ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

Χειμερινό Εξάμηνο 2024

ΕΠΛ 342 – Βάσεις Δεδομένων

Ομαδική Εργασία Εξαμήνου - Το Σύστημα EVManager

ΟΜΑΔΑ 10

Ονόματα και Ταυτότητες:

ΒΑΝΙΑ ΜΑΡΟΥΛΗ: UC1060955

ΕΛΕΝΗ ΛΟΙΖΟΥ: UC1058322

ΧΡΙΣΤΟΣ ΘΕΟΔΩΡΟΥ: UC1065855

Οδηγίες χρήσης της εφαρμογής :

1.(index.php)

LOG IN / REGISTER:

- **LOG IN**: Κατά τη διάρκεια της διαδικασίας log in, ο χρήστης πρέπει να συμπληρώσει τα εξής πεδία:
    1. Όνομα χρήστη (username)
    2. Κωδικός πρόσβασης (password)

Σε περίπτωση που ο χρήστης δεν έχει λογαριασμό, μπορεί να κάνει εγγραφή (register).

- **REGISTER**: Κατά τη διάρκεια της διαδικασίας εγγραφής, ο χρήστης πρέπει να συμπληρώσει τα εξής πεδία:
    1. Αναγνωριστικό χρήστη (user_id)
    2. Ρόλος (role) (Simple User, Dealer, TOM_Officer, Admin)
    3. Ημερομηνία γέννησης (birthdate)
    4. Όνομα (first_name)
    5. Επώνυμο (last_name)
    6. Όνομα χρήστη (username)
    7. Κωδικός πρόσβασης (password)

8. Email
9. Διεύθυνση (address)
10. Νομικό/Φυσικό πρόσωπο (nomiko/fisiko)

2.(main.php)

Αν ο χρήστης συνδεθεί ως **Simple User**, έχει 3 επιλογές:

- **View Grants(viewGrants.php)**:
  Εμφανίζονται όλες οι διαθέσιμες κατηγορίες επιχορηγήσεων μαζί με τις περιγραφές τους.
- **View My Applications(viewApplications.php)**:
  Ο χρήστης μπορεί να δει τις αιτήσεις που έχει καταχωρήσει, καθώς και την κατάσταση (status) τους.
- **Make Application (makeApplication.php)**:
  Ο χρήστης μπορεί να καταχωρίσει μία νέα αίτηση, συμπληρώνοντας όλα τα απαραίτητα πεδία.

3.(mainDealer.php)

Αν ο χρήστης συνδεθεί ως **Dealer**, έχει 2 επιλογές:

- **Make Order**:
  - Αρχικά, για να καταχωρήσει την παραγγελία, πρέπει να εντοπίσει την αίτηση χρησιμοποιώντας τα εξής στοιχεία:
    - **Subsidy Category**
    - **Application ID**
    - **User ID**
      Αρχείο: *dealer.php*

  - Αφού εντοπίσει την αίτηση, καταχωρεί τα απαραίτητα πεδία για να ολοκληρώσει την παραγγελία, καθώς και ένα έγγραφο. Μόλις υποβάλει την παραγγελία, αυτή καταχωρίζεται αυτόματα στις παραγγελίες του.
    Αρχείο: *submitOrder.php*

- **View Orders** (*viewOrders.php*):
  - Εμφανίζονται όλες οι παραγγελίες που έχει καταχωρήσει, μαζί με τις σχετικές πληροφορίες.

4.(mainOfficer.php)

Αν ο χρήστης συνδεθεί ως **TOM_Officer** μπορεί να δει όλες τις αιτήσεις που δεν έγιναν αποδεκτές ή απορρίφθηκαν. Δηλαδή τις ελέγχει.

5.(mainAdmin.php)

Αν ο χρήστης συνδεθεί ως **Admin**, έχει 2 επιλογές:

1. **View Applications** (*showApplications.php*):
    - Ο Admin μπορεί να δει όλες τις αιτήσεις (**applications**) μαζί με όλες τις πληροφορίες τους.
    - Έχει επίσης τη δυνατότητα να εγκρίνει (**approve**) ή να απορρίψει (**reject**) μια αίτηση.
2. **Make Report** (*reports.php*):
    - Ο διαχειριστής μπορεί να δημιουργήσει διάφορες αναφορές, χρησιμοποιώντας φίλτρα και ομαδοποιήσεις για να προσαρμόσει τα δεδομένα σύμφωνα με τις ανάγκες του.

Βασικές επιλογές στο σχεδιασμό της βάσης:

Χρησιμοποιήσαμε 5 πίνακες. Συγκεκριμένα, υλοποιήσαμε ένα πίνακα Simple user ο οποίος αποθηκεύει όλα τα στοιχεία που έχει εισάγει ο user κατά το register του. Επιπρόσθετα έχουμε ένα πίνακα Application οπού σε αυτόν υπάρχουν οι αιτήσεις που έχει κάνει ένας user για κάποια χορηγία. Ο πίνακας status περιέχει όλες τις φάσεις που πέρασαν οι αιτήσεις και έτσι προκύπτει το history status. Ακόμα έχουμε τον πίνακα subsidy που περιέχει όλες τις λεπτομέρειες για τις χορηγίες. Στον πίνακα document αποθηκεύονται όλα τα έγγραφα που σχετίζονται με τις αιτήσεις. Τέλος έχουμε ένα πίνακα order vehicle που χρησιμοποιείται για τις παραγγελίες οχημάτων .Οι πίνακες είναι καλά οργανωμένοι ώστε να καλύπτουν διαφορετικές πτυχές της διαδικασίας (αιτήσεις, χρήστες, επιδοτήσεις, έγγραφα). Υπάρχουν σαφείς συνδέσεις μεταξύ των πινάκων μέσω πρωτευόντων και ξένων κλειδιών (user_id, app_id, subsidy_name).Το σύστημα μπορεί εύκολα να επεκταθεί προσθέτοντας νέα είδη επιδοτήσεων ή καταστάσεις

Λειτουργείες που υλοποιήθηκαν:

Η εφαρμογή υλοποιεί ένα σύστημα διπροσωπίας το οποίο επιτρέπει στους χρήστες να συνδέονται με τη βάση δεδομένων και να εκτελούν όλες τις λειτουργίες.

Λειτουργία 1: δημιουργία χρηστών. Για κάθε χρήστη της εφαρμογής απαιτούνται προσωπικά στοιχεία (όνομα, ταυτότητα, ημερομηνία γέννησης, φύλο κ.λπ.) καθώς και

στοιχεία ταυτοποίησης (όνομα χρήστη/κωδικός πρόσβασης)Το σύστημα δημιουργεί τον κωδικό πρόσβασης σαν hashed. Κατηγοριοποιήσουμε τους χρήστες ως Φορέας Υλοποίησης , Λειτουργοί Τμήματος Οδικών Μεταφορών , Αντιπρόσωποι Αυτοκινήτων και Απλοί Χρήστες.

Λειτουργία 2: Διαχείριση κατηγοριών επιχορήγησης: Η λειτουργία αυτή αφορά στη διαχείριση των κατηγοριών επιχορήγησης που προσφέρονται στο σύστημα. Στόχος της είναι να επιτρέπει στους διαχειριστές του συστήματος καταργούν κατηγορίες επιχορήγησης, ώστε να οργανώνονται οι διαθέσιμες επιχορηγήσεις κατά τρόπο σαφή και ευέλικτο.

Λειτουργία 3: Διαχείριση αιτήσεων επιχορήγησης: Η λειτουργία αυτή αφορά στη διαχείριση των αιτήσεων που υποβάλλονται από τους ενδιαφερόμενους για την λήψη επιχορήγησης. Ο στόχος αυτής της λειτουργίας είναι να επιτρέπει στους διαχειριστές να επεξεργάζονται, να αξιολογούν και να παρακολουθούν την πορεία των αιτήσεων, διασφαλίζοντας την ορθή κατανομή των επιχορηγήσεων και την πλήρη παρακολούθηση της διαδικασίας.

Λειτουργία 4: Έγκριση/απόρριψη αιτήσεων: Η λειτουργία αυτή αφορά τη διαδικασία λήψης αποφάσεων σχετικά με τις αιτήσεις επιχορήγησης που έχουν υποβληθεί. Ο στόχος είναι να διασφαλιστεί ότι κάθε αίτηση αξιολογείται με βάση τα κριτήρια της εκάστοτε κατηγορίας επιχορήγησης και ότι οι αποφάσεις λαμβάνονται με διαφάνεια και δικαιοσύνη.

Λειτουργία 5: Παραγωγή αναφορών:Ο διαχειριστής του συστήματος με βάση κάποια φίλτρα μπορεί να παράξει τα αποτελέσματα για κάθε μια απο τις 9 ομαδες αναφορών:

ΑΝΑΦΟΡΕΣ ΕΠΙΧΟΡΗΓΗΣΕΩΝ

1) Επισκόπηση των συνολικών ποσών που δόθηκαν ως επιχορήγηση

PROCEDURE : CalculateTotalDisbursedSubsidyByNames

2) Επισκόπηση των υπολειπόμενων διαθέσιμών ποσών επιχορήγησης

PROCEDURE : CalculateRemainingSubsidyByNames

ΑΝΑΦΟΡΕΣ ΣΤΑΤΙΣΤΙΚΩΝ ΑΙΤΗΣΕΩΝ

3)Ανάλυση του αριθμού αιτήσεων που έγιναν

PROCEDURE : GetApplicationCountBySubsidyFiltered

4)Ποσοστό επιτυχών αιτήσεων:

PROCEDURE : GetApplicationCountPercantageBySubsidyCategory

5)Μέσο ποσό επιχορήγησης επιτυχών αιτήσεων

PROCEDURE: GetApplicationCountPercantageBySubsidyCategoryApproved

ΑΝΑΦΟΡΕΣ ΥΨΟΥΣ ΕΠΙΧΟΡΗΓΗΣΕΩΝ

6)Μέσο ποσό επιχορηγήσεων:

PROCEDURE: GetAverageSubsidyByCategory

7)Υψηλότερες και χαμηλότερες επιχορηγήσεις επιτυχών αιτήσεων:

PROCEDURE: IdentifySubsidyExtremes

ΑΝΑΦΟΡΕΣ ΑΠΟΔΟΣΗΣ

8)Μια αίτηση κάθε μήνα για τους τελευταίους 4 μήνες :

PROCEDURE: SubsidyApplicationsEveryMonthLastFourMonths

9)Αναφορά σε ημερολογιακό έτος για τουλάχιστον x αιτήσεις:

PROCEDURE: GetSubsidyCategoriesByApplicationCountAndYear

Δυσκολίες που αντιμετωπίσαμε κατά την διάρκεια της υλοποίησης:

Βρήκαμε δυσκολία στο να μάθουμε να δουλεύουμε με PHP, HTML, CSS και JavaScript, καθώς και στο πώς να συνδέσουμε τα stored procedures που δημιουργήσαμε στη βάση δεδομένων με την ιστοσελίδα μας. Επίσης βρήκαμε δυσκολία στην κατανόηση του οδηγού.

Δυνατότητες βελτίωσης του συστήματος:

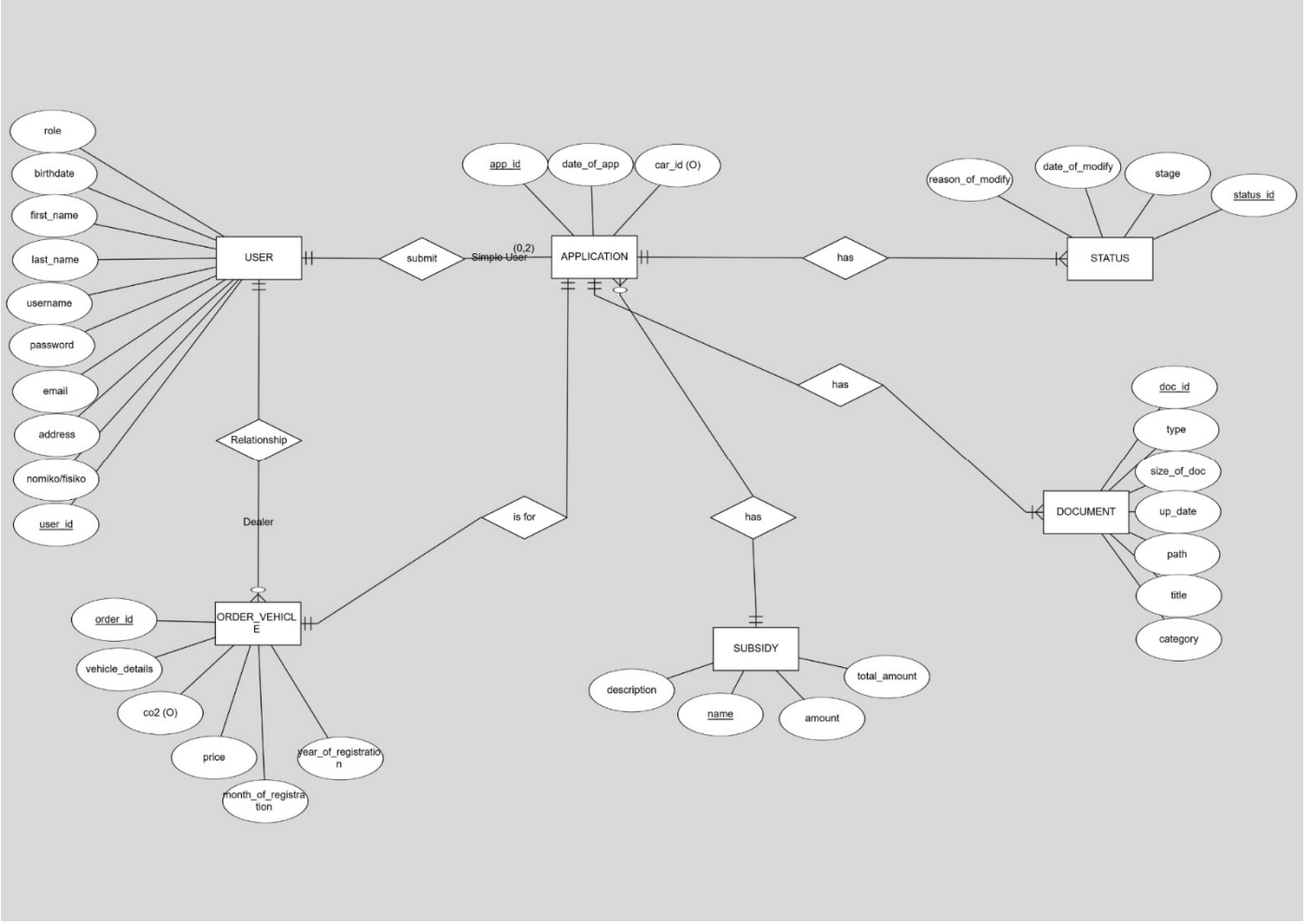
- Χρήση προηγμένης T-Sql

Αντιμετώπιση πιθανών σημείων συνωστισμού:

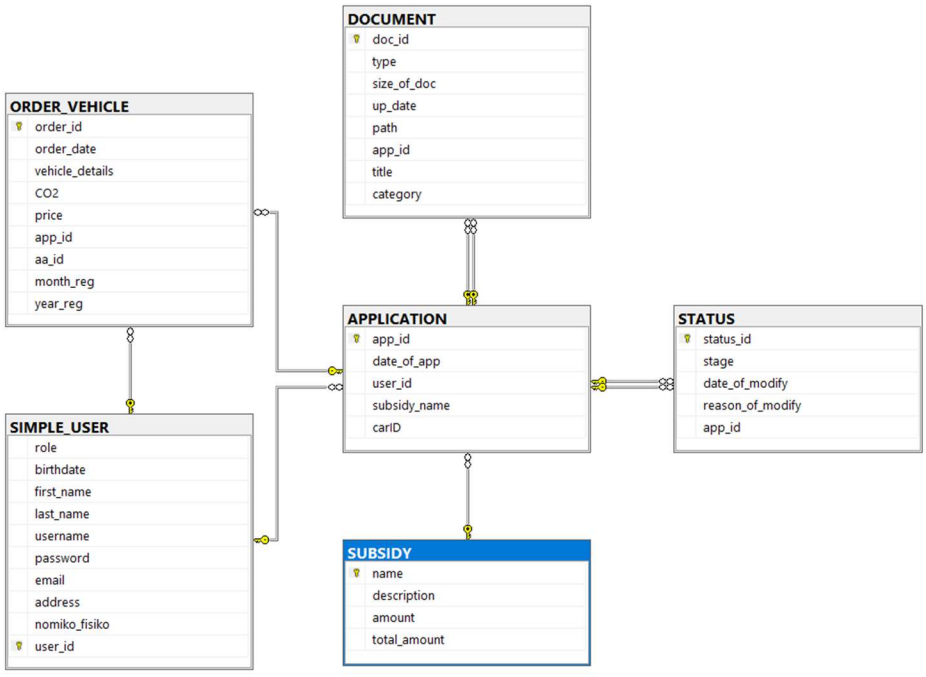
- Χρήση Indexes

DESIGN:

ER-Diagram:



Rational-Schema:



Normalized to 3NF/BCNF?

Πιστεύουμε πως όλοι οι πίνακες είναι σε 3NF και BCNF

General schema implementation:

DEFAULT constraints: Χρησιμοποιούμε στην βάση μας DEFAULT δημιουργία των πινάκων μας για να ορίσουμε αρχικά κάποιες τιμές.

NULL constraints: Χρησιμοποιούμε στην βάση μας NULL για να ορίσουμε ότι μια στήλη δεν μπορεί να πάρει τιμή NULL

DOMAIN constraints: Χρησιμοποιούμε στην βάση μας DOMAIN για την αποθήκευση συμβολοσειρών

IDENTITY usage: Χρησιμοποιούμε στην βάση μας IDENTITY για τον αυτόματο προσδιορισμό μοναδικών τιμών για ένα πεδίο, συνήθως για πεδία πρωτεύοντος κλειδιού

CHECK constraints: Χρησιμοποιούμε στην βάση μας CHECK για να εξασφαλίσουμε ότι οι τιμές που εισάγονται στη βάση δεδομένων πληρούν συγκεκριμένες προϋποθέσεις ή κριτήρια

Check for INDEX: Χρησιμοποιούμε στην βάση μας δείκτες (index) για να βελτιώσει την ταχύτητα των λειτουργιών ανάκτησης δεδομένων στους πίνακες.

Check for CASCADE UPDATES, DELETES: Χρησιμοποιούμε στην βάση μας delete cascade ετσι ωστε εάν διαγραφεί μια αίτηση , όλα τα σχετικά έγγραφα και καταστάσεις θα διαγραφούν επίσης αυτόματα.

Triggers:Χρησιμοποιήσαμε triggers έτσι ώστε αυτόματα όταν γίνετε μια αίτηση να καταγράφετε στον πίνακα status αυτή η αίτηση με stage:active. Επίσης όταν γίνετε approved μια αίτηση αυτόματα να μειώνετε το ολικό ποσό της χορηγίας αυτής.

Advanced TSQL usage (Cursor, SET NO COUNT) : Χρησιμοποιούμε στην βάση μας Set No Count για να βελτιώσουμε την απόδοση του store procedure στην t-sql

Handled SQL injections: Χρησιμοποιούμε στην βάση μας prepared statements και parameterized Queries και store procedures για την αποφυγή των injections

Εισαγωγή Πολλών Δεδομένων:

```
USE vmarou01;
GO

DECLARE @i INT = 0;  -- Start from 0 as per your instruction

WHILE @i < 10000  -- Ensures that exactly 10,000 users are inserted
BEGIN
    INSERT INTO SIMPLE_USER (
        user_id,
        role,               -- The role seems to be missing from your values list
        birthdate,
        first_name,
        last_name,
        username,
        password,
        email,
        address,
        nomiko_fisiko
    )
    VALUES (
        @i,
        'simple user',      -- Correct placement for 'role'
        '10/10/2000',
        CONCAT('FirstName', @i),  -- Unique first name
        CONCAT('LastName', @i),   -- Unique last name
        CONCAT('Userr', @i),       -- Unique username
        '$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9Jvl/1aVPn4vG ',
        CONCAT('user', @i, '@example.com'),  -- Unique email
        CONCAT('Address Line ', @i),  -- Example address
        @i % 2   -- Random boolean for nomiko_fisiko
    );

    SET @i = @i + 1;  -- Increment @i by 1
END;
```

83 %

Results | Messages

	role	birthdate	first_name	last_name	username	password	email	address	nomiko_fisiko	user_id
9984	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9983@example....	Address Line ...	1	9983
9985	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9984@example....	Address Line ...	0	9984
9986	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9985@example....	Address Line ...	1	9985
9987	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9986@example....	Address Line ...	0	9986
9988	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9987@example....	Address Line ...	1	9987
9989	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9988@example....	Address Line ...	0	9988
9990	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9989@example....	Address Line ...	1	9989
9991	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9990@example....	Address Line ...	0	9990
9992	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9991@example....	Address Line ...	1	9991
9993	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9992@example....	Address Line ...	0	9992
9994	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9993@example....	Address Line ...	1	9993
9995	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9994@example....	Address Line ...	0	9994
9996	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9995@example....	Address Line ...	1	9995
9997	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9996@example....	Address Line ...	0	9996
9998	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9997@example....	Address Line ...	1	9997
9999	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9998@example....	Address Line ...	0	9998
10000	simple user	2000-10-10	FirstName...	LastName...	Userr99...	$2y$10$Gf12YLgkcgNMaanscI2ekO6wYSQcwFI02S2nEhbA9...	user9999@example....	Address Line ...	1	9999

Query executed successfully. | mssql.cs.ucy.ac.cy (14.0 RTM) | vmarou01 (85) | vmarou01 | 00:00:00 | 10,000 rows

Data Creation:

Χρησιμοποιήσαμε chat gpt και Python Script για δημιουργία μεγάλου αριθμού δεδομένων

Data management:

-Στην βάση και στο UI κάναμε ολα τα insert και Update

-Τα reports δουλεύουν κανονικά

-Εχουμε ενα ικανοποιητικό Functional UI