2.

SHOW DATABASES;

3.

DROP DATABASE IF EXISTS MY\_ACCOUNTS;

CREATE DATABASE MY\_ACCOUNTS;

USE MY\_ACCOUNTS;

CREATE TABLE ACCOUNTS ( ACCTID INTEGER NOT NULL PRIMARY KEY,

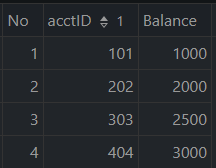
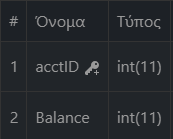
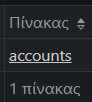
BALANCE INTEGER NOT NULL);

INSERT INTO ACCOUNTS (ACCTID, BALANCE) VALUES (101, 1000);

INSERT INTO ACCOUNTS (ACCTID, BALANCE) VALUES (202, 2000);

INSERT INTO ACCOUNTS (ACCTID, BALANCE) VALUES (303, 2500);

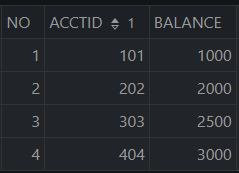
INSERT INTO ACCOUNTS (ACCTID, BALANCE) VALUES (404, 3000);



4.

SET @ROWNUM=0;

SELECT (@ROWNUM:=@ROWNUM+1) AS NO, ACCTID, BALANCE FROM ACCOUNTS ORDER BY ACCTID;



5.

Όχι, η αύξουσα αρίθμηση που εμφανίζεται στο σύνολο αποτελεσμάτων δεν θα υπάρχει στον ίδιο τον πίνακα Accounts, αφού η αρίθμηση υπολογίζεται και εμφανίζεται μόνο κατά τη στιγμή της αναζήτησης.

6.

CREATE TABLE CUSTOMERS (

CUSTNO INTEGER NOT NULL PRIMARY KEY,

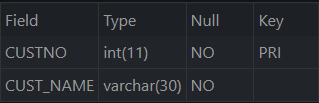
CUST\_NAME VARCHAR(30) NOT NULL);

INSERT INTO CUSTOMERS (CUSTNO, CUST\_NAME) VALUES (10, '101');

INSERT INTO CUSTOMERS (CUSTNO, CUST\_NAME) VALUES (20, '202');

SELECT \* FROM CUSTOMERS;

DESCRIBE CUSTOMERS;



7.

ALTER TABLE ACCOUNTS

ADD COLUMN CUSTNO INTEGER,

ADD CONSTRAINT FK\_CUSTNO

FOREIGN KEY (CUSTNO)

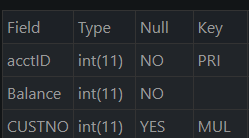
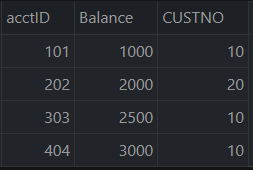
REFERENCES CUSTOMERS(CUSTNO);

UPDATE ACCOUNTS SET CUSTNO = 20 WHERE ACCTID = 202;

UPDATE ACCOUNTS SET CUSTNO = 10 WHERE ACCTID <> 202;

SELECT \* FROM ACCOUNTS;

DESCRIBE ACCOUNTS;



8.

SELECT CUSTNO, COUNT(\*), SUM(BALANCE)

FROM ACCOUNTS

WHERE CUSTNO NOT IN (20)

GROUP BY CUSTNO;

//ΠΑΡΑΛΛΑΓΉ ΜΕ ΧΡΉΣΗ ΜΕΤΑΒΛΗΤΉΣ

SET @CUST\_NO=20;

SELECT CUSTNO, COUNT(\*), SUM(BALANCE)

FROM ACCOUNTS

WHERE CUSTNO NOT IN (@CUST\_NO)

GROUP BY CUSTNO;

Για αρχή, υπολογίζει τον αριθμό των γραμμών για κάθε CUSTNO και το άθροισμα του balance όπου το CUSTNO δεν είναι 20 και το ομαδοποιεί κατά CUSTNO.

Στην συνέχεια δημιουργεί την μεταβλητή CUST\_NO και την αρχικοποιεί με 20. Μετά κάνει ότι και πριν απλώς αντί για 20 χρησιμοποιεί την μεταβλητή.

9.

SELECT COUNT(\*), SUM(BALANCE) FROM ACCOUNTS;

//ΠΑΡΑΛΛΑΓΉ ΜΕ ΧΡΉΣΗ ΜΕΤΑΒΛΗΤΏΝ

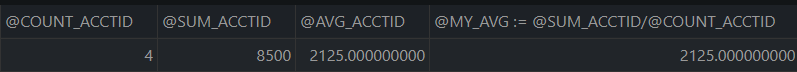
SET @COUNT\_ACCTID=0, @SUM\_ACCTID=0, @AVG\_ACCTID=0;

SELECT COUNT(\*), SUM(BALANCE), AVG(BALANCE)

INTO @COUNT\_ACCTID, @SUM\_ACCTID, @AVG\_ACCTID

FROM ACCOUNTS;

SELECT @COUNT\_ACCTID, @SUM\_ACCTID, @AVG\_ACCTID, @MY\_AVG := @SUM\_ACCTID/@COUNT\_ACCTID;



Για αρχή, υπολογίζει το COUNT(\*) και το συνολικό άθροισμα του BALANCE για όλες τις γραμμές του ACCOUNTS.

Στην συνέχεια δημιουργεί τις μεταβλητές COUNT\_ACCTID, SUM\_ACCTID, AVG\_ACCTID

και τις αρχικοποιεί με 0. Κάνει τους υπολογισμούς και βάζει το αποτέλεσμα στις αντίστοιχες μεταβλητές. Τις εμφανίζει και βρίσκει και εμφανίζει το μέσο όρο.

10.

DROP FUNCTION IF EXISTS FACTORIAL;

DELIMITER !

CREATE FUNCTION FACTORIAL(N INT)

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE F INT DEFAULT 1;

WHILE N > 0 DO

SET F = N \* F;

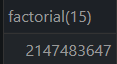
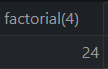
SET N = N - 1;

END WHILE;

RETURN F;

END !

DELIMITER ;



11.

DROP PROCEDURE MY\_PROCEDURE\_LOCAL\_VARIABLES;

DELIMITER $$

CREATE PROCEDURE MY\_PROCEDURE\_LOCAL\_VARIABLES()

BEGIN

SET @X = 25;

SET @Y = 10;

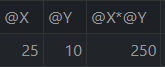
SELECT @X, @Y, @X\*@Y;

END $$

DELIMITER ;

// ΚΛΉΣΗ ΤΗΣ ΔΙΑΔΙΚΑΣΊΑΣ

CALL MY\_PROCEDURE\_LOCAL\_VARIABLES();



12.

// ΔΟΚΙΜΈΣ ΧΡΉΣΗΣ ΤΗΣ ΣΥΝΆΡΤΗΣΗΣ MOD

SET @P\_NO=3;

SELECT MOD(@P\_NO,2);

SET @P\_NO=8;

SELECT MOD(@P\_NO,2);

// ΔΗΜΙΟΥΡΓΊΑ ΒΆΣΗΣ ΚΑΙ ΠΊΝΑΚΑ

DROP TABLE IF EXISTS MYTRACE;

CREATE TABLE MYTRACE ( T\_NO INT,

T\_USER CHAR(20),

T\_DATE DATE,

T\_TIME TIME,

T\_PROC VARCHAR(16), T\_WHAT VARCHAR(30));

// ΔΗΜΙΟΥΡΓΊΑ ΑΠΟΘΗΚΕΥΜΈΝΗΣ ΔΙΑΔΙΚΑΣΊΑΣ MYPROC

DROP PROCEDURE IF EXISTS MYPROC;

DELIMITER !

CREATE PROCEDURE MYPROC (IN P\_NO INT, IN P\_IN VARCHAR(30),

OUT P\_OUT VARCHAR(30))

LANGUAGE SQL

BEGIN

SET P\_OUT=P\_IN;

INSERT INTO MYTRACE (T\_NO, T\_USER, T\_DATE, T\_TIME, T\_PROC, T\_WHAT)

VALUES (P\_NO, CURRENT\_USER, CURRENT\_DATE, CURRENT\_TIME, 'MYPROC', P\_IN);

IF (MOD(P\_NO,2)=0) THEN

COMMIT;

ELSE ROLLBACK;

END IF;

END !

DELIMITER ;

// ΚΛΉΣΗ ΤΗΣ ΔΙΑΔΙΚΑΣΊΑΣ

SET AUTOCOMMIT=0;

CALL MYPROC(1, 'HELLO1', @OUT);

CALL MYPROC(2, 'HELLO2', @OUT);

CALL MYPROC(3, 'HELLO3', @OUT);

CALL MYPROC(4, 'HELLO4', @OUT);

CALL MYPROC(5, 'HELLO5', @OUT);

CALL MYPROC(6, 'HELLO6', @OUT);

CALL MYPROC(7, 'HELLO7', @OUT);

SELECT \* FROM MYTRACE;

