



הנחיות להגשת התרגיל:

1. הגשה בזוגות בלבד
2. יש להגיש קובץ WORD ובו התשובות לתרגיל. על שם הקובץ להכיל את תעודות הזהות של המגישים
בפורמט הבא : HW3_firstID_secondID
3. העמוד הראשון של ההגשה צריך להכיל את שמות ותעודות הזהות של המגישים
4. הפתרון חייב להיות מוקלד באמצעות מעבד תמלילים WORD. פתרון בכתב יד יגרור קנס בגובה 20% מהציון.
5. קובץ ה-WORD אמור לכלול עבור כל סעיף את מספר הסעיף, פתרון ה-SQL המלא בטקסט (לא צילום מסך) ובנוסף צילום מסך של תוצאת הרצת השאילתה מ-SQL-Server.
6. את הקובץ יש להגיש דרך אתר הקורס במקום המתאים ב-moodle על ידי **אחד** מבני הזוג
7. איחור בהגשת התרגיל יגרור קנס בגובה 10% מהציון עבור כל יום איחור (פרט למקרים חריגים כגון מילואים. במקרים אלה יש לפנות טרם הגשת התרגיל)
8. נפתח פורום מיוחד עבור תרגיל בית זה. כל שאלה או בקשת הבהרה יש לפרסם בפורום, ואנו נענה בהקדם

תיאור הבעיה:

בהמשך לשאלה 2 מתרגיל בית 2, על מסעדת "שף אקספרס", להלן הטבלאות הקיימות במסד הנתונים של החברה (סקריפט ליצירת טבלאות אלו, הכולל גם טעינת רשומות אליהם פורסם באתר):

```
CREATE TABLE Employee (  
    EmployeeNum INT PRIMARY KEY,  
    Name VARCHAR(20),  
    Birthday DATE,  
    Salary INT  
);
```

טבלת עובדים

```
CREATE TABLE Manager (  
    EmployeeNum INT PRIMARY KEY,  
    Complaints INT,  
    FOREIGN KEY (EmployeeNum)  
        REFERENCES Employee (EmployeeNum)  
);
```

טבלת מנהלי משמרות

```
CREATE TABLE Chef (  
    EmployeeNum INT PRIMARY KEY,  
    Country VARCHAR(20),  
    FOREIGN KEY (EmployeeNum)  
        REFERENCES Employee (EmployeeNum)  
);
```

טבלת שפים



```
CREATE TABLE Cook (  
    EmployeeNum INT PRIMARY KEY,  
    Seniority INT,  
    FOREIGN KEY (EmployeeNum)  
        REFERENCES Employee (EmployeeNum)  
);
```

טבלת טבחים

```
CREATE TABLE Cashier (  
    EmployeeNum INT PRIMARY KEY,  
    Gender VARCHAR(1),  
    FOREIGN KEY (EmployeeNum)  
        REFERENCES Employee (EmployeeNum)  
);
```

טבלת קופאים

```
CREATE TABLE Shift (  
    ShiftNum INT PRIMARY KEY,  
    ShiftDesc VARCHAR(100),  
    NumOfCashiers INT NOT NULL,  
    NumOfCooks INT NOT NULL  
);
```

טבלת סוגי משמרות

```
CREATE TABLE ShiftInDate (  
    ShiftNum INT,  
    ShiftDate DATE,  
    StartingTime TIME,  
    EndingTime TIME,  
    ChefNum INT NOT NULL,  
    ManagerNum INT NOT NULL,  
    PRIMARY KEY (ShiftNum, ShiftDate),  
    FOREIGN KEY (ShiftNum)  
        REFERENCES Shift (ShiftNum)  
        ON DELETE CASCADE,  
    FOREIGN KEY (ChefNum) REFERENCES Chef (EmployeeNum),  
    FOREIGN KEY (ManagerNum) REFERENCES Manager (EmployeeNum)  
);
```

טבלת משמרות
בתאריך

```
CREATE TABLE CooksInShifts (  
    ShiftNum INT,  
    ShiftDate DATE,  
    CookNum INT,  
    PRIMARY KEY (ShiftNum, ShiftDate, CookNum),  
    FOREIGN KEY (ShiftNum, ShiftDate) REFERENCES ShiftInDate (ShiftNum,  
ShiftDate),  
    FOREIGN KEY (CookNum) REFERENCES Cook (EmployeeNum)  
);
```

טבלת טבחים
במשמרות בתאריך



טבלת קופאים
במשמרות בתאריך

```
CREATE TABLE CashiersInShifts (  
    ShiftNum INT,  
    ShiftDate DATE,  
    CashierNum INT,  
    PRIMARY KEY (ShiftNum, ShiftDate, CashierNum),  
    FOREIGN KEY (ShiftNum, ShiftDate) REFERENCES ShiftInDate (ShiftNum,  
ShiftDate),  
    FOREIGN KEY (CashierNum) REFERENCES Cashier (EmployeeNum)  
);
```

טבלת לקוחות

```
CREATE TABLE Customer (  
    CustomerNum INT PRIMARY KEY,  
    Name VARCHAR(20),  
    PhoneNum VARCHAR(11)  
);
```

טבלת הזמנות

```
CREATE TABLE Orders (  
    OrderNum INT PRIMARY KEY,  
    Payment VARCHAR(20),  
    CustomerNum INT NOT NULL,  
    ShiftNum INT NOT NULL,  
    ShiftDate DATE NOT NULL,  
    CashierNum INT NOT NULL,  
    FOREIGN KEY (ShiftNum, ShiftDate, CashierNum)  
        REFERENCES CashiersInShifts (ShiftNum, ShiftDate, CashierNum),  
    FOREIGN KEY (CustomerNum) REFERENCES Customer (CustomerNum)  
);
```

טבלת מנות

```
CREATE TABLE Course (  
    CourseNum INT PRIMARY KEY,  
    Name VARCHAR(20),  
    CourseDesc VARCHAR(100),  
    Price INT  
);
```

טבלת מנות בהזמנה

```
CREATE TABLE CoursesInOrder (  
    OrderNum INT,  
    CourseNum INT,  
    Quantity INT,  
    PRIMARY KEY (OrderNum, CourseNum),  
    FOREIGN KEY (OrderNum) REFERENCES Orders (OrderNum),  
    FOREIGN KEY (CourseNum) REFERENCES Course (CourseNum)  
);
```



משימות התרגיל:

הריצו תחילה את הסקריפט ב- SQL-Server על מנת ליצור את כל הטבלאות עם רשומות הנתונים.

א. (10 נק') כתבו שאילתה המחזירה את כל המנות (מספר מנה, שם ותיאור) שהוזמנו בתאריך

17/01/2015

ב. (10 נק') מצאו בכמה ימים שונים מכר הקופאי Tom Reed את מנה מספר 39603

ג. (25 נק') כתבו שאילתה המחזירה לכל לקוח (שם ומספר לקוח) שרכש את מנה מספר 99771

ביותר מ-2 הזמנות בשנת 2014, מה סך הכסף שהוציא במסעדה בשנה זו

ד. (30 נק') כתבו שאילתת SQL המחזירה לקוחות (מס' לקוח ושם) שמעולם לא קיבלו שירות מהקופאי טום ריד (מס' עובד 10027), ושביצעו לפחות 5 הזמנות במהלך שנת 2014 במשמרות בהן שמו של מנהל המשמרת היה גרי פריי (Garry Frye)

ה. (25 נק') אחד מעובדי המסעדה כתב את השאילתה הבאה, כדי להציג לכל משמרת כמה קופאים היו בפועל, כמה היו צריכים להיות לפי הגדרת סוג המשמרת, ומה ההפרש ביניהם (אם היו יותר קופאים ממה שצריך יוצג הפרש חיובי, ואם היו פחות יוצג הפרש שלילי):

```
SELECT s.ShiftNum, COUNT(*), s.NumOfCashiers, (s.NumOfCashiers – COUNT(*))  
AS The_Difference  
FROM CashiersInShifts cs INNER JOIN Shift s  
ON cs.ShiftNum = s.ShiftNum  
GROUP BY s.ShiftNum
```

לאחר סקירת השאילתה גיליתם שיש בה טעויות. פרטו את **כל השגיאות** שמצאתם, והסבירו אותן.