

Περιεχόμενα

Περιεχόμενα

1. Εισαγωγή - Σκοπός.....	2
1.1 Περίληψη.....	3
1.2 Κοστολόγηση του Project	3
2. Σύντομη Ιστορική αναδρομή της Oracle.....	4
3. Υλοποίηση της βάσης δεδομένων	5
3.1. Το διάγραμμα Entity Relationship.....	5
3.2 Συνδέσεις μεταξύ πινάκων	6
3.3 Τύποι δεδομένων	8
4 Υλοποίηση της βάσης με χρήση Oracle – Sql.....	8
4.1 Υλοποίηση της βάσης.....	8
4.2 Δημιουργία αυτοματοποιημένων ελέγχων και περιορισμών (Triggers).....	10
5. Εισαγωγή δεδομένων	15
6. Δημιουργία ειδικών ερωτημάτων - Queries	17
7. Συμπεράσματα - εκτιμήσεις	26
8. Βιβλιογραφία.....	27
Πίνακας επισυναπτόμενων αρχείων.....	27

1. Εισαγωγή - Σκοπός

Ο σκοπός της εργασίας είναι να γίνει μελέτη για την υλοποίηση βάσης δεδομένων που θα εξυπηρετεί μια νέα εταιρεία πληροφορικής που θα αναλαμβάνει έργα τεχνολογίας λογισμικού. Αντικείμενο της εργασίας θα είναι η κατασκευή μιας βάσης δεδομένων που θα αποσκοπεί στην καταγραφή έργων και υποέργων σε διάφορα στάδια ανάπτυξης, επεξεργασίας και διαχείρισης αυτών. Θα καταχωρούνται και θα επεξεργάζονται δεδομένα που αφορούν πληροφορίες σχετικά με θέματα προς συζήτηση, σχετικά έγγραφα, προσωπικά στοιχεία πελατών αλλά και των στελεχών της. Θα καταγράφονται επίσης λεπτομέρειες σχετικά με συνεννοήσεις, ημερομηνίες έναρξης έργων και καταληκτικές ημερομηνίες αυτών καθώς οικονομικά στοιχεία διαχείρισης της εταιρείας (μισθοί – κοστολόγια έργων – risk analysis). Απώτερος σκοπός θα είναι η θετική ή αρνητική κερδοφόρος ανάδειξη του κάθε έργου μετά την ωρίμανση του.

1.1 Περίληψη

Για την υλοποίηση της βάσης και την επεξεργασία των δεδομένων θα γίνει χρήση της Oracle και ειδικών ερωτημάτων με χρήση της PL/SQL, που αποτελεί extension της Structured Query Language (SQL) που χρησιμοποιείται στην Oracle.

Αρχικά γίνεται μια σύντομη ιστορική αναδρομή της Oracle. Έπειτα παρουσιάζεται το κοστολόγιο του συγκεκριμένου Project με αναλυτική περιγραφή των ωρών εργασίας που θα απαιτηθούν. Παρουσιάζει ο σχεδιασμός της βάσης δεδομένων με διαγραμματική απεικόνιση σε ER Diagram chart. Στην συνέχεια γίνεται περιγραφή όλων των πινάκων που θα δημιουργηθούν, τα πεδία τους, οι τύποι δεδομένων που θα χρησιμοποιηθούν και οι συσχετίσεις μεταξύ τους. Ακολούθως θα γίνει αναλυτική περιγραφή ειδικών περιορισμών με αυτοματοποιημένες μεθόδους (Triggers) σε συγκεκριμένα πεδία εφαρμογής. Αφού ολοκληρωθούν οι ανωτέρω διαδικασίες με επιτυχία θα γίνει εισαγωγή υποθετικών δεδομένων στους πίνακες της βάσης με συγκεκριμένη δομή και ιεραρχία. Τελευταία ενότητα της εργασίας αποτελεί συγγραφή ειδικών ερωτημάτων (queries) για την εμφάνιση συγκεκριμένων δεδομένων που απαιτούνται.

Αυτόματοι μέθοδοι και περιορισμοί που θα υλοποιηθούν σε Oracle – Sql:

1. Έλεγχος καταχωρήσεων για τη σωστή εισαγωγή δεδομένων τύπου email στο αντίστοιχο πεδίο.
2. Εισαγωγή δεδομένων μισθών υπαλλήλων με συγκεκριμένο αριθμητικό εύρος.
3. Έλεγχος καταχωρήσεων για την απασχόληση υπαλλήλων με ένα και μόνο έργο.
4. Έλεγχος καταχωρήσεων για την ημερομηνία έναρξης και λήξης των έργων.
5. Αυτοματοποιημένη μέθοδος για τον έλεγχο δικαιοδοσίας υποέργου σε υπεύθυνο.
6. Αυτοματοποιημένη μέθοδος για έλεγχο της αποκλειστικής ανάθεσης συγκεκριμένων υπευθύνων για την επικοινωνία με τον εκάστοτε πελάτη του κάθε έργου.
7. Έλεγχος καταχωρήσεων σχετικά με τα γεωγραφικά χαρακτηριστικά που απαιτούνται.

Ερωτήματα που θα υλοποιηθούν σε Oracle – Sql (PL/SQL). για την εμφάνισης συγκεκριμένων δεδομένων:

1. Εμφάνιση της λίστας με τα στοιχεία όλων των πελατών της εταιρείας.
2. Εμφάνισης λίστας με τα έργα ταξινομημένα ανά ημερομηνία ανάληψης.
3. Εμφάνιση λίστας με έργα – υποέργα – εργασίες ταξινομημένα με ημερομηνία εκκίνησης κάθε φάσης.
4. Εμφάνιση αριθμού έργων ανά κατάσταση, δηλαδή αυτά που απορρίφθηκαν, είναι ενεργά ή ολοκληρωμένα.
5. Εμφάνιση λίστας με τους υπαλλήλους ανά φάση του κάθε έργου με την χρονική διάρκεια αυτής. Απαραίτητη προϋπόθεση θα είναι η εισαγωγή κωδικού έργου.
6. Εμφάνιση λίστας ωρών εργασίας για κάθε έργο με ανάλυση ανά υπάλληλο.
7. Εμφάνιση με οικονομικά στοιχεία κάθε έργου, δηλαδή προϋπολογισμό που περιλαμβάνει, τελικό κόστος έργου σε ώρες εργασίας σε συνδυασμό με το κόστος κάθε ώρας εργασίας, ποσοστιαία ενασχόληση του υπαλλήλου στο έργο, το τελικό ποσό που αναλογεί στο μισθό του και η ενασχόληση του με το σχετικό έργο.

1.2 Κοστολόγηση του Project

Για την οργάνωση της βάσης δεδομένων που δημιουργήθηκε απασχολήθηκαν δύο προγραμματιστές με τον σχεδιασμό των βάσεων δεδομένων και την υλοποίηση του κώδικα σε γλώσσα Oracle – Sql. Έγινε χρήση ειδικών ερωτημάτων που ζητήθηκαν σε pl/sql προκειμένου

να εμφανίζονται συγκεκριμένα δεδομένα προς επεξεργασία. Συνολικά απαιτήθηκαν 80 ώρες εργασίας με το κόστος της κάθε ώρας να κοστολογείται στα 25 ευρώ. Το συνολικό κόστος ανέρχεται στα 2000 ευρώ.

2. Σύντομη Ιστορική αναδρομή της Oracle

Η Oracle Corporation ως μια μεγάλη η πολυεθνική εταιρεία τεχνολογίας ακολουθείτε από μια πορεία προόδου μέσω της ανάπτυξης και της εμπορίας συστημάτων υλικού υπολογιστών, εταιρικού λογισμικού και βάσεων δεδομένων.

Στις 16 Ιουνίου 1977 δημιουργήθηκαν τα Εργαστήρια Ανάπτυξης Λογισμικού (SDL) στο Santa Clara της Καλιφόρνια από τους Larry Ellison, Bob Miner και Ed Oates. Μέχρι και το έτος 1978 γεννήθηκε η πρώτη Oracle, γραμμένη σε γλώσσα assembly που τρέχει σε PDP-11 κάτω από το RSX-11 σε 128K μνήμης. Η έκδοση 1 της Oracle δεν κυκλοφόρησε ποτέ επίσημα και η υλοποίηση διαχώρισε τον κώδικα Oracle από τον κωδικό χρήστη. Η Oracle βαφτίστηκε έτσι καθώς ήταν η κωδική ονομασία ενός έργου της CIA στο οποίο εργάστηκαν και οι τρεις αρχικοί ιδρυτές όταν ήταν στην Amex Corporation.

Μέχρι τον Νοέμβριο του 1995 η Oracle είχε γίνει μια από τις πρώτες μεγάλες εταιρείες λογισμικού που ανέφεραν μια στρατηγική διαδικτύου και πρόσφερε το πρώτο RDBMS 64 bit σύστημα. Το 1996 κυκλοφόρησε το πρόγραμμα περιήγησης ιστού ως μέρος του Oracle PowerBrowser και κινήθηκε προς μια αρχιτεκτονική βασισμένη σε ανοιχτά πρότυπα, με δυνατότητα web. Τον Απρίλιο του 1997 η Oracle κυκλοφόρησε τη πρώτη της έκδοση του Discoverer και τον Ιούνιο του ίδιου έτους κυκλοφόρησε το Oracle v8 με τεχνολογία Internet, υποστήριξη για terabyte δεδομένων και τεχνολογία αντικειμένων SQL. Τον Σεπτέμβριο του ίδιου έτους, η Oracle Corporation παρουσίασε το Oracle Jdeveloper (ενσωματωμένο περιβάλλον ανάπτυξης Java της Oracle)

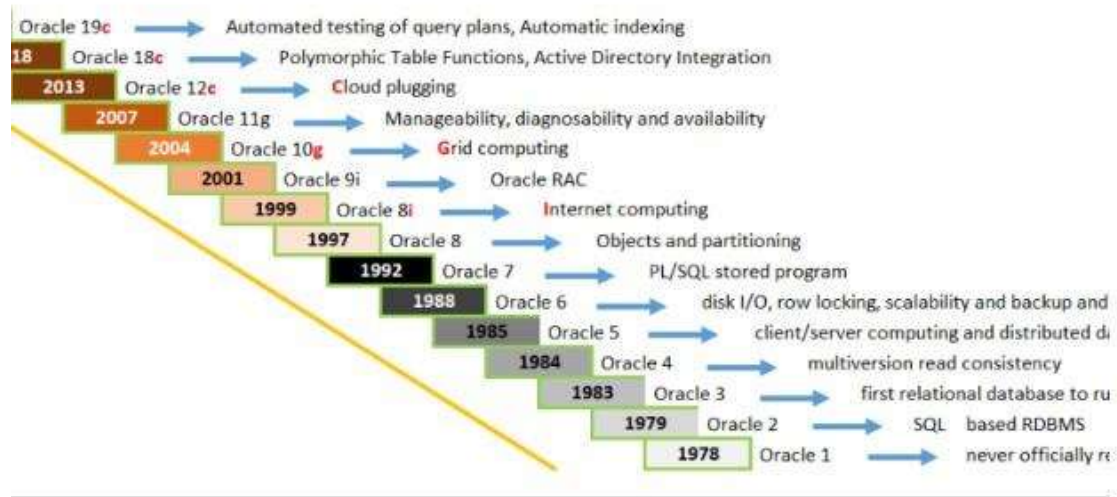
Το 1998 ήταν μια χρονιά με επιρροή για την Oracle Corporation καθώς τον Μάιο του ίδιου έτους κυκλοφόρησε το Oracle Applications 2 και τον Απρίλιο ανακοινώθηκε ότι θα ενσωματωθεί μια εικονική μηχανή Java με τη βάση δεδομένων Oracle.

Η Oracle σήμερα

Η Oracle ανακοινώνει την Oracle Database 21c, η οποία είναι η πιο πρόσφατη έκδοση της κορυφαίας converged βάσης δεδομένων στον κόσμο, είναι διαθέσιμη στο Oracle Cloud, συμπεριλαμβανομένης της παροχής της Oracle Autonomous Database.

Η Oracle Database 21c περιλαμβάνει περισσότερες από 200 νέες καινοτομίες, που περιλαμβάνουν αμετάβλητους πίνακες blockchain, In-Database JavaScript, εγγενή δυαδικό τύπο δεδομένων JSON, AutoML για μηχανική μάθηση (Machine Learning) εντός της βάσης δεδομένων και χώρο μόνιμης μνήμης, καθώς και βελτιώσεις για διεργασίες εντός μνήμης, απόδοση επεξεργασίας γραφημάτων, τεμαχισμός (sharding), δυνατότητες multitenant και ασφάλεια. Σε αντίθεση με τις βάσεις δεδομένων ειδικού σκοπού άλλων προμηθευτών στο cloud ή σε περιβάλλον εσωτερικής εγκατάστασης, η Oracle Database 21c παρέχει υποστήριξη για απαιτήσεις περί πολλαπλών μοντέλων, πολλαπλών φόρτων εργασίας και multi-tenant – όλα αυτά με έναν σύγχρονο, converged μηχανισμό βάσης δεδομένων.

History of Oracle Database Versions



Oracle History

Πηγή: <https://ittutorial.org/oracle-version-history-oracle-database-release-versions/>

3. Υλοποίηση της βάσης δεδομένων

3.1. Το διάγραμμα Entity Relationship

Αρχικά έγινε ο σχεδιασμός της βάσης δεδομένων απεικονίζοντας διαγραμματικά με χρήση Entity Relationship (ER) diagram τους πίνακες της βάσης και τις συσχετίσεις μεταξύ τους. Έτσι σχεδιάστηκαν δεκαέξι πίνακες:

1. **CUSTOMERS:** Περιέχει όλα τα δεδομένα που αφορούν τους πελάτες της εταιρείας (προσωπικά στοιχεία, στοιχεία επικοινωνίας).
2. **CUSTOMERS_CATEGORIES:** Περιλαμβάνει την κατηγοριοποίηση του είδους της επιχείρησης και το είδος του έργου προς ανάληψη.
3. **CATEGORIES:** Περιέχει τα δεδομένα που αφορούν τη περιγραφή της κάθε κατηγορίας.
4. **PROJECTS:** Περιέχει όλα τα δεδομένα που αφορούν την περιγραφή του έργου, ημερομηνίες έναρξης – λήξης των ενεργών και ανενεργών έργων και την εταιρεία πελάτη που το έχει αναθέσει.
5. **SUBPROJECTS:** Περιλαμβάνει τα δεδομένα που αφορούν ημερομηνίες έναρξης και λήξης των υποέργων και την κατάσταση που βρίσκονται.
6. **PROJECTS_STATUS:** Περιλαμβάνει την περιγραφή της κατάστασης του κάθε έργου.
7. **TASKS:** Περιλαμβάνει όλα τα δεδομένα που αφορούν τις διεργασίες που γίνονται, τις ώρες εργασίας, αρχεία, ημερομηνίες των διεργασιών και την περιγραφή τους.
8. **EMPLOYEES:** Περιλαμβάνει τα δεδομένα που αφορούν το προσωπικό της εταιρείας (προσωπικά στοιχεία, μισθός, ώρες εργασίας).
9. **EMPLOYEE_CATEGORIES:** Περιλαμβάνει την περιγραφή της κατηγορίας του κάθε έργου.
10. **EMPLOYEES_PROJECTS:** Περιλαμβάνει δεδομένα που αφορούν τα έργα και την κατάσταση που βρίσκονται.
11. **COUNTRIES:** Περιλαμβάνει δεδομένα που καταγράφονται χώρες.
12. **STATES:** Περιλαμβάνει δεδομένα που αφορούν τις πολιτείες.
13. **TOWNS:** Περιλαμβάνει δεδομένα που αφορούν ονόματα πόλεων.

14. **LEADS**: Περιλαμβάνει στοιχεία με όλα τα δεδομένα που αφορούν το έργο όπως ημερομηνίες, εργαζόμενο που το έχει αναλάβει, απαραίτητα αρχεία, παρατηρήσεις κλπ.
15. **LEADS_STATUS**: Περιλαμβάνει στοιχεία με όλα τα απαραίτητα δεδομένα που αφορούν την κατάσταση στην οποία βρίσκεται με όλες τις απαραίτητες παρατηρήσεις.
16. **PROJECT_CONTACTS**: Περιλαμβάνει στοιχεία με όλα τα απαραίτητα δεδομένα που αφορούν το έργο και τον εργαζόμενο που το έχει αναλάβει.

Οι πίνακες με τα πεδία τους καταγράφηκαν λεπτομερώς σε αρχικό στάδιο σε αρχείο excel το οποίο και επισυνάπτεται .¹

3.2 Συνδέσεις μεταξύ πινάκων.

Για τον σχεδιασμό της βάσης δεδομένων και τη συσχέτιση μεταξύ των πινάκων θα πρέπει να ακολουθηθεί σωστή διασύνδεση των οντοτήτων (connection links), προκειμένου να γίνει σωστός συγχρονισμός, διαμοιρασμός και επεξεργασία δεδομένων.

Στο διάγραμμα ER της εργασίας καθώς και στο σχήμα της βάσης θα πραγματοποιηθούν οι κάτωθι συνδέσεις:

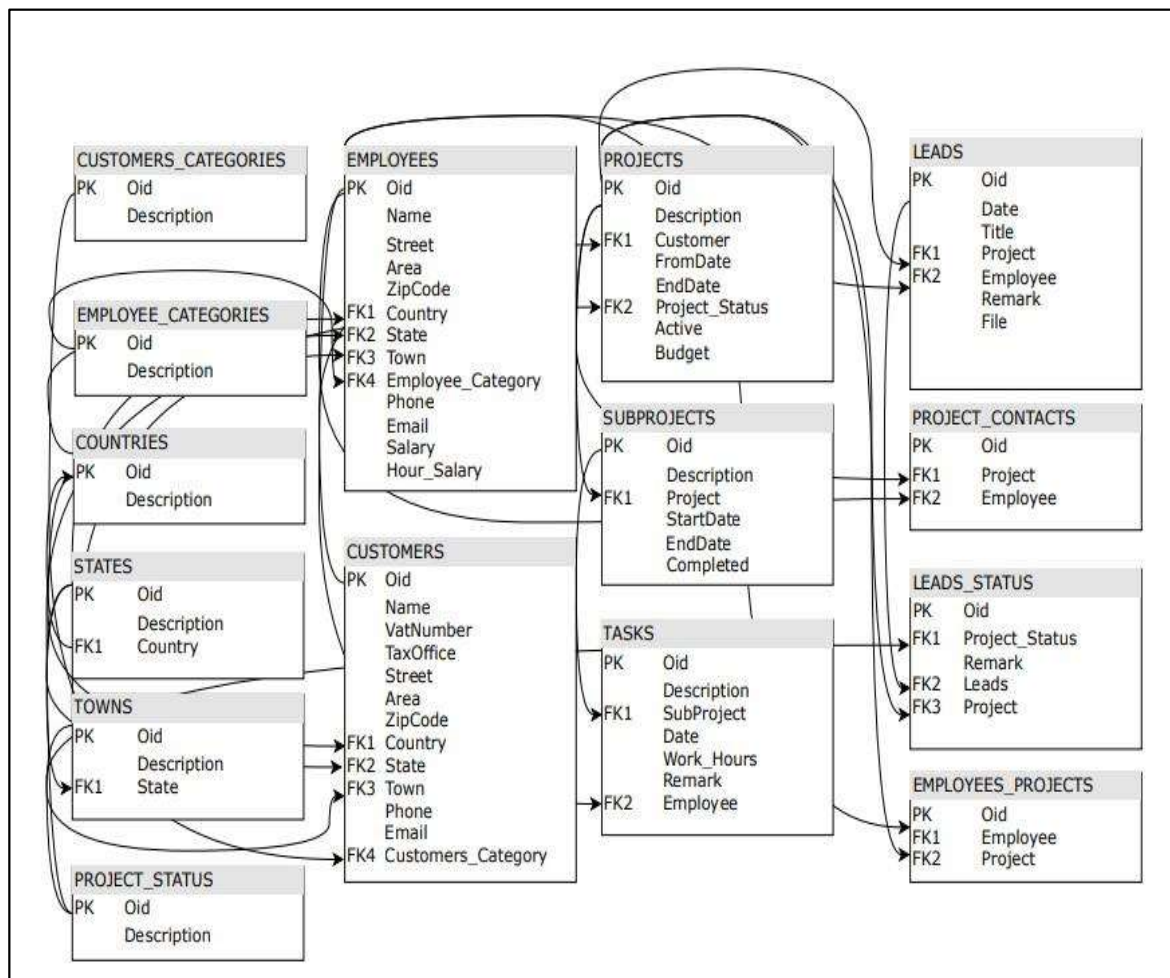
1. Ο πίνακας **CUSTOMERS** συνδέεται με τους πίνακες **CUSTOMERS_CATEGORIES**, **COUNTRIES**, **STATE** και **TOWN** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να γίνει συγχρονισμός όλων των δεδομένων για να εμφανίζονται οι αντίστοιχες πληροφορίες που αφορούν χώρα, πολιτεία, πόλη και κατηγορία που ανήκει ο κάθε πελάτης.
2. Ο πίνακας **EMPLOYESS** συνδέεται με τους πίνακες **EMPLOYEE_CATEGORIES**, **COUNTRIES**, **STATE** και **TOWN** προκειμένου να μπορεί να γίνει συγχρονισμός όλων των δεδομένων για να εμφανίζονται οι αντίστοιχες πληροφορίες που αφορούν χώρα, πολιτεία, πόλη και κατηγορία που ανήκει ο κάθε εργαζόμενος.
3. Ο πίνακας **PROJECTS** συνδέεται με τους πίνακες **PROJECTS_STATUS & CUSTOMER** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να εμφανίζονται και να συγχρονίζονται οι αντίστοιχες πληροφορίες που αφορούν δεδομένα του κάθε έργου, την κατάσταση στην οποία βρίσκεται και τον πελάτη που το έχει αναθέσει.
4. Ο πίνακας **TASKS** συνδέεται με τον πίνακα **SUBPROJECTS** και **EMPLOYEE** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να εμφανίζονται οι αντίστοιχες πληροφορίες που αφορούν δεδομένα του κάθε υποέργου ανά διεργασία σε συνδυασμό με τα στοιχεία του εργοδότη που το έχει αναλάβει.
5. Ο πίνακας **SUBPROJECTS** συνδέεται με τον πίνακα **PROJECT** με τη χρήση ξένου κλειδιού (foreign key), προκειμένου να εμφανίζονται και να συγχρονίζονται όλες οι πληροφορίες που αφορούν δεδομένα του κάθε υποέργου αλλά και του κυρίως έργου στο οποίο ανήκει.

¹ Tables.xls

6. Ο πίνακας **EMPLOYEES_PROJECT** συνδέεται με τον πίνακα **PROJECTS** και **EMPLOYEE** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να συγχρονιστούν όλα τα δεδομένα που αφορούν το κάθε έργο και υπάλληλο του.
7. Ο πίνακας **LEADS** συνδέεται με τον πίνακα **PROJECT** και **EMPLOYEE** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να συγχρονιστούν όλα τα δεδομένα που αφορούν πληροφορίες για το κάθε έργο τον υπάλληλο και περιγραφή της κατάσταση στην οποία βρίσκεται.
8. Ο πίνακας **LEADS_STATUS** συνδέεται με τον πίνακα **PROJECTS_STATUS**, **LEADS** και **PROJECT** με χρήση ξένων κλειδιών (foreign keys), προκειμένου να συγχρονιστούν όλα τα δεδομένα που αφορούν το κάθε έργο και την κατάσταση που βρίσκεται
9. Τέλος θα δημιουργηθούν οι πίνακες **COUNTRIES**, **STATES** και **TOWNS** που θα συγχρονίζονται με χρήση ξένων κλειδιών προκειμένου να γίνει χρήση δεδομένων που αφορούν τη χώρα τη πολιτεία και την πόλη για μετέπειτα χρήση τους και πεδίο εφαρμογής σε άλλους πίνακες που προαναφέρθηκαν.

Στο διάγραμμα² [1] που ακολουθεί παρουσιάζεται το διάγραμμα Entity Relationship της βάσης δεδομένων και οι συνδέσεις μεταξύ των πινάκων.

² DatabaseSchema.pdf



ΔΙΑΓΡΑΜΜΑ 1: Entity Relationship Diagram

3.3 Τύποι δεδομένων.

Σε κάθε πεδίο των πινάκων ορίζεται ο τύπος δεδομένων. Ο τύπος *NVARCHAR2* αφορά τη δυνατότητα αποθήκευσης Unicode χαρακτήρων, με τη δυνατότητα προκαθορισμού του αριθμού των χαρακτήρων, χρησιμοποιώντας όμως περισσότερο χώρο κατά την αποθήκευση. Ο τύπος *INT* αποτελεί συντομογραφία της λέξης *INTEGER* και αφορά την εγγραφή δεδομένων που αποτελούν ακέραιους αριθμούς. Ο τύπος *BLOB* αποτελεί αρκτικόλεξο του Binary Large Object που αναφέρεται σε αρχεία μεγάλου όγκου που βρίσκονται σε δυαδική μορφή και αποθηκεύονται στην βάση δεδομένων. Ο τύπος *DATE* αποτελεί δεδομένα που αφορούν τιμές τύπου ημερολογίου. Ο περιορισμός *NOT NULL* υποχρεώνει τον χρήστη να εισάγει δεδομένα και με αυτό το τρόπο να μη μείνει κενό το πεδίο του πίνακα. Τέλος ο τύπος *DECIMAL* εμφανίζει αριθμούς με τη μορφή δύο δεκαδικών ψηφίων.

4 Υλοποίηση της βάσης με χρήση Oracle – Sql

4.1 Υλοποίηση της βάσης.

Βασιζόμενοι στον σχεδιασμό και την υλοποίηση της βάσης και των πινάκων που σχεδιάστηκαν θα ακολουθηθεί υλοποίηση όλων των πινάκων της βάσης (*CREATE TABLES*) με Oracle - Sql. Προκειμένου να γίνει σωστή δημιουργία του σχήματος θα πρέπει να γίνει η κατασκευή των πινάκων με κατάλληλη σειρά για να πραγματοποιηθεί σωστός συγχρονισμός των primary & foreign keys.

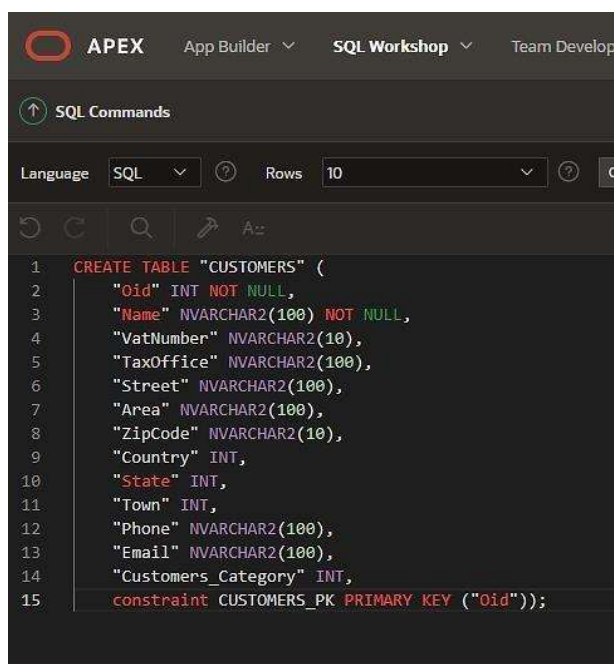
Στην εικόνα 1 που ακολουθεί παρουσιάζεται ο κώδικας Oracle - Sql για την υλοποίηση του πίνακα CUSTOMERS.

Επιπλέον προκειμένου να δημιουργηθεί αυτοματοποιημένη αρίθμηση με μοναδικό ID εγγραφών στον πίνακα CUSTOMERS θα συνταχθεί κομμάτι κώδικα που θα πραγματοποιεί αυτοματοποιημένη αρίθμηση με χρήση της εντολής Create Sequence. Στην συνέχεια θα συνταχθεί Trigger που θα παίρνει σαν όρισμα το ID που δημιουργήθηκε από την διαδικασία Sequence και θα την αναπαράγει με τη χρήση της for για κάθε γραμμή του πίνακα (for each row). Επιπροσθέτως θα γίνει μέριμνα ώστε ο Trigger να ξεκινάει με την εντολή «before insert» προκειμένου να γίνει η διαδικασία προτού της εισαγωγής δεδομένων από τον χρήστη.

Η διασύνδεση του πίνακα CUSTOMERS με τον πίνακα TOWNS θα γίνει με την διαδικασία χρήσης ξένου κλειδιού (foreign key) από τον πίνακα TOWN στον πίνακα CUSTOMERS. Αυτή η διαδικασία θα πραγματοποιηθεί με την εντολή ALTER TABLE για να γίνει παρεμβολή στον επιθυμητό πίνακα και με τη χρήση των εντολών ADD CONSTRAINT, FOREIGN KEY('table'), REFERENCES ('tableID').

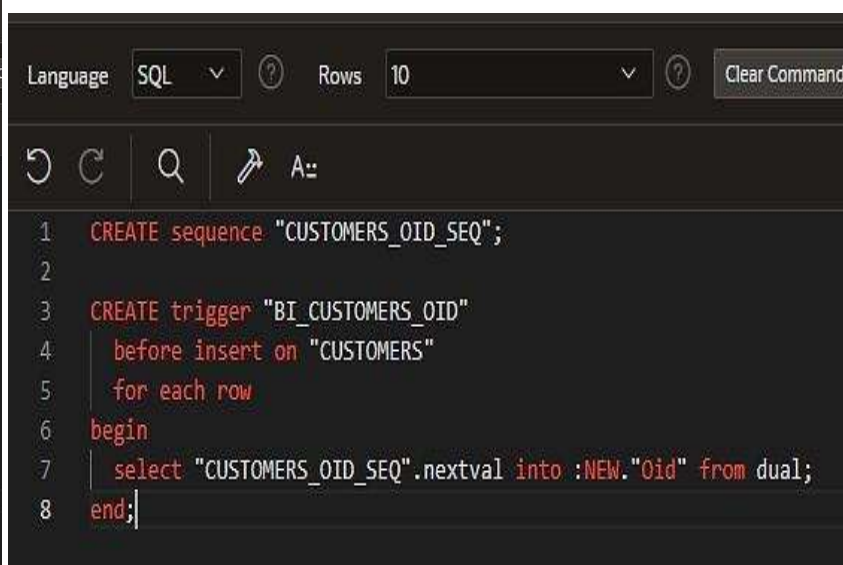
Στην εικόνα 2 που ακολουθεί παρουσιάζεται η εντολή Create Sequence καθώς και ο Trigger για τον πίνακα CUSTOMERS. Στην εικόνα 2-α παρουσιάζεται ο κώδικας για την διασύνδεση των πινάκων CUSTOMERS και TOWNS με τη χρήση foreign key. Η ίδια διαδικασία θα ακολουθηθεί και για τους άλλους πίνακες με την σειρά που απαιτείται.

Στο επισυναπτόμενο αρχείο³, παρουσιάζεται ο κώδικας για όλους τους πίνακες της βάσης που θα δημιουργηθούν.



```
1 CREATE TABLE "CUSTOMERS" (  
2   "oid" INT NOT NULL,  
3   "Name" NVARCHAR2(100) NOT NULL,  
4   "VatNumber" NVARCHAR2(10),  
5   "TaxOffice" NVARCHAR2(100),  
6   "Street" NVARCHAR2(100),  
7   "Area" NVARCHAR2(100),  
8   "ZipCode" NVARCHAR2(10),  
9   "Country" INT,  
10  "State" INT,  
11  "Town" INT,  
12  "Phone" NVARCHAR2(100),  
13  "Email" NVARCHAR2(100),  
14  "Customers_Category" INT,  
15  constraint CUSTOMERS_PK PRIMARY KEY ("oid"));
```

(EIKONA 1)



```
1 CREATE sequence "CUSTOMERS_OID_SEQ";  
2  
3 CREATE trigger "BI_CUSTOMERS_OID"  
4   before insert on "CUSTOMERS"  
5   for each row  
6 begin  
7   select "CUSTOMERS_OID_SEQ".nextval into :NEW."oid" from dual;  
8 end;
```

(EIKONA2)



```
1 ALTER TABLE "CUSTOMERS" ADD CONSTRAINT "CUSTOMERS_fk3" FOREIGN KEY ("Town") REFERENCES "TOWNS"("oid");
```

(EIKONA 2-α)

³ CreateDatabase.txt

4.2 Δημιουργία αυτοματοποιημένων ελέγχων και περιορισμών (Triggers)

Προτού παραδοθεί η βάση δεδομένων για εισαγωγή δεδομένων θα πρέπει να γίνει χρήση περιορισμών προκειμένου να αποφευχθούν λάθη και αστοχία καταχωρήσεων που θα οδηγήσει σε μετέπειτα λάθη που μπορεί να αποβούν χρονοβόρα αλλά και κοστοβόρα για την νεοσύστατη εταιρεία. Ο έλεγχος αυτός μπορεί να γίνει αυτοματοποιημένα με τη χρήση Triggers.

- Αρχικά ζητείται να καταγραφεί ο μισθός των υπαλλήλων που πρέπει να είναι μεταξύ των 800 και 5000 ευρώ. Η λειτουργία αυτή θα γίνει με χρήση συνθηκών if statement για κάθε γραμμή (for each row), προτού της εισαγωγής η ενημέρωσης των δεδομένων από τους χρήστες. Σε περίπτωση λάθους θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100, με την εμφάνιση κατάλληλου μηνύματος.
Στην εικόνα 3 που ακολουθεί παρουσιάζεται ο Trigger BI_EMPLOYEES_CHECK_SALARY που εφαρμόστηκε στον πίνακα EMPLOYEES με χρήση κώδικα Oracle – Sql.

```
Script Name Project_Database_CN7039

1 create or replace NONEDITIONABLE trigger "BI_EMPLOYEES_CHECK_SALARY"
2 before insert or update on "EMPLOYEES"
3 for each row
4 begin
5 IF (:NEW."Salary" is not null and ( :NEW."Salary"<800 or :NEW."Salary">5000)) THEN
6 RAISE_APPLICATION_ERROR(-20100, 'SALARY MUST BETWEEN 800 AND 5000');
7 END IF;
8 end;
```

(ΕΙΚΟΝΑ 3)

- Στην συνέχεια θα δημιουργηθεί Trigger ο οποίος θα ελέγχει τις καταχωρήσεις στους πίνακες Customers, Employees στο πεδίο εισόδου δεδομένων email προτού της εισαγωγής η ενημέρωσης των δεδομένων από τους χρήστες.
Ειδικότερα θα ελέγχει την καταχώρηση από τους χρήστες που αφορά το κενό χαρακτήρα καθώς και την καταχώρηση του χαρακτήρα @. Η λειτουργία αυτή θα υλοποιηθεί με χρήση συνθηκών if statement που θα ελέγχει τη μη κενή καταχώρηση (not null) καθώς και την ύπαρξη του χαρακτήρα @ στο πεδίο email. Σε περίπτωση λάθους θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100, με την εμφάνιση κατάλληλου μηνύματος.
Στις εικόνες 4 και 5 που ακολουθούν παρουσιάζονται οι Triggers BI_EMPLOYEES_CHECK_EMAIL με εφαρμογή στον πίνακα EMPLOYEES και BI_CUSTOMERS_CHECK_EMAIL με εφαρμογή στον πίνακα CUSTOMERS.

The screenshot shows the 'SQL Scripts \ Script Editor' window. The 'Script Name' field contains 'Project_Database_CN7039'. The SQL code is as follows:

```
1 create or replace NONEDITIONABLE trigger "BI_EMPLOYEES_CHECK_EMAIL"
2   before insert or update on "EMPLOYEES"
3   for each row
4   begin
5     IF (:NEW."Email" is not null and :NEW."Email" not like '%@%') THEN
6       RAISE_APPLICATION_ERROR(-20100, 'WRONG EMAIL. EMAIL MUST CONTAIN @ SYMBOL');
7     END IF;
8   end;
```

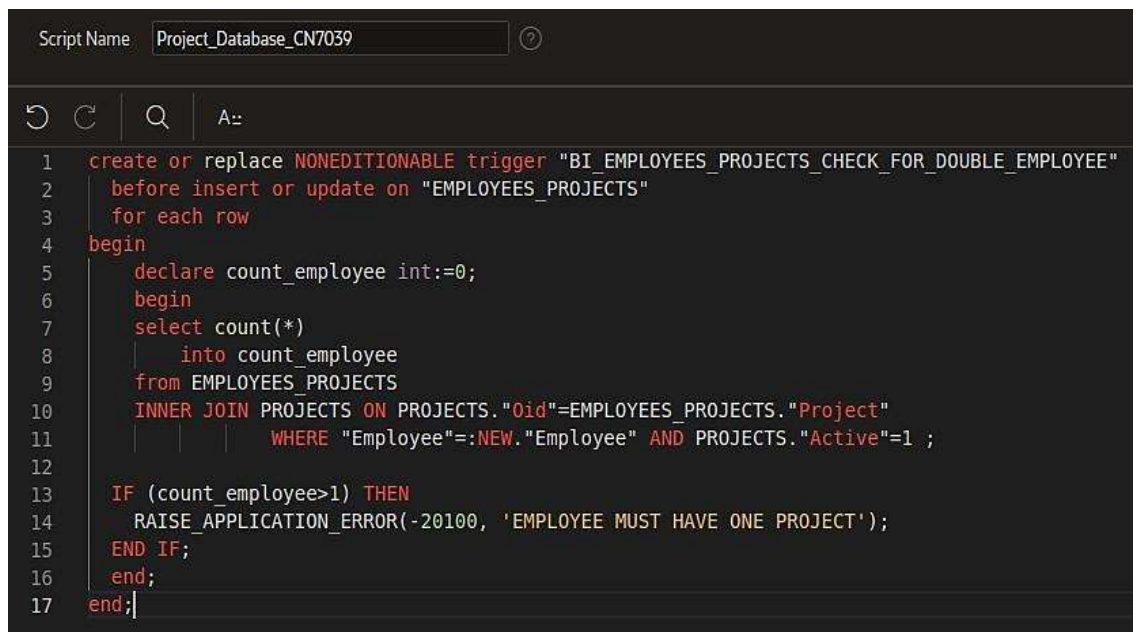
(EIKONA 4)

The screenshot shows the 'SQL Scripts \ Script Editor' window. The 'Script Name' field contains 'Project_Database_CN7039'. The SQL code is as follows:

```
1 create or replace NONEDITIONABLE trigger "BI_CUSTOMERS_CHECK_EMAIL"
2   before insert or update on "CUSTOMERS"
3   for each row
4   begin
5     IF (:NEW."Email" is not null and :NEW."Email" not like '%@%') THEN
6       RAISE_APPLICATION_ERROR(-20100, 'WRONG EMAIL. EMAIL MUST CONTAIN @ SYMBOL');
7     END IF;
8   end;
```

(EIKONA 5)

- Επόμενο ζήτημα θα είναι ο έλεγχος της απασχόλησης των έργων του κάθε υπαλλήλου που δεν θα πρέπει ξεπερνάει τα δύο. Ο έλεγχος θα πραγματοποιείται προτού της εισαγωγής ή ενημέρωσης των δεδομένων από τους χρήστες. Ο περιορισμός θα γίνει με την χρήση των εντολών INNER JOIN αξιοποιώντας και επεξεργάζονται τα κοινά δεδομένα ανάμεσα στους πίνακες PROJECTS & EMPLOYEES_PROJECTS προτού γίνει εισαγωγή ή ενημέρωση στον πίνακα EMPLOYEES PROJECTS. Σε περίπτωση λάθους θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100 με την εμφάνιση κατάλληλου μηνύματος.
- Στην εικόνα 6 που ακολουθεί παρουσιάζεται ο Trigger BI_EMPLOYEES_PROJECTS_CHECK_FOR_DOUBLE με χρήση κώδικα Oracle – Sql.



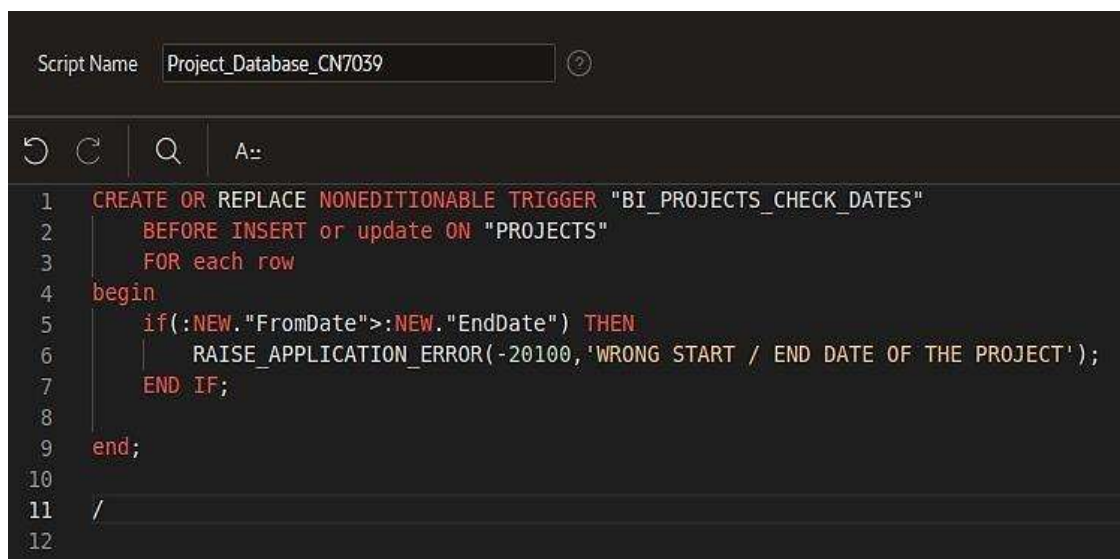
```
Script Name Project_Database_CN7039

1 create or replace NONEDITIONABLE trigger "BI_EMPLOYEES_PROJECTS_CHECK_FOR_DOUBLE_EMPLOYEE"
2   before insert or update on "EMPLOYEES_PROJECTS"
3   for each row
4   begin
5     declare count_employee int:=0;
6     begin
7       select count(*)
8         into count_employee
9       from EMPLOYEES_PROJECTS
10      INNER JOIN PROJECTS ON PROJECTS."Oid"=EMPLOYEES_PROJECTS."Project"
11      WHERE "Employee"=:NEW."Employee" AND PROJECTS."Active"=1 ;
12
13      IF (count_employee>1) THEN
14        RAISE_APPLICATION_ERROR(-20100, 'EMPLOYEE MUST HAVE ONE PROJECT');
15      END IF;
16    end;
17  end;
```

(EIKONA 6)

- Στη συνέχεια θα ελεγχθεί η ημερομηνία έναρξης και λήξης των έργων προτού της εισαγωγής η ενημέρωσης των δεδομένων από τους χρήστες. Η ημερομηνία λήξης δεν θα πρέπει να είναι πριν την ημερομηνία εκκίνησης του έργου. Με την κατάλληλη χρήση συνθηκών if statement θα ελέγχονται τα πεδία των ημερομηνιών του πίνακα PROJECTS. Ο ίδιος ακριβώς έλεγχος με την ίδια μεθοδολογία θα πραγματοποιηθεί και στον πίνακα SUBPROJECTS. Σε περίπτωση λάθους θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100 με την εμφάνιση κατάλληλου μηνύματος.

Στις εικόνες 7 και 8 που ακολουθούν παρουσιάζονται οι Triggers BI_PROJECTS_CHECK_DATES με εφαρμογή στον πίνακα PROJECTS και BI_SUBPROJECT_CHECK_DATE με εφαρμογή στον πίνακα SUBPROJECT.



```
Script Name Project_Database_CN7039

1 CREATE OR REPLACE NONEDITIONABLE TRIGGER "BI_PROJECTS_CHECK_DATES"
2   BEFORE INSERT or update ON "PROJECTS"
3   FOR each row
4   begin
5     if (:NEW."FromDate">:NEW."EndDate") THEN
6       RAISE_APPLICATION_ERROR(-20100, 'WRONG START / END DATE OF THE PROJECT');
7     END IF;
8
9   end;
10
11 /
12
```

(EIKONA 7)

```

1 create or replace noneditionable trigger "BI_SUBPROJECTS_CHECK_DATE"
2   BEFORE INSERT OR UPDATE ON "SUBPROJECTS"
3   FOR EACH ROW
4 BEGIN
5   IF (:NEW."StartDate">:NEW."EndDate") THEN
6     RAISE_APPLICATION_ERROR(-20100,'WRONG START DATE / END DATE OF THE SUBPROJECT');
7   END IF;
8 END;
9
10

```

(EIKONA 8)

- Επόμενος απαραίτητος έλεγχος που θα πραγματοποιηθεί θα είναι εάν κάποιο υποέργο ανήκει στην δικαιοδοσία και επίβλεψη συγκεκριμένου εργαζομένου που του είχε ανατεθεί προτού της εισαγωγής η ενημέρωσης των δεδομένων από τους χρήστες. Η λειτουργία αυτή θα επιτευχθεί με τη χρήση των εντολών INNER JOIN αξιοποιώντας την επεξεργασία που θα προκύψει από τη χρήση κοινών δεδομένων στους πίνακες PROJECTS, SUBPROJECTS και EMPLOYEES_PROJECTS με την αξιοποίηση κατάλληλου υποερωτήματος. Σε περίπτωση λάθους, δηλαδή μη ανάθεσης συγκεκριμένου υποέργου σε εργαζόμενο θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100 με την εμφάνιση κατάλληλου μηνύματος.

Στην εικόνα 9 που ακολουθεί παρουσιάζεται ο Trigger BI_TASKS_CHECK_SUBPROJECT_EMPLOYEE με χρήση κώδικα Oracle – Sql με εφαρμογή στον πίνακα TASKS.

```

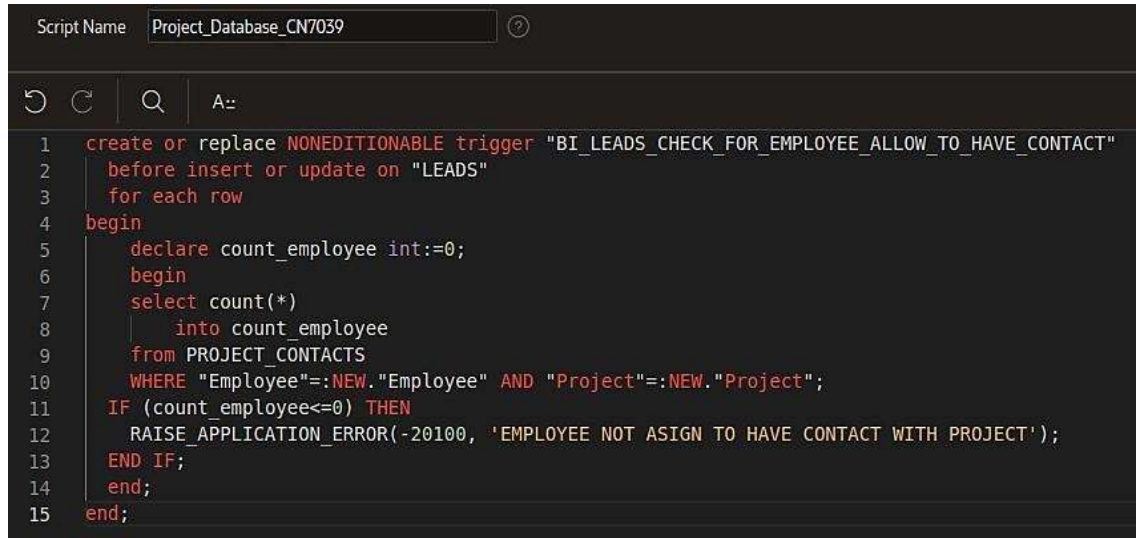
1 CREATE OR REPLACE NONEDITIONABLE TRIGGER "BI_TASKS_CHECK_SUBPROJECT_EMPLOYEE"
2   BEFORE INSERT OR UPDATE ON "TASKS"
3   FOR EACH ROW
4 BEGIN
5   DECLARE COUNT_PROJECTS INT:=0;
6   BEGIN
7     SELECT COUNT(*) INTO COUNT_PROJECTS FROM EMPLOYEES_PROJECTS P
8     INNER JOIN PROJECTS PR ON PR."Oid"=P."Project"
9     INNER JOIN SUBPROJECTS S ON S."Project"=PR."Oid" and s."Oid"=:NEW."SubProject"
10    WHERE P."Employee"=:NEW."Employee" ;
11    if(COUNT_PROJECTS=0) THEN
12      RAISE_APPLICATION_ERROR(-20100,'SUBPROJECT NOT BELONGS TO EMPLOYEE');
13    END IF;
14  END;
15
16 END;

```

(EIKONA 9)

- Ένας άλλος απαραίτητος έλεγχος που θα πραγματοποιηθεί θα είναι η αποκλειστική ανάθεση συγκεκριμένων υπευθύνων για την επικοινωνία με τον εκάστοτε πελάτη του κάθε έργου. Η λειτουργία αυτή θα επιτευχθεί με τον έλεγχο συνθήκης στους πίνακες Employee και Project. Σε περίπτωση λάθους, δηλαδή μη αρμόδιου ατόμου για επικοινωνία θα χρησιμοποιηθεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100 με την εμφάνιση κατάλληλου μηνύματος.

Στην εικόνα 10 που ακολουθεί παρουσιάζεται ο Trigger BI_LEADS_CHECK_FOR_EMPLOYEE_ALLOW_TO_HAVE_CONTACT με χρήση κώδικα Oracle – Sql με πεδίο εφαρμογής τον πίνακα LEADS.



```

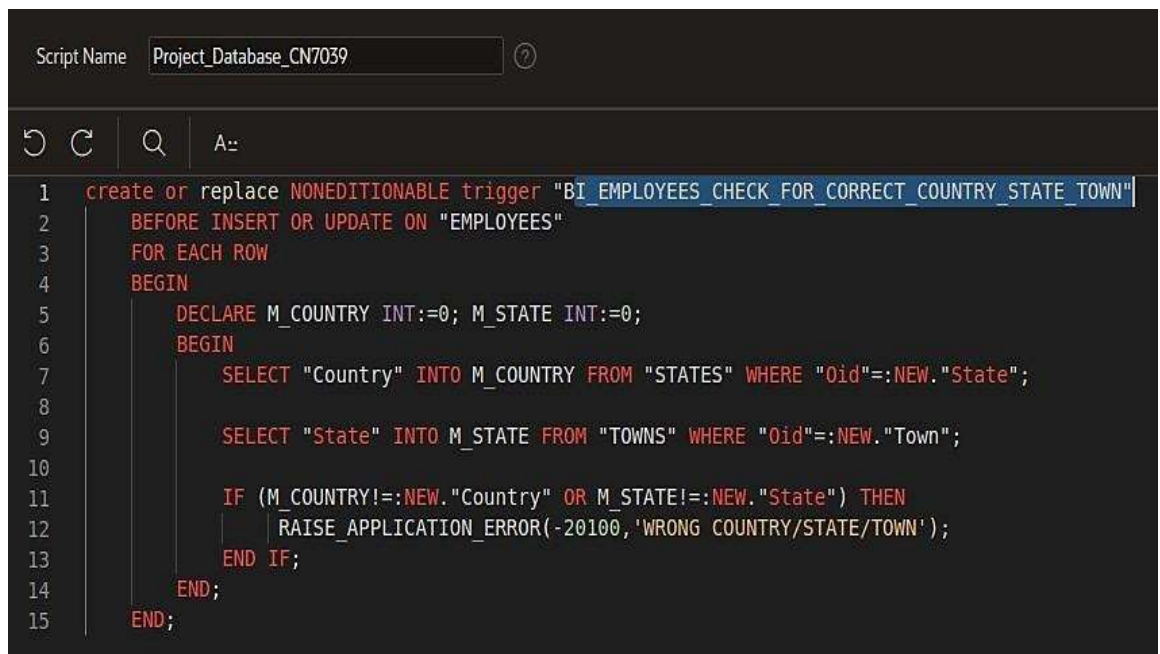
Script Name Project_Database_CN7039

1 create or replace NONEDITIONABLE trigger "BI_LEADS_CHECK_FOR_EMPLOYEE_ALLOW_TO_HAVE_CONTACT"
2   before insert or update on "LEADS"
3   for each row
4   begin
5     declare count_employee int:=0;
6     begin
7       select count(*)
8         into count_employee
9       from PROJECT_CONTACTS
10      WHERE "Employee"=:NEW."Employee" AND "Project"=:NEW."Project";
11      IF (count_employee<=0) THEN
12        RAISE_APPLICATION_ERROR(-20100, 'EMPLOYEE NOT ASIGN TO HAVE CONTACT WITH PROJECT');
13      END IF;
14    end;
15  end;
  
```

(ΕΙΚΟΝΑ 10)

- Τέλος ένας άλλος απαραίτητος έλεγχος που θα πραγματοποιηθεί είναι αυτός των δεδομένων εισόδου στον πίνακα EMPLOYEES σχετικά με την συσχέτιση χώρας, πολιτείας και πόλης του κάθε υπαλλήλου για την ορθή καταχώρηση των δεδομένων. Έτσι θα πραγματοποιηθούν δομές ελέγχου σε δεδομένα μεταξύ των πινάκων Country, State και Towns. Σε περίπτωση λάθους, δηλαδή μη ταιριάσματός των πεδίων των πινάκων θα εφαρμοστεί η λειτουργία RAISE_APPLICATION_ERROR με κωδικό σφάλματος -20100 με την εμφάνιση κατάλληλου μηνύματος. Η ίδια μεθοδολογία εφαρμοστεί και στον πίνακα CUSTOMERS.

Στις εικόνες 11 και 11-α που ακολουθούν παρουσιάζεται οι Triggers BI_EMPLOYEES_CHECK_FOR_CORRECT_COUNTRY_STATE_TOWN & BI_EMPLOYEES_CHECK_FOR_CORRECT_COUNTRY_STATE_TOWN με χρήση κώδικα Oracle – Sql με πεδίο εφαρμογής τους πίνακες EMPLOYEES και CUSTOMERS αντίστοιχα.



```
Script Name Project_Database_CN7039 ?

1 create or replace NONEDITIONABLE trigger "BI_EMPLOYEES_CHECK_FOR_CORRECT_COUNTRY_STATE_TOWN"
2 BEFORE INSERT OR UPDATE ON "EMPLOYEES"
3 FOR EACH ROW
4 BEGIN
5     DECLARE M_COUNTRY INT:=0; M_STATE INT:=0;
6     BEGIN
7         SELECT "Country" INTO M_COUNTRY FROM "STATES" WHERE "Oid"=:NEW."State";
8
9         SELECT "State" INTO M_STATE FROM "TOWNS" WHERE "Oid"=:NEW."Town";
10
11        IF (M_COUNTRY!:=:NEW."Country" OR M_STATE!:=:NEW."State") THEN
12            RAISE_APPLICATION_ERROR(-20100,'WRONG COUNTRY/STATE/TOWN');
13        END IF;
14    END;
15 END;
```

(EIKONA 11)



```
1 create or replace NONEDITIONABLE trigger "BI_CUSTOMERS_CHECK_FOR_CORRECT_COUNTRY_STATE_TOWN"
2 BEFORE INSERT OR UPDATE ON "CUSTOMERS"
3 FOR EACH ROW
4 BEGIN
5     DECLARE M_COUNTRY INT:=0; M_STATE INT:=0;
6     BEGIN
7         SELECT "Country" INTO M_COUNTRY FROM "STATES" WHERE "Oid"=:NEW."State";
8
9         SELECT "State" INTO M_STATE FROM "TOWNS" WHERE "Oid"=:NEW."Town";
10
11        IF (M_COUNTRY!:=:NEW."Country" OR M_STATE!:=:NEW."State") THEN
12            RAISE_APPLICATION_ERROR(-20100,'WRONG COUNTRY/STATE/TOWN');
13        END IF;
14    END;
15 END;
```

(EIKONA 11-a)

Στο αρχείο που επισυνάπτεται ⁴ παρουσιάζεται ο κώδικας Oracle – Sql για όλους τους Triggers που χρησιμοποιήθηκαν.

5. Εισαγωγή δεδομένων

Στο κεφάλαιο που ακολουθεί θα γίνει περιγραφή υποθετικής εισαγωγής δεδομένων στους πίνακες της βάσης του Project.

Αρχικά με την χρήση των εντολών INSERT INTO, VALUES και παρατηρώντας προσεκτικά τον τύπο δεδομένων του κάθε πεδίου θα γίνει χειροκίνητα εισαγωγή υποθετικών δεδομένων στους βασικούς πίνακες της βάσης.

Η εισαγωγή δεδομένων στον πίνακες θα πρέπει να γίνει με συγκεκριμένη δομή και ιεραρχία για τη σωστή καταχώρηση και μετέπειτα επεξεργασία τους. Για την επίτευξη αυτού του σκοπού θα ακολουθηθεί η λογική της αρχικής καταχώρησης δεδομένων στους πίνακες που δεν

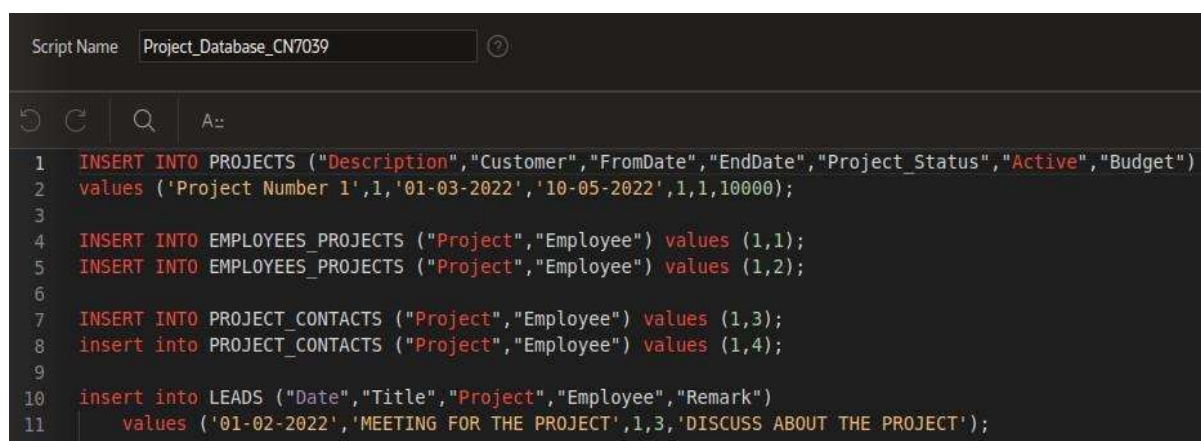
⁴ TRIGGERS.txt

έχουν relate fields και μετέπειτα καταχώρηση τους πίνακες που έχουν relate fields, όπου υπάρχουν διαθέσιμες εγγραφές. Π.χ για να καταχωρήσουμε των πίνακα states που έχει relate field με τον πίνακα country, θα πρέπει πρώτα να καταχωρήσουμε τον πίνακα country και μετά τον πίνακα states.

Στην συνέχεια θα γίνει εισαγωγή δεδομένων για τους πίνακες με πεδίο εφαρμογής το όλο των φάσμα των έργων που χρησιμοποιεί το σύστημα (PROJECTS, EMPLOYEES_PROJECTS, PROJECT_CONTACTS, LEADS, LEADS_STATUS, SUBPROJECTS, TASKS).

Στις εικόνες 12 και 13 που ακολουθούν παρουσιάζεται ενδεικτικά η εισαγωγή δεδομένων για τους πίνακες EMPLOYEES_CATEGORIES και PROJECTS.

Επίσης στα επισυναπτόμενα αρχεία⁵ εμφανίζεται ο κώδικας Oracle – Sql. για το σύνολο της εισαγωγής των υποθετικών δεδομένων

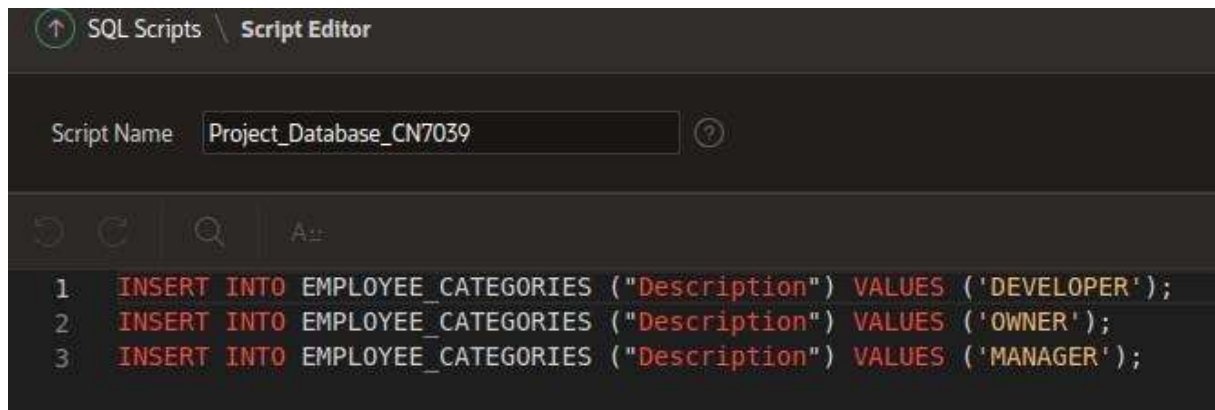


```
Script Name Project_Database_CN7039

1 INSERT INTO PROJECTS ("Description","Customer","FromDate","EndDate","Project_Status","Active","Budget")
2 values ('Project Number 1',1,'01-03-2022','10-05-2022',1,1,10000);
3
4 INSERT INTO EMPLOYEES_PROJECTS ("Project","Employee") values (1,1);
5 INSERT INTO EMPLOYEES_PROJECTS ("Project","Employee") values (1,2);
6
7 INSERT INTO PROJECT_CONTACTS ("Project","Employee") values (1,3);
8 insert into PROJECT_CONTACTS ("Project","Employee") values (1,4);
9
10 insert into LEADS ("Date","Title","Project","Employee","Remark")
11 values ('01-02-2022','MEETING FOR THE PROJECT',1,3,'DISCUSS ABOUT THE PROJECT');
```

(EIKONA 12)

⁵ INSERT_BASE_TABLES.txt , INSERT_PROJECTS.txt



(ΕΙΚΟΝΑ 13)

6. Δημιουργία ειδικών ερωτημάτων - Queries

Η εκτέλεση των Queries πραγματοποιήθηκε στην έκδοση 21.4.3 της Oracle

6.1 Εμφάνιση της λίστας με τα στοιχεία όλων των υπαλλήλων της εταιρείας.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα εμφανίζει τα στοιχεία των υπαλλήλων της εταιρείας με έμφαση ότι οι υπάλληλοι θεωρούνται και ιδιοκτήτες. Αυτό θα πραγματοποιηθεί με χρήση των εντολών LEFT OUTER JOIN, inner join στους πίνακες COUNTRIES, TOWNS και STATES και με χρήση των πεδίων του πίνακα EMPLOYEES. Στην εικόνα 14 ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 14-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων.

Στο επισυναπτόμενο αρχείο6 εμφανίζεται ο κώδικας Oracle – Sql για εκτέλεση του Querye.

```

SELECT
    emp."Oid",
    "Name",
    "Street",
    "Area",
    "ZipCode",
    c."Description" "Country",
    s."Description" "State",
    t."Description" "Town",
    "Phone",
    "Email",
    "Salary",
    ca."Description" category
FROM
    employees emp
    LEFT OUTER JOIN countries c ON c."Oid" = emp."Country"
    LEFT OUTER JOIN states s ON s."Oid" = emp."State"
    LEFT OUTER JOIN towns t ON t."Oid" = emp."Town"
    INNER JOIN employee_categories ca ON ca."Oid" = emp."Employee_Category";
  
```

(ΕΙΚΟΝΑ 14)

⁶ 5_i_query.txt

Oid	Name	Street	Area	ZipCode	Country	State	Town	Phone	Email	Salary	CATEGORY
1	1 TZANIS TASOS	PATISION 10	PAGKRATI	11523	Greece	Attica	Athens	2102926587	TZANIS@G-LOGIC.GR	3000	OWNER
2	2 GEORGIADIS NIKOS KIFISIAS	5 PSICHIKO	13523	Greece	Attica	Kifisia		2106815574	GN@G-LOGIC.GR	2000	MANAGER
3	3 GATES BILL	SCROU 20	MAROUSI	11523	Greece	Attica	Faliro	2105566985	BG@G-LOGIC.GR	1500	DEVELOPER
4	4 ELON MUSK	ALEXANDROU 3	KALLITHEA	11523	Greece	Thessaloniki	Thessaloniki Center	2103366547	EM@G-LOGIC.GR	1800	DEVELOPER

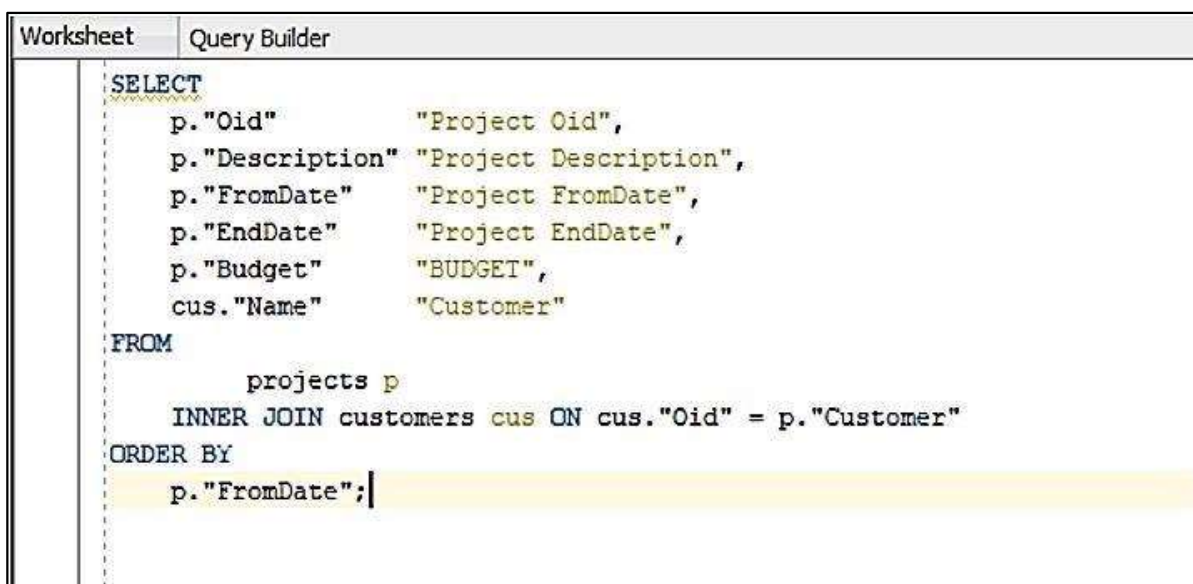
(ΕΙΚΟΝΑ 14-1)

6.2 Εμφάνισης λίστας με τα έργα ταξινομημένα ανά ημερομηνία ανάληψης.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα εμφανίζει τα Project ανά πελάτη ταξινομημένα ανά ημερομηνία. Αυτό θα πραγματοποιηθεί με χρήση των εφαρμογών INNER JOIN, στους πίνακες PROJECTS και CUSTOMERS. Με την χρήση ORDER BY θα γίνει ταξινόμηση δεδομένων ανά ημερομηνία ανάληψης.

Στην εικόνα 15 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql στην εικόνα 15-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων.

Επιπροσθέτως στο επισυναπτόμενο αρχείο⁷ παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.



(EIKONA 15)

	Project Oid	Project Description	Project FromDate	Project EndDate	BUDGET	Customer
1	5	Project Number 4	01/01/22	10/03/22	22000	AMAZON
2	1	Project Number 1	01/03/22	10/05/22	10000	DELTA
3	2	Project Number 2	01/04/22	10/06/22	12000	KOTISOVOLOS A.E.
4	3	Project Number 3	01/05/22	10/05/22	8000	PLAISIO A.E.
5	4	Project Number 4	01/08/22	10/09/22	18000	LIDL

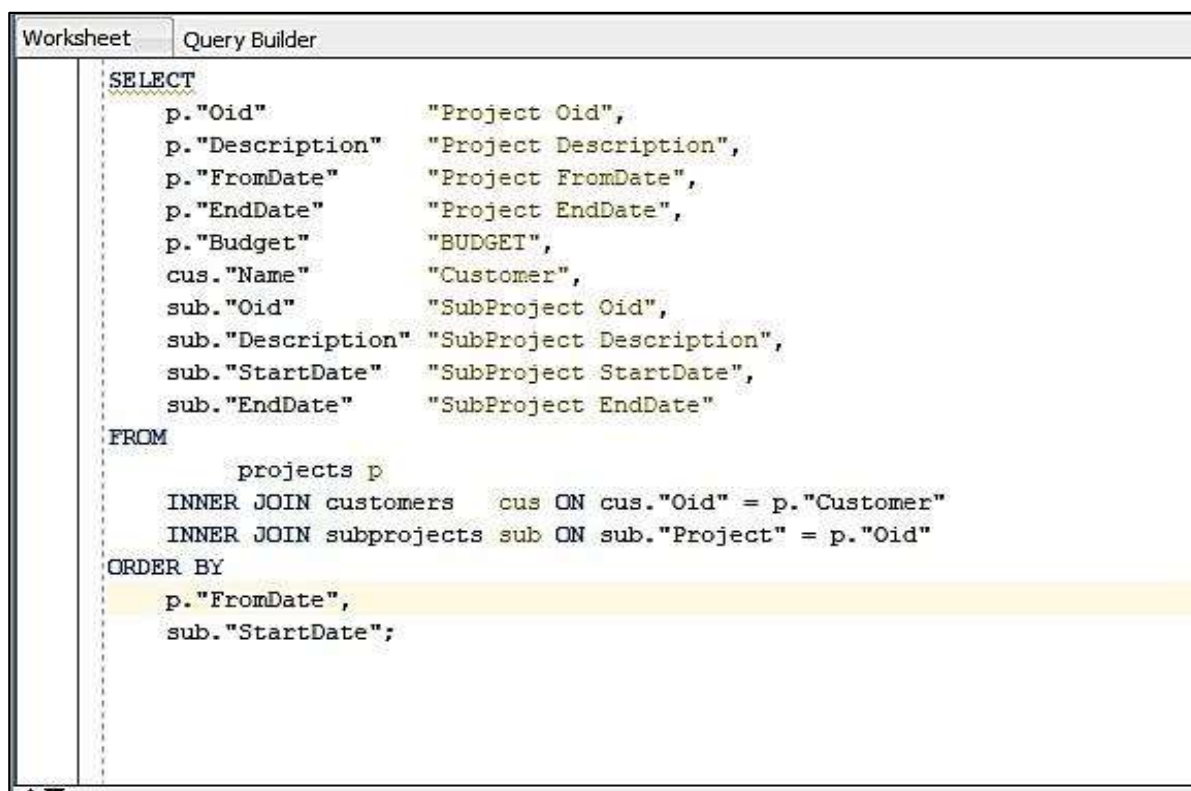
(EIKONA 15-1)

⁷ 5_ii_query.txt

6.3 Εμφάνιση λίστας με έργα – υποέργα – εργασίες ταξινομημένα με ημερομηνία εκκίνησης κάθε φάσης.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα πραγματοποιηθεί με χρήση των εφαρμογών INNER JOIN, στους πίνακες SUBPROJECTS και CUSTOMERS με αξιοποίηση δεδομένων από τον πίνακα PROJECTS. Η ταξινόμηση ανά ημερομηνία εκκίνησης στους πίνακες PROJECTS & SUBPROJECTS θα επιτευχθεί με τη χρήση της εντολής ORDER BY. Στην εικόνα 16 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 16-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων.

Επιπροσθέτως στο επισυναπτόμενο αρχείο⁸ παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.



(ΕΙΚΟΝΑ 16)

Project Oid	Project Description	Project FromDate	Project EndDate	BUDGET	Customer	SubProject Oid	SubProject Description	SubProject StartDate	SubProject EndDate
1	1 Project Number 1	01/03/22	10/05/22	10000 DELTA		1 Sub Project 1-1	01/03/22	15/03/22	
2	1 Project Number 1	01/03/22	10/05/22	10000 DELTA		2 Sub Project 1-2	15/03/22	20/03/22	
3	1 Project Number 1	01/03/22	10/05/22	10000 DELTA		3 Sub Project 1-3	21/03/22	10/04/22	
4	1 Project Number 1	01/03/22	10/05/22	10000 DELTA		4 Sub Project 1-4	11/04/22	20/04/22	
5	1 Project Number 1	01/03/22	10/05/22	10000 DELTA		5 Sub Project 1-5	21/04/22	10/05/22	
6	2 Project Number 2	01/04/22	10/06/22	12000 KOTSOVOLOS A.E.		6 Sub Project 2-1	01/04/22	15/04/22	
7	2 Project Number 2	01/04/22	10/06/22	12000 KOTSOVOLOS A.E.		7 Sub Project 2-2	15/04/22	20/04/22	
8	2 Project Number 2	01/04/22	10/06/22	12000 KOTSOVOLOS A.E.		8 Sub Project 2-3	21/04/22	10/05/22	
9	2 Project Number 2	01/04/22	10/06/22	12000 KOTSOVOLOS A.E.		9 Sub Project 2-4	11/05/22	20/05/22	
10	2 Project Number 2	01/04/22	10/06/22	12000 KOTSOVOLOS A.E.		10 Sub Project 2-5	21/05/22	10/06/22	

(ΕΙΚΟΝΑ 16-1)

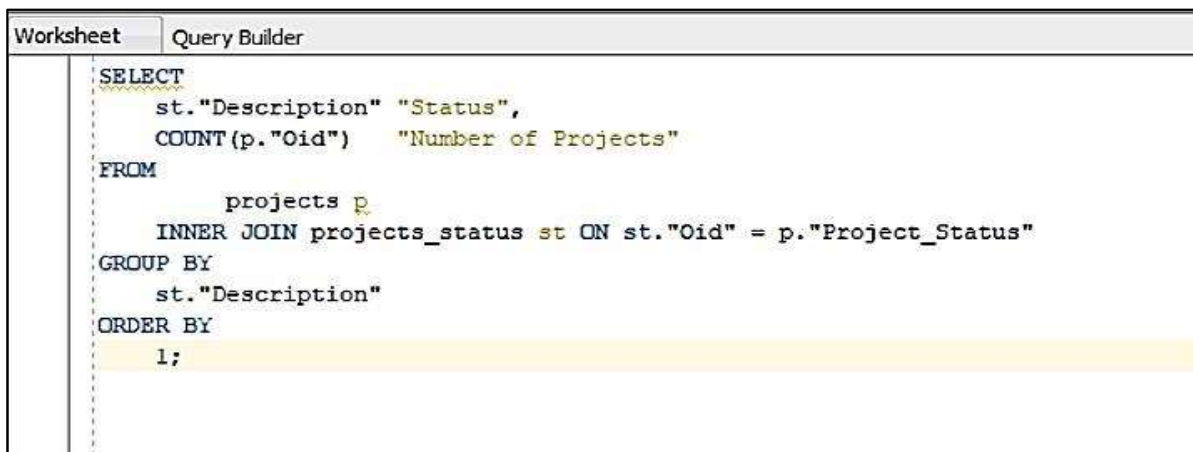
⁸ 5_iii_query.txt

6.4 Εμφάνιση αριθμού έργων ανά κατάσταση, δηλαδή αυτά που απορρίφθηκαν, είναι ενεργά ή ολοκληρωμένα.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα πραγματοποιηθεί με χρήση των εφαρμογών INNER JOIN, στον πίνακα PROJECTS_STATUS και με χρήση δεδομένων από τον πίνακα PROJECTS. Με τη χρήση της εντολής GROUP BY θα γίνει ομαδοποίηση του πεδίου Description του πίνακα PROJECTS_STATUS και με τη χρήση ORDER BY θα γίνει η ταξινόμηση των έργων (PROJECTS).

Στην εικόνα 17 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 17-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων.

Επιπροσθέτως στο επισυναπτόμενο αρχείο⁹ παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.

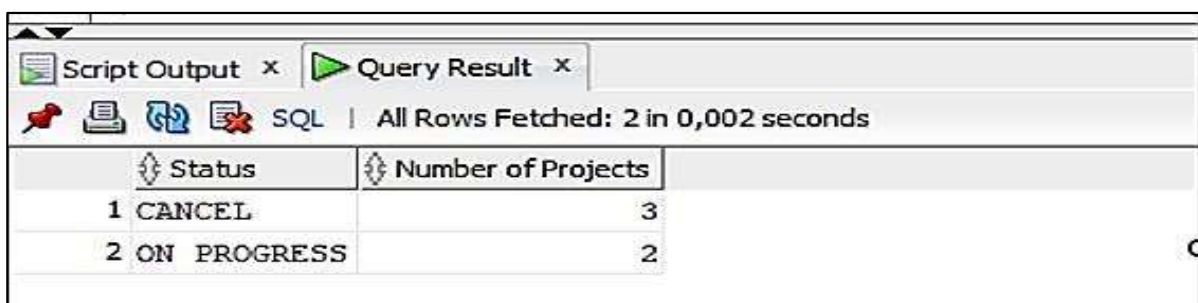


```

SELECT
    st."Description" "Status",
    COUNT(p."Oid") "Number of Projects"
FROM
    projects p
    INNER JOIN projects_status st ON st."Oid" = p."Project_Status"
GROUP BY
    st."Description"
ORDER BY
    1;

```

(ΕΙΚΟΝΑ 17)



Status	Number of Projects
1 CANCEL	3
2 ON PROGRESS	2

(ΕΙΚΟΝΑ 17-1)

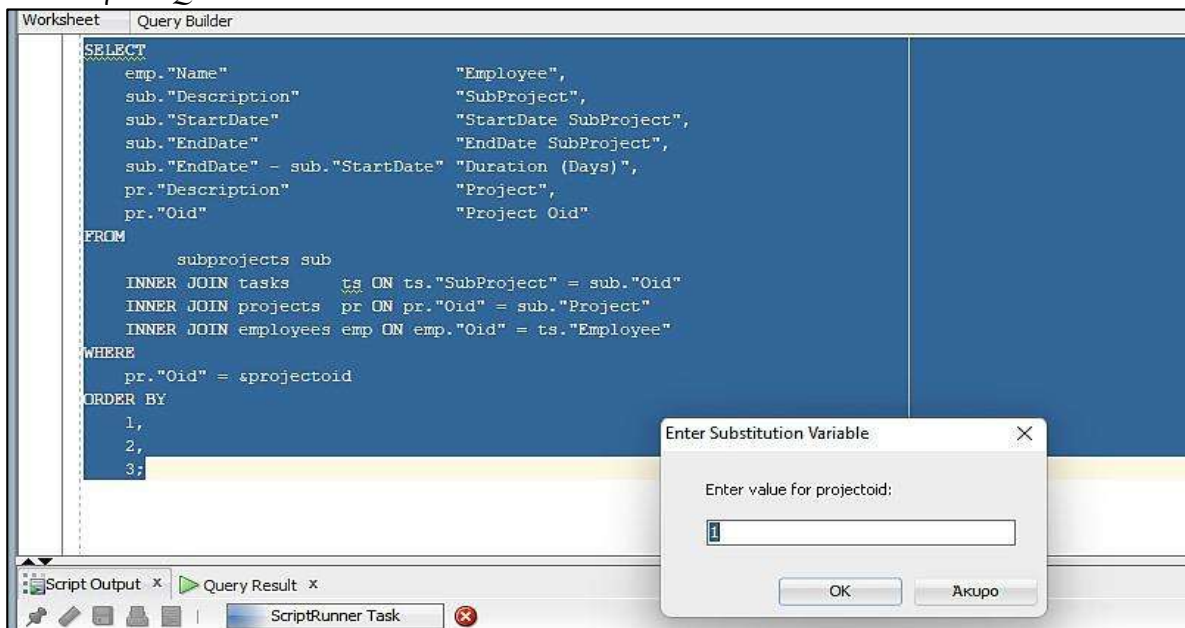
6.5 Εμφάνιση λίστας με τους υπαλλήλους ανά φάση του κάθε έργου με την χρονική διάρκεια αυτής. Απαραίτητη προϋπόθεση θα είναι η εισαγωγή κωδικού έργου.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα πραγματοποιηθεί με χρήση των εφαρμογών INNER JOIN, στους πίνακες TASKS, PROJECTS και EMPLOYEES με χρήση δεδομένων από τον πίνακα SUBPROJECTS. Με τη χρήση της συνθήκης WHERE PR."Oid"=&PROJECTOID θα πραγματοποιείται έλεγχος για την εισαγωγή κωδικού και με την χρήση ORDER BY θα γίνει η ταξινόμηση των τριών πινάκων. Επιπροσθέτως θα γίνει δημιουργία νέων στηλών για την παρουσίαση των επιθυμητών αποτελεσμάτων.

⁹ 5_iv_query.txt

Στην εικόνα 18 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 18-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων καθώς το πεδίο εισαγωγής αναζήτησης κωδικού έργου.

Επιπροσθέτως στο επισυναπτόμενο αρχείο¹⁰ παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.



(ΕΙΚΟΝΑ 18)

The screenshot shows the "Query Result" window in Oracle SQL Developer. It displays the results of the query executed in the previous image. The results are shown in a table with 8 rows and 7 columns. The columns are: Employee, SubProject, StartDate SubProject, EndDate SubProject, Duration (Days), Project, and Project Oid.

	Employee	SubProject	StartDate SubProject	EndDate SubProject	Duration (Days)	Project	Project Oid
1	GEORGIADIS NIKOS	Sub Project 1-2	15/03/22	20/03/22	5	Project Number 1	1
2	GEORGIADIS NIKOS	Sub Project 1-3	21/03/22	10/04/22	20	Project Number 1	1
3	GEORGIADIS NIKOS	Sub Project 1-4	11/04/22	20/04/22	9	Project Number 1	1
4	TZANIS TASOS	Sub Project 1-1	01/03/22	15/03/22	14	Project Number 1	1
5	TZANIS TASOS	Sub Project 1-1	01/03/22	15/03/22	14	Project Number 1	1
6	TZANIS TASOS	Sub Project 1-2	15/03/22	20/03/22	5	Project Number 1	1
7	TZANIS TASOS	Sub Project 1-3	21/03/22	10/04/22	20	Project Number 1	1
8	TZANIS TASOS	Sub Project 1-5	21/04/22	10/05/22	19	Project Number 1	1

(ΕΙΚΟΝΑ 18-1)

6.6 Εμφάνιση λίστας ωρών εργασίας για κάθε έργο με ανάλυση ανά υπάλληλο.

Με την χρήση της Oracle – Sql θα υλοποιηθεί κώδικας που θα πραγματοποιηθεί με χρήση της λειτουργίας INNER JOIN, στους πίνακες TASKS, PROJECTS και EMPLOYEES με χρήση δεδομένων από τον πίνακα SUBPROJECTS. Τα αποτελέσματα των νέων δεδομένων θα εμφανίζονται σε μία νέα στήλη Work Hours. Θα γίνει χρήση της εντολής Group by για ομαδοποίηση των δεδομένων ως προς το πεδίο της περιγραφής του πίνακα PROJECTS και του πεδίου Name του πίνακα EMPLOYEES.

Στην εικόνα 19 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 19-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων.

¹⁰ 5_v_query.txt

Επιπροσθέτως στο επισυναπτόμενο αρχείο¹¹ παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.

```

SELECT
    pr."Description"      "Project",
    emp."Name"            "Employee",
    SUM(ts."Work_Hours")  "Work Hours"
FROM
    subprojects sub
    INNER JOIN tasks    ts ON ts."SubProject" = sub."Oid"
    INNER JOIN projects pr ON pr."Oid" = sub."Project"
    INNER JOIN employees emp ON emp."Oid" = ts."Employee"
GROUP BY
    pr."Description",
    emp."Name"
ORDER BY
    1,
    2;
  
```

(ΕΙΚΟΝΑ 19)

	Project	Employee	Work Hours
1	Project Number 1	GEORGIADIS NIKOS	30
2	Project Number 1	TZANIS TASOS	50
3	Project Number 2	ELON MUSK	40
4	Project Number 2	GATES BILL	40

(ΕΙΚΟΝΑ 19-1) 6.7 Εμφάνιση με οικονομικά

στοιχεία κάθε έργου, δηλαδή προϋπολογισμό που περιλαμβάνει, τελικό κόστος έργου σε ώρες εργασίας σε συνδυασμό με το κόστος κάθε ώρας εργασίας, ποσοστιαία ενασχόληση του υπαλλήλου στο έργο, το τελικό ποσό που αναλογεί στο μισθό του και η ενασχόληση του με το σχετικό έργο. Αρχικά θα γίνει εισαγωγή και επεξεργασία δεδομένων για τα πεδία των πινάκων Subprojects, tasks, employess, projects με τα κατάλληλα αποτελέσματα να εμφανίζονται σε νέες στήλες. Θα γίνει χρήση της λειτουργίας inner join για χρήση κοινών δεδομένων. Επίσης Θα γίνει χρήση της αριθμητικής μεθόδου πρόσθεσης δεδομένων (sum) με ανάλογες πράξεις αριθμητικών πράξεων με τα τελικά δεδομένα και αποτελέσματα να εμφανίζονται σε νέες στήλες .

Στην γραμμή 7 θα υπολογίζονται οι ώρες εργασίας για κάθε εργαζόμενο.

Στην γραμμή 8 θα υπολογίζεται η ποσοστιαία ενασχόληση του κάθε εργαζόμενου σε κάθε έργο με τα αποτελέσματα να εμφανίζονται σε ξεχωριστή στήλη.

Στην γραμμή 9 θα υπολογίζεται το κόστος ωρών εργασίας του κάθε υπαλλήλου ανά έργο.

¹¹ 5_vi_query.txt

Με τις λειτουργίες INNER JOIN στους πίνακες TASKS, PROJECTS, EMPLOYEES, SUBPROJECTS, θα μπορεί να γίνει επεξεργασία των κοινών δεδομένων των πεδίων των πινάκων για τη σωστή εμφάνιση των θεμιτών αποτελεσμάτων.

Στην εικόνα 20 που ακολουθεί εμφανίζεται ο κώδικας του ερωτήματος που υλοποιήθηκε σε Oracle – Sql και στην εικόνα 20-1 τα αποτελέσματα της εκτέλεσης του ερωτήματος με την επιθυμητή παρουσίαση των δεδομένων

Επιπροσθέτως στο επισυναπτόμενο αρχείο¹² παρουσιάζεται ο κώδικας Oracle – Sql για την εκτέλεση του Query.

```

SELECT
    pr."Description"                "Project",
    pr."Budget"                    "Budget",
    subq.workhours                 "Project WorkHours",
    subq.projectcost               "Project Cost",
    emp."Name"                    "Employee",
    SUM(ts."Work_Hours")           "Employee Work Hours",
    SUM(ts."Work_Hours") / subq.workhours * 100 "Work Hours Employee on Project %",
    SUM(ts."Work_Hours" * emp."Hour_Salary") "Employee Work Hours Cost on Project"
FROM
    subprojects sub
    INNER JOIN tasks ts ON ts."SubProject" = sub."Oid"
    INNER JOIN projects pr ON pr."Oid" = sub."Project"
    INNER JOIN employees emp ON emp."Oid" = ts."Employee"
JOIN (
    SELECT
        s."Project"                projectoid,
        SUM(st."Work_Hours")       workhours,
        SUM(st."Work_Hours" * e."Hour_Salary") projectcost
    FROM
        tasks st
        INNER JOIN subprojects s ON s."Oid" = st."SubProject"
        INNER JOIN employees e ON e."Oid" = st."Employee"
    GROUP BY
        s."Project"
) subq ON subq.projectoid = pr."Oid"
GROUP BY
    pr."Description",
    pr."Budget",
    emp."Name",
    subq.workhours,
    subq.projectcost,
    emp."Salary"
ORDER BY
    1,
    5;
  
```

(ΕΙΚΟΝΑ 20)

Project	Budget	Project WorkHours	Project Cost	Employee	Employee Work Hours	Work Hours Employee on Project %	Employee Work Hours Cost on Project
1 Project Number 1	10000	80	1050	GEORGIADIS NIKOS	30	37,5	300
2 Project Number 1	10000	80	1050	TZANIS TASOS	50	62,5	750
3 Project Number 2	12000	80	680	ELON MUSK	40	50	360
4 Project Number 2	12000	80	680	GATES BILL	40	50	320

(ΕΙΚΟΝΑ 20-1)

¹² 5_vii_query.txt

7. Συμπεράσματα - εκτιμήσεις

Κατά τη μελέτη ανάπτυξης της βάσης δεδομένων δίνεται έμφαση στον σχεδιασμό και την υλοποίηση του αρχικού σχήματος της βάσης. Η σωστή οργάνωση της βάσης, η εκτέλεση των σωστών αυτοματοποιημένων μεθόδων και περιορισμών έχει θετικά αποτελέσματα ως προς την εκτέλεση των ερωτημάτων και την παρουσίαση των δεδομένων που απαιτούνται. Αυτή η μεθοδολογία έχει θεμιτά αποτελέσματα για τους χρήστες της βάσης και τις απαιτήσεις του συστήματος. Οι απαντήσεις σε προβληματισμούς που αφορούν θέματα οργάνωσης της νεοσύστατης εταιρείας κατανέμονται με σωστό τρόπο χωρίς να χάνεται πολύτιμος χρόνος πράγμα που συντελεί στην σωστή οργάνωση, διαχείριση και κατ' επέκταση κερδοφορία της εταιρείας. Έτσι με τη διαχείριση της βάσης πραγματοποιείται πλήρης καταγραφή των δεδομένων της εταιρείας, με τους κατάλληλους και ασφαλείς περιορισμούς για αποφυγή λαθών από τους χρήστες. Πραγματοποιείται πλήρης διαχείριση και επίβλεψη των οικονομικών στοιχείων σε όλο το φάσμα καθώς και πλήρης διαχείριση του ανθρώπινου δυναμικού και των πελατών της εταιρείας με το καλύτερο δυνατό αποτέλεσμα.

1. DatabaseSchema.pdf	Σχήμα της βάσης σε διάγραμμα - κεφ. 3.2
2. Tables.xls	Πίνακες της βάσης σε excel – κεφ 3.2
3. CreateDatabase.txt	Υλοποίηση κώδικα για τον δημιουργία της βάσης - κεφ. 4.2
4. TRIGGERS.txt	Αυτόματοι μέθοδοι ελέγχου και εισαγωγής δεδομένων - κεφ. 4.2
5. INSERT_BASE_TABLES.txt	Εισαγωγή δεδομένων – κεφ. 5

8.	6. INSERT_PROJECTS.txt	Εισαγωγή δεδομένων – κεφ. 5
	7. 5_i_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.1

Βιβλιογραφία

Websites – ebooks

1. Oracle PL/SQL Οδηγός Προγραμματισμού Christopher Allen, 09/07/2003.
2. Εγχειρίδιο Προγραμματισμού με PL/SQL, Γκαγκάτσιου Ελένη, 01/11/2012.
3. Oracle Database 10g: A Beginner's Guide, Abbey Michael, Abramson Ian Corey Michael, 15/04/2005.
4. <https://www.infocom.gr/>
5. <https://www.oracle.com/corporate/>

Software – Tools

1. Oracle – Sql έκδοση 21.4.3
2. <https://apex.oracle.com/> (Online Oracle Script Builder)

Πίνακας επισυναπτόμενων αρχείων

8. 5_ii_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.2
9. 5_iii_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.3
10. 5_iv_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.4
11. 5_v_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.5
12. 5_vi_query.txt	Εκτέλεση ερωτήματος – κεφ. 6.6

✚ Γίνεται μνεία ότι τα αρχεία .txt επισυνάπτονται και σε .sql