מטלה 2 – מסדי נתונים

חלק א'-

הערה: אנחנו לא מצליחת לסדר את הקוד שהסוגריים יהיו מסודרים כמו שצריך. מצרפות לתיקיית הזיפ גם את קבצי ה-SQL שיצרנו.

**שאלה 1:**

יצרנו Stored Procedure שמבצע את הפעולה הנדרשת. (כמובן שניתן היה גם לכתוב את ה-SELECT עצמו, עשינו כך למען נוכחות הכתיבה)

DELIMITER $$

CREATE PROCEDURE `my\_queue`

(in idDoc int)

BEGIN

SELECT a.Patient\_id ,p.Patient\_name, Appointment\_time from appointment as a join patients as p

where a.Patient\_id=p.Patient\_id and doctor\_id=idDoc and Appointment\_time>now()

group by doctor\_id ,appointment\_time ;

END $$

DELIMITER ;

והפונקציה בג'אווה. עם הפעלת הפונקציה המשתמש מתבקש להזין ת"ז של הרופא ותתקבל טבלה של כל המטופלים שעוד צריכים להיכנס לרופא מהתאריך הנ"ל-

**public** **static** **void** doctor\_queue(){

**try**{

Class.*forName*("com.mysql.jdbc.Driver");

**try**(Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/macabi", "root", "shsz1997")){

Statement stmt = con.createStatement();

Scanner myObj = **new** Scanner(System.***in***);

String id\_doctor;

System.***out***.println("Enter id\_doctor");

id\_doctor = myObj.nextLine();

System.***out***.println("Your Patients queue is below");

System.***out***.println("-------------------------------------------------");

ResultSet rs = stmt.executeQuery("call my\_queue("+id\_doctor+");");

**int** numOfColumns = rs.getMetaData().getColumnCount();

**while** (rs.next()){

**for** (**int** col = 1; col <= numOfColumns; col++){

System.***out***.print(rs.getString(col) + " ");

}

System.***out***.println();

}

System.***out***.println("-------------------------------------------------");

}} **catch** (Exception ex){ex.printStackTrace();}

}

**שאלה 2:**

יצרנו Stored Procedure שמבצע את הפעולה הנדרשת.

DELIMITER $$

CREATE PROCEDURE `actual\_time`

(in idPatient int)

BEGIN

select @numAp := Appointment\_id from appointment where Patient\_id in(

select Patient\_id from appointment where Patient\_id=idPatient);

update queue set actual\_time=now() where Appointment\_id = @numAp;

END $$

DELIMITER ;

והפונקציה בג'אווה- בתוך הפונקציה הרופא מתבקש להכניס ת"ז של המטופל והיא מעדכנת באופן אוטומטי את זמן הכניסה הנוכחי של המטופל בטבלת QUEUE.

**public** **static** **void** update\_insert\_time\_of\_patsient(){

**try**{

Class.*forName*("com.mysql.jdbc.Driver");

**try**(Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/macabi", "root", "shsz1997")){

Statement stmt = con.createStatement();

Scanner myObj = **new** Scanner(System.***in***);

String id\_patsient;

System.***out***.println("Enter id\_patsient");

id\_patsient = myObj.nextLine();

ResultSet rs = stmt.executeQuery("call actual\_time("+id\_patsient+");");

}} **catch** (Exception ex){ex.printStackTrace();}

}

**שאלה 3:**

יצרנו VIEW המציג את 10 האנשים שחיכו הכי הרבה זמן בתור.

CREATE VIEW max\_waiting AS

select Patient\_id, TIMEDIFF(q.actual\_time,a.Appointment\_time) as Waiting\_Time

from appointment as a join queue as q

on a.Appointment\_id = q.Appointment\_id

order by Waiting\_Time desc limit 10;

**שאלה 4:**

יצרנו שני טריגרים – אחד בשביל ההכנסה ואחד בשביל המחיקה שמעדכן את הטבלה queue\_summary בהתאם.

DELIMITER $$

CREATE TRIGGER new\_patient\_received

AFTER INSERT ON queue

FOR EACH ROW

BEGIN

update queue\_summary set num\_of\_Patient=num\_of\_Patient+1

where

(doctor\_id = (select doctor\_id from appointment where Appointment\_id=new.Appointment\_id))

and

(cast(date\_ as date) = (select cast(Appointment\_time as date) from appointment where Appointment\_id=new.Appointment\_id));

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER patient\_delete

AFTER DELETE ON queue

FOR EACH ROW

BEGIN

update queue\_summary set num\_of\_Patient=num\_of\_Patient-1

where

(doctor\_id = (select doctor\_id from appointment where Appointment\_id = OLD.Appointment\_id))

and

(cast(date\_ as date) = (select cast(Appointment\_time as date) from appointment where Appointment\_id = OLD.Appointment\_id));

END$$

DELIMITER ;

חלק ב'-

**שאלה 5:**

select doctor\_name, salary

from doctors inner join queue\_summary

where (cast(date\_ as date)='2020-04-20' and num\_of\_Patient>4);

השאילתה מחזירה טבלה של שמות הרופאים והמשכורות שלהם בתנאי שביום 20-04-2020 היה להם יותר מ-4 מטופלים.

**שאלה 6:**

candidate keys- {B,F}, {B,C,E}, {B,D,C}

רמת הנרמול היא 3NF.

נוכיח: ה-NONPRIME היחיד שנותר הוא A.

התלות הנתונה עבורו היא {B, F, E}→A.

בשביל :2NF נרצה לבדוק ש-A לא תלוי בקבוצה חלקית של ה-C.K ואכן התלות היחידה שקיימת היא עם קבוצה שמהווה SUPERKEY.

בשביל 3NF: נרצה לבדוק ש-A לא תלוי בקבוצה שהיא לא SUPERKEY ואכן הוא תלוי בקבוצה שמהווה SUPERKEY.

בשביל 3.5NF: נרצה לבדוק עבור כל התכונות הקיימות שיהיו תלויות רק בקבוצות מסוג-SUPERKEY אבל נתון לנו כי E→D כאשר D תלוי בקבוצה שאינה מהווה SUPERKEY וזה מפר את רמה זו.

לכן רמת הנרמול הגבוה ביותר היא 3NF .