της τιμής ενός χαρακτηριστικού, βάση ορισμένου κανόνα.

- *Counters*

  Προσθέτουν ένα νέο μετρήσιμο για τον αριθμό των επαναλήψεων ενός χαρακτηριστικού, π.χ. πόσες φορές εμφανίζεται ένα προϊόν.

Το SAP HANA υποστηρίζει τους παρακάτω τύπους Information Views:

- *Attribute Views*

Τα Attribute Views χρησιμοποιούνται για τη μοντελοποίηση μιας οντότητας βασισμένα σε χαρακτηριστικά τα οποία περιέχονται σε διάφορους πίνακες. Τα Attribute Views μπορούν να περιέχουν Columns, Calculated Columns και Hierarchies. Στη συνέχεια μπορούν να χρησιμοποιηθούν από Analytic Views ή Calculation Views προκειμένου να δημιουργηθεί ένα Εικονικό Σχήμα Αστέρα (Virtual Star Schema) στα δεδομένα του SAP HANA.

- *Analytic Views*

  Χρησιμοποιούνται για τη μοντελοποίηση δεδομένων που περιέχουν μετρήσιμα. Παράδειγμα είναι η αναφορά του ιστορικού πωλήσεων που περιέχει μετρήσιμα για την ποσότητα, την τιμή κλπ

  Ως πηγή δεδομένων μπορεί να είναι περισσότεροι του ενός πίνακες, αλλά τα μετρήσιμα χαρακτηριστικά του Analytic View πρέπει να προέρχονται από μόνο έναν από τους πίνακες.

  Τα Analytic Views μπορεί να είναι ένας συνδυασμός από πίνακες που περιέχουν και χαρακτηριστικά και μετρήσιμα δεδομένα. Επιπλέον μπορούν να περιέχουν και Attribute Views.

  Τα στοιχεία που περιέχονται σε ένα Analytic View είναι Columns, Calculated Columns και Restricted Columns, Variables και Input Parameters.

- *Calculation Views*

  Ένα Calculation View χρησιμοποιείται για τον ορισμό πιο προχωρημένων και σύνθετων αναλύσεων στα δεδομένα του SAP HANA. Τα Calculation Views μπορούν να είναι απλά και να αντικατοπτρίζουν τη λειτουργικότητα που υπάρχει στα Attribute Views και στα Analytic Views. Επιπλέον όμως χρησιμοποιούνται εκεί που οι ανάγκες απαιτούν πιο προχωρημένη λογική η οποία δεν καλύπτεται από τις προηγούμενες κατηγορίες Information Views.

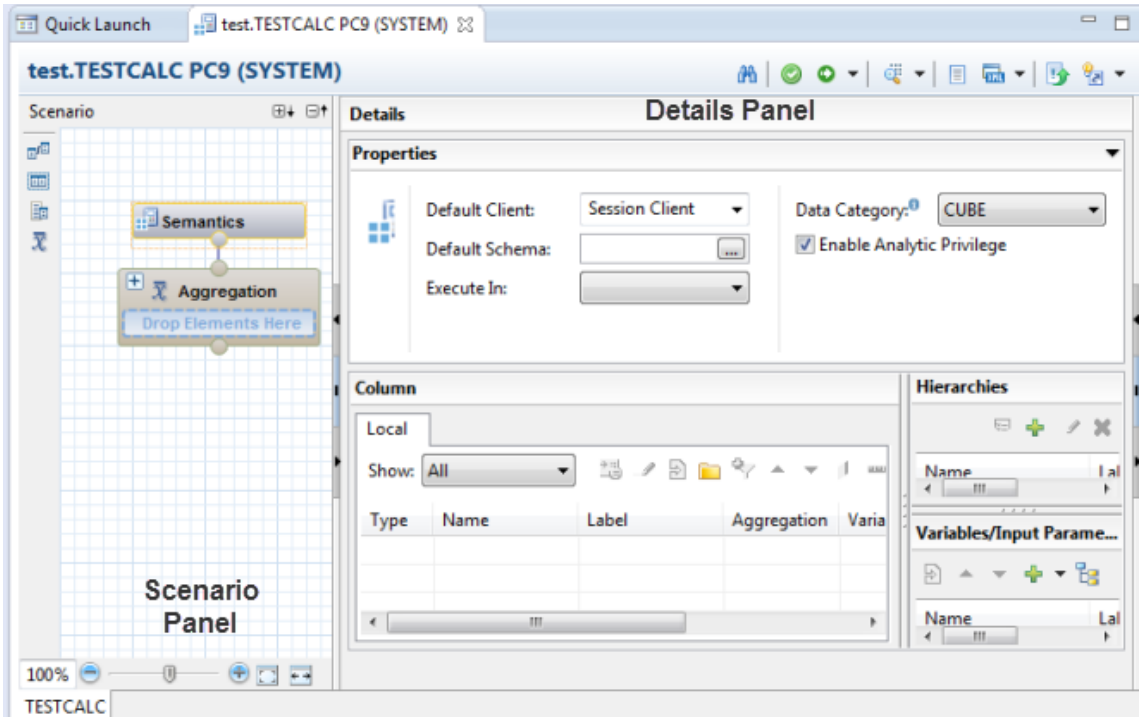
  Για παράδειγμα μπορούν να περιέχουν λογικά επίπεδα, μπορούν σε αντίθεση με τα Analytic Views να περιέχουν μετρήσιμα χαρακτηριστικά από πολλούς πίνακες και να περιλαμβάνουν σύνθετα SQL ερωτήματα. Η πηγή των δεδομένων ενός Calculation View

μπορεί να περιλαμβάνει οποιοδήποτε συνδυασμό Πινάκων, Attribute Views και Analytic Views και σε αυτά μπορούν να δημιουργηθούν επίπεδα Joins, Unions, Projections, και Aggregation.

Τα Calculation Views μπορούν να περιέχουν Attributes, Measures, Calculated Measures, Counters, Hierarchies, Variables και Input Parameters.

Τα Calculation Views μπορούν να περιέχουν μετρήσιμα χαρακτηριστικά και να χρησιμοποιηθούν για πολυδιάστατες αναφορές. Μπορούν να δημιουργηθούν είτε με τη χρήση Γραφικού Περιβάλλοντος είτε με SQL.

Για τη δημιουργία των SAP HANA Information Views υπάρχει το Modeler Perspective στο SAP HANA Studio που αποτελεί το Γραφικό Περιβάλλον Ανάπτυξης καθώς και η δυνατότητα περαιτέρω ανάπτυξης μέσω SQL και SQL Script.



Εικόνα 33. Graphical Calculation View in SAP HANA Studio [30]

3.4 Ανάπτυξη Διαδικασιών (Procedures) στο SAP HANA

SQLScript

Για να ακολουθηθεί το μοντέλο ανάπτυξης Code to Data και να γίνει το Code Pushdown, πρέπει να υπάρχουν τα κατάλληλα εργαλεία για την ανάπτυξη της εφαρμογής στο επίπεδο της βάσης δεδομένων. Η SQLScript είναι μια συλλογή από επεκτάσεις στην standard SQL. Βασικό κίνητρο για τη δημιουργία της, είναι οι περιορισμοί της γλώσσας SQL ως γενικό εργαλείο προγραμματισμού. Αλγόριθμοι οι οποίοι με ευκολία αναπτύσσονται σε κλασσικές γλώσσες προγραμματισμού όπως η ABAP και η Java, είναι πολύ δύσκολο να γραφούν και να βελτιστοποιηθούν για παράλληλη επεξεργασία στη γλώσσα SQL η οποία είναι προσανατολισμένη στα σύνολα δεδομένων.

Εννοιολογικά η SQLScript πλησιάζει τις Stored Procedures του προτύπου SQL, αλλά είναι πολύ πιο ισχυρή και βελτιστοποιημένη για την παράλληλη επεξεργασία των δεδομένων στο επίπεδο της βάσης δεδομένων. Έτσι αποφεύγονται οι μετακινήσεις τεράστιων όγκων δεδομένων από και προς τον application server και ταυτόχρονα αξιοποιούνται οι εξειδικευμένες στρατηγικές παράλληλης εκτέλεσης του SAP HANA. Η SQL Script χρησιμοποιείται εκεί που τα άλλα εργαλεία μοντελοποίησης όπως τα SAP HANA Calculation Views και Analytic Views δεν επαρκούν.

Συνοπτικά η SQLScript επιλύει τα παρακάτω προβλήματα:

- Τα ερωτήματα SQL δεν έχουν δυνατότητες για την έκφραση επιχειρηματικής λογικής (business logic). Παράδειγμα αποτελεί μια σύνθετη μετατροπή νομισματικής ισοτιμίας. Ως αποτέλεσμα τέτοιοι υπολογισμοί είναι αδύνατο να γίνουν στο επίπεδο της βάσης δεδομένων.

- Ένα SQL Query μπορεί να επιστρέψει μόνο ένα αποτέλεσμα τη φορά. Έτσι ο υπολογισμός συσχετιζόμενων αποτελεσμάτων πρέπει να διασπαστεί σε ξεχωριστά queries.

- Με την SQLScript μπορεί να γίνει μίξη αλγορίθμων επιτακτικής λογικής (imperative logic), όπως αλγορίθμους επαναληπτικής προσέγγισης, με δηλωτικούς (declarative) αλγορίθμους της SQL.

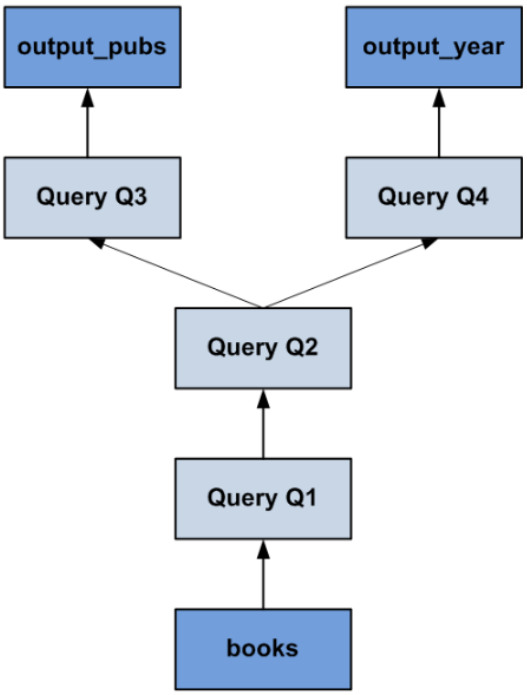
- Η αποσύνθεση ενός ερωτήματος SQL μπορεί να γίνει με τη χρήση SQL Views. Όμως εάν το ερώτημα είναι πολύ σύνθετο, απαιτεί πολλά ενδιάμεσα αποτελέσματα, τα οποία είναι ορατά. Επίσης τα SQL Views δεν δέχονται παραμέτρους, οπότε η επαναχρησιμοποίησή τους είναι περιορισμένη.

Ακολουθεί το παράδειγμα μιας SQLScript Procedure. Σε αυτό ορίζεται μια διαδικασία η οποία είναι μόνο για ανάγνωση. Ως είσοδο δέχεται δύο μεταβλητές, και επιστρέφει δύο μεταβλητές τύπου πίνακα. Επειδή το SAP HANA υποστηρίζει και άλλες γλώσσες στις Procedures, η χρήση της SQL Script δηλώνεται.

```
CREATE PROCEDURE getOutput( IN cnt INTEGER, IN currency VARCHAR(3),
                            OUT output_pubs tt_publishers,
                            OUT output_year tt_years)
     LANGUAGE SQLSCRIPT READS SQL DATA AS
BEGIN
  big_pub_ids = SELECT publisher AS pid FROM books      -- Query Q1
                GROUP BY publisher HAVING COUNT(isbn) > :cnt;
  big_pub_books = SELECT title, name, publisher,        -- Query Q2
                          year, price
                  FROM :big_pub_ids, publishers, books
                  WHERE pub_id = pid AND pub_id = publisher
                  AND crcy = :currency;
  output_pubs = SELECT publisher, name,                 -- Query Q3
                       SUM(price) AS price, COUNT(title) AS cnt
                FROM :big_pub_books GROUP BY publisher, name;
  output_year = SELECT year, SUM(price) AS price,       -- Query Q4
                COUNT(title) AS cnt
                FROM :big_pub_books GROUP BY year;
END;
```

Εικόνα 34. Παράδειγμα διαδικασίας SQLScript [29]

Στη συνέχεια υπάρχουν 4 ερωτήματα όπου το κάθε ένα χρησιμοποιείται από το επόμενο και τα δύο τελευταία επιστρέφονται ως πίνακες. Η διαδικασία αυτή, μπορεί να αναπαρασταθεί και από το παρακάτω γράφημα.



Εικόνα 35. Το γράφημα της παραπάνω διαδικασίας [29]

3.5 Server Side JavaScript

Το Consumption Model που προσφέρουν τα SAP HANA XS επικεντρώνεται σε Server Side εφαρμογές γραμμένες σε JavaScript. Οι εφαρμογές αυτές μπορούν να χρησιμοποιήσουν το πλούσιο και ισχυρό σύνολο εντολών του SAP HANA και να συνδυάσουν εντολές προς τη βάση δεδομένων με εντολές που αφορούν τις αιτήσεις του web.

Οι εφαρμογές του SAP HANA XS επιτρέπουν τον έλεγχο της ροής των δεδομένων, ανάμεσα στο επίπεδο παρουσίασης (presentation layer), που θα μπορούσε να εκτελείται σε ένα web-browser, και στο επίπεδο δεδομένων (data-processing layer) του SAP HANA, εκεί όπου εκτελούνται οι υπολογισμοί πάνω στα δεδομένα σε SQL ή SQL Script. Όπως αναφέρθηκε η ενσωματωμένη αυτή εκτέλεση δίνει πολύ μεγάλη ώθηση στην απόδοση της εφαρμογής.

Χρήση Server Side JavaScript στο SAP HANA XS

Τα JavaScript APIs επιτρέπουν όχι μόνο την ανάγνωση των δεδομένων μέσω web αλλά και την εισαγωγή, ενημέρωση και διαγραφή αυτών. Οι ενέργειες που μπορούν να γίνουν με τα JavaScript APIs είναι [31]:

- Αλληλεπίδραση με το περιβάλλον SAP XANA XS Runtime.
- Άμεση πρόσβαση στις δυνατότητες της βάσης δεδομένων του SAP HANA.
- Αλληλεπίδραση με υπηρεσίες σε καθορισμένους προορισμούς HTTP.

Τα προγράμματα JavaScript βρίσκονται αποθηκευμένα στο Repository του SAP HANA μαζί με τα υπόλοιπα αντικείμενα. Όταν τα προγράμματα JavaScript ενεργοποιούνται, ο κώδικας τους αποθηκεύεται ως Runtime Object.

Παράδειγμα ανάπτυξης Server Side JavaScript

Για τη δημιουργία μιας εφαρμογής JavaScript στο SAP HANA, χρησιμοποιείται το SAP HANA Studio. Τα βήματα που ακολουθούνται είναι [31]:

1. Δημιουργία ενός νέου Project τύπου «XS Project».
2. Δημιουργία ενός root package για την εφαρμογή, έστω `helloxsjs`.
3. Δημιουργία ενός application descriptor αρχείου και τοποθέτησή του στο παραπάνω πακέτο. Το αρχείο αυτό χρησιμοποιούταν για να περιγράψει την διαθεσιμότητα της εφαρμογής στο SAP HANA XS. Πλέον δεν έχει όνομα και περιεχόμενο αλλά μόνο την κατάληξη `.xsapp` . Για λόγους συμβατότητας με παλαιότερες εκδόσεις επιτρέπεται περιεχόμενο σε αυτό αλλά αγνοείται.
4. Δημιουργία ενός application-access αρχείου και το τοποθέτησή του στο πακέτο που χρειάζεται η απόκτηση δικαιωμάτων πρόσβασης. Το αρχείο δεν έχει όνομα παρά μόνο την κατάληξη `.xsaccess` . Τα περιεχόμενα του πρέπει να ακολουθούν τους κανόνες της

σύνταξης JSON. Οι κανόνες πρόσβασης που ορίζονται στο αρχείο .xsaccess αφορούν το πακέτο στο οποίο βρίσκεται καθώς και τα υποπακέτα αυτού.

Στο παράδειγμα αυτό το περιεχόμενο του αρχείου είναι:

```
{
 "exposed" : true,
 "authentication" : { "method": "Form" }
 "authorization": // Optional: see xsprivileges file
  [
    "sap.xse.test::Execute",
    "sap.xse.test::Admin"
  ]
 }
```

Στη συνέχεια πρέπει το αρχείο αυτό να ενεργοποιηθεί

5. Εφόσον το αρχείο .xsaccess περιέχει τη λέξη κλειδί “authorization”, τότε πρέπει να δημιουργηθεί και ένα αρχείο application-privileges με κατάληξη .xsprivileges .

   Στο αρχείο .xsaccess ορίζεται ποια επίπεδα authorization ανατίθενται σε ποιο πακέτο.

   Στο αρχείο .xsprivileges αναφέρονται ποια είναι τα διαθέσιμα επίπεδα authorization σε αυτό το πακέτο.

   Τα περιεχόμενα του αρχείου .xsprivileges ακολουθούν και αυτά το πρότυπο JSON, και οι κανόνες του αναφέρονται στο πακέτο που βρίσκεται και στα υποπακέτα του.

   Στο παράδειγμα αυτό το περιεχόμενο του αρχείου είναι:

```
{
 "privileges" :
```

```
  [
   { "name" : "Execute", "description" : "Basic execution privilege" },
   { "name" : "Admin", "description" : "Administration privilege" }
  ]
}
```

Στη συνέχεια πρέπει το αρχείο αυτό να ενεργοποιηθεί

6. Ο κώδικας με τη λογική της εφαρμογής JavaScript βρίσκεται στα αρχεία με κατάληξη .xsjs. Ο κώδικας αυτός εκτελείται όταν τα SAP HANA XS διαχειρίζονται μια αίτηση HTTP. Στο παράδειγμα το αρχείο ονομάζεται hello.xsjs και ο κώδικάς του είναι ο παρακάτω:

```
$.response.contentType = "text/plain";
$.response.setBody( "Hello, World!");
```

Στη συνέχεια πρέπει το αρχείο αυτό να ενεργοποιηθεί

7. Τελικά το application package του παραδείγματος έχει την παρακάτω μορφή:

```
.
 \
   helloxsjs
             \
               .xsapp
               .xsaccess
               .xsprivileges // optional
               Hello.xsjs
```

8. Το πρόγραμμα είναι προσβάσιμο μέσω web browser. Ο SAP HANA XS Web Server επιτρέπει την προβολή των αποτελεσμάτων αμέσως μετά την ενεργοποίηση. Για το παραπάνω παράδειγμα το URL είναι το παρακάτω:

   http://<SAPHANA_hostname>:80<DB_Instance_Number>/helloxsjs/hello.xsjs

Θέματα Ασφαλείας Server Side JavaScript

Κατά την ανάπτυξη της εφαρμογής είναι απαραίτητο να λαμβάνονται μέτρα για την αποτροπή προβλημάτων ασφαλείας που υπάρχουν γενικότερα στην ανάπτυξη web εφαρμογών. Τα σημαντικότερα θέματα καθώς και η προτεινόμενες λύσεις, που αφορούν τις εφαρμογές Server Side JavaScript του SAP HANA XS, είναι [32]:

- **Packet Sniffing - SSL/HTTPS**

  Πρέπει ο Web Dispatcher να αποδέχεται μόνο αιτήσεις HTTPS, ώστε η επικοινωνία να γίνεται μέσω κρυπτογραφημένου καναλιού, για να αποτραπεί η υποκλοπή ευαίσθητων δεδομένων από κάποιον επιτιθέμενο, ο οποίος παρακολουθεί τα πακέτα του δικτύου (Packet Sniffing).

- **Injection flaws**

  Χρειάζεται προσοχή στις παραμέτρους των ερωτημάτων SQL, που εδώ βρίσκονται στο URL, ώστε να μην μπορεί να αλλάξει το ερώτημα SQL και να έχουμε SQL Injection. Το παρακάτω παράδειγμα δείχνει ένα ευπαθές πρόγραμμα.

  Έστω ότι το URL http://xsengine/customer.xsjs?id=3 εκτελεί το παρακάτω πρόγραμμα του αρχείου `customer.xsjs`.

```
var conn = $.db.getConnection();

var pstmt = conn.prepareStatement( " SELECT * FROM accounts
WHERE custID='" + $.request.parameters.get("id"));

var rs = pstmt.executeQuery();
```

Στην περίπτωση που ένας επιτιθέμενος αλλάξει το URL σε http://xsengine/customer.xsjs?id='3 OR 1=1' , τότε μπορεί να δει όχι μόνο ένα αλλά όλους τους λογαριασμούς στη βάση.

Η λύση είναι να χρησιμοποιούνται prepared statements οι παράμετροι των οποίων ελέγχονται για εγκυρότητα. Με βάση αυτό, το ερώτημα θα πρέπει να γραφεί ως εξής:

```
var conn = $.db.getConnection();

var pstmt = conn.prepareStatement( " SELECT * FROM accounts
WHERE custID=?' );

pstmt.setInt(1, $.request.parameters.get("id"), 10);

var rs = pstmt.executeQuery();
```

- **Cross-site scripting (XSS)**

  Το Cross-site scripting αναφέρεται στην εκτέλεση κώδικα JavaScript που έβαλε σε μια σελίδα ένας επιτιθέμενος με σκοπό να εκτελεστεί στον υπολογιστή στόχο.

  Η ευπάθεια αυτή έχει τις εξής μορφές:

  - Reflected (non-persistent)

    Ο κακόβουλος κώδικας επηρεάζει μόνο ένα χρήστη στον τοπικό του υπολογιστή

  - Stored (persistent)

    Ο κακόβουλος κώδικας είναι αποθηκευμένος στον Server και επηρεάζει όλους τους χρήστες που επισκέπτονται τη μολυσμένη σελίδα.

  Το SAP HANA XS JavaScript API δεν έχει κάποια βιβλιοθήκη για τον έλεγχο HTML, οπότε προτείνεται η χρήση της βιβλιοθήκης JavaScript ESAPI. Επιπλέον η επικοινωνία πρέπει να γίνεται με standard πρωτόκολλα όπως είναι το OData ή το JSON.

- **Broken authentication and session management**

Το πρόβλημα αυτό αναφέρεται στην προσπάθεια ενός επιτιθέμενου να πραγματοποιήσει είσοδο ως χρήστης της εφαρμογής, βασιζόμενος σε ευπάθεια του μηχανισμού ταυτοποίησης χρήστη καθώς και του μηχανισμού συνεδριών.

Για την αποφυγή του προτείνεται να χρησιμοποιείται μόνο ο παρεχόμενος από το SAP HANA μηχανισμός. Επίσης προτείνεται η χρήση της λέξης κλειδί "authentication" στο αρχείο `.xsaccess`, ώστε να καθορισθεί ο μηχανισμός ταυτοποίησης καθώς και τα δικαιώματα των χρηστών στην εφαρμογή.

- **Insecure direct object references**

  Όλα τα αντικείμενα πρέπει να έχουν κάποιο μηχανισμό προστασίας ώστε να μην είναι δυνατή η άμεση κλήση τους από κάποιον επιτιθέμενο. Αυτό επιτυγχάνεται με χρήση της λέξης κλειδί "authentication" στο αρχείο `.xsaccess`, και τον καθορισμό των δικαιωμάτων στο αντικείμενο.

- **Cross-site request forgery (XSRF)**

  Το πρόβλημα αυτό υπάρχει όταν δύο ή περισσότερα web sites μοιράζονται το ίδιο browser session, ειδικά όταν αυτά χρησιμοποιούν cookies για την αποθήκευση των στοιχείων εισόδου. Έτσι μπορεί, για παράδειγμα, κάποιος επιτιθέμενος μέσω ενός συνδέσμου να οδηγήσει τον χρήστη σε URL του site που έχει την ευπάθεια, και να προβεί σε ενέργειες για λογαριασμό του μέσω κλήσης HTTP GET με παραμέτρους. Πολλές φορές η συγκεκριμένη επίθεση είναι δύσκολο να εντοπιστεί γιατί η ενέργεια αυτή δεν ξεχωρίζει από τις συνηθισμένες ενέργειες του χρήστη.

  Στο SAP HANA για την προστασία από αυτή την επίθεση, χρησιμοποιείται ένα τυχαίο TOKEN το οποίο περιλαμβάνεται σε κάθε POST κλήση, ώστε ο Server να μπορεί να το επαληθεύσει. Το TOKEN δημιουργείται τυχαία και με ασφάλεια και είναι αδύνατο για έναν επιτιθέμενο να το προβλέψει.

Για να προστατευθεί η εφαρμογή στο SAP XANA XS από επιθέσεις XSRF, πρέπει στο αρχείο .xsacess να δοθεί η τιμή `true` στην παράμετρο `prevent_xsrf`, όπως φαίνεται παρακάτω.

```
{
    "prevent_xsrf" : true
}
```

Η επιλογή αυτή επιβάλει τους ελέγχους για την ύπαρξη έγκυρου Security Token σε κάθε Session για κάθε κλήση. Έτσι μπορεί να επιβεβαιωθεί ότι οι αιτήσεις γίνονται μόνο από τον πραγματικό χρήστη της εφαρμογής. Πρέπει να σημειωθεί ότι η προεπιλεγμένη τιμή αυτής της μεταβλητής είναι false, οπότε εάν δεν δηλωθεί ρητά η τιμή της σε true, δεν υπάρχει προστασία από επιθέσεις XSRF.

Στο παρακάτω παράδειγμα υπάρχει ο κώδικας μιας client html σελίδας που πραγματοποιεί κλήση AJAX και χρησιμοποιεί το XSRF Security Token [31].

```
<html>
<head>
   <title>Example</title>
   <script id="sap-ui-bootstrap" type="text/JavaScript"
      src="/sap/ui5/1/resources/sap-ui-core.js"
      data-sap-ui-language="en"
      data-sap-ui-theme="sap_goldreflection"
      data-sap-ui-libs =
      "sap.ui.core,sap.ui.commons,sap.ui.ux3,sap.ui.table">
    </script>
    <script type="text/JavaScript"
        src="/sap/ui5/1/resources/jquery-sap.js"/>
    <script>
      function doSomething() {
```

```
    $.ajax({
    url: "logic.xsjs",
    type: "GET",
    beforeSend: function(xhr) {
       xhr.setRequestHeader("X-CSRF-Token", "Fetch");
    },
    success: function(data, textStatus, XMLHttpRequest) {
       var token = XMLHttpRequest.getResponseHeader('X-CSRF-
Token');
       var data = "somePayLoad";
       $.ajax({
         url: "logic.xsjs",
         type: "POST",
         data: data,
         beforeSend: function(xhr) {
           xhr.setRequestHeader("X-CSRF-Token", token);
         },
         success: function() {
           alert("works");
         },
         error: function() {
           alert("works not");
         }
       });
      }
   });
  }
 </script>
 </head>
```

```
<body>
 <div>
  <a href="#" onClick="doSomething();">Do something</a>
 </div>
</body>
</html>
```

- **Μη ασφαλής αποθήκευση στη συσκευή του πελάτη**

  Η εφαρμογή που εκτελείται στον πελάτη, μπορεί να αποθηκεύσει διάφορα δεδομένα στη συσκευή κάποια από τα οποία μπορεί να έχουν πληροφορίες ασφαλείας, πχ τα στοιχεία εισόδου του χρήστη.

  Για να αποφευχθεί κάποιος να αποκτήσει πρόσβαση σε αυτά, πρέπει να αποθηκευθούν μόνο σε κρυπτογραφημένη μορφή.

- **Μη σωστές ρυθμίσεις ασφαλείας του HANA Server**

  Καθόλου ή μη επαρκής υλοποίηση μηχανισμού ασφαλείας του HANA στην εφαρμογή εκθέτει το σύστημα σε επιθέσεις που σχετίζονται με αυτό.

  Πρέπει να ενσωματωθεί στην εφαρμογή ο μηχανισμός ταυτοποίησης του SAP HANA, καθώς και να χρησιμοποιηθεί το SAP HANA XS cookie και το Session Handling που προσφέρει. Οι προγραμματιστές πρέπει επίσης να είναι προσεκτικοί και να ελέγχουν ποια server paths εκτίθενται μέσω του HTTP προς το internet και ποια από αυτά απαιτούν ελεγχόμενη πρόσβαση και ταυτοποίηση.

- **Missing restrictions on URL Access**

  Μη εξουσιοδοτημένοι χρήστες δεν πρέπει να έχουν πρόσβαση σε URLs τα οποία περιέχουν ευαίσθητες πληροφορίες. Πέρα από αυτά που περιγράφονται στο «Insecure direct object references», πρέπει επιπρόσθετα να εξετάζεται εάν ο χρήστης έχει δικαιώματα στο URL πριν από την εκτέλεση του κώδικα JavaScript που αντιστοιχεί σε αυτό το URL, προσθέτοντας τον αντίστοιχο κώδικα για τον έλεγχο σε κάθε αρχείο JavaScript.

- **Μη επαρκής προστασία του Επιπέδου Μεταφοράς (Transport Layer)**

  Πρέπει να ενεργοποιηθεί το Transport Layer Protection στο SAP HANA XS, ώστε το επίπεδο μεταφοράς να μην μπορεί να παρακολουθηθεί από κάποιον επιτιθέμενο.

- **Invalid redirects and forwards**

  Οι εφαρμογές Ιστού συχνά ανακατευθύνουν τους χρήστες σε άλλες σελίδες είτε εξωτερικές είτε εσωτερικές. Το πρόβλημα εντοπίζεται όταν ο επιτιθέμενος αποκτήσει τον έλεγχο της διαδικασίας είτε για να ανακατευθύνει τους χρήστες σε δικό του URL για σκοπούς phishing των στοιχείων τους, είτε για να αποκτήσει πρόσβαση σε URL όπου δεν θα έπρεπε να έχει πρόσβαση.

  Για αυτό προτείνονται:

  - Εάν είναι δυνατόν να αποφεύγεται η ανακατεύθυνση, ειδικά σε εξωτερικά URLs τα οποία δεν βρίσκονται υπό τον έλεγχο της εφαρμογής.
  - Δεν πρέπει η ανακατεύθυνση να είναι παράμετρος στο URL, γιατί είναι πολύ απλό για κάποιον επιτιθέμενο να την αλλάξει.
  - Πριν την ανακατεύθυνση θα πρέπει να γίνεται έλεγχος εάν ο προορισμός βρίσκεται σε λίστα επιτρεπτών προορισμών. Εναλλακτικά πρέπει να αποφευχθεί η απευθείας είσοδος από το χρήστη σχετικά με τον τελικό προορισμό, όπως φαίνεται στο παρακάτω παράδειγμα:

```
var destination = $.request.parameters.get("dest");

switch (destination) {

  case "1": $.response.headers.set( "location" ,
"http://FirstWhitelistedURL.com"); break;

  case "2": $.response.headers.set("location",
"http://SecondWhitelistedURL.com"); break;
```

```
   default: $.response.headers.set("location",
"http://DefaultWhitelistedURL.com");}
```

- **Θέματα ασφαλείας κατά την επεξεργασία XML**

Κατά την επεξεργασία των αρχείων XML, είτε αφορά την είσοδο είτε την έξοδο αυτών, μπορεί να δημιουργηθούν κενά στην ασφάλεια του συστήματος.

Η πιο συνηθισμένη επίθεση XML είναι η επίθεση τύπου Billion Laughs. Κατά την επίθεση αυτή, δίνεται ως πηγή xml αρχείο κατά την ανάγνωση του οποίου, εξαντλούνται γρήγορα οι υπολογιστικοί πόροι του server με αποτέλεσμα σε κάποιες περιπτώσεις ανάλογα με τη ρύθμιση του να καταρρέει ολόκληρος.

Στο παρακάτω παράδειγμα υπάρχει ο κακόβουλος κώδικας ενός τέτοιου αρχείου μεγέθους περίπου 1KB:

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

Εικόνα 36 Παράδειγμα κώδικα Billion Laughs Attack [33]

Στο παράδειγμα αυτό κάθε οντότητα περιέχει 100 στοιχεία της προηγούμενης, οπότε όταν στο τέλος ζητείται η οντότητα `&lol9;` της ρίζας `lolz` , τότε πρέπει να φορτωθούν συνολικά 10^9 `lol` τα οποία καταλαμβάνουν περίπου 3 GB κύριας μνήμης. Καθώς αυξάνονται οι οντότητες, εκθετικά αυξάνονται και οι απαιτήσεις σε μνήμη, οδηγώντας σύντομα τον server σε κατάρρευση.

Για να αποφευχθούν επίθεσεις που αφορούν επεξεργασία XML σε μια εφαρμογή SAP HANA, προτείνονται:

- Όταν διαβάζουμε XML από αναξιόπιστες πηγές, πρέπει να είναι απενεργοποιημένα τα DTD Processing και Entity Expansion, εκτός εάν αυτό είναι απολύτως απαραίτητο.
- Για να αποφευχθεί η εισαγωγή στο Server μη εγκεκριμένων αρχείων XML πρέπει να περιοριστεί η δυνατότητα να επεξεργάζονται αρχεία ή URLs ακόμα και εάν αυτά προέρχονται από αξιόπιστες πηγές.
- Να τοποθετηθούν λογικά όρια στις παραμέτρους των μηχανών επεξεργασίας XML, όπως είναι η μέγιστη μνήμη που μπορούν να χρησιμοποιήσουν, το πόσες εμφωλευμένες οντότητες μπορούν να υπάρχουν και το μέγιστο μέγεθος των ονομάτων των οντοτήτων και των ιδιοτήτων.

3.6 Σύνδεση εφαρμογών με το SAP HANA

Το SAP HANA παρέχει διάφορες δυνατότητες για να μπορούν να επικοινωνήσουν οι εφαρμογές-πελάτες με αυτό. Παρέχει πρόσβαση Web-based μέσω ενός μοντέλου κατανάλωσης, το οποίο δίνει πρόσβαση στις εξωτερικές εφαρμογές μέσω του πρωτοκόλλου HTTP. Αυτό μπορεί να γίνει είτε μέσω υπηρεσιών ODATA, είτε μέσω Rest Web Services, που θα δημιουργηθούν με τη χρήση SAP HANA XS γραμμένα με Server Side JavaScript. Εναλλακτικά μπορούν να συνδεθούν εφαρμογές σε αυτό μέσω κλασσικών οδηγών JDBC/ODBC.

OData

Το πρωτόκολλο OData (Open Data Protocol), είναι ένα Restful πρωτόκολλο για την ανταλλαγή δεδομένων μέσω HTTP. Αρχικά σχεδιάστηκε και υλοποιήθηκε από τη Microsoft για την Cloud Πλατφόρμα Azure, και στη συνέχεια υιοθετήθηκε από την SAP για την επικοινωνία εξωτερικών εφαρμογών με το SAP NetWeaver.

Το ODATA είναι ένα πρωτόκολλο για την ανταλλαγή δεδομένων από και προς τις βάσεις δεδομένων. Αυτό σημαίνει ότι με αυτό μπορεί να υλοποιηθούν εύκολα οι λειτουργίες CRUD (Create Read Update Delete) μιας απομακρυσμένης εφαρμογής – πελάτη με τη βάση δεδομένων.

Το OData είναι βασισμένο πάνω σε ήδη υπάρχοντα, διαδεδομένα και δοκιμασμένα πρωτόκολλα. Αυτά είναι τα [34]:

- HTTP (Hypertext Transfer Protocol)

  Είναι το κυρίαρχο πρωτόκολλο του Internet και του Παγκοσμίου Ιστού και το OData στηρίχθηκε σε αυτό.

- Atom (Atom Syndication Format)

  Το Atom είναι γνωστό για την χρήση σε RSS Feeds, ώστε να μοιράζει το περιεχόμενο σε άλλους μέσω HTTP. Επιλέχθηκε γιατί η λειτουργία του RSS Feed είναι παρόμοια με μια βάση δεδομένων. Ένα RSS Feed που περιέχει μια συλλογή από posts είναι παρόμοιο με έναν πίνακα, και ένα blog post είναι παρόμοιο με μια εγγραφή. Οι ιδιότητες που περιέχει το post όπως είναι ο τίτλος, το σώμα και η ημερομηνία δημοσίευσης, μοιάζουν με τις στήλες ενός πίνακα.

- REST (Representational state transfer)

  Το OData δημιουργήθηκε με βάση την προδιαγραφή REST. Ένα Restful Web Service είναι ένα Web Service το οποίο υλοποιείται με βάση το HTTP και τις προδιαγραφές REST. Είναι μια συλλογή από πηγές, με τέσσερις πτυχές:

  - Το βασικό URI για το Web Service, όπως είναι το http://example.com/resources/
  - Το Internet media type που υποστηρίζεται από το Web Service. Συνήθως είναι JSON ή XML αλλά μπορεί να είναι και άλλοι τύποι.
  - Το σύνολο των λειτουργιών που υποστηρίζεται από το πρωτόκολλο HTTP - GET, PUT, POST και DELETE.
  - Οι κλήσεις API πρέπει να γίνονται με βάση το hypertext.

Σκοπός του OData είναι να ορίσει ένα αφηρημένο μοντέλο δεδομένων και ένα πρωτόκολλο, τα οποία σε συνδυασμό να επιτρέπουν σε οποιοδήποτε πρόγραμμα – πελάτη να έχει πρόσβαση σε δεδομένα οποιασδήποτε πηγής δεδομένων. Οι πιθανοί πελάτες είναι οι Web Browsers, έξυπνα κινητά, εργαλεία Business Intelligence, και διάφορες εφαρμογές σε οποιαδήποτε γλώσσα. Οι πηγές δεδομένων μπορεί να είναι Βάσεις Δεδομένων, Συστήματα Διαχείρισης Περιεχομένου, Cloud Data ή Επιχειρηματικές Εφαρμογές.

Η προσέγγιση OData σχετικά με την ανταλλαγή δεδομένων περιέχει τα ακόλουθα στοιχεία[32]:

- **OData data model**

  Δίνει ένα γενικό τρόπο για την οργάνωση και την περιγραφή των δεδομένων. Το OData χρησιμοποιεί το Entity 1 Data Model (EDM).

- **OData protocol**

  Επιτρέπει σε μια εφαρμογή-πελάτη να επικοινωνήσει με μια υπηρεσία OData. Το πρωτόκολλο OData είναι ένα σύνολο ενεργειών που περιλαμβάνουν τα REST-based CRUD, μαζί με μια γλώσσα ερωτημάτων του OData. Τα δεδομένα της υπηρεσίας OData στέλνονται είτε σε μορφή JSON, είτε σε XML-based Atom format.

- **OData client libraries**

  Είναι βιβλιοθήκες για τα προγράμματα-πελάτες, ώστε να επιταχύνουν και να απλοποιούν την εργασία των προγραμματιστών για την επικοινωνία της εφαρμογής τους με την υπηρεσία OData. Υπάρχουν διαθέσιμες βιβλιοθήκες OData για τις περισσότερες γλώσσες και πλατφόρμες προγραμματισμού όπως είναι οι Java, JavaScript, .NET, PHP, Ruby, Android / Java, IOS / Objective C. Η χρήση αυτών των βιβλιοθηκών είναι προαιρετική, αφού ο προγραμματιστής μπορεί να τις αντιμετωπίσει και ως κλασσικά Rest Web Services, χάνοντας όμως τους επιπλέον αυτοματισμούς του OData.

- **OData services**

Είναι ένα τελικό σημείο (End Point) το οποίο επιτρέπει την πρόσβαση στα δεδομένα. Η υπηρεσία OData, υλοποιεί το πρωτόκολλο OData, και χρησιμοποιεί το Data Access Layer για να αντιστοιχίσει τα δεδομένα μεταξύ της πηγής δεδομένων και ενός format που μπορεί να καταλάβει ο πελάτης που κάνει την αίτηση.

Το SAP HANA XS υποστηρίζει την έκδοση 2.0 του OData. Η Κωδικοποίηση Γλώσσας που υποστηρίζεται είναι αποκλειστικά η UTF-8. Για ενέργειες ανάγνωσης, τα δεδομένα μπορούν να είναι είτε σε μορφή XML, είτε σε μορφή JSON. Για ενέργειες που τροποποιούν τα δεδομένα (πχ CREATE, UPDATE, DELETE), το SAP HANA XS υποστηρίζει μόνο τη μορφή JSON ("`content-type: application/json`") [32].

Με τη χρήση των υπηρεσιών OData στο SAP HANA, οι εφαρμογές αξιοποιώντας την άμεση πρόσβαση που προσφέρει στις συναρτήσεις υπολογισμού και στα δεδομένα, μπορούν να πραγματοποιήσουν το μοντέλο Code To Data και να αυξήσουν κατακόρυφα τις συνολική τους απόδοση.

Οι υπηρεσίες OData δηλώνονται στο SAP HANA στα αρχεία με κατάληξη `.xsodata`. Το αρχείο αυτό πρέπει να έχει τουλάχιστον την καταχώρηση `Service { }`, η οποία θα δημιουργήσει ένα πλήρως λειτουργικό OData Service. Στο παρακάτω παράδειγμα δημιουργείται η υπηρεσία OData για τον πίνακα 'myTable' του DB Schema 'mySchema' [32].

```
service {
  "mySchema"."myTable" as "MyTable";
}
```

Από προεπιλογή όλες οι υπηρεσίες OData του SAP HANA είναι εγγράψιμες, δηλαδή μπορούν να εκτελεστούν σε αυτές αιτήσεις CREATE, UPDATE ή DELETE. Εάν χρειαστεί να απαγορευθεί κάποια ενέργεια από την υπηρεσία, τότε αυτή πρέπει να δηλωθεί ρητά με το όνομα της ενέργειας και τη λέξη κλειδί FORBIDDEN. Στο ακόλουθο παράδειγμα [32] η υπηρεσία είναι μόνο για ανάγνωση, αφού απαγορεύονται οι ενέργειες CREATE, UPDATE και DELETE.

```
service {
```

```
  "sap.test::myTable"
  create forbidden
  update forbidden
  delete forbidden;
}
```

Η κλήση της υπηρεσίας OData, μπορεί να γίνει από οποιαδήποτε πλατφόρμα υποστηρίζει το πρωτόκολλο αυτό. Ακολουθούν παραδείγματα κλήσεων με HTTP Request [27].

1. **Αίτηση για πηγές**

Θα χρησιμοποιηθεί το παράδειγμα της υπηρεσίας TripPin, η οποία εξομοιώνει ένα σύστημα Trip Management. Με την ακόλουθη κλήση ζητείται μια λίστα με τα στοιχεία του συνόλου οντοτήτων People. Η απάντηση θα είναι μια λίστα με τις οντότητες του People σε μορφή JSON.

HTTP Request

```
GET http://services.odata.org/v4/TripPinServiceRW/People
HTTP/1.1
```

2. **Αίτηση για συγκεκριμένη πηγή**

Μπορούμε να ζητήσουμε συγκεκριμένη οντότητα του People με χρήση κλειδιών. Έστω ότι κλειδί είναι το username. Η ακόλουθη κλήση θα φέρει τα στοιχεία της οντότητας με κλειδί 'russelwhyte' εάν αυτή υπάρχει.

HTTP Request

```
GET
http://services.odata.org/v4/TripPinServiceRW/People('russellwhyte
')
HTTP/1.1
```

3. Ερωτήματα

Το OData υποστηρίζει σύνθετα ερωτήματα πάνω στα σύνολα οντοτήτων. Με το επόμενο παράδειγμα θα επιστραφούν τα 2 πρώτα άτομα που έχουν τουλάχιστον 1 ταξίδι που κόστισε πάνω από 3000 και θα εμφανίσει μόνο το όνομα και το επίθετο.

HTTP Request

```
GET   http://services.odata.org/v4/TripPinServiceRW/People?$top=2 &
$select=FirstName, LastName & $filter=Trips/any(d:d/Budget gt 3000)
HTTP/1.1
```

4. Δημιουργία Νέου

Στο παρακάτω παράδειγμα, δημιουργείται νέα οντότητα στο People. Αυτή είναι POST Request και η αναπαράσταση των στοιχείων είναι σε μορφή JSON.

HTTP Request

```
POST
http://services.odata.org/v4/(S(34wtn2c0hkuk5ekg0pjr513b))/TripPinS
erviceRW/People HTTP/1.1
Content-Length: 428
Content-Type: application/json
{
   "UserName":"lewisblack",
   "FirstName":"Lewis",
   "LastName":"Black",
   "Emails":[
        "lewisblack@example.com"
   ],
   "AddressInfo":[
           {
```

```
            Address: "187 Suffolk Ln.",
            City: {
                CountryRegion: "United States",
                Name: "Boise",
                Region: "ID"
            }
        }
    ],
    "Gender": "Male",
    "Concurrency":635519729375200400
}
```

Η υπηρεσία θα απαντήσει με το παρακάτω μήνυμα:

Response

```
HTTP/1.1 201 Created
Content-Length: 652
Content-Type:
 application/json;odata.metadata=minimal;odata.streaming=true;IEEE7
54Compatible=false;charset=utf-8
ETag: W/"08D1D3800FC572E3"
Location:
http://services.odata.org/V4/(S(34wtn2c0hkuk5ekg0pjr513b))/TripPinS
erviceRW/People('lewisblack')
```

5. Συσχετίσεις Οντοτήτων

Το OData υποστηρίζει την εκτέλεση ενεργειών σε συσχετισμένες οντότητες. Στο παρακάτω παράδειγμα, ο χρήστης 'russelwhite' προσκαλεί το χρήστη 'lewisblack' στο ταξίδι του, συσχετίζοντας το ταξίδι αυτό με τον 'lewisblack'.

HTTP Request

```
POST
http://services.odata.org/v4/(S(34wtn2c0hkuk5ekg0pjr513b))/TripPinS
erviceRW/People('lewisblack')/Trips/$ref HTTP/1.1
Content-Length: 123
Content-Type: application/json

{

   "@odata.id":"http://services.odata.org/V4/(S(34wtn2c0hkuk5ekg0

   pjr513b))/TripPinServiceRW/People('russellwhyte')/Trips(0)"

}
```

6. Κλήση Συνάρτησης

Το πρωτόκολλο OData υποστηρίζει την κλήση συναρτήσεων τα οποία πραγματοποιούν σύνθετες λειτουργίες. Στο παρακάτω παράδειγμα καλείται η συνάρτηση GetInvolvedPeople η οποία επιστρέφει εκείνους που σχετίζονται με ένα συγκεκριμένο ταξίδι.

HTTP Request

```
GET
http://services.odata.org/v4/(S(34wtn2c0hkuk5ekg0pjr513b))/TripPinS
erviceRW/People('russellwhyte')/Trips(0)/Microsoft.OData.SampleServ
ice.Models.TripPin.GetInvolvedPeople() HTTP/1.1
```

Σύνδεση μέσω οδηγών JDBC / ODBC

Το SAP HANA υποστηρίζει τη σύνδεση εφαρμογών σε αυτό για την ανάκτηση και την εγγραφή δεδομένων και μέσω των κλασσικών οδηγών JDBC / ODBC.

Συγκεκριμένα το SAP HANA παρέχει τους παρακάτω οδηγούς:

- JDBC για προγράμματα Java
- ODBC για σύνδεση εφαρμογών 32 & 64 bit
- SAP HANA Data Provider για την πλατφόρμα Microsoft .NET
- ODBO για τη σύνδεση σε αυτό εφαρμογών ανάλυσης και εκτέλεσης ερωτημάτων MDX. Παράδειγμα τέτοιας εφαρμογής είναι το Microsoft Excel 2013, το οποίο επιτρέπει στους χρήστες να καταναλώνουν τα Analytic Views του SAP HANA και να δημιουργούν Pivot Tables και γραφήματα.

Στο παρακάτω παράδειγμα υπάρχει ο κώδικας Java για τη σύνδεση μέσω JDBC σε SAP HANA Server με όνομα `myhdb`, με όνομα χρήστη `myname` και κωδικό `mysecret`, και την εκτέλεση ενός απλού ερωτήματος. Για τη λειτουργία του παραδείγματος πρέπει να υπάρχει στο classpath ο JDBC οδηγός `ngdbc.jar`. [31]

```
import Java.sql.*;

public class jdemo {

 public static void main(String[] argv){

  Connection connection = null;

  try{

      connection = DriverManager.getConnection(
"jdbc:sap://myhdb:30715/?autocommit=false",myname,mysecret);

   } catch (SQLException e) {

     System.err.println("Connection Failed. User/Password Error?");

     return;
```

```
  }
  if (connection != null) {
    try {
      System.out.println("Connection to HANA successful!");
      Statement stmt = connection.createStatement();
      ResultSet resultSet = stmt.executeQuery("Select 'hello world'
from dummy");
      resultSet.next();
      String hello = resultSet.getString(1);
      System.out.println(hello);
    } catch (SQLException e) {
      System.err.println("Query failed!");
    }
  }
}
```

3.7 Mobile Clients

Οι mobile εφαρμογές μπορούν να συνδεθούν με το SAP HANA μέσω του πρωτοκόλλου ODATA. Η εφαρμογές μπορούν να είναι γραμμένες είτε στην πλατφόρμα ανάπτυξης για την αντίστοιχη συσκευή (Native Apps), είτε να είναι υβριδικές εφαρμογές HTML/JavaScript (Hybrid Apps), οι οποίες μπορούν να λειτουργούν σε πολλές συσκευές μέσω του Web Browser της εκάστοτε συσκευής.

Native Apps

Η ανάπτυξη Native Εφαρμογών που επικοινωνούν με το SAP HANA Cloud, γίνεται ανεξάρτητα από αυτό με τα εργαλεία που επιθυμεί ο προγραμματιστής για τη συγκεκριμένη πλατφόρμα της εφαρμογής του. Η native εφαρμογή θα επικοινωνεί με το SAP HANA Cloud μέσω των κλήσεων στα OData Services και XSJS Services που προσφέρει το SAP HANA Cloud.

Για την κλήση των OData Services, ο προγραμματιστής μπορεί είτε να χρησιμοποιήσει βιβλιοθήκη της πλατφόρμας του για OData Services, είτε να τα καλέσει ως κλασικά Rest Web Services. Για την ταυτοποίηση των χρηστών απέναντι στο SAP HANA Cloud, μπορεί να χρησιμοποιηθεί είτε Basic HTTP Authentication, είτε να χρησιμοποιηθεί το πρωτόκολλο OAuth 2.0, με το τελευταίο να είναι το προτεινόμενο για λόγους που θα αναλυθούν σε επόμενο κεφάλαιο.

Hybrid Apps – SAPUI5

Οι υβριδικές εφαρμογές αποτελούν συνδυασμό τεχνολογιών ιστού, δηλαδή συνδυάζουν κυρίως κώδικα HTML, CSS και JavaScript. Εκτελούνται είτε ως εφαρμογές ιστού μέσω του προγράμματος περιήγησης της εκάστοτε συσκευής, είτε ως υβρίδια εφαρμογών όπου το μεγαλύτερο μέρος του κώδικα είναι γραμμένο σε HTML, CSS και JavaScript αλλά ένα μικρό μέρος που αποτελεί την εφαρμογή wrapper είναι σε γραμμένο σε native πλατφόρμα.

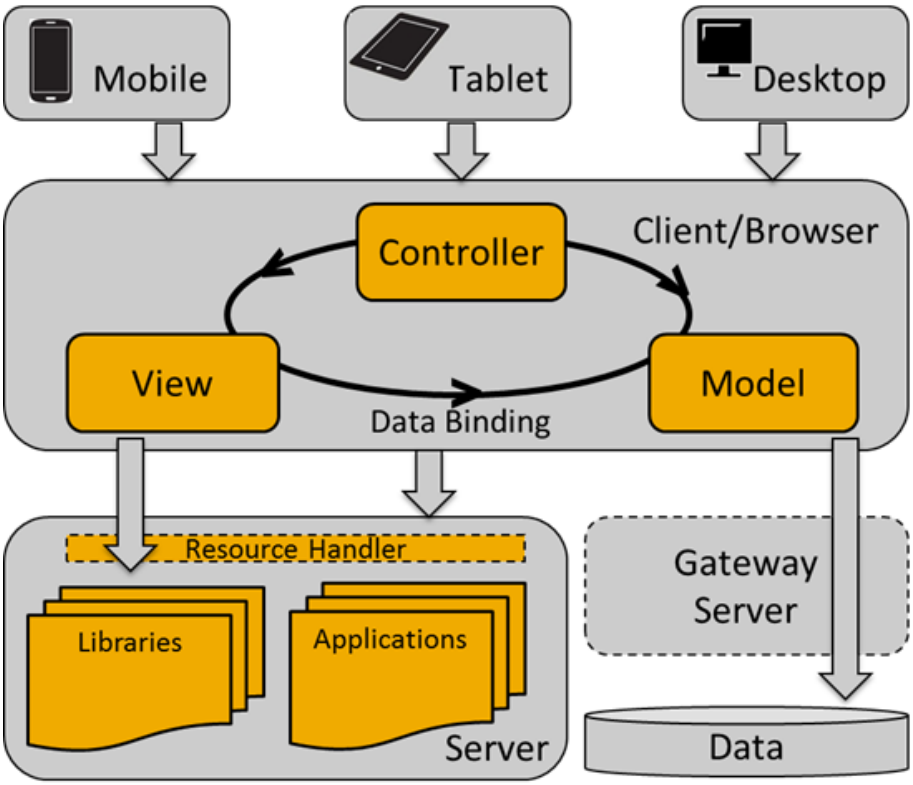
Οι mobile εφαρμογές που εκτελούνται εξ ολοκλήρου στον Web Browser, είναι online web εφαρμογές οι οποίες έχουν responsive layout, δηλαδή προσαρμόζονται στη διάσταση της οθόνης της εκάστοτε συσκευής. Μεγάλο πλεονέκτημα αυτών είναι η δημιουργία και η συντήρηση μιας μόνο εφαρμογής η οποία μπορεί να εκτελεστεί από Κινητά Τηλέφωνα, Ταμπλέτες και Υπολογιστές. Στα αρνητικά τους είναι οι περιορισμένες δυνατότητες για λειτουργία χωρίς σύνδεση (offline), οι περιορισμοί στη χρήση native δυνατοτήτων των συσκευών και η μη δυνατότητα διάθεσης μέσω των online καταστημάτων κινητών εφαρμογών.

Οι Υβριδικές Mobile εφαρμογές λόγω του ότι εκτελούνται μέσα σε εφαρμογή wrapper, έχουν τη δυνατότητα να εκτελεστούν χωρίς σύνδεση, αφού όλος ο κώδικας βρίσκεται στη συσκευή και η σύνδεση απαιτείται μόνο για την ανταλλαγή δεδομένων με τον κεντρικό Server της εφαρμογής εάν και

όταν αυτό απαιτείται. Μέσω βιβλιοθηκών όπως είναι το Apache Cordova μπορούν να χρησιμοποιήσουν Native δυνατότητες των συσκευών στις οποίες εκτελούνται. Οι υβριδικές εφαρμογές μπορούν να διατεθούν μέσω των online καταστημάτων κινητών εφαρμογών. Στα αρνητικά τους σε σχέση με τις online εφαρμογές είναι η χρήση, έστω και μικρού μέρους, native κώδικα ο οποίος πρέπει να αναπτυχθεί και να συντηρηθεί σε κάθε πλατφόρμα, ενώ σε σχέση με τις εξ ολοκλήρου native λύσεις υστερούν σε απόδοση και δεν μοιάζουν εμφανισιακά με τις υπόλοιπες Native εφαρμογές της συσκευής.

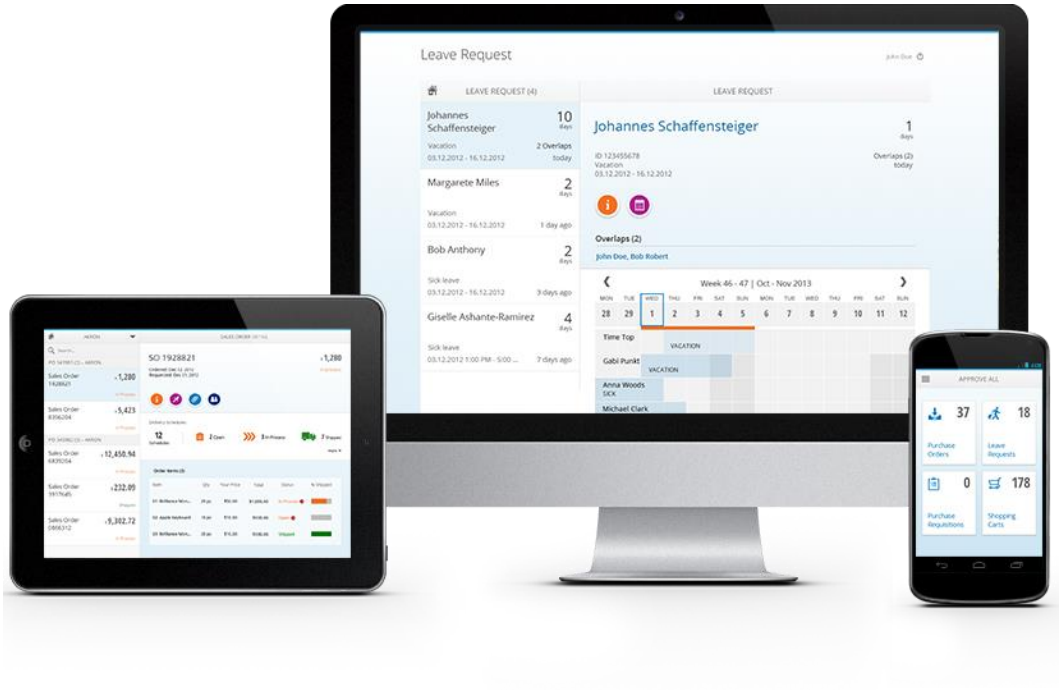
Για να επικοινωνήσει μια εφαρμογή HTML / JS με το SAP HANA πρέπει να κάνει κλήσεις μέσω βιβλιοθηκών JavaScript στις OData υπηρεσίες του SAP HANA. Οι βιβλιοθήκες αυτές μπορεί να είναι οι κλασσικές κλήσεις βιβλιοθηκών JavaScript, όπως είναι το JQuery, σε Υπηρεσίες Ιστού REST. Για τη βέλτιστη λειτουργία της εφαρμογής με το SAP HANA προτείνεται η χρήση κάποιας βιβλιοθήκης JavaScript που υποστηρίζει εγγενώς το πρωτόκολλο OData.

Επειδή η αρχιτεκτονική των εφαρμογών για το SAP HANA είναι Model – View – Controller, όπου το Model και ο Controller βρίσκεται μέσα στο SAP HANA ενώ στον εκάστοτε πελάτη εκτελείται το View, η SAP δημιούργησε μια βιβλιοθήκη HTML / JavaScript η οποία ονομάζεται SAP UI5 για να διευκολύνει και να επιταχύνει την ανάπτυξη των εφαρμογών.



Εικόνα 37. Αρχιτεκτονική εφαρμογής SAPUI5 [35]

Το SAP UI5 είναι μια συλλογή από αντικείμενα και χειριστήρια για τη δημιουργία γραφικού περιβάλλοντος, το οποίο είναι responsive, και επικοινωνεί απευθείας με τα OData Services του SAP HANA. Επίσης μπορεί να προσαρμόζεται αυτόματα σε Υπολογιστή, Ταμπλέτα και Κινητό Τηλέφωνο. Διατίθεται σε δύο εκδόσεις, το SAP UI5 που είναι η εμπορική έκδοση και το OPEN UI5 που είναι η δωρεάν έκδοση ανοικτού κώδικα. Το SAP UI5 αποτελεί την βάση και για το νέο γραφικό περιβάλλον των εφαρμογών του SAP Business Suite το οποίο ονομάζεται SAP Fiori.



Εικόνα 38. Παράδειγμα εφαρμογών SAPUI5/SAP Fiori [35]

Κεφάλαιο 4 - SAP HANA Cloud Platform

4.1 Περιγραφή

Το SAP HANA Cloud Platform αποτελεί την προσέγγιση της SAP για την προσφορά υπηρεσιών In-Memory Platform-as-a-Service (PaaS). Αποτελεί μια προσέγγιση η οποία διευκολύνει τη γρήγορη υλοποίηση της λύσης του SAP HANA προσφέροντας την απαραίτητη υποδομή και υποστήριξη για αυτό, υποστηρίζοντας παράλληλα εύκολη επεκτασιμότητα.

Το SAP HANA Cloud Platform υποστηρίζει το πλήρες σετ λειτουργιών και γλωσσών ανάπτυξης εφαρμογών, καθώς και επιπλέον υπηρεσίες που αφορούν τη διαχείριση και ανάπτυξη των εφαρμογών, της βάσης δεδομένων και της υποδομής του SAP HANA. Το SAP HANA Cloud Platform μπορεί να συνδεθεί με την εσωτερική μηχανογραφική υποδομή της εταιρείας, είτε μέσω των Connectivity Services, είτε μέσω του SAP HANA Cloud Connector, για τα συστήματα SAP.

Το SAP HANA Cloud Platform σήμερα (2015), προσφέρεται ως πλήρες πακέτο μέσω της υπηρεσίας SAP HANA Enterprise Cloud. Εναλλακτικά το SAP HANA προσφέρεται σε λύση Cloud και μέσω των υποδομών Cloud Amazon AWS και Microsoft Azure.

4.2 Σενάρια Ανάπτυξης Εφαρμογών

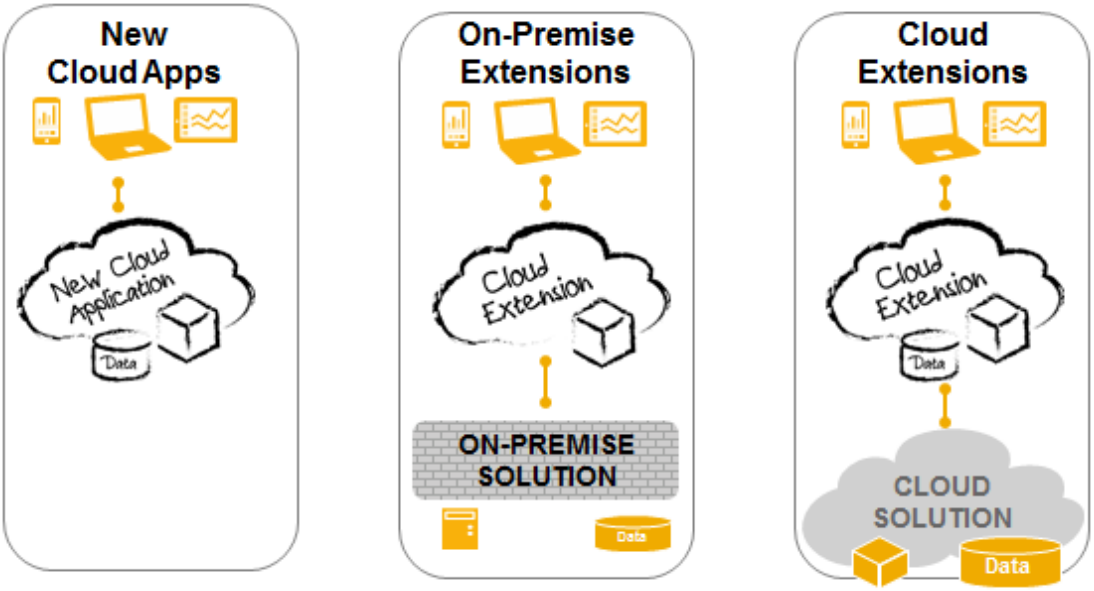
Τα σενάρια για την ανάπτυξη εφαρμογών στο SAP HANA Cloud Platform είναι τα εξής[35]:

- *Ανάπτυξη νέων εφαρμογών στο Cloud*
  Αυτό το σενάριο αφορά την ανάπτυξη εφαρμογών στο cloud, οι οποίες δεν χρησιμοποιούν κάποια από τις υπάρχουσες δομές του οργανισμού, και έχουν ως στόχο την άμεση πρόσβαση σε αυτές των τελικών χρηστών.

- *Ανάπτυξη επεκτάσεων στο Cloud για την υπάρχουσα υποδομή*
  Αυτό το σενάριο αφορά εταιρείες και οργανισμούς οι οποίοι διαθέτουν μεγάλη υποδομή IT, SAP και non SAP, στο χώρο τους. Αυτοί μπορούν να επεκτείνουν την υποδομή τους με

εφαρμογές στο cloud οι οποίες θα ενσωματωθούν στην υπάρχουσα υποδομή επικοινωνώντας μαζί της με το SAP HANA Cloud Connector και τα Connectivity Services.

- *Ανάπτυξη επεκτάσεων στο Cloud για την υπάρχουσα υποδομή Cloud*
  Αφορά την ανάπτυξη επεκτάσεων για υπάρχουσα υποδομή σε άλλα Συστήματα Cloud οι οποίες θα επικοινωνούν με το SAP HANA Cloud Platform.



Εικόνα 39. Σενάρια Ανάπτυξης εφαρμογών στο SAP HANA Cloud Platform [35]

4.3 Μοντέλα Ανάπτυξης Εφαρμογών

Τα παρακάτω μοντέλα ανάπτυξης υποστηρίζονται εγγενώς από το SAP HANA Cloud Platform.

- *SAP HANA*
  Υποστηρίζονται όλα τα προγραμματιστικά εργαλεία για την ανάπτυξη στο SAP HANA, όσον αφορά τη Βάση Δεδομένων, τα OData Services και τα Αναλυτικά Μοντέλα. Διαθέτει και περιβάλλον ανάπτυξης στο Web για αυτά, το SAP HANA Web IDE.

- *Java*

  Το SAP HANA Cloud Platform είναι ένας πιστοποιημένος Application Server για την εκτέλεση εφαρμογών Java EE 6 Web Profile, επομένως μπορεί να χρησιμοποιηθεί για την ανάπτυξη Java EE εφαρμογών όπως κάθε άλλος Java Server. Επίσης μπορούν να μεταφερθούν εκεί ήδη υπάρχουσες εφαρμογές Java EE, ώστε να εκμεταλλευτούν τις δυνατότητες της In-Memory Database.

- *HTML5*

  Με το SAP HANA Web IDE, μπορούν να αναπτυχθούν και να φιλοξενηθούν εφαρμογές HTML5.

- *SAPUI5*

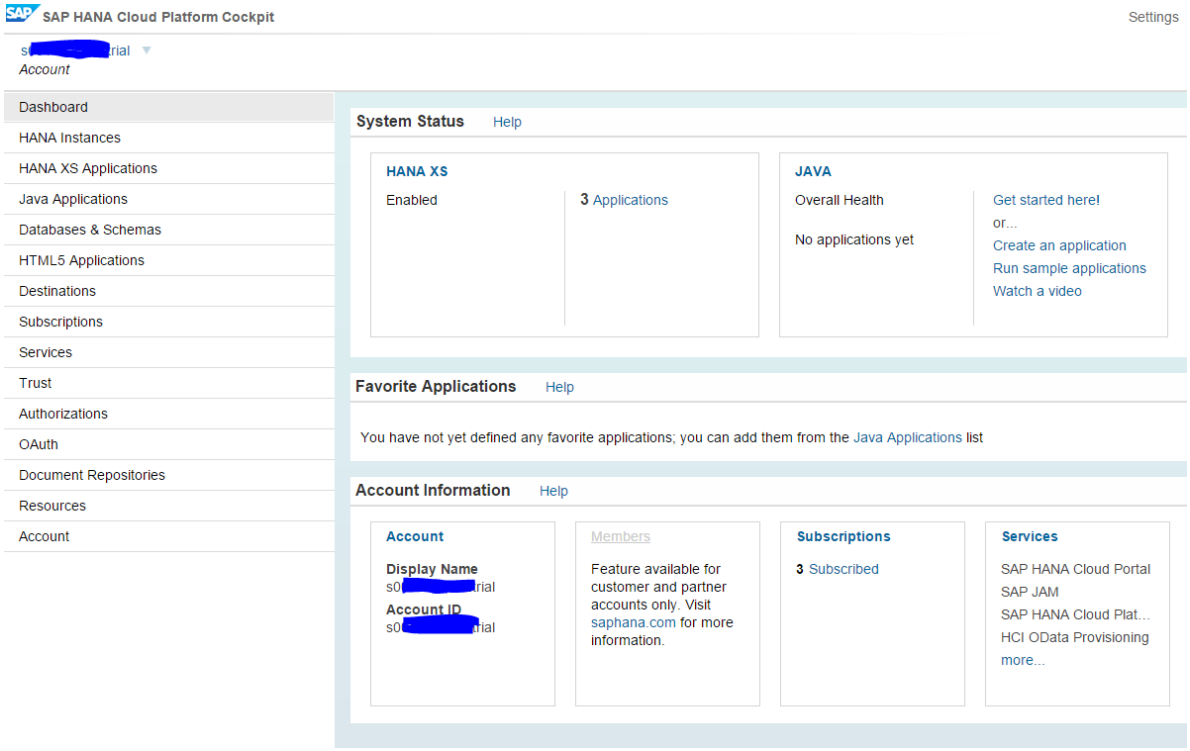
  Με το SAP HANA Web IDE, μπορούν να αναπτυχθούν και να φιλοξενηθούν εφαρμογές SAPUI5.

4.4 Εργαλεία του SAP HANA Cloud Platform

SAP HANA Cloud Cockpit

Είναι το κεντρικό σημείο, όπου διαθέτει web-based interface για τη διαχείριση όλων των δραστηριοτήτων που σχετίζονται με το SAP HANA Cloud Instance και τις εφαρμογές του.

Αποτελεί τον πίνακα ελέγχου μέσω του οποίου γίνεται η διαχείριση και ο έλεγχος των εφαρμογών HANA, Java, SAPUI5/HTML5, των χρηστών, των ομάδων και των δικαιωμάτων τους, καθορίζεται η επικοινωνία με τα on-premise συστήματα καθώς και ο τρόπος ταυτοποίησης των χρηστών.



Εικόνα 40. SAP HANA Cloud Cockpit

SAP HANA Cloud Connector

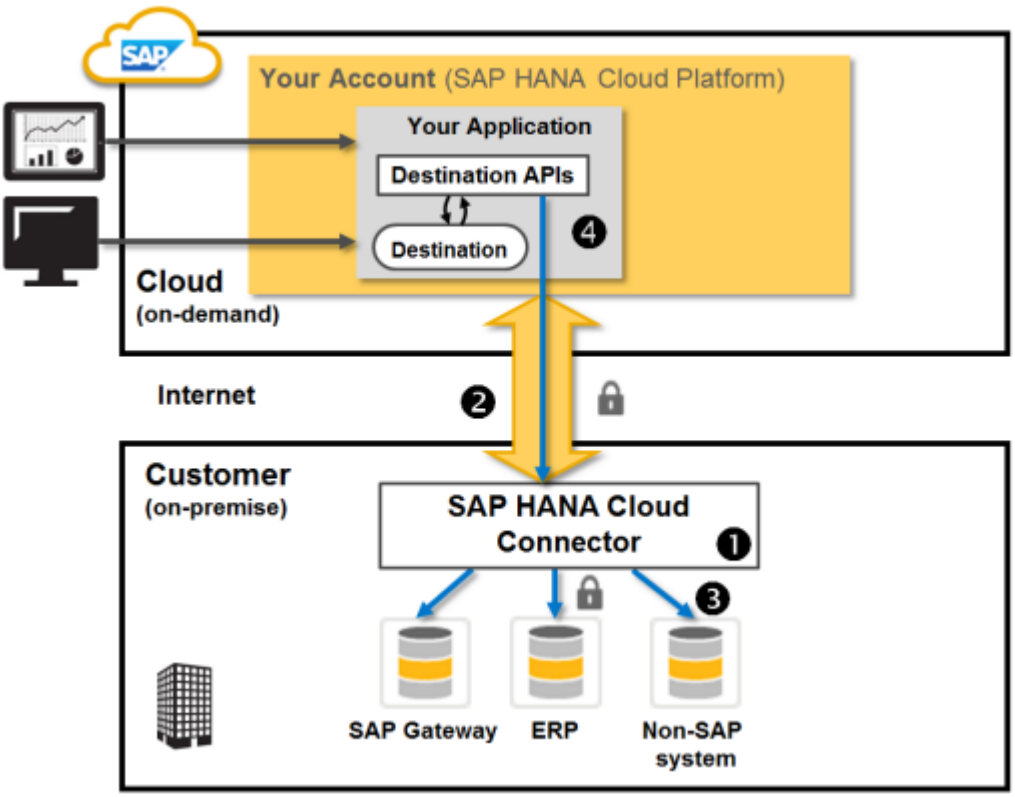
Ο SAP HANA Cloud Connector, εξυπηρετεί ως ο ενδιάμεσος σύνδεσμος ανάμεσα στο SAP HANA Cloud Platform και τα on-premise SAP και non-SAP συστήματα. Είναι ένα πρόγραμμα – agent το οποίο εκτελείται μέσα στο ασφαλές ενδοεταιρικό δίκτυο, και λειτουργεί ως reverse invoke proxy, προσφέροντας αμφίδρομη επικοινωνία από και προς το SAP HANA Cloud Platform.

Ο SAP HANA Cloud Connector διαθέτει αυτόματη επανασύνδεση, παρέχει καταγραφή της κίνησης και των αλλαγών στις ρυθμίσεις του για λόγους ελέγχου (audit) και υποστηρίζει εκτέλεση με διαμόρφωση High Availability.

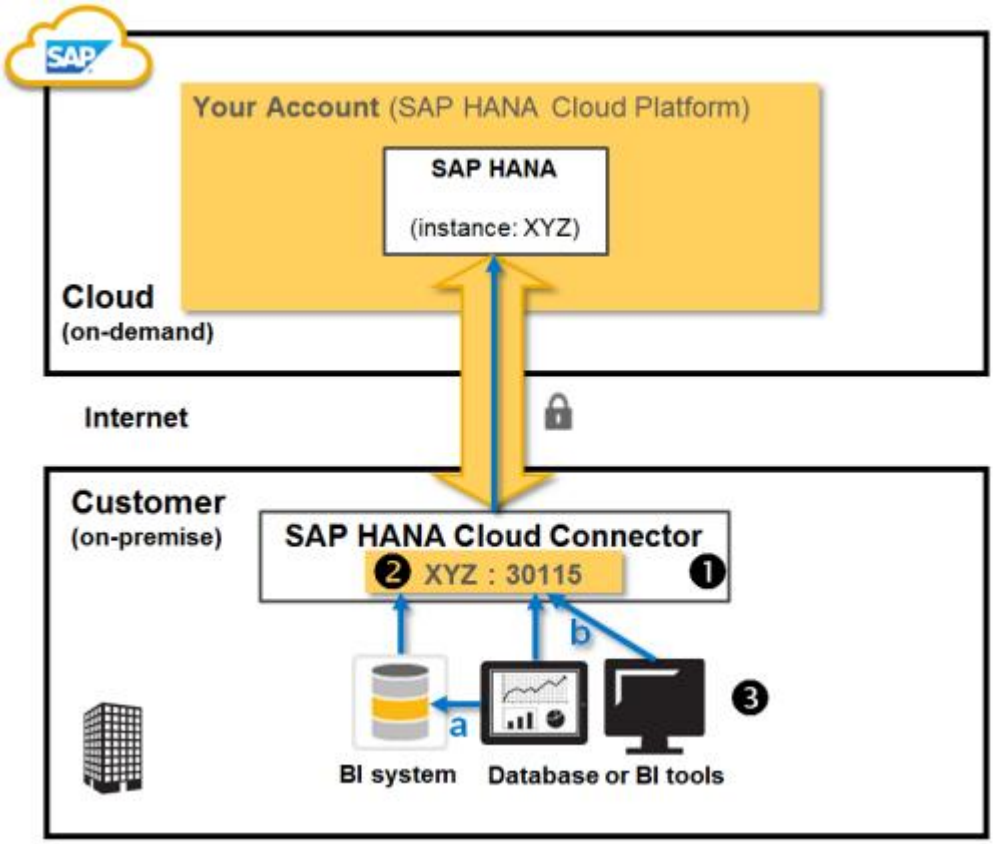
Τα σενάρια που μπορεί να χρησιμοποιηθεί είναι δύο.

- Πρόσβαση στα εσωτερικά εταιρικά συστήματα των εφαρμογών του HANA Cloud.

- Πρόσβαση των εσωτερικών εφαρμογών στα δεδομένα του SAP HANA Cloud.



Εικόνα 41. Σενάριο 1 - Πρόσβαση εφαρμογών του SAP HANA Cloud στο εταιρικό δίκτυο [35]



Εικόνα 42. Σενάριο 2 - Πρόσβαση εφαρμογών από το εταιρικό δίκτυο στα δεδομένα του SAP HANA Cloud [35]

Ο SAP HANA Cloud Connector έχει πλεονεκτήματα σε σχέση με τη λύση της σύνδεσης των συστημάτων με άλλες μεθόδους, όπως το άνοιγμα θυρών στο firewall και η χρήση reverse proxies. Συγκεκριμένα[35]:

- Δε χρειάζεται να ανοίξουν θύρες στο firewall. Δεν απαιτείται κάποια τροποποίηση για την επικοινωνία με το SAP HANA Cloud.
- Υποστηρίζονται επιπλέον πρωτόκολλα, πέρα από το HTTP. Για παράδειγμα το RFC πρωτόκολλο, επιτρέπει την εγγενή σύνδεση σε συστήματα SAP μέσω των κλήσεων Συναρτήσεων ABAP.
- Υποστηρίζει αμφίδρομη επικοινωνία από και προς το SAP HANA Cloud.

- Επιτρέπει την αποστολή της ταυτότητας των χρηστών με ασφαλή τρόπο.
- Ο SAP HANA Cloud Connector είναι εύκολος και γρήγορος στην εγκατάστασή του, και η SAP παρέχει υποστήριξη για αυτόν.

SAP Web IDE

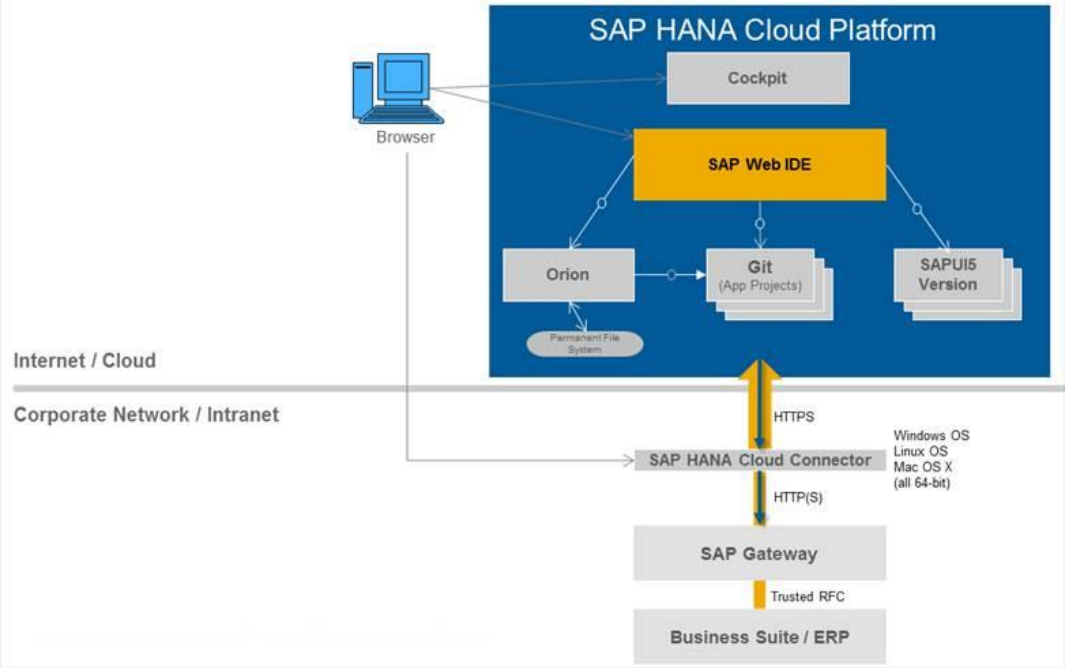
Το SAP HANA Cloud Platform διαθέτει δικό του Web based περιβάλλον ανάπτυξης εφαρμογών που ονομάζεται SAP WEB IDE. Μέσω του περιβάλλοντος αυτού είναι δυνατή η ανάπτυξη end-to-end λύσεων για το SAP HANA.

Συγκεκριμένα το SAP Web IDE υποστηρίζει:

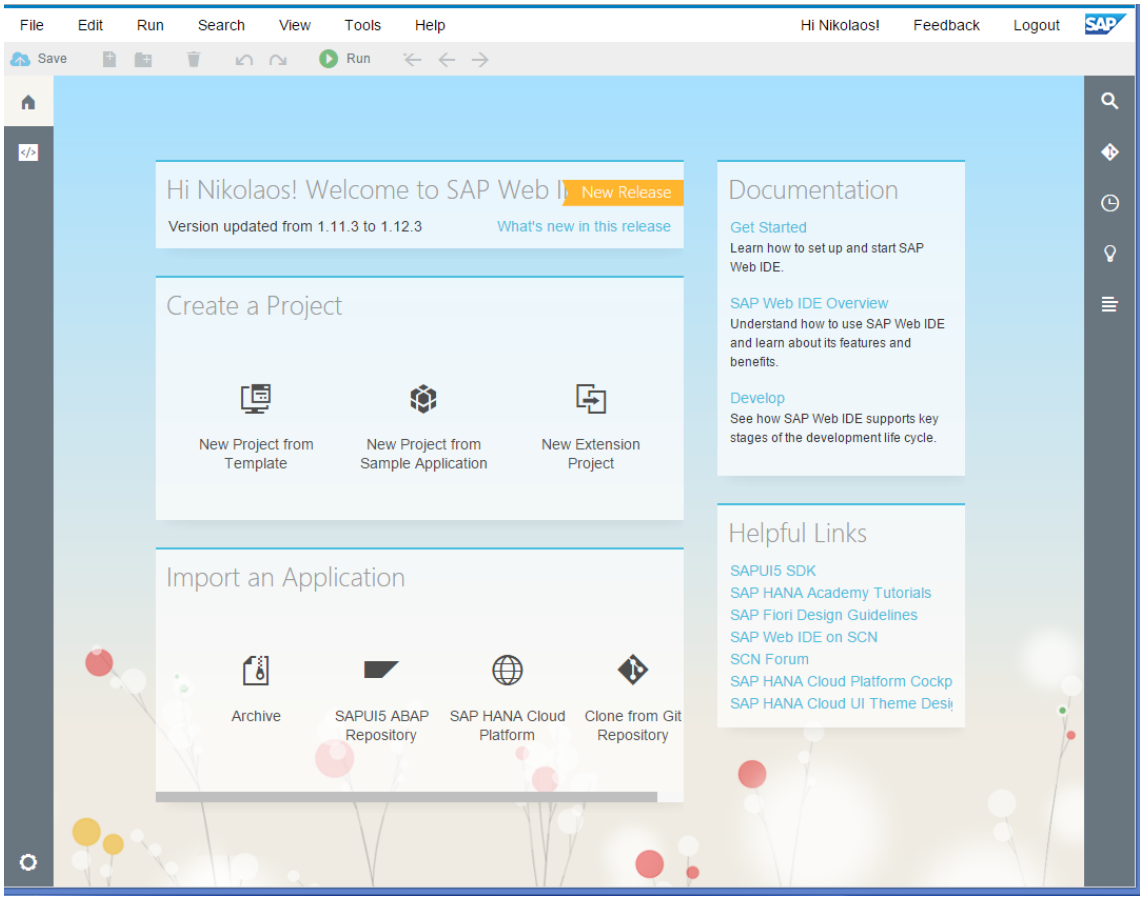
- *Ανάπτυξη SAP HANA*
  Δημιουργία και συντήρηση του μοντέλου της Βάσης Δεδομένων, τόσο κατά το χρόνο σχεδίασης με χρήση CDS και HDBTable .
  Ανάπτυξη των ODATA Services καθώς και υπηρεσιών Server Side JavaScript (xsjs).

- *Ανάπτυξη SAPUI5 / Fiori εφαρμογών*
  Διαθέτει WYSIWYG περιβάλλον για την ανάπτυξη SAPUI5 εφαρμογών. Διαθέτει wizards, templates και περιβάλλον για δημιουργία πρωτοτύπων με δοκιμαστικά δεδομένα.
  Υποστηρίζει την διάθεση των εφαρμογών αυτών στο SAP HANA Cloud και στο Backend SAP Σύστημα ως BSP εφαρμογών.

- *Σύνδεση με on premise συστήματα*
  Υποστηρίζει τη σύνδεση με on premise συστήματα, τόσο για τη λήψη δεδομένων όσο και για την επέκταση εφαρμογών και επαναδιάθεση τους σε αυτά.



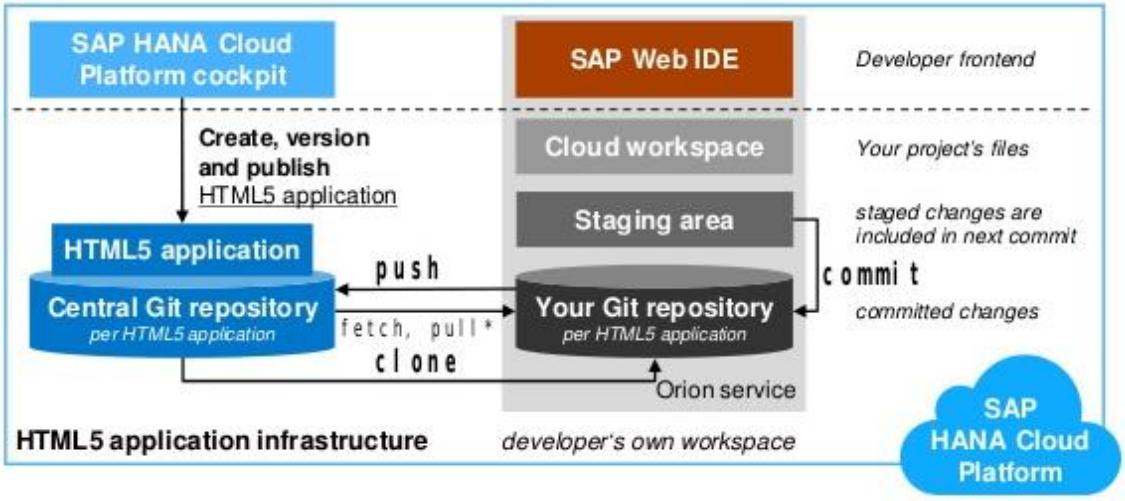
Εικόνα 43. Αρχιτεκτονική SAP Web IDE [35]



Εικόνα 44. Η αρχική οθόνη του SAP Web IDE

Git Repository

Για τον έλεγχο των εκδόσεων και τη συνεργασία μεταξύ των προγραμματιστών, το SAP HANA Cloud προσφέρει το Git Repository. Το SAP Web IDE διαθέτει ενσωματωμένο Git Client που ρυθμίζεται ώστε να το χρησιμοποιεί για τις εφαρμογές SAPUI5 / HTML5.

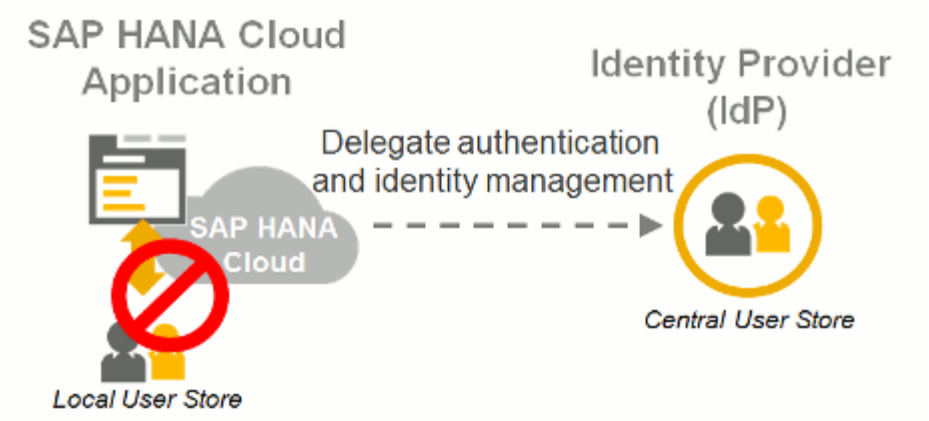


Εικόνα 45. Χρήση του Git για την ανάπτυξη HTML5 εφαρμογής στο SAP HANA Cloud [36]

4.5 Ασφάλεια και Ταυτοποίηση Χρηστών

Για τη διαχείριση των χρηστών στο SAP HANA Cloud Platform, υποστηρίζεται τόσο η διαχείρισή τους από το SAP HANA Cloud Platform μέσω του το SAP ID Service , όσο και η ενσωμάτωση των υπαρχόντων συστημάτων διαχείρισης χρηστών του οργανισμού.

Για την ενσωμάτωση των υπαρχόντων συστημάτων διαχείρισης χρηστών, το SAP HANA Cloud Platform υποστηρίζει single-sign-on (SSO) και δυνατότητες Identity Federation. Σε αυτή την περίπτωση οι πληροφορίες για την ταυτότητα του χρήστη δίνονται από τους Identity Providers (IdP) και δεν αποθηκεύονται στο ίδιο το SAP HANA Cloud Platform.

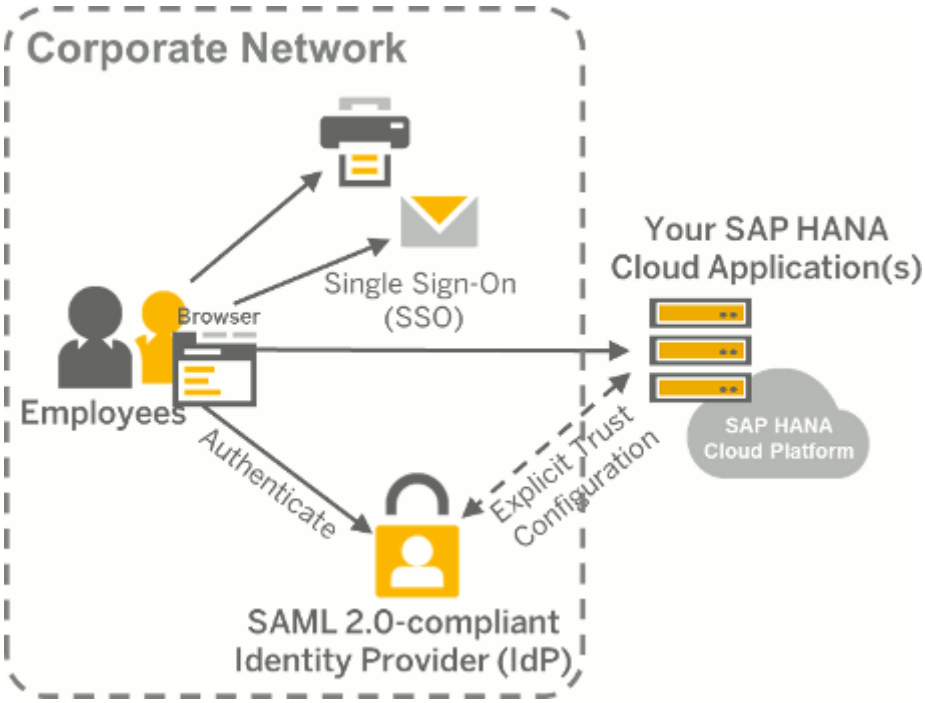


Εικόνα 46. Διαχείριση χρηστών με εξωτερικό IdP στο SAP HANA Cloud [35]

Identity Federation με χρήση Εταιρικού Identity Provider

Το SAP HANA Cloud Platform μπορεί να συνεργαστεί με τον υπάρχοντα εταιρικό Identity Provider για την ταυτοποίηση των χρηστών. Σκοπός είναι να υπάρχει μόνο ένα σημείο στην εταιρεία το οποίο θα διαχειρίζεται τους χρήστες, οι οποίοι σε μια εταιρεία μπορεί να είναι οι εργαζόμενοι, οι πελάτες, οι συνεργάτες κλπ, χωρίς την ανάγκη ξεχωριστής διαχείρισης στο SAP HANA Cloud Platform.

Κατά την διαδικασία ταυτοποίησης χρήστη (Logon), όλες οι σχετικές πληροφορίες μπορούν να μεταβιβαστούν από τον IdP στο SAP HANA Cloud Platform, χρησιμοποιώντας standard πρωτόκολλα ασφαλείας. Δε χρειάζεται συγχρονισμός των στοιχείων ή κάποια άλλη σύνδεση μεταξύ του εταιρικού IdP και του SAP HANA Cloud Platform.



Εικόνα 47.Χρήση εταιρικού IdP στο SAP HANA Cloud Platform [35]

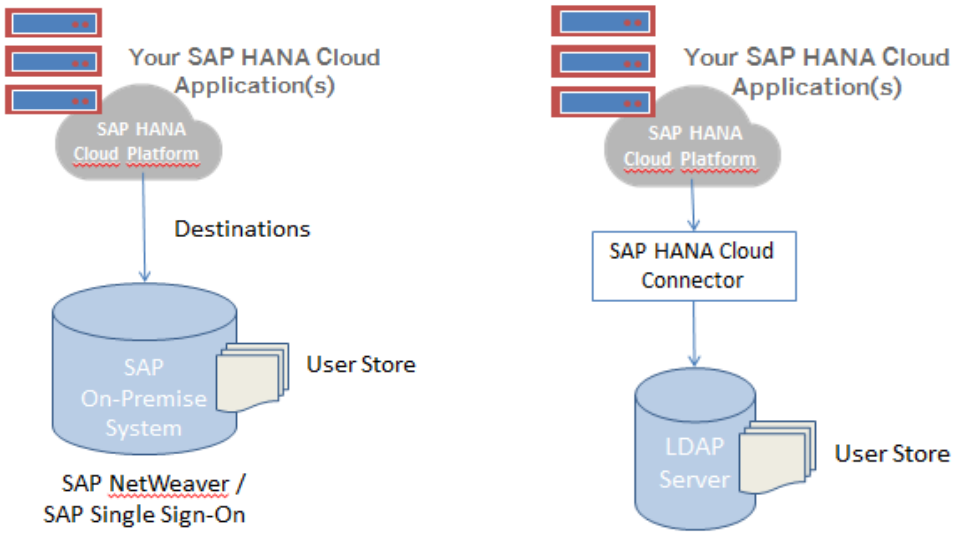
Ταυτοποίηση χρηστών εφαρμογών SAP HANA Cloud με χρήση SAML 2.0

Το SAML 2.0 ή Security Assertion Markup Language (SAML) 2.0 είναι ένα πρωτόκολλο για την ταυτοποίηση χρηστών και για την υλοποίηση single-sign-on (SSO). Το SAP HANA Cloud λειτουργεί ως Service Provider (SP), όπως ορίζεται από το πρωτόκολλο.

Αυτό σημαίνει ότι πρόσβαση στις εφαρμογές του SAP HANA Cloud έχουν μόνο οι χρήστες που έχουν δικαιώματα σε αυτές και κανείς άλλος. Για την ταυτοποίηση των χρηστών χρειάζεται ένας Identity Provider (IdP), όπως αυτός ορίζεται από το SAML 2.0. Ο IdP αποθηκεύει στη βάση δεδομένων του, τα στοιχεία όλων των χρηστών που έχουν πρόσβαση στον SP, μαζί με τα στοιχεία ταυτοποίησης. Τα στοιχεία ταυτοποίησης μπορεί να είναι απλά όνομα και κωδικός ασφαλείας, αλλά να συνδυάζονται με επιπλέον στοιχεία για μεγαλύτερη ασφάλεια όπως Ψηφιακό Πιστοποιητικό, κωδικός PIN, βιομετρικά στοιχεία κλπ.

Το SAP HANA Cloud δεν αποθηκεύει αυτά τα στοιχεία, αλλά εμπιστεύεται τον IdP για αυτά. Κατά την πρώτη είσοδο ενός χρήστη σε μια εφαρμογή του SAP HANA Cloud, ο χρήστης μεταφέρεται σε μια σελίδα web του IdP για ταυτοποίηση. Μετά από επιτυχή ταυτοποίηση στον IdP, δημιουργείται σύνοδος και ο χρήστης δε χρειάζεται να εισάγει ξανά τα στοιχεία του όσο η σύνοδος είναι ενεργή, έχοντας πρόσβαση σε όλες τις εφαρμογές του SAP HANA Cloud Platform. Αυτή η δυνατότητα ονομάζεται single-sign-on (SSO) [35].

Για την ταυτοποίηση χρηστών του SAP HANA Cloud Platform και εφαρμογών Web που φιλοξενούνται σε αυτό (Java, HTML5) προτείνεται η χρήση ταυτοποίησης μέσω SAML 2.0, που είναι και η προεπιλεγμένη για αυτές. Το SAP HANA Cloud Platform ως προεπιλεγμένο IdP, χρησιμοποιεί το SAP ID Service που αναγνωρίζει τους χρήστες του SAP Community Network (SCN). Με την κατάλληλη παραμετροποίησης μπορεί να χρησιμοποιηθεί το εταιρικό IdP, το οποίο μπορεί να είναι το SAP Single-Sign-On και το Microsoft Active Directory.

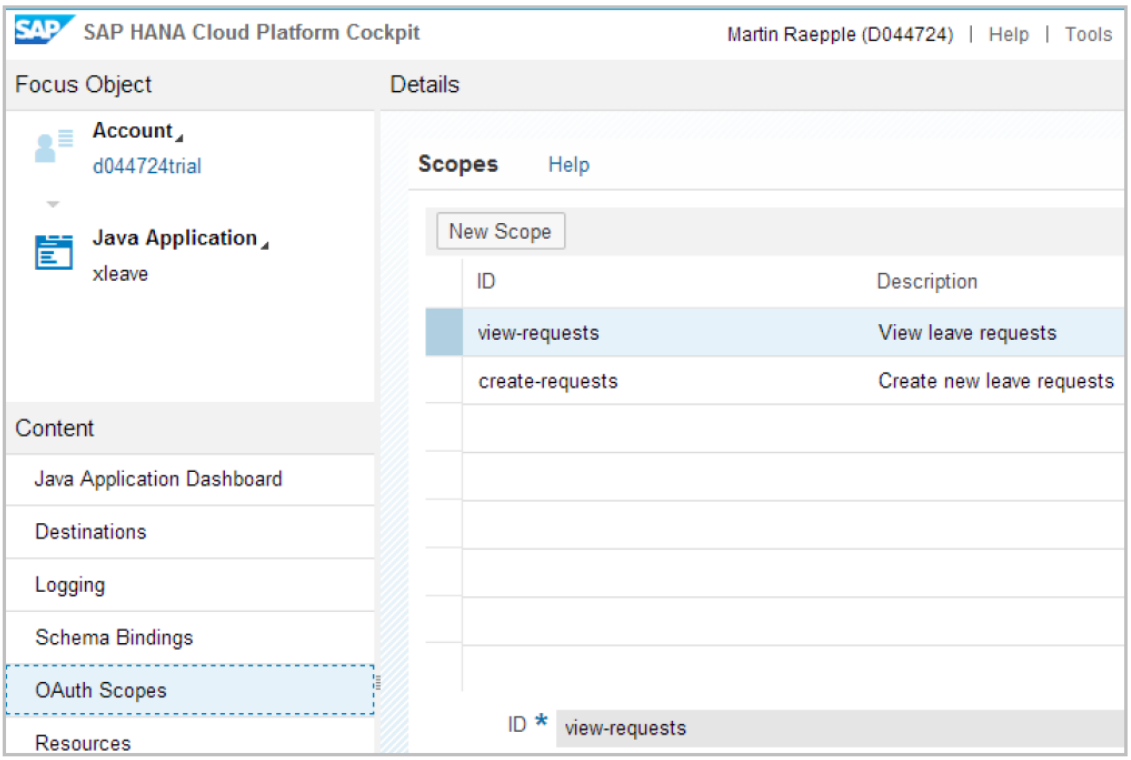


Εικόνα 48. Χρήση ως IdP των SAP SSO και Microsoft AD[35]

Ταυτοποίηση χρηστών τρίτων εφαρμογών με χρήση OAuth 2.0

Το OAuth 2.0 είναι ένα ανοικτό πρότυπο το οποίο επιτρέπει την εξουσιοδότηση πρόσβασης εξωτερικών εφαρμογών (Android, iOS, Windows κλπ) στον server, χωρίς το διαμοιρασμό των στοιχείων ταυτοποίησης του χρήστη. Η προδιαγραφή του OAuth 2.0 βρίσκεται στο IETF RFC 6749.

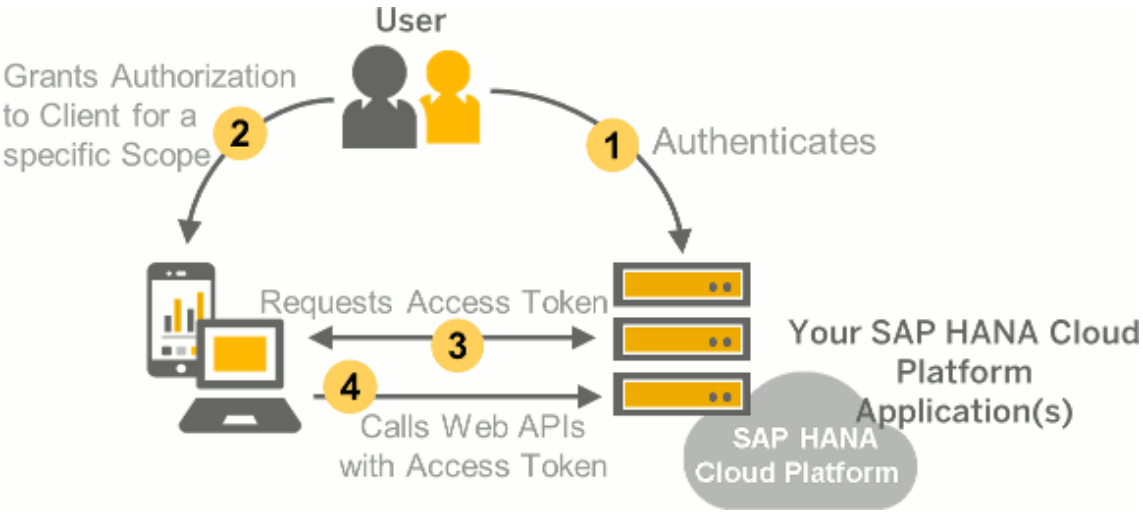
Βασική λειτουργία του OAuth είναι η αντικατάσταση των στοιχείων ταυτοποίησης του χρήστη με ένα token, το οποίο έχει μικρή διάρκεια ζωής και περιορισμένο σκοπό και πρόσβαση σε πηγές σε σχέση με τα στοιχεία εισόδου του χρήστη. Για αυτό ακόμα και εάν κλαπεί το token, η ζημιά θα είναι πολύ μικρότερη σε σχέση με την υποκλοπή του κωδικού του χρήστη.



Εικόνα 49. Διαχείριση των σκοπών μέσα από το SAP HANA Cloud Cockpit [36]

Για τη σύνδεση και ταυτοποίηση χρηστών μέσω εφαρμογών πελατών στο SAP HANA Cloud Platform, προτείνεται η χρήση του OAuth 2.0. Το SAP HANA Cloud Platform διαθέτει API για την

ανάπτυξη εφαρμογών που προστατεύονται με το OAuth 2.0. Η διαχείριση γίνεται μέσα από το SAP HANA Cloud Cockpit.



Εικόνα 50. Εφαρμογές με χρήση του OAuth στο SAP HANA Cloud [35]

Κεφάλαιο 5 - Υλοποίηση Εφαρμογής Android

5.1 Περιγραφή Εφαρμογής

Στο πλαίσιο της παρούσας διατριβής, αναπτύχθηκε native εφαρμογή για συστήματα android, η οποία χρησιμοποιεί ως Backend Σύστημα το SAP HANA Cloud. Η εφαρμογή αφορά ένα σύστημα παραγγελιοληψίας (Sales Force Automation), το οποίο θα αποθηκεύει το σύνολο των δεδομένων του στο SAP HANA Cloud, με σκοπό τη σύνδεση αυτού με το on-premise backend σύστημα της εταιρείας.

Χρήστες της εφαρμογής είναι οι πωλητές της εταιρείας, οι οποίοι επισκέπτονται τα υποκαταστήματα των πελατών, για να λάβουν παραγγελίες και να προωθήσουν τα προϊόντα. Οι δυνατότητες που έχει η εφαρμογή είναι:

- *Δημιουργία Παραγγελίας*

  Ο πωλητής έχει τη δυνατότητα να καταχωρήσει νέα παραγγελία στο σύστημα. Αφού διαλέξει εταιρεία και υποκατάστημα, επιλέγει τα προϊόντα από τον κατάλογο προϊόντων και τα προσθέτει στο καλάθι αγορών. Για την ολοκλήρωση της παραγγελίας, βλέπει συγκεντρωτικά το καλάθι αγορών, προσθέτει σχόλια και αποστέλλει την παραγγελία.

- *Αναζήτηση Πελατών – Υποκαταστημάτων*

  Ο πωλητής μπορεί να χρησιμοποιήσει την εφαρμογή, για να δει τα στοιχεία των πελατών και των υποκαταστημάτων τους και να πλοηγηθεί μέσω GPS σε αυτά.

- *Προβολή Ιστορικού Παραγγελιών*

  Ο πωλητής μπορεί να δει το ιστορικό των παραγγελιών που έχει καταχωρήσει, καθώς και την κατάσταση τους.

- *Προβολή Στατιστικών Στοιχείων*

Εμφανίζονται διάφορα στατιστικά στοιχεία, σχετικά με την πορεία πωλήσεων των προϊόντων και τους τζίρους των πελατών.

Στην τρέχουσα υλοποίηση της εφαρμογής υποστηρίζεται μόνο η on-line λειτουργία, δηλαδή απαιτείται συνεχής σύνδεση με το SAP HANA Cloud μέσω του Internet. Στην εφαρμογή android υπάρχει πρόβλεψη για τη λειτουργία και σε κατάσταση επίδειξης με δοκιμαστικά δεδομένα, χωρίς τη σύνδεση με το SAP HANA Cloud.

5.2 Τεχνολογίες

SAP HANA Cloud

Για την εφαρμογή χρησιμοποιήθηκε η έκδοση του SAP HANA, Developer Edition SPS 09, στο Amazon AWS Cloud. Σε αυτό δημιουργήθηκε η βάση δεδομένων της εφαρμογής, τα OData Services καθώς και τα HANA Server Side Javasript Services (xsjs). Επίσης οι χρήστες της εφαρμογής και τα δικαιώματά τους συντηρούνται αποκλειστικά μέσα στο SAP HANA. Για τη ανάπτυξη στο SAP HANA Cloud χρησιμοποιήθηκε αποκλειστικά το SAP Web IDE.

Android L

Η εφαρμογή – πελάτης δημιουργήθηκε για Android L, με προτεινόμενη συσκευή Android Tablet. Πέρα από τα στοιχεία παραμετροποίησης, η εφαρμογή δεν αποθηκεύει άλλα δεδομένα καθώς λειτουργεί σε σύνδεση με το SAP HANA Cloud. Η επικοινωνία με το SAP HANA Cloud γίνεται μέσω κλήσεων σε OData Services. Οι ταυτοποίηση των χρηστών στο SAP HANA Cloud γίνεται με χρήση Basic HTTP Authentication. Για τη δημιουργία της εφαρμογής Android χρησιμοποιήθηκε το Android Studio IDE στην έκδοση 1.2.2.

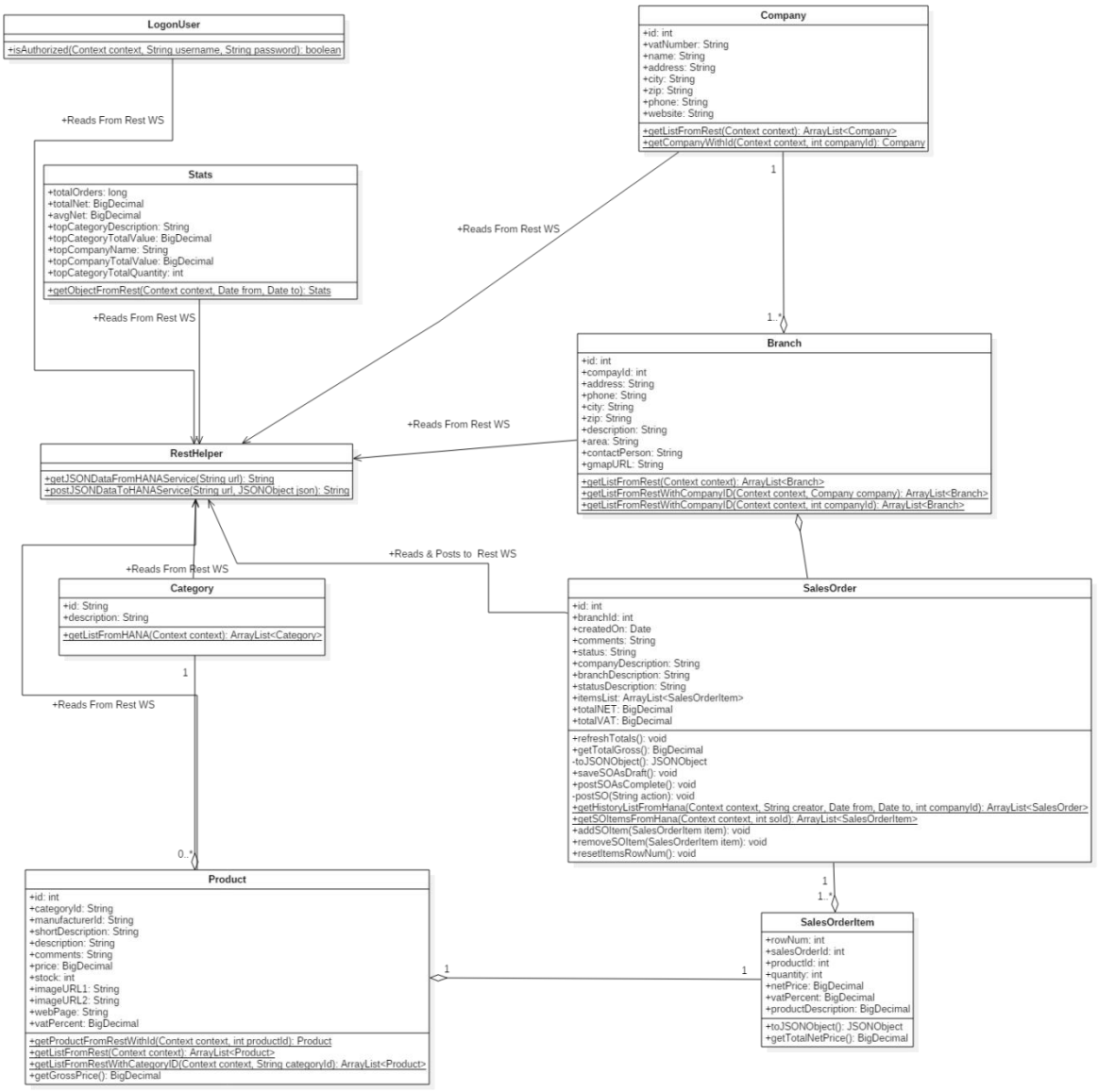
Αρχιτεκτονική

Στην παρακάτω εικόνα φαίνεται η αρχιτεκτονική της εφαρμογής.

Εικόνα 51. Αρχιτεκτονική της εφαρμογής

Διάγραμμα Κλάσεων

Στην παρακάτω εικόνα παρουσιάζεται το διάγραμμα κλάσεων του μοντέλου της Android Εφαρμογής.



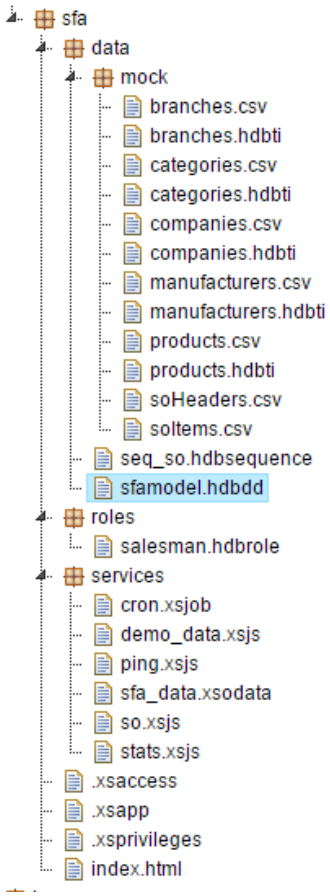
Εικόνα 52. Το διάγραμμα κλάσεων της εφαρμογής Android

5.3 Υλοποίηση

Η εφαρμογή υλοποιήθηκε σε δύο φάσεις. Στην πρώτη φάση έγινε η ανάπτυξη στο SAP HANA Cloud και δημιουργήθηκαν οι απαραίτητοι πίνακες στη βάση δεδομένων και δημιουργήθηκαν τα OData Services. Στη δεύτερη φάση δημιουργήθηκε η εφαρμογή πελάτης στο Android.

Υλοποίηση στο SAP HANA Cloud

Χρησιμοποιώντας το SAP Web IDE δημιουργήθηκε ένα νέο πακέτο που αφορά την εφαρμογή, το οποίο ονομάστηκε SFA. Μέσα σε αυτό δημιουργήθηκαν τα πακέτα data, roles και services. Στο πακέτο data υπάρχουν όλα τα απαραίτητα στοιχεία για τη δημιουργία του μοντέλου της βάσης δεδομένων, στο πακέτο roles υπάρχουν οι εντολές για τη δημιουργία των ρόλων της εφαρμογής και στο πακέτο services, υπάρχουν τα αρχεία για την υλοποίηση των απαραίτητων OData και XSJS Services.



Εικόνα 53. Τα αρχεία της εφαρμογής SFA στο SAP Web IDE

Στο αρχείο sfamodel.hdbdd υπάρχει ο κώδικας για τη δημιουργία των πινάκων στο HANA DB μέσω εντολών CDS. Οι πίνακες της εφαρμογής είναι οι:

- Category: Περιέχει τις κατηγορίες των προϊόντων
- Product: Περιέχει τα προϊόντα και τις τιμές τους
- Manufacturer: Περιέχει τους κατασκευαστές των προϊόντων
- Company: Περιέχει τους πελάτες της εταιρείας
- Branch: Περιέχει τα υποκαταστήματα των πελατών

- SalesOrderHeader: Περιέχει τα δεδομένα κεφαλίδας των παραγγελιών
- SalesOrderItem: Περιέχει τις αναλυτικές γραμμές των παραγγελιών

Ο κώδικας του αρχείου sfamodel.hdbdd για την δημιουργία των παραπάνω πινάκων είναι ο εξής:

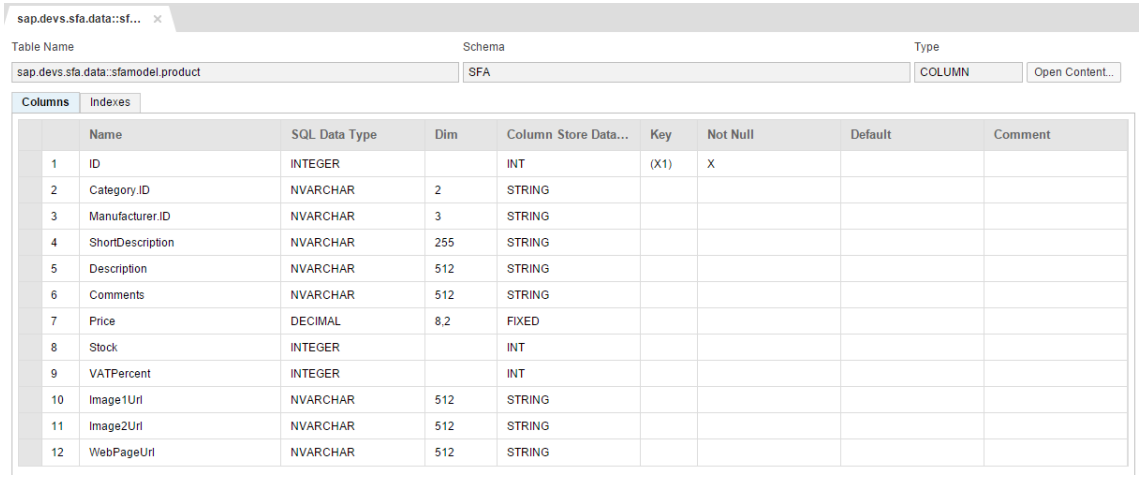
```
namespace sap.devs.sfa.data;
@Schema: 'SFA'
context sfamodel {
    type XLString: String(512);
    type LString: String(255);
    type SString: String(50);

    @Catalog.tableType: #COLUMN
    Entity manufacturer {
        key ID: String(3);
        Description: SString;
    };

    @Catalog.tableType: #COLUMN
    Entity category {
        key ID: String(2);
        Description: SString;
    };

    @Catalog.tableType: #COLUMN
    Entity product {
        key ID: Integer;
```

```
        Category : Association  to category;
        Manufacturer : Association to manufacturer;
        ShortDescription: LString;
        Description: XLString;
        Comments: XLString;
        Price: Decimal(8,2);
        Stock: Integer;
        VATPercent: Integer;
        Image1Url: XLString;
        Image2Url: XLString;
        WebPageUrl: XLString;
    };

    @Catalog.tableType: #COLUMN
    Entity company {
        key ID: Integer;
        VATNumber: SString;
        Name:SString;
        Address:LString;
        City:SString;
        ZIP:String(10);
        Phone:String(15);
        Website:LString;
    };

    @Catalog.tableType: #COLUMN
```

```
Entity branch {
    key ID: Integer;
    Company: Association  to company;
    Description: SString;
    Address:LString;
    Area:SString;
    City:SString;
    ZIP:String(10);
    Phone:String(15);
    ContactPerson:SString;
    Gmap:XLString;
};

@Catalog.tableType: #COLUMN
Entity salesOrderHeader {
    key ID: Integer;
    Branch : Association  to branch;
    CreatedOn: LocalDate;
    Creator:SString;
    Comments:LString;
    Status:String(1);
};

@Catalog.tableType: #COLUMN
Entity salesOrderItem {
    key SalesOrderHeader: Association  to salesOrderHeader;
```

```
            key RowNum: Integer;
            Product: Association  to product;
            Quantity: Integer;
            NetPrice: Decimal(8,2);
            VATPercent: Integer;
        };
    };
```

Πριν από κάθε entity υπάρχει η εντολή `@Catalog.tableType: #COLUMN` για να ορίσει ένα ο πίνακας θα είναι αποθηκευμένος κατά γραμμή ή στήλη. Με την αποθήκευση του αρχείου και την ενεργοποίησή του δημιουργούνται οι πίνακες στο HANA DB.



	Name	SQL Data Type	Dim	Column Store Data...	Key	Not Null	Default	Comment
1	ID	INTEGER		INT	(X1)	X		
2	Category.ID	NVARCHAR	2	STRING				
3	Manufacturer.ID	NVARCHAR	3	STRING				
4	ShortDescription	NVARCHAR	255	STRING				
5	Description	NVARCHAR	512	STRING				
6	Comments	NVARCHAR	512	STRING				
7	Price	DECIMAL	8,2	FIXED				
8	Stock	INTEGER		INT				
9	VATPercent	INTEGER		INT				
10	Image1Url	NVARCHAR	512	STRING				
11	Image2Url	NVARCHAR	512	STRING				
12	WebPageUrl	NVARCHAR	512	STRING				

Εικόνα 54. Ο πίνακας product όπως φαίνεται στο HANA DB

Για τον πίνακα SalesOrderHeader χρειαζόμαστε μια γεννήτρια ακολουθίας συνεχόμενων ακεραίων οι οποίοι θα δίνονται μοναδικά σε κάθε γραμμή του πίνακα. Αυτή δημιουργείται από το αρχείο seq_so.hdbsequence . Ο κώδικας του αρχείου είναι ο παρακάτω:

```
schema= "SFA";
start_with= 1;
```

```
  nomaxvalue=true;
  nominvalue=true;
  cycles= false;
  depends_on_table= "sap.devs.sfa.data::sfamodel.salesOrderHeader";
```

Σε παραγωγική λειτουργία το πρόγραμμα θα φορτώνει τα βασικά δεδομένα του από το EPR της εταιρείας. Για τις ανάγκες της διατριβής, φορτώθηκαν mock data από αρχεία csv. Τα αρχεία αυτά βρίσκονται στο πακέτο mock. Για τη φόρτωση κάθε αρχείου csv στον αντίστοιχο πίνακα, υπάρχει το αντίστοιχο αρχείο με κατάληξη hdbti. Για παράδειγμα, για τον πίνακα των κατηγοριών υπάρχει το αρχείο categories.csv με τα παρακάτω περιεχόμενα:

```
  DE,Επιτραπέζιοι Υπολογιστές
  LA,Φορητοί Υπολογιστες
  SE,Διακομιστές
  SM,Έξυπνα Κινητά
  TA,Ταμπλέτες
```

Για να φορτωθεί αυτό στον αντίστοιχο πίνακα χρειάζεται το αρχείο categories.hdbti, ο κώδικας του οποίου είναι ο παρακάτω:

```
  import = [{
    table = "sap.devs.sfa.data::sfamodel.category";
    schema = "SFA";
    file = "sap.devs.sfa.data.mock:categories.csv";
    header = false;
  }];
```

Για τα δικαιώματα των χρηστών, δημιουργήθηκε ο ρόλος salesman, όπως περιγράφεται στο αρχείο salesman.hdbrole. Οι χρήστες της εφαρμογής πρέπει να έχουν αυτό το ρόλο. Ο κώδικας του salesman.hdbrole είναι:

```
role sap.devs.sfa.roles::salesman{
```

```
  sql object sap.devs.sfa.data::sfamodel.category: SELECT;
  sql object sap.devs.sfa.data::sfamodel.company: SELECT;
  sql object sap.devs.sfa.data::sfamodel.branch: SELECT;
  sql object sap.devs.sfa.data::sfamodel.manufacturer: SELECT;
  sql object sap.devs.sfa.data::sfamodel.product: SELECT;
  sql object sap.devs.sfa.data::sfamodel.salesOrderHeader: SELECT,
INSERT, UPDATE, DELETE;
  sql object sap.devs.sfa.data::sfamodel.salesOrderItem: SELECT,
UPDATE, INSERT, DELETE;
  application privilege: "sap.devs.sfa::Execute";
}
```

Για τη δημιουργία των OData Services, δημιουργήθηκε το αρχείο sfa_data.xsodata, στο οποίο δηλώθηκαν όλα τα Entities της εφαρμογής. Ο κώδικας του αρχείου είναι:

```
  service {
    "sap.devs.sfa.data::sfamodel.category" as "Categories";
    "sap.devs.sfa.data::sfamodel.company" as "Companies";
    "sap.devs.sfa.data::sfamodel.branch" as "Branches";
    "sap.devs.sfa.data::sfamodel.manufacturer" as "Manufacturers";
    "sap.devs.sfa.data::sfamodel.product" as "Products";
    "sap.devs.sfa.data::sfamodel.salesOrderHeader" as "SOHeaders";
    "sap.devs.sfa.data::sfamodel.salesOrderItem" as "SOItems";
  }
```

Αυτόματα θα δημιουργηθεί το OData endpoint το οποίο θα περιέχει όλες αυτές τις οντότητες. Εάν καλέσουμε από browser το URL του αρχείου, δηλαδή εάν καλέσουμε το:

http://<HANA_Cloud_IP>/sap/devs/sfa/services/sfa_data.xsodata

Η απάντηση σε μορφή XML που περιγράφει τις διαθέσιμες οντότητες του service είναι:

```
  <service xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://
www.w3.org/2007/app" xmlns="http://www.w3.org/2007/app" xml:base="http
://52.28.109.247/sap/devs/sfa/services/sfa_data.xsodata/">
  <workspace>
    <atom:title>Default</atom:title>
    <collection href="Categories">
         <atom:title>Categories</atom:title>
    </collection>
    <collection href="Companies">
         <atom:title>Companies</atom:title>
    </collection>
    <collection href="Branches">
         <atom:title>Branches</atom:title>
    </collection>
    <collection href="Manufacturers">
         <atom:title>Manufacturers</atom:title>
    </collection>
    <collection href="Products">
         <atom:title>Products</atom:title>
    </collection>
    <collection href="SOHeaders">
         <atom:title>SOHeaders</atom:title>
    </collection>
    <collection href="SOItems">
         <atom:title>SOItems</atom:title>
    </collection>
  </workspace>
```

```
</service>
```

Εάν στη συνέχεια προσθέσουμε κάποια από αυτές στο URL π.χ.

http://<HANA_Cloud_IP>/sap/devs/sfa/services/sfa_data.xsodata/Categories

τότε θα μας επιστραφεί η λίστα με τις κατηγορίες κ.ο.κ

Εκτός από τα OData Services, για την εφαρμογή χρειάστηκε να αναπτυχθούν και XSJS Services, τα αρχεία των οποίων έχουν κατάληξη .xsjs. Για τη δημιουργία της παραγγελίας δημιουργήθηκε το αρχείο so.xsjs το οποίο εκτός των άλλων δημιουργεί την παραγγελία γράφοντας στους πίνακες salesOrderHeader και salesOrderItem και σε περίπτωση προβλήματος κατά την εγγραφή στον δεύτερο πίνακα ακυρώνει και τις εγγραφές και στον salesOrderHeader. Ο κώδικας ο οποίος δημιουργεί την παραγγελία είναι:

```
function createSalesOrder() {
    var conn = $.db.getConnection();
    try {
        //get next id from sequence
        var pstmt = conn.prepareStatement('select
"SFA"."sap.devs.sfa.data::seq_so".NEXTVAL fROM DUMMY ');
        var rs = pstmt.executeQuery();
        rs.next();
        var newSOId = rs.getString(1);
        //create so_header
        pstmt = conn.prepareStatement(' INSERT INTO
"SFA"."sap.devs.sfa.data::sfamodel.salesOrderHeader" '
                                    + '
("ID","Branch.ID","CreatedOn","Creator","Comments","Status") '
                                    + ' VALUES (?,?,?,?,?,?) ');
        var index = 1;
        pstmt.setString(index++, newSOId);
        pstmt.setInt(index++, JSONObj.BranchID);
        pstmt.setDate(index++, new Date());
        pstmt.setString(index++, $.session.getUsername());
```

```
        pstmt.setString(index++, JSONObj.comments);
        //set status
        if (action === ACTION_POST_SO) { //create
            pstmt.setString(index++, STATUS_IN_PROGRESS); //completed
        } else {
            pstmt.setString(index++, STATUS_DRAFT); // save draft
        }
        pstmt.execute();

        //create so items
        //set new sales order id
        var itemsArray = JSONObj.items;
        for (var i = 0; i < itemsArray.length; i++) {
            var item = itemsArray[i];
            item.id = newSOId;
            pstmt = conn
                .prepareStatement(' INSERT INTO
"SFA"."sap.devs.sfa.data::sfamodel.salesOrderItem" '
                                + ' (
"SalesOrderHeader.ID","RowNum","Product.ID","Quantity","NetPrice","VAT
Percent") '
                                  + ' VALUES (?,?,?,?,?,?) ');
            index = 1;
            pstmt.setString(index++, newSOId);
            pstmt.setInt(index++, (i + 1));
            pstmt.setInt(index++, item.ProductID);
            pstmt.setInt(index++, item.Quantity);
            pstmt.setDouble(index++, item.NetPrice);
            pstmt.setInt(index++, item.VATPercent);
            pstmt.execute();
        }
        conn.commit();
        // cleanup
```

```
        rs.close();
        pstmt.close();
        conn.close();
        $.response.setBody("{\"SalesOrder.ID\":" + newSOId + "}");
        $.response.status = $.net.http.OK;
    } catch (e) {
        conn.rollback();
        $.response.status = $.net.http.INTERNAL_SERVER_ERROR;
    }
}
```

Για την δοκιμή της απόδοσης του SAP HANA Cloud δημιουργήθηκαν 1 εκατομμύριο παραγγελίες για το έτος 2014, που αντιστοιχούν σε περίπου 3 εκ. γραμμές στους πίνακες SalesOrderHeader και SalesOrderItem. Σκοπός αυτών είναι να διαβαστούν από το service stats.xsjs για να δοθούν στην εφαρμογή android συγκεντρωτικά στοιχεία για αυτές τις εγγραφές σε πραγματικό χρόνο.

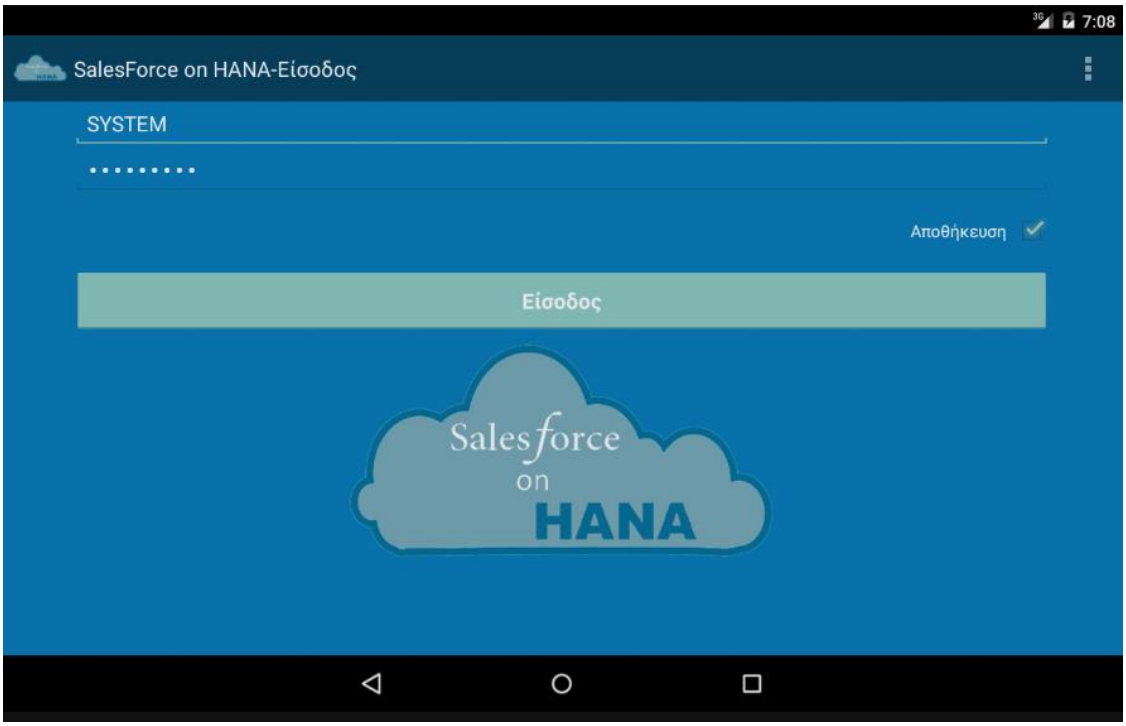
Τέλος το αρχείο .xsaccess που βρίσκεται στο πακέτο SFA καθορίζει τον τρόπο ταυτοποίησης των χρηστών στην εφαρμογή. Στην εφαρμογή αυτή έχει επιλεγεί Basic HTTP Authentication. Ο κώδικας του αρχείου είναι:

```
{
    "exposed": true,
    "authentication": [{
        "method": "Basic"
    }],
    "authorization": ["sap.devs.sfa::Execute"]
}
```

Υλοποίηση στο Android

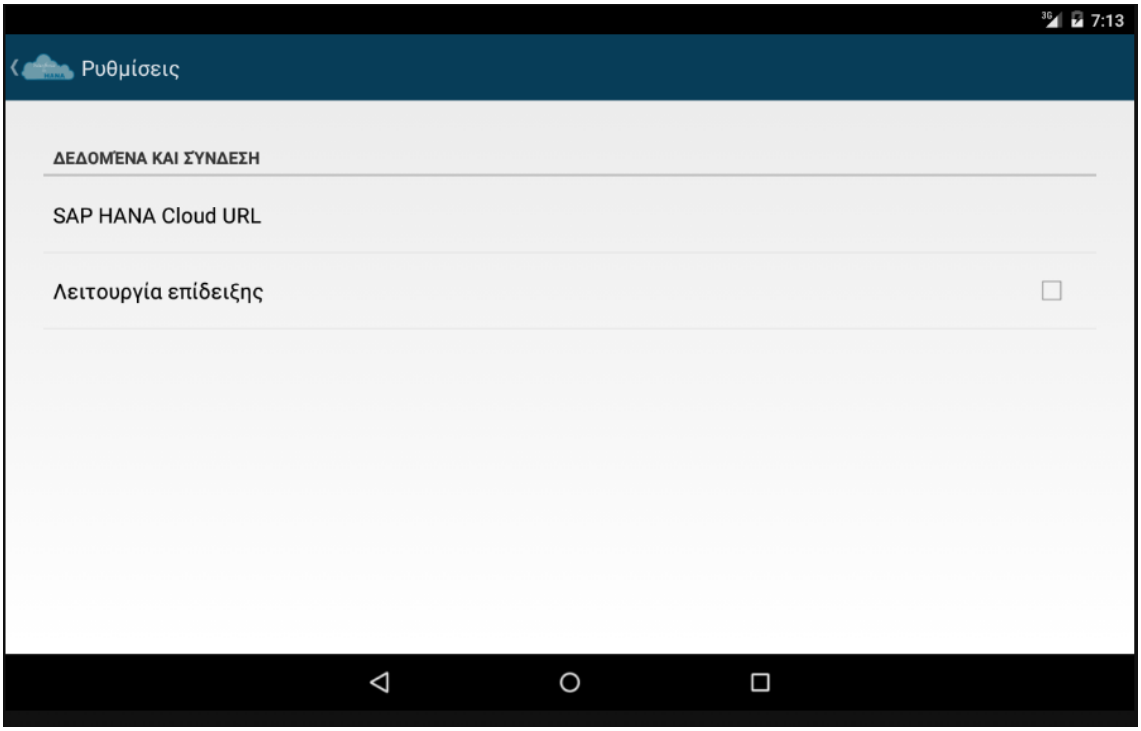
Για την υλοποίηση της εφαρμογής στο Android χρησιμοποιήθηκε στο Android Studio 1.2.2. Όταν χρειάζεται η επικοινωνία με το SAP HANA Cloud για τη μεταφορά δεδομένων, οι υπηρεσίες OData καταναλώνονται ως κλασσικά Rest Services με Basic HTTP Authentication.

Η πρώτη οθόνη της εφαρμογής είναι η παρακάτω:



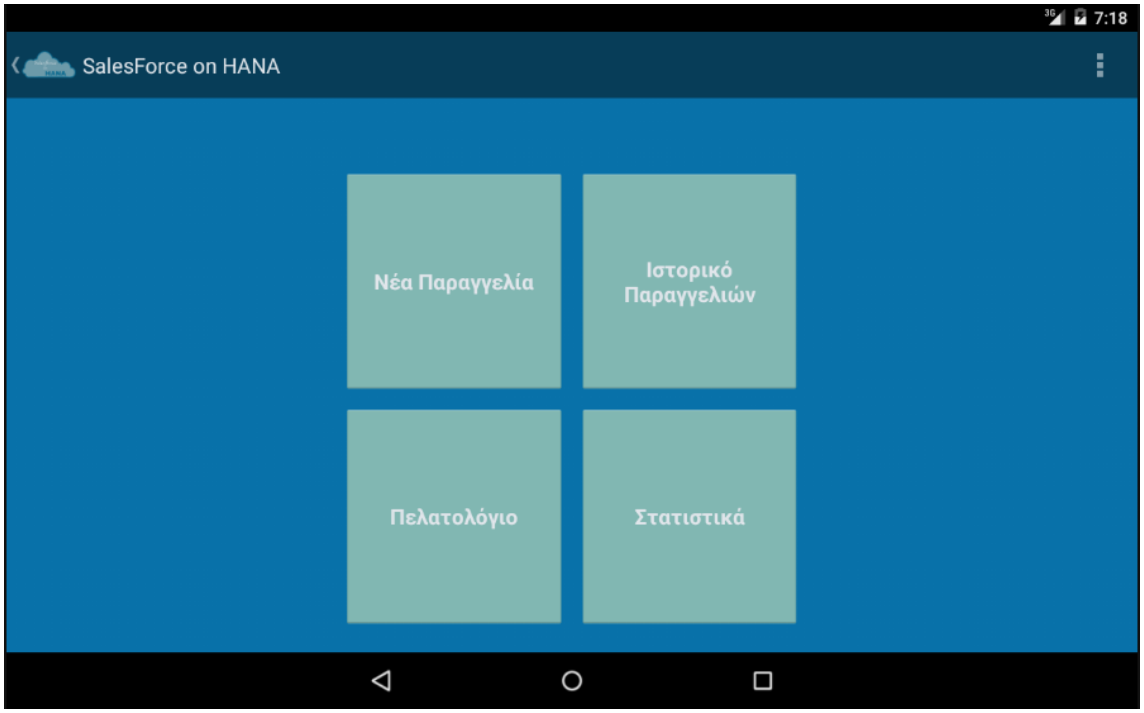
Εικόνα 55. Η οθόνη εισόδου

Κατά την πρώτη εκτέλεση της εφαρμογής θα ανοίξει η οθόνη ρυθμίσεων, η οποία μπορεί να ζητηθεί από το χρήστη και αργότερα στις επιλογές της οθόνης εισόδου. Στην οθόνη των ρυθμίσεων καταχωρείται το URL end point του SAP HANA Cloud, και επιλέγεται εάν θα εκτελείται η εφαρμογή σε κατάσταση πραγματικής λειτουργίας με το SAP HANA, ή σε λειτουργία επίδειξης με δοκιμαστικά δεδομένα.



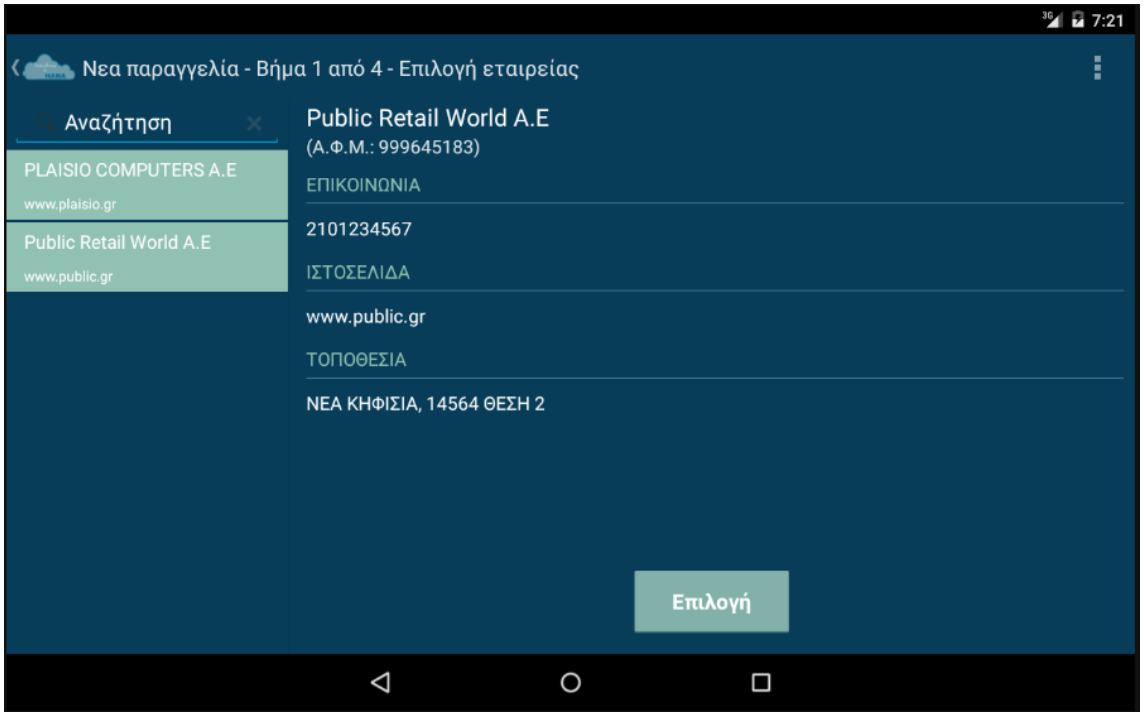
Εικόνα 56. Οθόνη ρυθμίσεων

Κατά την είσοδο στην εφαρμογή ο χρήστης μπορεί να επιλέξει εάν θέλει να αποθηκευτούν σε ασφαλή τοποθεσία στη συσκευή τα στοιχεία εισόδου. Πατώντας το κουμπί είσοδος εκτελείται ταυτοποίηση του χρήστη καλώντας το ping HANA Service. Εάν η συσκευή είναι συνδεδεμένη στο internet και τα στοιχεία του χρήστη γίνουν αποδεκτά από το SAP HANA Cloud, τότε ο χρήστης εισέρχεται στο κεντρικό μενού της εφαρμογής.



Εικόνα 57. Η κεντρική οθόνη της εφαρμογής

Πατώντας το κουμπί νέα παραγγελία μεταφέρεται στο πρώτο από τα 4 βήματα στο οποίο καλείται να διαλέξει τον πελάτη.

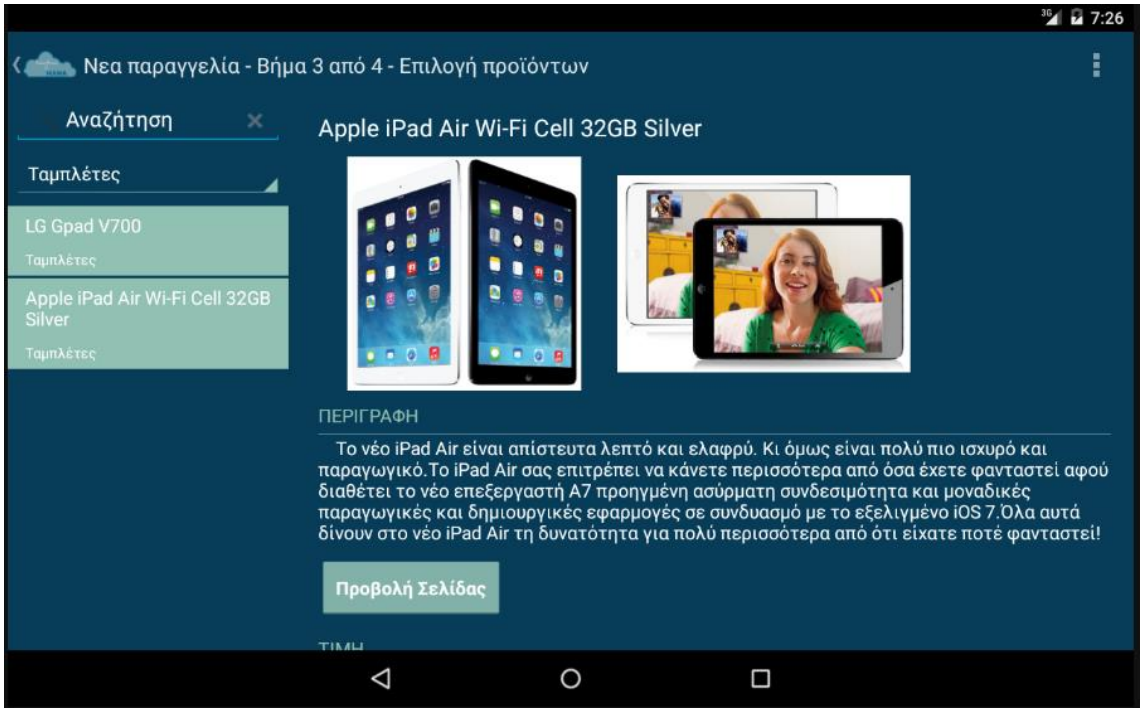


Εικόνα 58. Νέα παραγγελία - Βήμα 1 από 4 - Επιλογή Πελάτη

Αφού επιλεγεί ο πελάτης, μεταφερόμαστε στο βήμα 2 όπου πρέπει να επιλεγεί το υποκατάστημα.

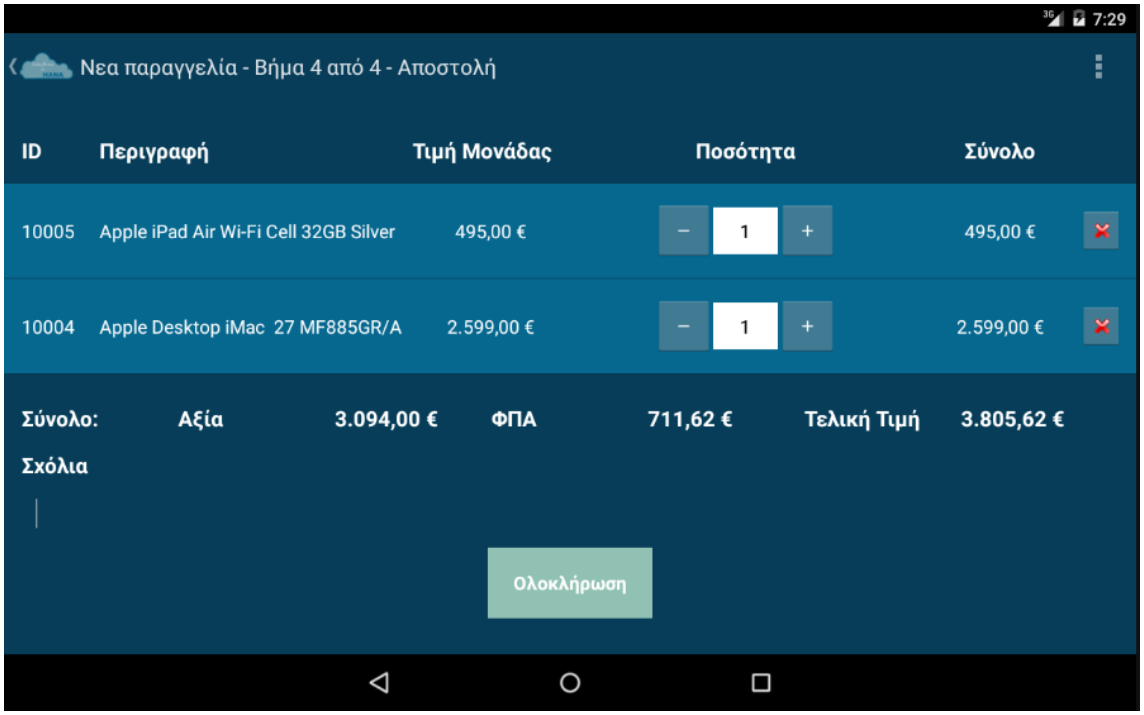
Εικόνα 59. Νέα παραγγελία - Βήμα 2 από 4 - Επιλογή Υποκαταστήματος

Στο βήμα 3 εμφανίζεται ο κατάλογος των προϊόντων και ο πωλητής προσθέτει στο καλάθι τα προϊόντα της παραγγελίας. Κατά την εμφάνιση των προϊόντων ο πωλητής μπορεί να μεγεθύνει τις φωτογραφίες και να μεταβεί στη ιστοσελίδα του κατασκευαστή του προϊόντος.



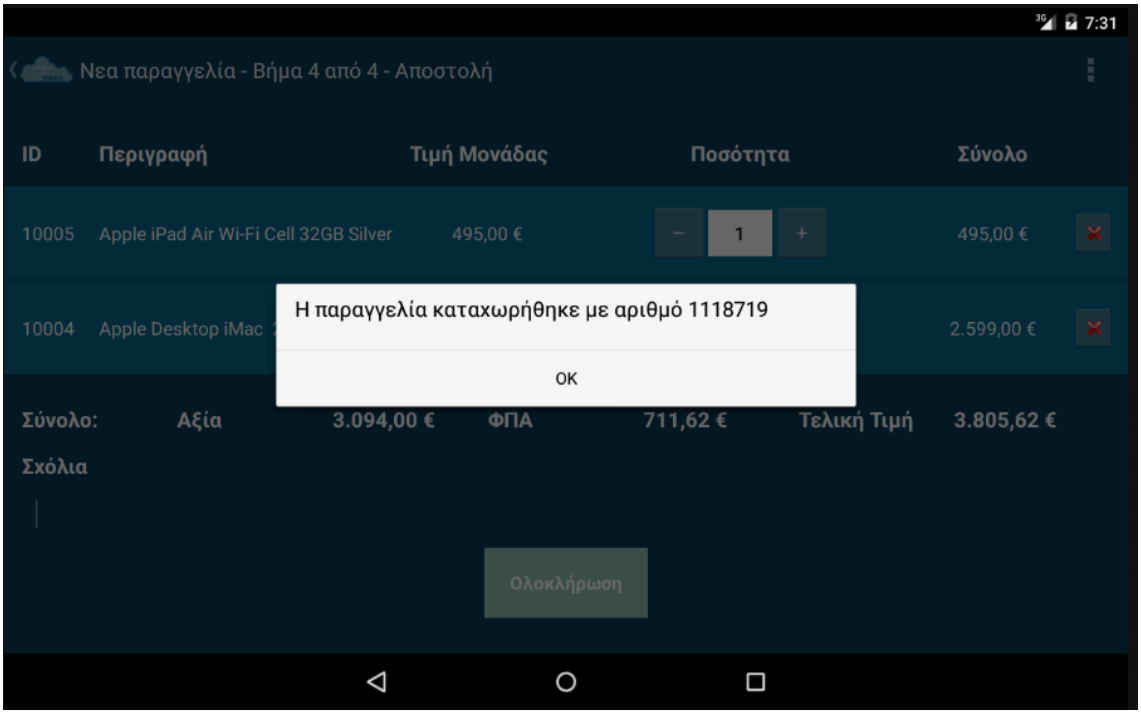
Εικόνα 60. Νέα παραγγελία – Βήμα 3 από 4 – Προσθήκη Προϊόντων στο καλάθι της παραγγελίας

Μόλις προσθέσει όλα τα προϊόντα στο καλάθι, μπορεί να μεταβεί στο 4^{ο} και τελευταίο βήμα της διαδικασίας, όπου καταχωρεί την παραγγελία στο SAP HANA Cloud.



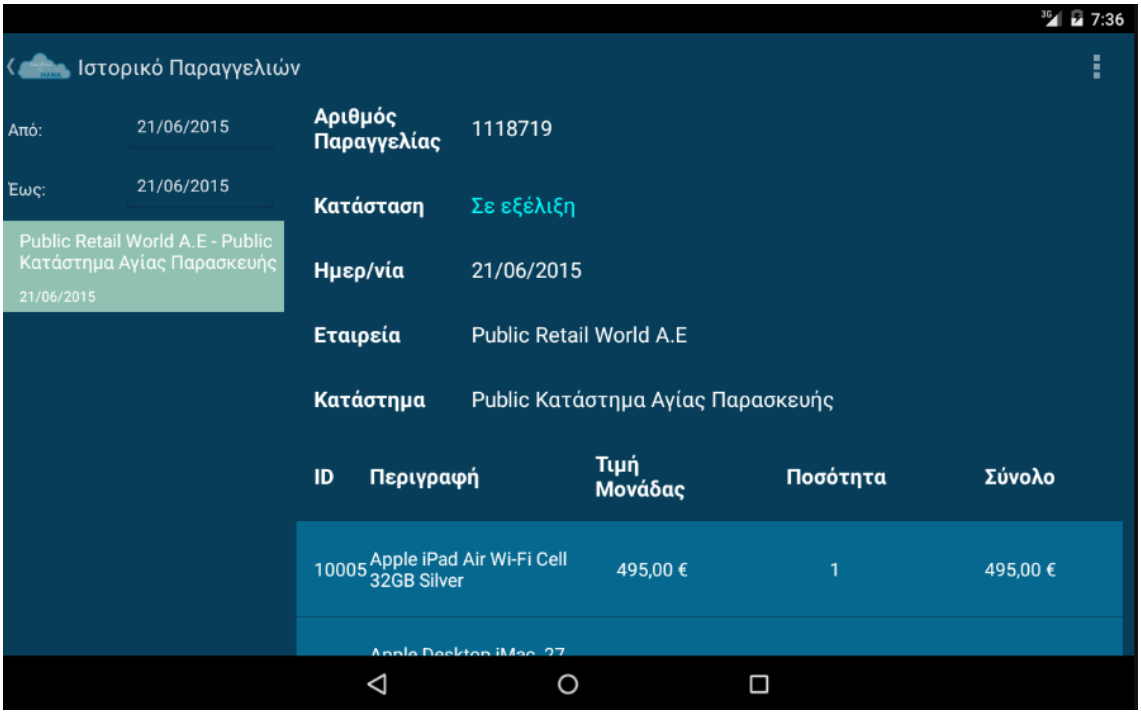
Εικόνα 61. Νέα παραγγελία – Βήμα 4 από 4 – Επισκόπηση παραγγελίας και αποστολή στο SAP HANA Cloud

Πατώντας το κουμπί ολοκλήρωση, εμφανίζεται η απάντηση από το SAP HANA Cloud με τον αριθμό που δόθηκε στην παραγγελία.



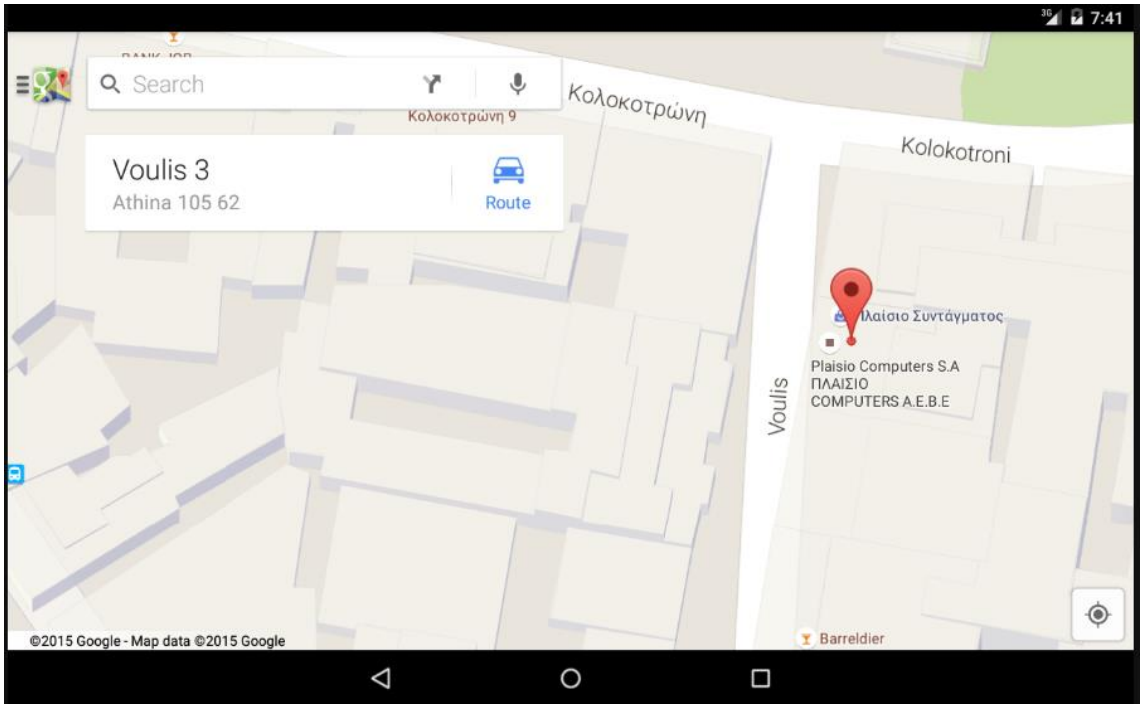
Εικόνα 62. Επιτυχής καταχώρηση στο SAP HANA Cloud

Στη συνέχεια ο χρήστης μεταφέρεται στο αρχικό μενού. Εάν επιλέξει προβολή ιστορικού παραγγελιών, θα δει το ιστορικό των παραγγελιών που έχει καταχωρήσει και ανάμεσά τους και την αυτήν που μόλις καταχώρησε.



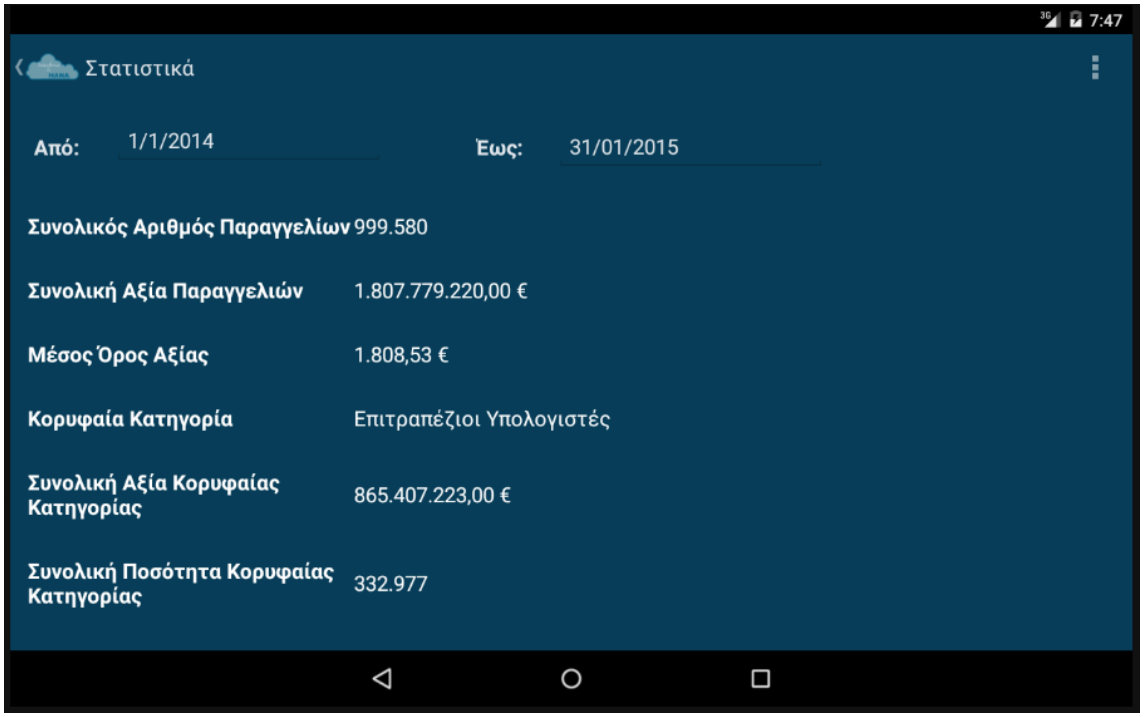
Εικόνα 63. Προβολή Ιστορικού Παραγγελιών

Από το κεντρικό μενού εάν επιλέξει το κουμπί πελατολόγιο, μπορεί να δει τα στοιχεία των πελατών και τα υποκαταστήματά τους, καθώς και να τα εμφανίσει στο προεπιλεγμένο πρόγραμμα χαρτών της συσκευής και να πλοηγηθεί μέσω GPS σε αυτά.



Εικόνα 64. Προβολή υποκαταστήματος στο Χάρτη

Τέλος από την κεντρική οθόνη επιλέγοντας στατιστικά, μπορεί να δει KPIs πωλήσεων για μια συγκεκριμένη περίοδο. Για λόγους επίδειξης και δοκιμής της απόδοσης του SAP HANA Cloud, όπως αναφέρθηκε, καταχωρήθηκαν αυτοματοποιημένα πολλές εγγραφές (~3 εκ). Η απάντηση του SAP HANA Cloud σε αυτά τα στατιστικά είναι άμεση, και δεν είναι ορατή κάποια διαφορά στην απάντηση από το SAP HANA Cloud σε σχέση με μια απλή κλήση σε ODATA Service.



Εικόνα 65. Οθόνη προβολής των στατιστικών

Συμπεράσματα

Το SAP HANA ανατρέπει την υπάρχουσα αρχιτεκτονική των υπολογιστικών συστημάτων, βάζοντας στο επίκεντρο της εφαρμογής τη Βάση Δεδομένων. Πλέον το επίπεδο του Application Server ενώνεται με το επίπεδο της Database, προσφέροντας στις εφαρμογές εξωπραγματικές επιδόσεις. Από τις δοκιμές που πραγματοποιήθηκαν μέσω της εφαρμογής android ο χρόνος ανάκτησης συγκεντρωτικών δεδομένων από πίνακες με εκατομμύρια εγγραφές είναι ακαριαίος.

Για να επιτευχθούν αυτές οι επιδόσεις πρέπει μεγάλο μέρος της εφαρμογής να εκτελείται στο SAP HANA, ώστε να ακολουθηθεί το μοντέλο Code To Data. Για νέες εφαρμογές αυτό δεν αποτελεί πρόβλημα, όμως για τη μεταφορά υφιστάμενων εφαρμογών απαιτείται αρκετή προσπάθεια. Αυτό μπορεί να οδηγήσει κάποιες εφαρμογές στην ενδιάμεση λύση της χρήσης του SAP HANA ως κλασσικής RDBMS. Οι εφαρμογές αυτές θα έχουν μεγάλη αύξηση της ταχύτητάς τους λόγω του In-Memory της Βάσης Δεδομένων, αλλά δε θα φθάσουν στα επίπεδα απόδοσης που θα προέκυπταν από τον ανασχεδιασμό τους σε μοντέλο Code to Data.

Η πλατφόρμα του SAP HANA, υιοθετεί σύγχρονα και ανοικτά πρότυπα για την επικοινωνία των εφαρμογών πελατών με αυτό. Υπάρχει σαφής διαχωρισμός ανάμεσα στα προγράμματα του Server που αφορούν τα δεδομένα και στην πρόσβαση σε αυτά από εφαρμογές Client μέσω OData και Rest Web Services. Αυτός ο διαχωρισμός δημιουργεί ένα μοναδικό σημείο για όλα τα δεδομένα, στα οποία έχουν πρόσβαση οι εξωτερικές εφαρμογές που εκτελούνται σε μια πληθώρα διαφορετικών συσκευών. Αυτή η αρχιτεκτονική είναι ιδανική για χρήση στο Internet Of Things (IoT), την εξέλιξη του διαδικτύου στην οποία οι συσκευές θα επικοινωνούν απευθείας με άλλες συσκευές μέσω web services.

Το SAP HANA Cloud Platform προσφέρει μια λύση για γρήγορη εκκίνηση στη χρήση του SAP HANA, χωρίς αλλαγές στην εσωτερική μηχανογραφική υποδομή. Προσφέρει μια πληθώρα σύγχρονων χαρακτηριστικών, η οποία βοηθά τόσο στην πλήρη διαχείριση του SAP HANA μέσα από το Cloud, όσο και στην ενσωμάτωση του SAP HANA Cloud στην υπάρχουσα μηχανογραφική υποδομή της εταιρείας. Στα αρνητικά του SAP HANA Cloud είναι η αρκετά υψηλή τιμή, ειδικά για την Ελληνική Αγορά, γεγονός που εν μέρει οφείλεται στις πολύ υψηλές απαιτήσεις Hardware που έχει η πλατφόρμα SAP HANA.

Η SAP AG ανασχεδιάζει τη σουίτα των εφαρμογών της (SAP ERP,CRM κλπ), ώστε να μπορούν να εκτελούνται στο SAP HANA με βέλτιστη απόδοση. Η μετάβαση των υφιστάμενων εγκαταστάσεων SAP Business Suite σε υποδομή HANA θα οδηγήσει σε καθολική επικράτησή του, στο περιβάλλον των επιχειρήσεων.

Για τη χρήση του SAP HANA ως βάση δεδομένων για εφαρμογές εκτός SAP, υπάρχει πλέον ο ανταγωνισμός από τα κλασσικά RDBMS (MSSQL,Oracle) τα οποία προσφέρουν υβριδικές υλοποιήσεις In-memory, δίνοντας τη δυνατότητα επιλεγμένοι πίνακες να μεταφερθούν στη μνήμη, βοηθώντας την αποσυμφόρηση της βάσης για μεγάλους πίνακες. Αν και υστερούν σε απόδοση σε σχέση με το SAP HANA που τρέχει πλήρως στην κύρια μνήμη και δεν υποστηρίζουν το Code To Data Paradigm, επειδή δεν απαιτούν κάποια αλλαγή στις εφαρμογές και λειτουργούν με το υφιστάμενο hardware, αποτελούν ελκυστικές λύσεις ειδικά για τις υφιστάμενες εφαρμογές.

Βιβλιογραφία

[1] H. Plattner, A. Zeier, 2012, “In-Memory Data Management, Technology and Applications”, 2nd Edition, Springer

[2] H. Plattner, 2014, “In-Memory Data Management (2014) - Implications on Enterprise Systems”, Course Material, Hasso Plattner Institute, https://open.hpi.de/courses/imdb2014

[3] D. Abadi, S. Madden, and M. Ferreira, 2006, “Integrating compression and execution in column-oriented database systems”. In Proceedings of the 2006 ACM SIGMOD international conference on Management of data, SIGMOD ’06, pages 671–682, New York, NY, USA, 2006. ACM.

[4] J. Gruschke, J. Weiler, 2014, “ABAP Development for SAP HANA”, Course Material, https://open.sap.com

[5] B. Berg, P. Silvia, 2014, “SAP HANA an Introduction”, 3rd Edition, SAP Press

[6] T. Schneider, E. Westenberger, H. Gahm, 2014, “ABAP Development for SAP HANA”, SAP Press

[7] J. Haun, C. Hickman, D. Loden, R. Wells, 2013, “Implementing SAP HANA”, SAP Press

[8] SAP AG, “SAP HANA Developer Guide”, SAP HANA Platform SPS 08, 2014, Version 1.1, http://help.sap.com/hana/SAP_HANA_Developer_Guide_en.pdf

[9] SAP AG, SAP Help Portal, 2014, http://help.sap.com

[10] V. Sikka , 2013, “Re-thinking the performance of information processing systems”, Data Engineering (ICDE), IEEE 29th International Conference 8-12 April 2013

[11] F. Färber, , N.May, W. Lehner, P.Große, I. Müller, H. Rauhe, & , J. Dees (2012). “The SAP HANA Database - An Architecture Overview”. IEEE Data Eng. Bull., 35(1), 28-33.

[12] J. Boese, C. Tosun, C. Mathis, and F. Faerber. 2012. “Data management with SAPs in-memory computing engine”. In *Proceedings of the 15th International Conference on Extending Database Technology* (EDBT '12), ACM

[13] V. Sikka, F. Färber, W. Lehner, S. Kyun Cha, T. Peh, and C. Bornhövd. 2012. “Efficient transaction processing in SAP HANA database: the end of a column store myth”. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (SIGMOD '12).

[14] F. Färber, S. Kyun Cha, J. Primsch, C.f Bornhövd, S. Sigg, and W. Lehner. 2012. “SAP HANA database: data management for modern business applications”. *SIGMOD Rec.* 40, 4 (January 2012).

[15] I. Müller, C. Ratsch, F. Faerber, 2014, “Adaptive String Dictionary Compression in In-Memory Column-Store Database Systems”, 7th International Conference on Extending Database Technology (EDBT) – March 2014, Athens Greece, Page(s): 283–294.

[16] SAP AG, “SAP HANA Administration Guide”, 2014, Version 1.1, http://help.sap.com/hana/SAP_HANA_Administration_Guide_en.pdf

[17] H. Rauhe, 2011, “HANA In-Memory Computing Engine, Concepts and Architecture Overview”, SAP AG, Technische Universität Ilmenau, FG Datenbanken

[18] C. Hallenbeck, C. Bornhoevd, R. Pledereder, 2013, “SAP HANA - Real Time Computing”, SAP AG, Presentation, http://web.stanford.edu/class/ee380/Abstracts/130522-slides.pdf

[19] S. Hildenbrand, 2012, “Scaling Out Column Stores: Data, Queries, and Transactions”, Dissertation ETH NO. 20314

[20] J. Word, 2013, “SAP HANA Essentials”, Epistemy Press LLC.

[21] J. Lee, Y. Sik Kwon, F. Färber, M. Muehle, C. Lee, C. Bensberg, J. Yeon Lee, A. Lee, W. Lehner, 2013, “SAP HANA Distributed In-Memory Database System: Transaction, Session, and Metadata Management”, Data Engineering (ICDE), IEEE 29th International Conference 8-12 April 2013

[22] Mark Walker, 2013, “Software Development on the SAP HANA Platform”, Pact Publishing

[23] SAP AG, 2013, 2013, “How To Delta Merge for SAP HANA and SAP NetWeaver BW powered by SAP HANA”, SAP AG, http://scn.sap.com/docs/DOC-27558

[24] SAP Community Network, 2014, http://scn.sap.com

[25] SAP HANA Website, 2014, http://www.saphana.com

[26] M. Pizzo, R. Handl, M. Zurmuehl, 2014, “OData Version 4.0 Part 1: Protocol Plus Errata 01, OASIS Standard incorporating Approved Errata 01”, https://www.oasis-open.org/

[27] OData Website, 2014, http://www.odata.org/

[28] SAP AG, “SAP HANA Troubleshooting and Performance Analysis Guide”, 2014, http://help.sap.com/saphelp_hanaplatform/helpdata/en/da/1332489e1c44009e8f050352751852/content.htm

[29] SAP AG, “SAP HANA SQLScript Reference”, 2014, http://help.sap.com/hana/SAP_HANA_SQL_Script_Reference_en.pdf

[30] SAP AG, “SAP HANA Modeling Guide”, 2014, http://help.sap.com/hana/SAP_HANA_Modeling_Guide_en.pdf

[31] SAP AG, “SAP HANA Developer Guide for SAP HANA Studio”, SAP HANA Platform SPS 09, 2014, Version 1.0, http://help.sap.com/hana/SAP_HANA_Developer_Guide_for_SAP_HANA_Studio_en.pdf

[32] SAP AG, “SAP HANA Security Guide”, SAP HANA Platform SPS 09, 2014, Version 1.0, http://help.sap.com/hana/SAP_HANA_Security_Guide_en.pdf

[33] Wikipedia, www.wikipedia.org

[34] Chris Woodruff , 2015, “31 Days of OData”, chriswoodruff.com

[35] SAP AG, “SAP HANA Cloud Documentation”, 2015, https://help.hana.ondemand.com/help/frameset.htm

[36] SAP AG, 2015, “Next Steps in Software Development on SAP HANA”, Course Material, https://open.sap.com