

2024

# מערכת לניהול הזמנות

מיני פרויקט בבסיסי נתונים ה'תשפ"ד

עמינדב ברנר ויואב סרי

## תוכן עניינים

3.....	תיאור מערכת בית המלון
3.....	תיאור מערכת המידע:
3.....	תיאור הישויות:
4.....	הקשרים במערכת המידע:
5.....	תרשימי הישויות והקשרים
6.....	יצירת הטבלאות לבסיס הנתונים
6.....	יצירת הטבלה Employee:
7.....	יצירת הטבלה ReservationAgent:
8.....	יצירת הטבלה Receptionist:
9.....	יצירת הטבלה Room:
10.....	יצירת הטבלה Guest:
12.....	יצירת הטבלה Booking:
13.....	יצירת הטבלה Request:
14.....	יצירת הטבלה Inform:
16.....	סיכום יצירה, הכנסה, ומחיקה
16.....	סיכום יצירת הטבלאות:
17.....	סיכום הכנסת ערכים לטבלאות:
18.....	מחיקת טבלאות:
19.....	הצגת הטבלאות:
20.....	אכלוס בסיס הנתונים
20.....	אכלוס הטבלה Guest:
20.....	אכלוס הטבלה Room:
21.....	אכלוס הטבלה Employee:
22.....	אכלוס הטבלה ReservationAgent:
22.....	אכלוס הטבלה Receptionist:
23.....	אכלוס הטבלה Booking:
23.....	אכלוס הטבלה Request:
23.....	אכלוס הטבלה Inform:
25.....	גיבוי ושחזור בסיס הנתונים
25.....	גיבוי בסיס הנתונים:

25.....	ייבוא בסיס נתונים:
28.....	שאילתות על בסיס הנתונים
28.....	שאילתות select:
28.....	שאילתה 1:
29.....	שאילתה 2:
29.....	שאילתה 3:
30.....	שאילתה 4:
31.....	שאילתות delete:
31.....	שאילתה 1:
32.....	שאילתה 2:
34.....	שאילתות update:
34.....	שאילתה 1:
35.....	שאילתה 2:
37.....	שאילתות עם פרמטרים:
37.....	שאילתה 1:
38.....	שאילתה 2:
40.....	שאילתה 3:
41.....	שאילתה 4:
43.....	הוספת אילוצים לטבלאות:
43.....	אילוץ ראשון:
43.....	אילוץ שני:
44.....	אילוץ שלישי:

# תיאור מערכת בית המלון

## תיאור מערכת המידע:

אנחנו יוצרים מאגר נתונים עבור בית המלון הידוע "AirPods Plus Plus" בברית הניה.

בית מלון, כמו בית מלון, מורכב ממחלקות שונות ומשונות שתפקידן המשותף הוא לדאוג לרווחת האורחים. במחלקת ניהול הזמנות הסתבכו עד לא מזמן עם טפסים ידניים ונקראנו לדגל כדי להציל את המצב!

תפקידה של מחלקת ההזמנות הוא לטפל בכל מה שקשור להזמנות חדרים שמבוצעות על ידי אורחי המלון. כאשר אורח מתקשר למלון לבצע הזמנה הוא מטופל על ידי אחד העובדים מצוות הדלפקאים של המלון. אם צריך, הדלפקאי איתו הוא מדבר יכול להפנות אותו לאחד מסוכני ההזמנות של בית המלון בכדי שיידע אותו בכל מה שצריך לגבי הזמנת חדר.

## תיאור הישיות:

• **אורח (Guest):** טבלת האורחים תכיל את כל האורחים שנקלטו במערכת המלון משנת 1970

עד היום (כולל אלו העתידיים להתארח).

- guest\_id – מספר תעודת זהות.

- first\_name – שם פרטי.

- last\_name – שם משפחה.

- phone – מספר טלפון ליצירת קשר.

- date\_of\_birth – תאריך לידה.

- address – כתובת מגורים.

• **חדר (Room):** טבלת חדרי השינה תכיל את כל חדרי השינה שבמלון.

- room\_number – מספר החדר.

- beds – מספר מיטות בחדר.

- balcony – מרפסת (יש / אין).

- price – מחיר ללילה.

• **עובד (Employee):** טבלת העובדים תכיל את כל עובדי בית המלון לסוגם.

- em\_id – מספר העובד.

- first\_name – שם פרטי.

- last\_name – שם משפחה.

- salary – משכורת חודשית.

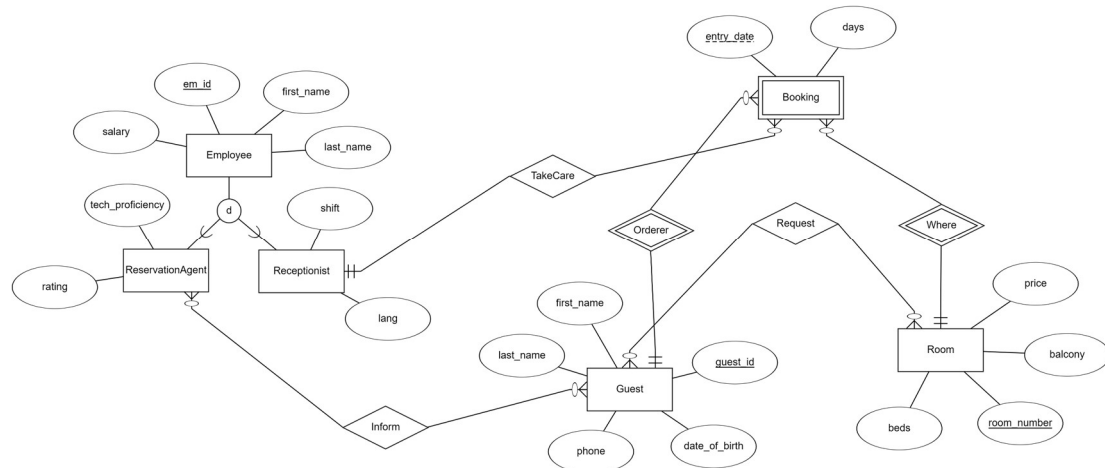
- **סוכן (ReservationAgent):** טבלת הסוכנים תכיל את כל הסוכנים של בית המלון.
  - em\_id – מספר העובד.
  - rating – דירוג שקיבל הסוכן מהלקוחות.
  - tech\_proficiency – התמחות טכנולוגית (רמה גבוהה, ממוצעת, נמוכה).
- **מארח (Receptionist):** טבלת המארחים תכיל את כל המארחים של בית המלון.
  - em\_id – מספר העובד.
  - lang – שפת אם לתקשורת עם לקוחות.
  - shift – משמרת לעבודה (בוקר / ערב / לילה).
- **הזמנה (Booking):** טבלת ההזמנות תכיל את כל ההזמנות של האורחים בבית המלון.
  - guest\_id – תעודת זהות המזמין.
  - room\_number – מספר החדר.
  - entry\_date – תאריך כניסה.
  - days – מספר ימי השהות במלון.

#### הקשרים במערכת המידע:

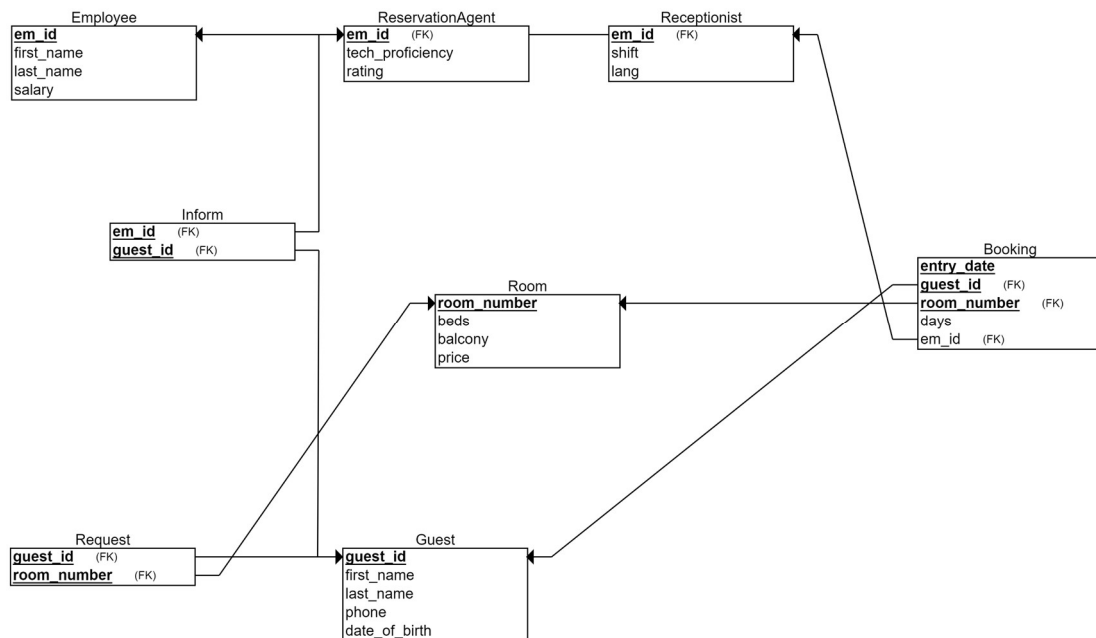
- קשר **Inform**: לא כל האורחים שרוצים להזמין חדרים במלון בקיאים בסוד העניינים. יש כאלו שזו פעם ראשונה שהם מזמינים בעצמם חדר/ים בבית מלון. לצורך כך המלון מעסיק סוכנים שתפקידם לסייע לאותם אנשים לבצע את ההזמנה עליה הם חולמים.
- קשר **TakeCare**: כאשר מתקבלת הזמנה במערכת, היא מועברת לטיפול אחד מהמארחים של בית המלון. כל הזמנה מטופלת על ידי מארח אחד בלבד לאורך כל חייה.
- קשר **Request**: במהלך שהותם של האורחים במלון הם יכולים להיתקל בחדרים שדורשים טיפול או תיקון (גם אם זה לא החדר שהם הזמינו). טיפול יכול להיות גם לבקש מגבת נוספת או כל דבר שחסר. במקרה כזה הם ימלאו טופס דיווח על תקלה שישלח לאגף התחזוקה של בית המלון. כל אורח יכול לדווח על כל אחד מהחדרים, ולחילופין – רבים לרבים.

# תרשימי הישויות והקשרים

התרשים הראשון הוא תרשים ERD שמתאר לנו את התכונות של כל ישות ואת הקשרים בין ישות אחת לחברתה.



התרשים השני של המערכת הוא תרשים DSD שמציג את הקשרים בין התכונות של הישויות. בתרשים זה נוכל לראות איזו ישות תלויה באיזו ישות מבחינת המפתח הזר שלה וכו'.



את שני התרשימים יצרנו בעזרת האתר <https://erdplus.com>

## יצירת הטבלאות לבסיס הנתונים

ביצירת הטבלאות נשים לב שיש טבלאות שלא ניתן לייצר לפני שייצרנו טבלה קודמת, למשל את הטבלה Receptionist לא נוכל לייצר לפני שייצרנו את הטבלה Employee, כיוון שהתכונה em\_id ב-Receptionist היא מפתח זר.

הסדר שבו נבחר ליצור את הטבלאות הוא (משמאל לימין):

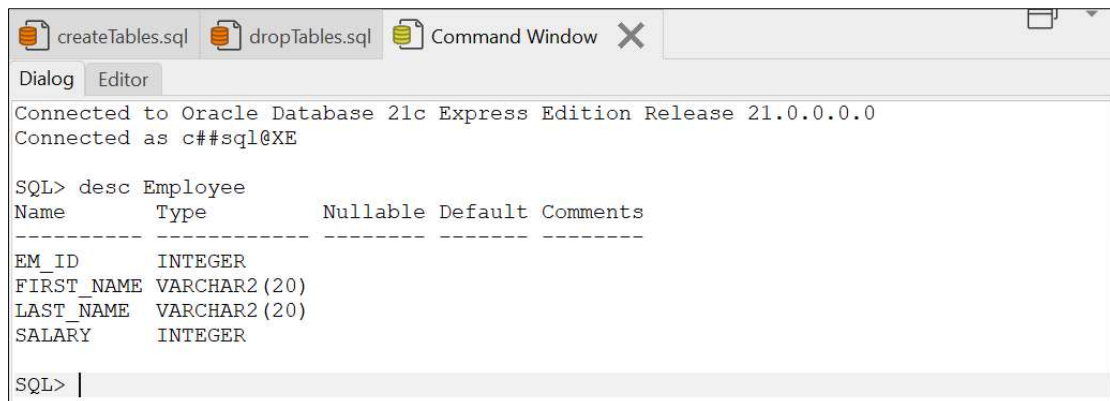
Employee, ReservationAgent, Receptionist, Room, Guest, Booking, Request, Inform

### יצירת הטבלה Employee:

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE Employee
(
  em_id INT NOT NULL,
  first_name VARCHAR2(20) NOT NULL,
  last_name VARCHAR2(20) NOT NULL,
  salary INT NOT NULL,
  PRIMARY KEY (em_id)
);
```

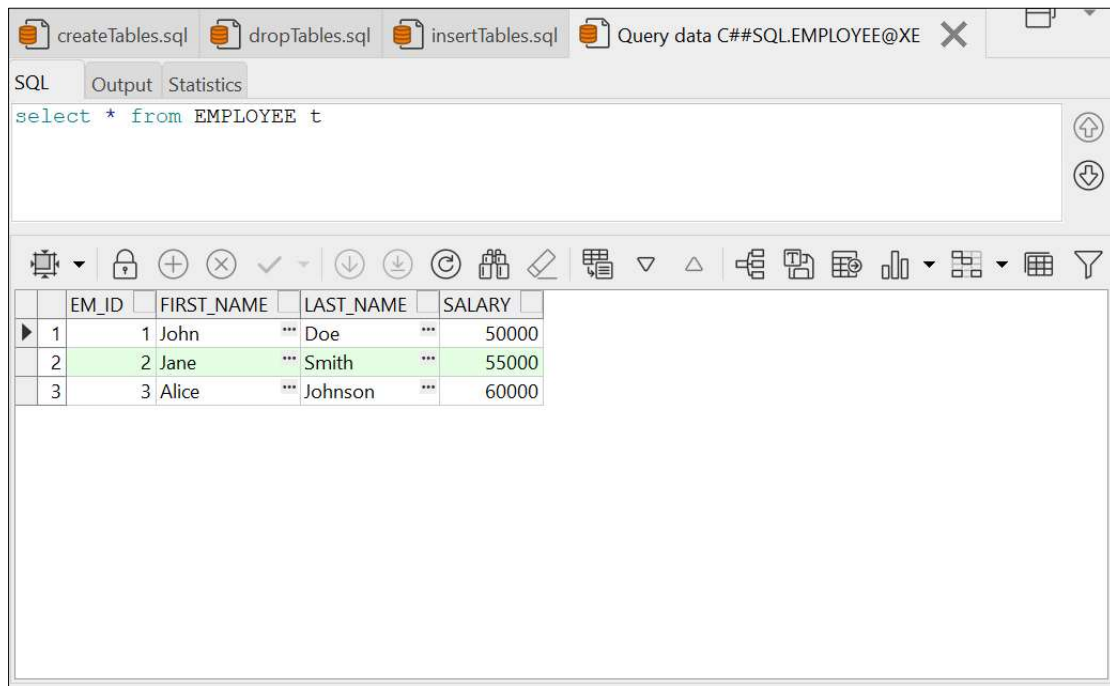
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (1, 'John', 'Doe', 50000);
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (2, 'Jane', 'Smith', 55000);
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (3, 'Alice', 'Johnson', 60000);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:



## יצירת הטבלה ReservationAgent:

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE ReservationAgent
(
  em_id INT NOT NULL,
  tech_proficiency VARCHAR2(15) NOT NULL,
  rating INT NOT NULL,
  PRIMARY KEY (em_id),
  FOREIGN KEY (em_id) REFERENCES Employee(em_id)
);
```

נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



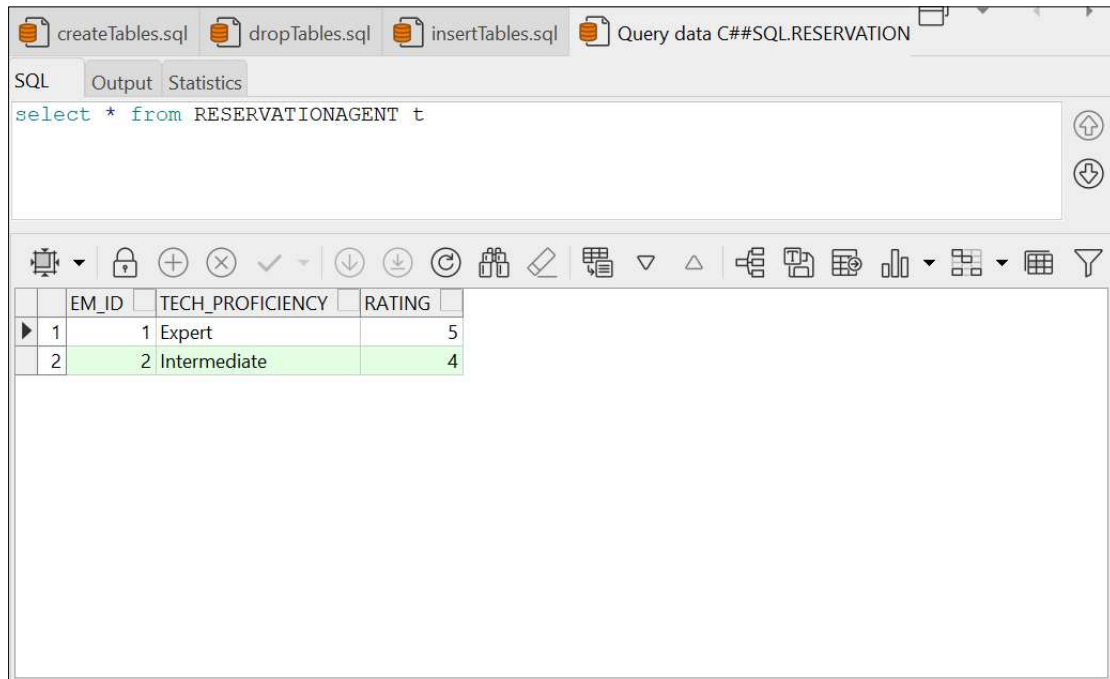
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO ReservationAgent (em_id, tech_proficiency, rating)
VALUES (1, 'Expert', 5);
INSERT INTO ReservationAgent (em_id, tech_proficiency, rating)
```



```
VALUES (2, 'Intermediate', 4);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:



### יצירת הטבלה Receptionist

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE Receptionist
(
  em_id INT NOT NULL,
  shift VARCHAR2(10) NOT NULL,
  lang VARCHAR2(30) NOT NULL,
  PRIMARY KEY (em_id),
  FOREIGN KEY (em_id) REFERENCES Employee (em_id)
);
```

נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



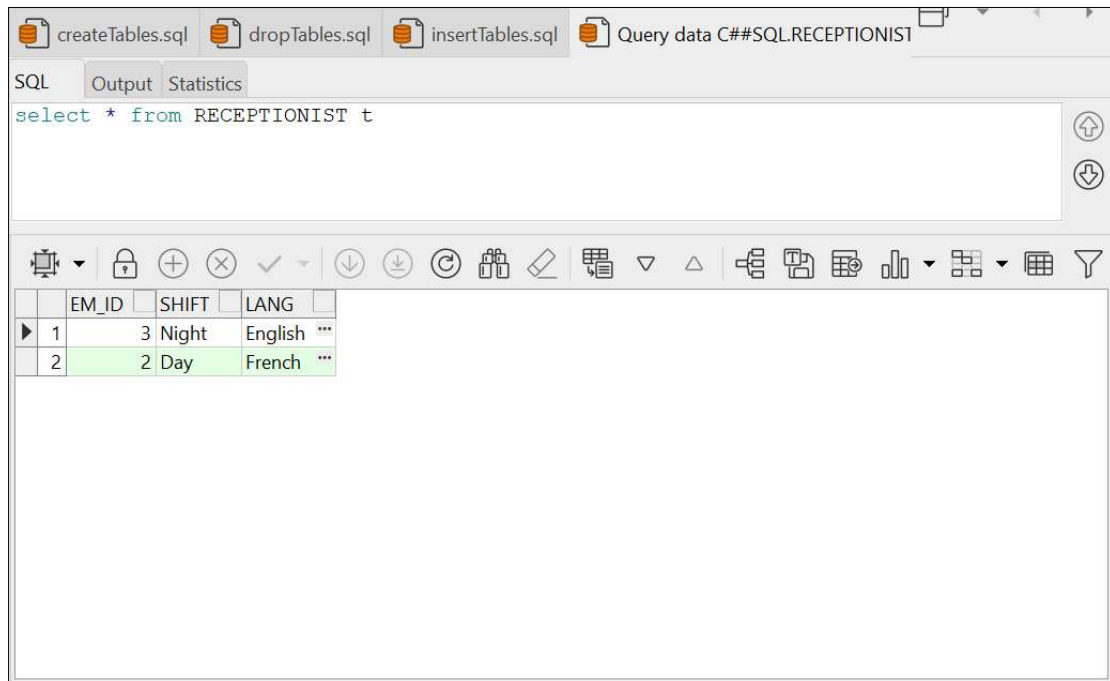
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```

INSERT INTO Receptionist (em_id, shift, lang)
VALUES (3, 'Night', 'English');
INSERT INTO Receptionist (em_id, shift, lang)
VALUES (2, 'Day', 'French');

```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:



The screenshot shows a SQL IDE window with a query editor and an output pane. The query editor contains the SQL statement: `select * from RECEPTIONIST t`. The output pane displays the results of the query in a table format.

	EM_ID	SHIFT	LANG
1	3	Night	English
2	2	Day	French

### יצירת הטבלה Room:

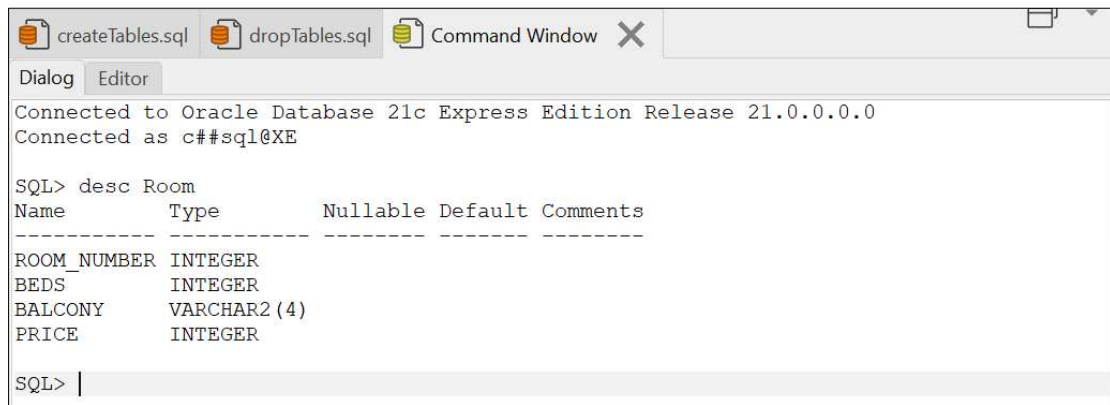
קוד ה-SQL שיוצר לנו את הטבלה הוא:

```

CREATE TABLE Room
(
    room_number INT NOT NULL,
    beds INT NOT NULL,
    balcony VARCHAR2(4) NOT NULL,
    price INT NOT NULL,
    PRIMARY KEY (room_number)
);

```

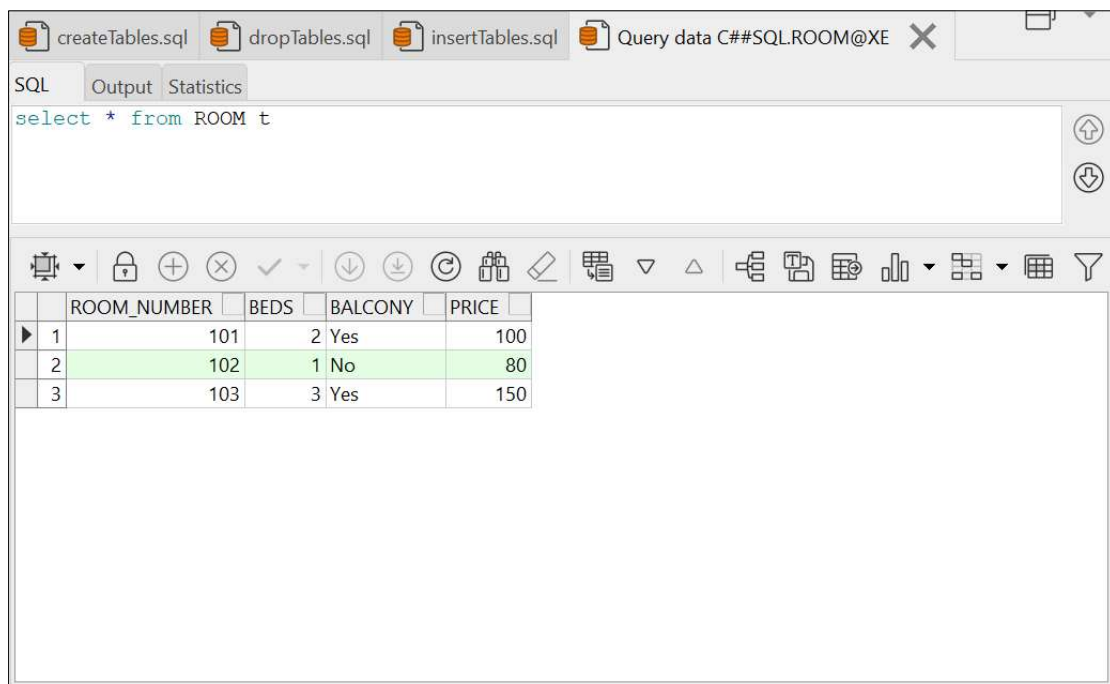
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO Room (room_number, beds, balcony, price)
VALUES (101, 2, 'Yes', 100);
INSERT INTO Room (room_number, beds, balcony, price)
VALUES (102, 1, 'No', 80);
INSERT INTO Room (room_number, beds, balcony, price)
VALUES (103, 3, 'Yes', 150);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:



יצירת הטבלה Guest:

קוד ה-SQL שיוצר לנו את הטבלה הוא:

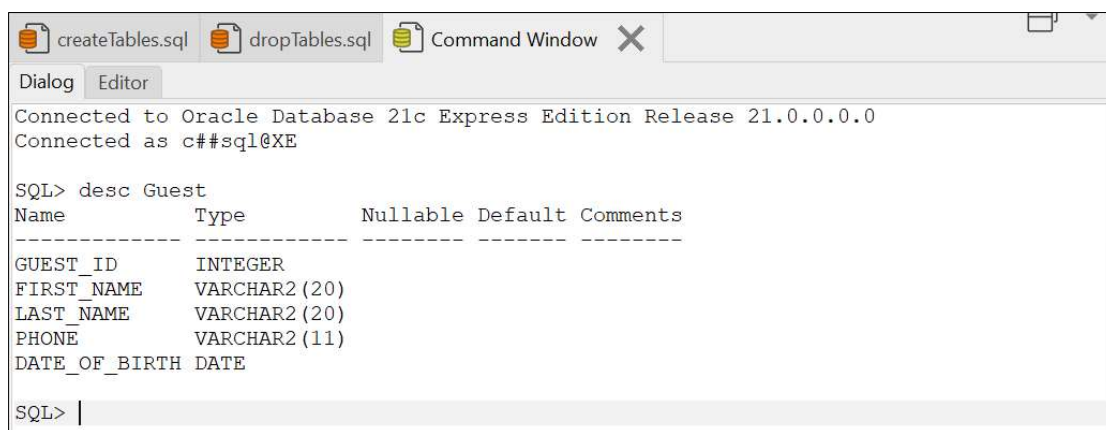
```
CREATE TABLE Guest
(
    guest_id INT NOT NULL,
    first_name VARCHAR2(20) NOT NULL,
```

```

last_name VARCHAR2(20) NOT NULL,
phone VARCHAR2(11) NOT NULL,
date_of_birth DATE NOT NULL,
PRIMARY KEY (guest_id)
);

```

נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



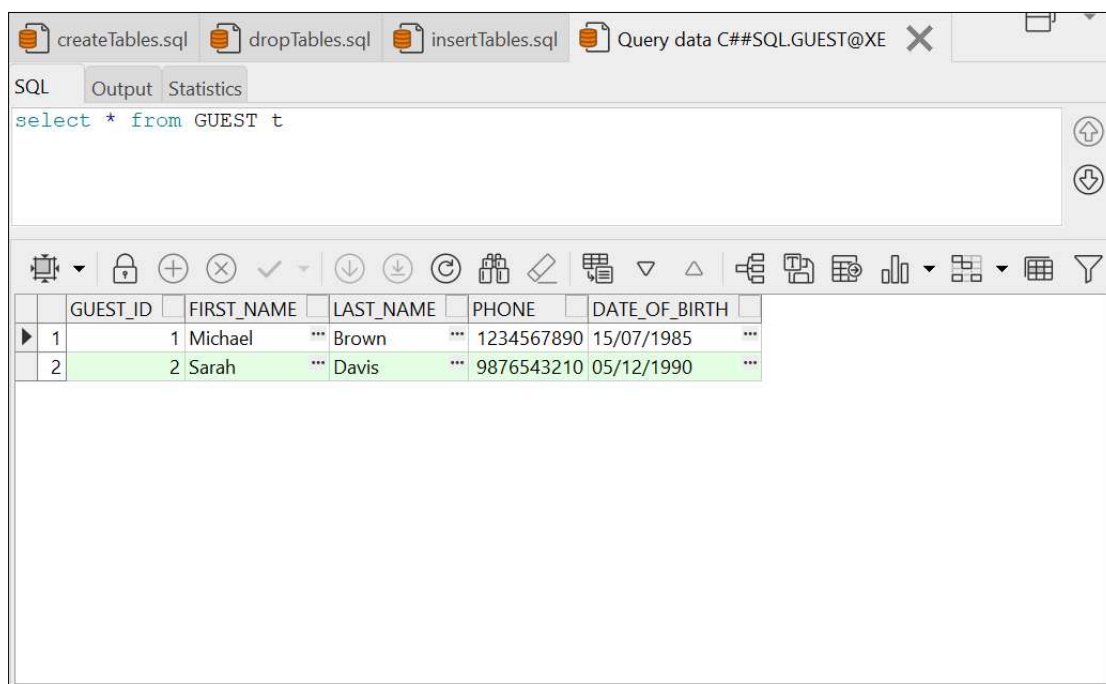
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```

INSERT INTO Guest (guest_id, first_name, last_name, phone,
date_of_birth)
VALUES (1, 'Michael', 'Brown', 1234567890, DATE '1985-07-15');
INSERT INTO Guest (guest_id, first_name, last_name, phone,
date_of_birth)
VALUES (2, 'Sarah', 'Davis', 9876543210, DATE '1990-12-05');

```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:

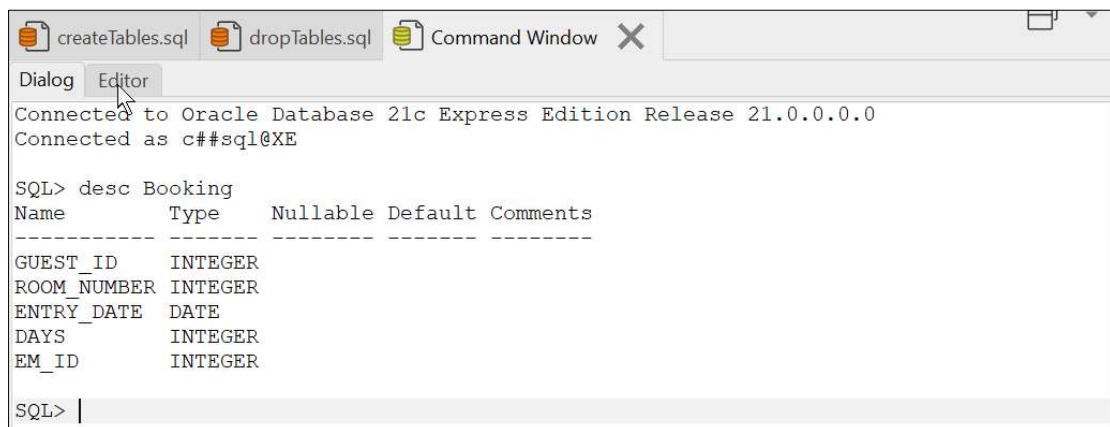


## יצירת הטבלה Booking:

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE Booking
(
    guest_id INT NOT NULL,
    room_number INT NOT NULL,
    entry_date DATE NOT NULL,
    days INT NOT NULL,
    em_id INT NOT NULL,
    PRIMARY KEY (entry_date, guest_id, room_number),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_number) REFERENCES Room(room_number),
    FOREIGN KEY (em_id) REFERENCES Receptionist(em_id)
);
```

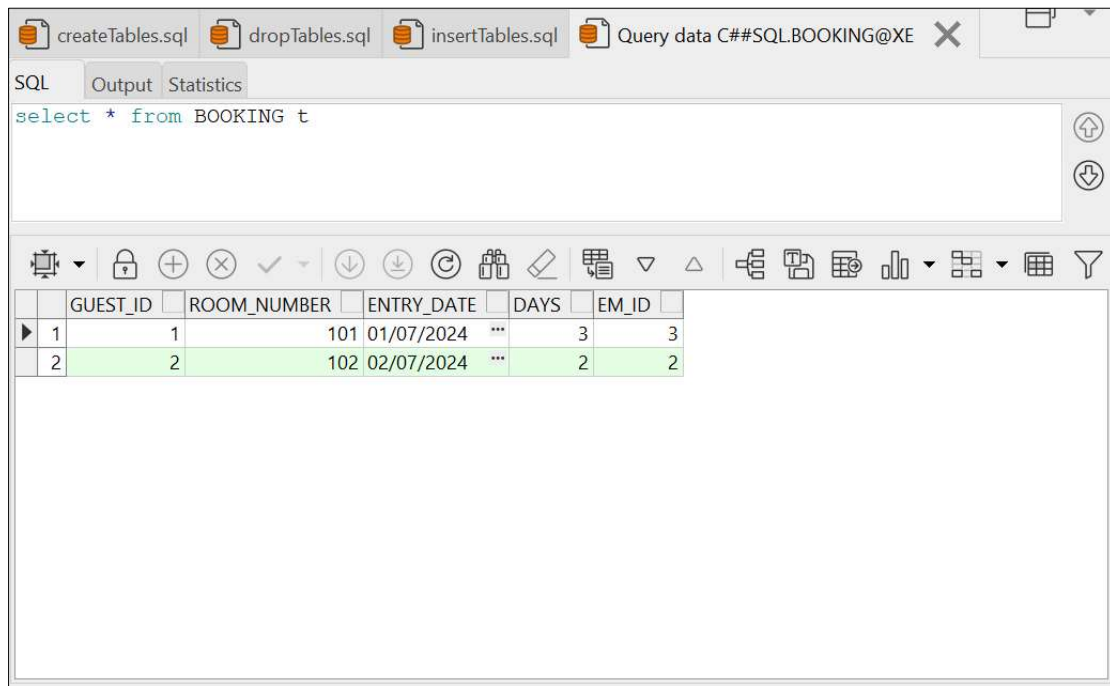
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO Booking (guest_id, room_number, entry_date, days, em_id)
VALUES (1, 101, DATE '2024-07-01', 3, 3);
INSERT INTO Booking (guest_id, room_number, entry_date, days, em_id)
VALUES (2, 102, DATE '2024-07-02', 2, 2);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:

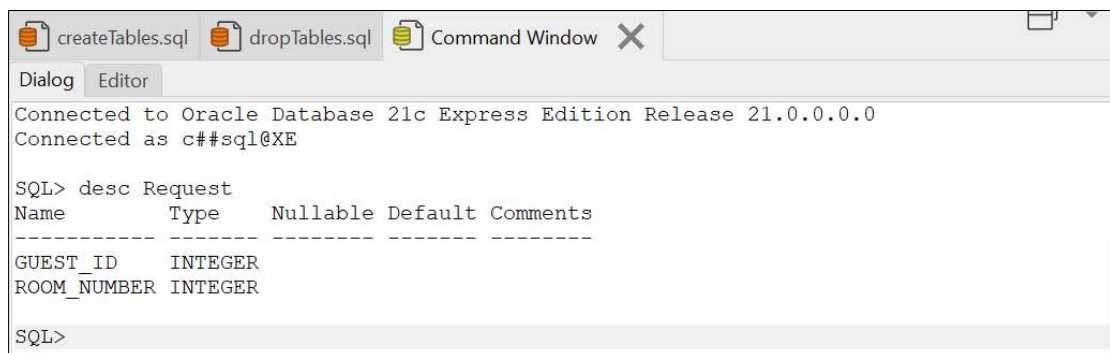


### יצירת הטבלה Request:

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE Request
(
  guest_id INT NOT NULL,
  room_number INT NOT NULL,
  PRIMARY KEY (guest_id, room_number),
  FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
  FOREIGN KEY (room_number) REFERENCES Room(room_number)
);
```

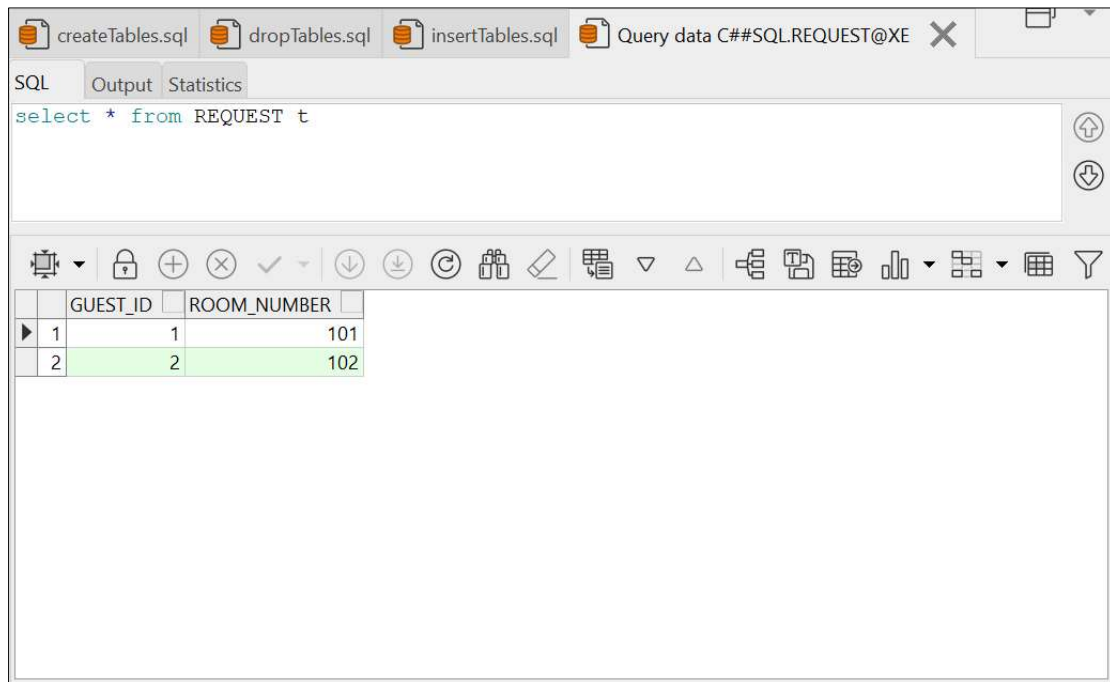
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO Request (guest_id, room_number)
VALUES (1, 101);
INSERT INTO Request (guest_id, room_number)
VALUES (2, 102);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:

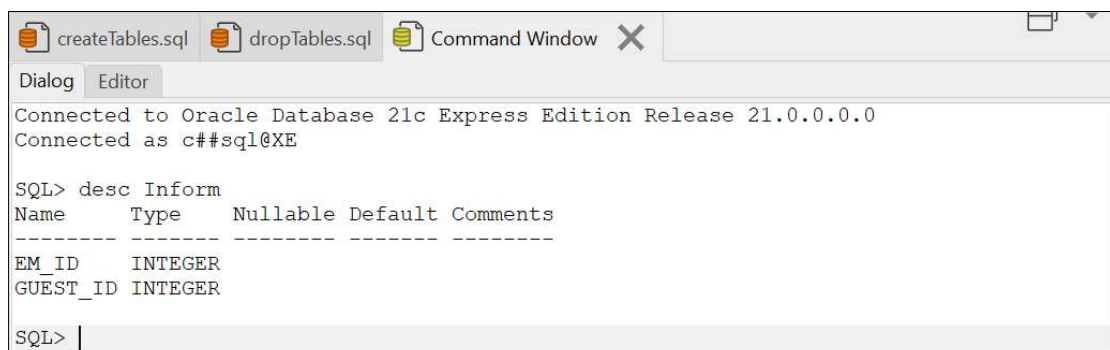


### יצירת הטבלה Inform

קוד ה-SQL שיוצר לנו את הטבלה הוא:

```
CREATE TABLE Inform
(
  em_id INT NOT NULL,
  guest_id INT NOT NULL,
  PRIMARY KEY (em_id, guest_id),
  FOREIGN KEY (em_id) REFERENCES ReservationAgent(em_id),
  FOREIGN KEY (guest_id) REFERENCES Guest(guest_id)
);
```

נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:

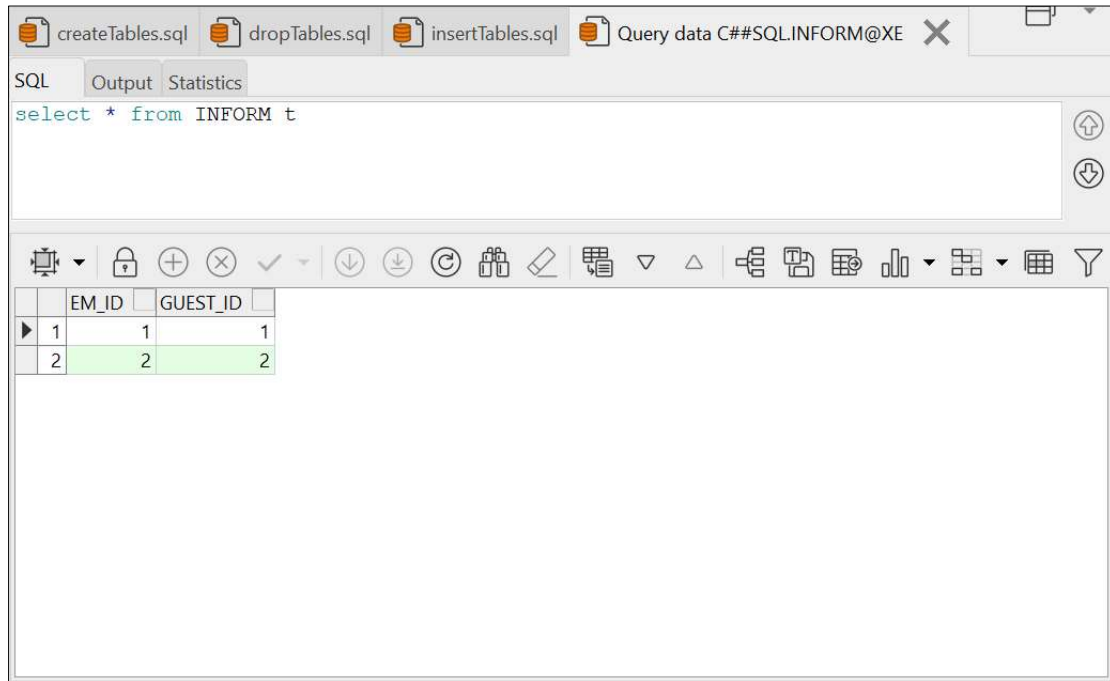


נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
INSERT INTO Inform (em_id, guest_id)
```

```
VALUES (1, 1);  
INSERT INTO Inform (em_id, guest_id)  
VALUES (2, 2);
```

נבדוק שאכן הערכים הוכנסו לטבלה כמו שצריך:



The screenshot shows the SQL Developer interface. The top tab bar includes 'createTables.sql', 'dropTables.sql', 'insertTables.sql', and 'Query data C##SQL.INFORM@XE'. The 'SQL' tab is active, displaying the query 'select \* from INFORM t'. Below the query editor is a toolbar with various icons. The results pane shows a table with two columns, 'EM\_ID' and 'GUEST\_ID', and two rows of data. The first row has values 1 and 1, and the second row has values 2 and 2. The second row is highlighted in green.

	EM_ID	GUEST_ID
▶	1	1
	2	2



# סיכום יצירה, הכנסה, ומחיקה

## סיכום יצירת הטבלאות:

נאחד את כל קטעי הקוד שכתבנו ליצירת הטבלאות לקובץ אחד בשם `createTables.sql`.

```
-- Table to store employee details
CREATE TABLE Employee
(
    em_id INT NOT NULL,
    first_name VARCHAR2(20) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    salary INT NOT NULL,
    PRIMARY KEY (em_id)
);

-- Table to store reservation agent details, extending Employee
CREATE TABLE ReservationAgent
(
    em_id INT NOT NULL,
    tech_proficiency VARCHAR2(15) NOT NULL,
    rating INT NOT NULL,
    PRIMARY KEY (em_id),
    FOREIGN KEY (em_id) REFERENCES Employee(em_id)
);

-- Table to store receptionist details, extending Employee
CREATE TABLE Receptionist
(
    em_id INT NOT NULL,
    shift VARCHAR2(10) NOT NULL,
    lang VARCHAR2(30) NOT NULL,
    PRIMARY KEY (em_id),
    FOREIGN KEY (em_id) REFERENCES Employee(em_id)
);

-- Table to store room details
CREATE TABLE Room
(
    room_number INT NOT NULL,
    beds INT NOT NULL,
    balcony VARCHAR2(4) NOT NULL,
    price INT NOT NULL,
    PRIMARY KEY (room_number)
);

-- Table to store guest details
CREATE TABLE Guest
(
    guest_id INT NOT NULL,
    first_name VARCHAR2(20) NOT NULL,
    last_name VARCHAR2(20) NOT NULL,
    phone VARCHAR2(11) NOT NULL,
    date_of_birth DATE NOT NULL,
    PRIMARY KEY (guest_id)
);

-- Table to store booking details
```

```

CREATE TABLE Booking
(
    guest_id INT NOT NULL,
    room_number INT NOT NULL,
    entry_date DATE NOT NULL,
    days INT NOT NULL,
    em_id INT NOT NULL,
    PRIMARY KEY (entry_date, guest_id, room_number),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_number) REFERENCES Room(room_number),
    FOREIGN KEY (em_id) REFERENCES Receptionist(em_id)
);

-- Table to store requests made by guests
CREATE TABLE Request
(
    guest_id INT NOT NULL,
    room_number INT NOT NULL,
    PRIMARY KEY (guest_id, room_number),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_number) REFERENCES Room(room_number)
);

-- Table to store information about interactions between reservation
agents and guests
CREATE TABLE Inform
(
    em_id INT NOT NULL,
    guest_id INT NOT NULL,
    PRIMARY KEY (em_id, guest_id),
    FOREIGN KEY (em_id) REFERENCES ReservationAgent(em_id),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id)
);

```

## סיכום הכנסת ערכים לטבלאות:

נאחד את כל קטעי הקוד שכתבנו להכנסת ערכים לטבלאות לקובץ אחד בשם insertTables.sql.

```

-- Inserting into Employee table
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (1, 'John', 'Doe', 50000);
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (2, 'Jane', 'Smith', 55000);
INSERT INTO Employee (em_id, first_name, last_name, salary)
VALUES (3, 'Alice', 'Johnson', 60000);

-- Inserting into ReservationAgent table
INSERT INTO ReservationAgent (em_id, tech_proficiency, rating)
VALUES (1, 'Expert', 5);
INSERT INTO ReservationAgent (em_id, tech_proficiency, rating)
VALUES (2, 'Intermediate', 4);

-- Inserting into Receptionist table
INSERT INTO Receptionist (em_id, shift, lang)
VALUES (3, 'Night', 'English');
INSERT INTO Receptionist (em_id, shift, lang)
VALUES (2, 'Day', 'French');

-- Inserting into Room table
INSERT INTO Room (room_number, beds, balcony, price)

```

```

VALUES (101, 2, 'Yes', 100);
INSERT INTO Room (room_number, beds, balcony, price)
VALUES (102, 1, 'No', 80);
INSERT INTO Room (room_number, beds, balcony, price)
VALUES (103, 3, 'Yes', 150);

-- Inserting into Guest table
INSERT INTO Guest (guest_id, first_name, last_name, phone,
date_of_birth)
VALUES (1, 'Michael', 'Brown', 1234567890, DATE '1985-07-15');
INSERT INTO Guest (guest_id, first_name, last_name, phone,
date_of_birth)
VALUES (2, 'Sarah', 'Davis', 9876543210, DATE '1990-12-05');

-- Inserting into Booking table
INSERT INTO Booking (guest_id, room_number, entry_date, days, em_id)
VALUES (1, 101, DATE '2024-07-01', 3, 3);
INSERT INTO Booking (guest_id, room_number, entry_date, days, em_id)
VALUES (2, 102, DATE '2024-07-02', 2, 2);

-- Inserting into Request table
INSERT INTO Request (guest_id, room_number)
VALUES (1, 101);
INSERT INTO Request (guest_id, room_number)
VALUES (2, 102);

-- Inserting into Inform table
INSERT INTO Inform (em_id, guest_id)
VALUES (1, 1);
INSERT INTO Inform (em_id, guest_id)
VALUES (2, 2);

```

## מחיקת טבלאות:

כפי שביצירת הטבלאות היינו צריכים להקפיד מאוד על הסדר הנכון, כך גם במחיקת הטבלאות עלינו להקפיד על הסדר מאותה הסיבה. לא נוכל למחוק טבלה שאחת התכונות שלה היא מפתח זר בטבלה אחרת... הסדר בו נמחק את הטבלאות יהיה בסדר ההפוך לסדר אפשרי ליצירתן.

```

-- Drop Inform table
DROP TABLE Inform;

-- Drop Request table
DROP TABLE Request;

-- Drop Booking table
DROP TABLE Booking;

-- Drop Guest table
DROP TABLE Guest;

-- Drop Room table
DROP TABLE Room;

-- Drop Receptionist table
DROP TABLE Receptionist;

-- Drop ReservationAgent table
DROP TABLE ReservationAgent;

```

```
-- Drop Employee table
DROP TABLE Employee;
```

### הצגת הטבלאות:

נרצה לכתוב קוד SQL שיציג לנו את כל אחת מהטבלאות שיצרנו ואכלסנו (עוד לא ממש אכלסנו, רק הכנסנו מעט מאוד ערכים לבדיקה).

```
-- Select all from Employee table
SELECT * FROM Employee;

-- Select all from ReservationAgent table
SELECT * FROM ReservationAgent;

-- Select all from Receptionist table
SELECT * FROM Receptionist;

-- Select all from Room table
SELECT * FROM Room;

-- Select all from Guest table
SELECT * FROM Guest;

-- Select all from Booking table
SELECT * FROM Booking;

-- Select all from Request table
SELECT * FROM Request;

-- Select all from Inform table
SELECT * FROM Inform;
```

# אכלוס בסיס הנתונים

נשתמש בשלוש דרכים כדי לג'נרט (לחולל בעברית) נתונים לבסיס המידע שלנו:

- הכלי Data Generator שמובנה בסביבת העבודה של PL/SQL.
- קבצי SQL עם פקודות ליצירת רשומות שניצור בעזרת האתר *Mockaroo*.
- קבצי Excel עם טבלאות נתונים אותם נייבא לבסיס הנתונים שלנו. גם את קבצי ה-Excel ניצור באמצעות האתר *Mockaroo*.

## אכלוס הטבלה Guest:

באתר *Mockaroo* ניצור סכמה חדשה בשם Guest ונגדיר עבורה שדות מתאימים.

The screenshot shows the Mockaroo web interface. At the top, there's a navigation bar with links like SCHEMAS, DATASETS, MOCK APIS, SCENARIOS, PROJECTS, and FUNCTIONS. Below this, the 'My Schemas / Guest' section is active. The main area is titled 'Guest' and contains a table with columns: Field Name, Type, and Options. The table lists five fields: 'guest\_id' (Regular Expression, \d{9}), 'first\_name' (First Name), 'last\_name' (Last Name), 'phone' (Regular Expression, 05\d{1}-\d{7}), and 'date\_of\_birth' (Datetime, 01/01/1950 to 01/01/2006). Each field has a 'blank' percentage set to 0% and a 'Generate' button. Below the table, there are buttons for '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. At the bottom, there are settings for '# Rows' (1000), 'Format' (SQL), 'Table Name' (Guest), and an 'Include CREATE TABLE' checkbox. The bottom bar contains buttons for 'GENERATE DATA', 'PREVIEW', 'CHANGES SAVED', 'CREATE API', and 'MORE'.

בתחתית המסך נבקש שהנתונים יוצרו בפורמט SQL, כלומר קובץ SQL עם המון פקודות INSERT INTO. את הקובץ נוריד ונשמור בשם `Guest.sql`.

## אכלוס הטבלה Room:

באתר *Mockaroo* ניצור סכמה חדשה בשם Room ונגדיר עבורה שדות מתאימים.

בתחתית המסך נבקש שהנתונים יוצרו בפורמט SQL, כלומר קובץ SQL עם המון פקודות INSERT INTO. את הקובץ נוריד ונשמור בשם Room.sql.

## אבלוס הטבלה Employee:

באתר *Mockaroo* ניצור סכמה חדשה בשם Employee ונגדיר עבודה שדות מתאימים.

את הפורמט נבחר להיות CSV כיוון שרק ככה האתר מאפשר לשמור את הטבלה אצלו בתור dataset. בעת נוכל להיכנס ללשונית DATASETS ולהורית אותו משם כקובץ csv. כיוון ש-PL/SQL מאפשר לנו לייבא רק קבצי אקסל – נמיר את קובץ ה-csv. לקובץ xlsx. ונקרא לו EmployeeTable.xlsx.

## אכלוס הטבלה ReservationAgent:

באתר *Mockaroo* ניצור סכמה חדשה בשם *Employee* ונגדיר עבורה שדות מתאימים. נשים לב שאת שדה *em\_id* נצטרך לייחס ל-*em\_id* של הסכמה *Employee*, וזאת הסיבה ששמרנו את הטבלה *Employee* שנוצרה, כ-*dataset* באתר.

The screenshot shows the Mockaroo interface for creating a dataset named 'ReservationAgent'. The schema is defined with three fields: 'em\_id' (Dataset Column, linked to Employee, random, 0% blank), 'tech\_proficiency' (Custom List, Expert, Beginner, Intermediate, random, 0% blank), and 'rating' (Number, min: 1, max: 10, decimals: 0, 0% blank). The number of rows is set to 400, and the format is Excel. The 'Append Dataset' dropdown is set to 'choose a dataset...'. The bottom bar shows 'Generating data via cURL' and buttons for 'GENERATE DATA', 'PREVIEW', 'CHANGES SAVED', 'CREATE API', and 'MORE'.

בתחתית המסך נבקש שהנתונים ייווצרו בפורמט קובץ Excel. את קובץ האקסל שנוצר נוריד ונשמור בשם *ReservationAgentTable.xlsx*.

## אכלוס הטבלה Receptionist:

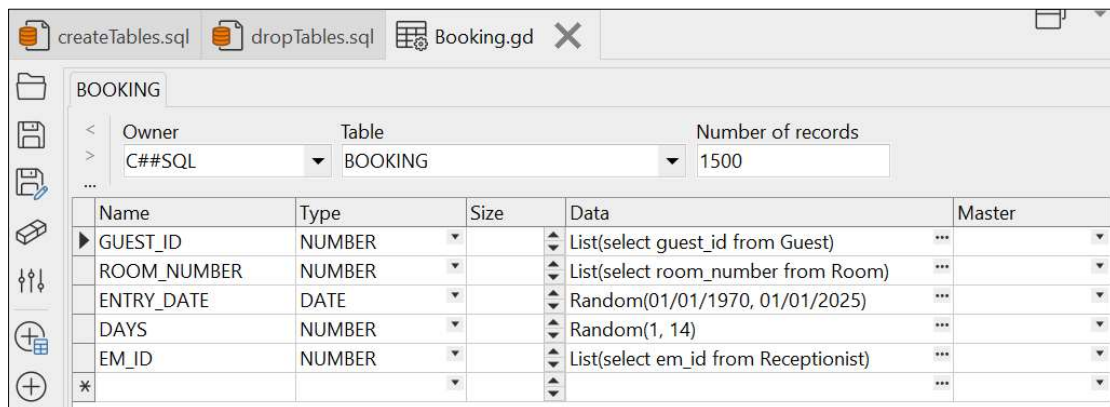
באתר *Mockaroo* ניצור סכמה חדשה בשם *Employee* ונגדיר עבורה שדות מתאימים. נשים לב שאת שדה *em\_id* נצטרך לייחס ל-*em\_id* של הסכמה *Employee*, וזאת הסיבה ששמרנו את הטבלה *Employee* שנוצרה, כ-*dataset* באתר.

The screenshot shows the Mockaroo interface for creating a dataset named 'Receptionist'. The schema is defined with three fields: 'em\_id' (Dataset Column, linked to Employee, random, 0% blank), 'shift' (Custom List, Night, Morning, Evening, random, 0% blank), and 'lang' (Language, 0% blank). The number of rows is set to 400, and the format is Excel. The 'Append Dataset' dropdown is set to 'choose a dataset...'. The bottom bar shows 'Generating data via cURL' and buttons for 'GENERATE DATA', 'PREVIEW', 'SAVE CHANGES', 'CREATE API', and 'MORE'.

בתחתית המסך נבקש שהנתונים ייווצרו בפורמט קובץ Excel. את קובץ האקסל שנוצר נוריד ונשמור בשם ReceptionistTable.xlsx.

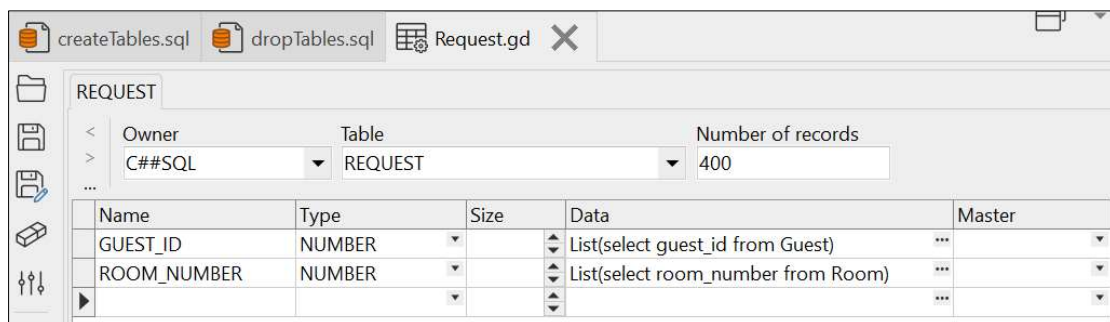
### אכלוס הטבלה Booking:

נשמש בכלי Data Generator המובנה בתוך PL/SQL. את הקובץ יצירה נשמור בשם Booking.gd כדי שנוכל להשתמש בו פעם נוספת אם נרצה.



### אכלוס הטבלה Request:

נשמש בכלי Data Generator המובנה בתוך PL/SQL. את הקובץ יצירה נשמור בשם Request.gd כדי שנוכל להשתמש בו פעם נוספת אם נרצה.



### אכלוס הטבלה Inform:

נשמש בכלי Data Generator המובנה בתוך PL/SQL. את הקובץ יצירה נשמור בשם Inform.gd כדי שנוכל להשתמש בו פעם נוספת אם נרצה.



createTables.sql

dropTables.sql

Inform.gd

X

INFORM

<

Owner

Table

Number of records

>

C##SQL

INFORM

500

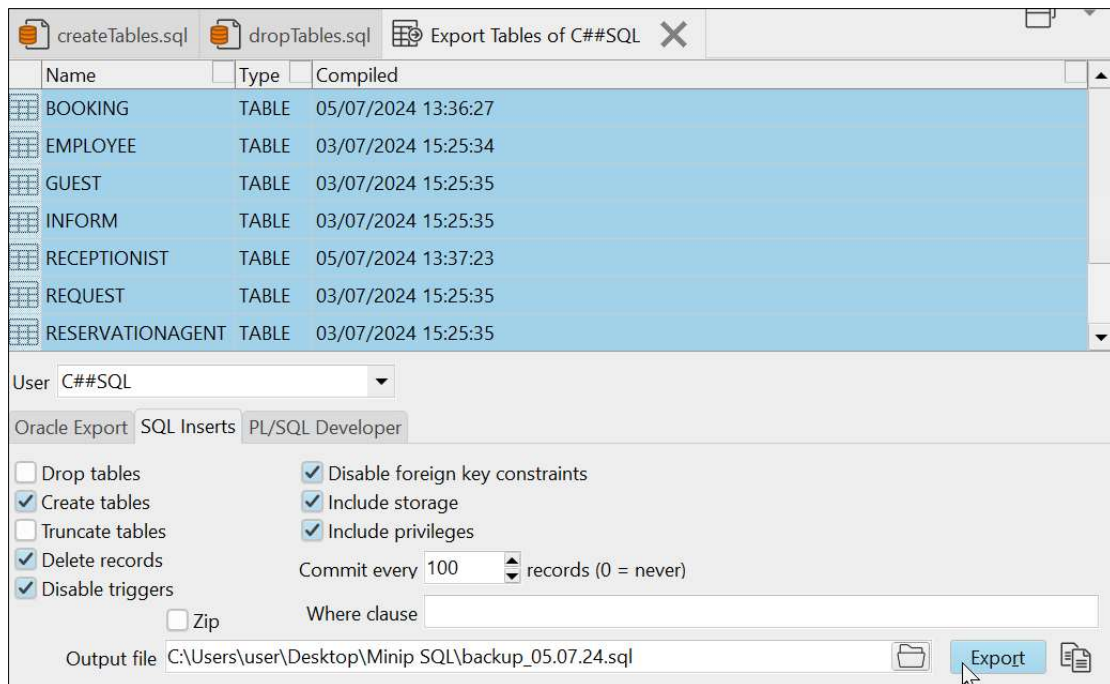
...

Name	Type	Size	Data	Master
EM_ID	NUMBER		List(select em_id from ReservationAgent)	...
GUEST_ID	NUMBER		List(select guest_id from Guest)	...
*				...

## גיבוי ושחזור בסיס הנתונים

### גיבוי בסיס הנתונים:

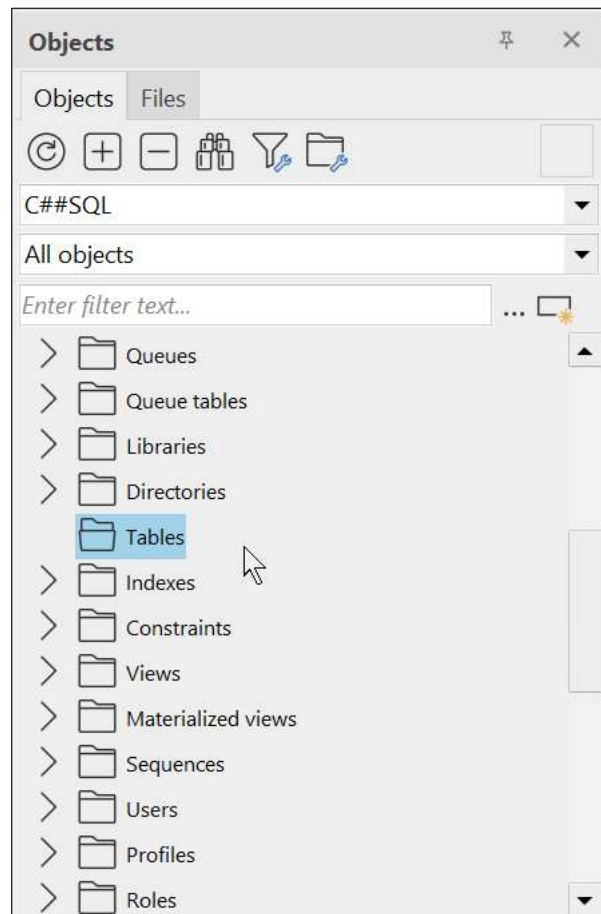
נגבה את כל נתוני יצירת ואכלוס הטבלאות שלנו אל תוך קובץ SQL ייעודי על מנת שנוכל לשחזר את הטבלאות או ליצור אותן על בסיס נתונים אחר מתי שנרצה.



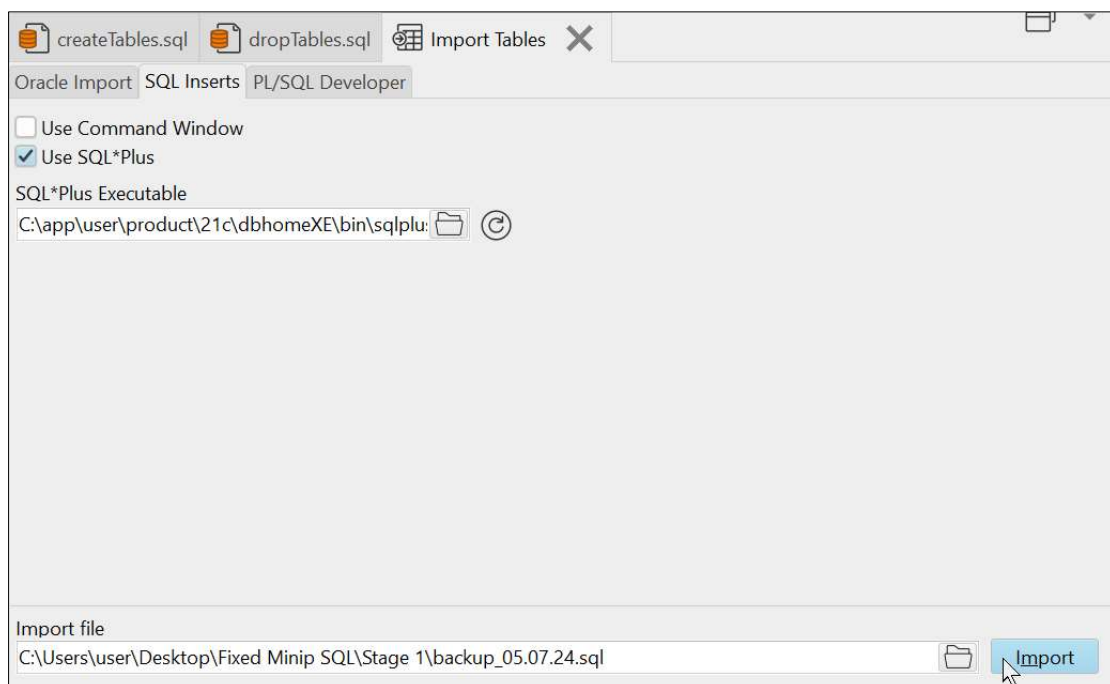
כל הקוד ליצירת בסיס הנתונים שלנו נמצא בקובץ backup\_05.07.2024.sql.

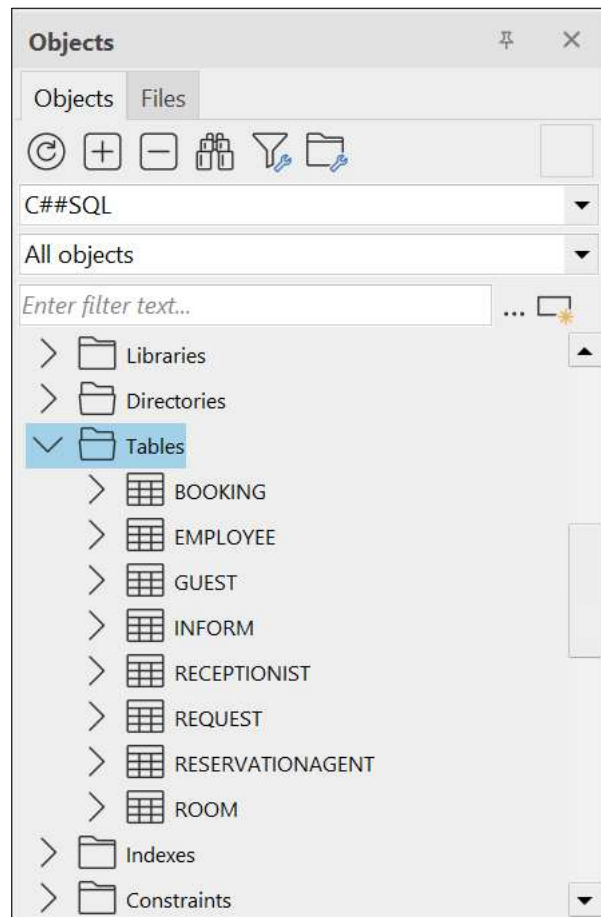
### ייבוא בסיס נתונים:

כאמור, נרצה לייבא את בסיס הנתונים שלנו או כי מחקנו אותו ואנחנו רוצים לשחזר אותו, או אחרי שנחליט לטעון אותו למערכת חדשה.



הרצנו את הקובץ dropTables.sql וכל הטבלאות שלנו באמת נמחקו. נכנס ל-Import Tables ונייבא את הקובץ הנ"ל.





נוכל לראות שאכן כל הטבלאות נוצרו ואוכלונו בנתונים הקודמים.

# שאלות על בסיס הנתונים

## שאלות select:

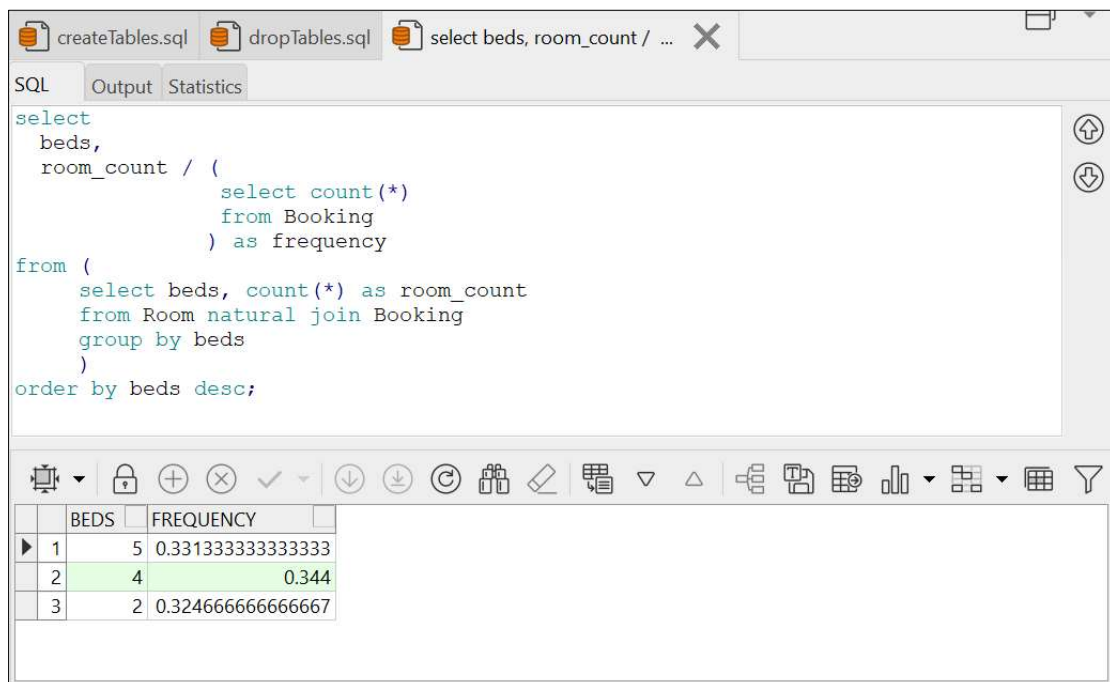
### שאלה 1:

הנהלת בית המלון קיבלה אישור מממשלת ברית הניה לבנות קומה נוספת לבית המלון. על מנת להחליט בצורה נכונה ומדויקת כמה מיטות עליהם להציב בכל חדר, הוחלט ללמוד משנים עברו אילו חדרים היו הכי פופולריים.

לצורך כך נכתוב שאלתה שתמצא לנו כמה אחוז מהמזמינים הזמינו סוג חדר מסוים עבור כל אחד מסוגי החדרים השונים.

```
select
  beds,
  room_count / (
    select count(*)
    from Booking
  ) as frequency
from (
  select beds, count(*) as room_count
  from Room natural join Booking
  group by beds
)
order by beds desc;
```

הרצת השאלתה:



The screenshot shows a SQL IDE window with the following tabs: createTables.sql, dropTables.sql, and select beds, room\_count / ... X. The SQL tab is active, displaying the query. The Output tab shows the results of the query, which is a table with two columns: BEDS and FREQUENCY. The results are ordered by BEDS in descending order.

	BEDS	FREQUENCY
1	5	0.331333333333333
2	4	0.344
3	2	0.324666666666667

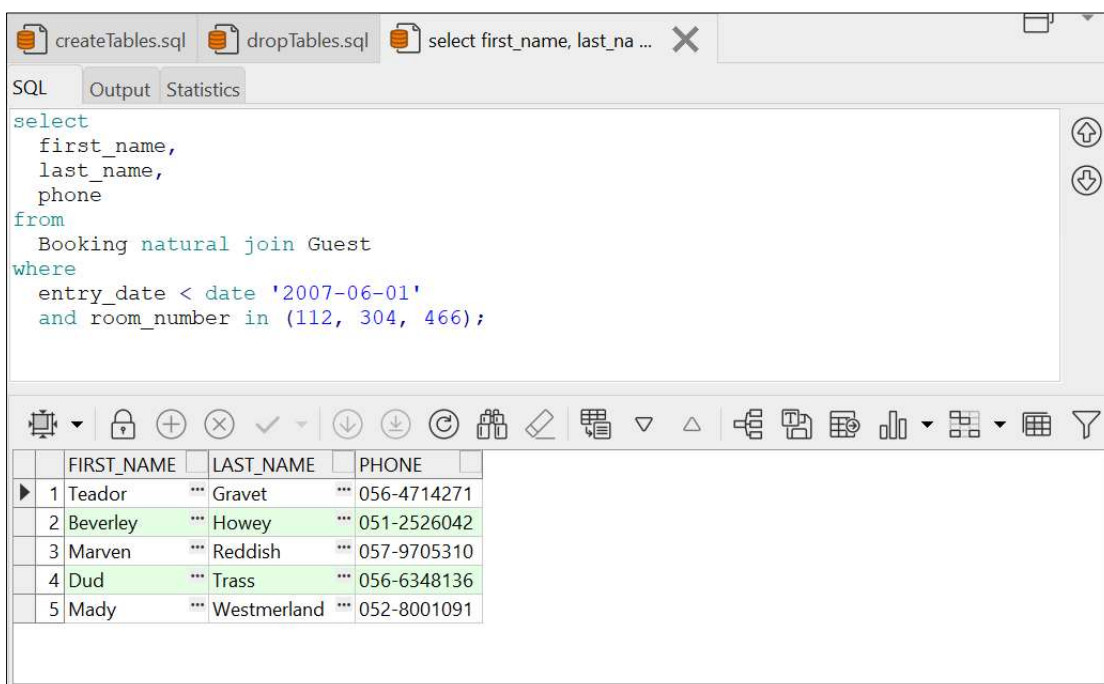
## שאלתה 2:

לאחר ביקורת של משרד התברואה בבית המלון התגלו במספר חדרים במלון חורים זעירים בקיר שבין החדר למרפסת החדר. החורים גרמו לעכברים שהגיעו מבחוץ להיכנס אל תוך החדר, ולהפריע את מנוחתם של האורחים.

הנהלת בית המלון החליטה לפצות את כל האורחים שהיו בחדרים 304, 112, 466, בשובר לסינמה פיצה בנוו פלאנט שנסגרה לפני שנתיים. לצורך כך נכתב שאילתה שתאתר לנו את כל אותם האורחים.

```
select
  first_name,
  last_name,
  phone
from
  Booking natural join Guest
where
  entry_date < date '2007-06-01'
  and room_number in (112, 304, 466);
```

הרצת השאילתה:



	FIRST_NAME	LAST_NAME	PHONE
1	Teador	Gravet	056-4714271
2	Beverley	Howey	051-2526042
3	Marven	Reddish	057-9705310
4	Dud	Trass	056-6348136
5	Mady	Westmerland	052-8001091

## שאלתה 3:

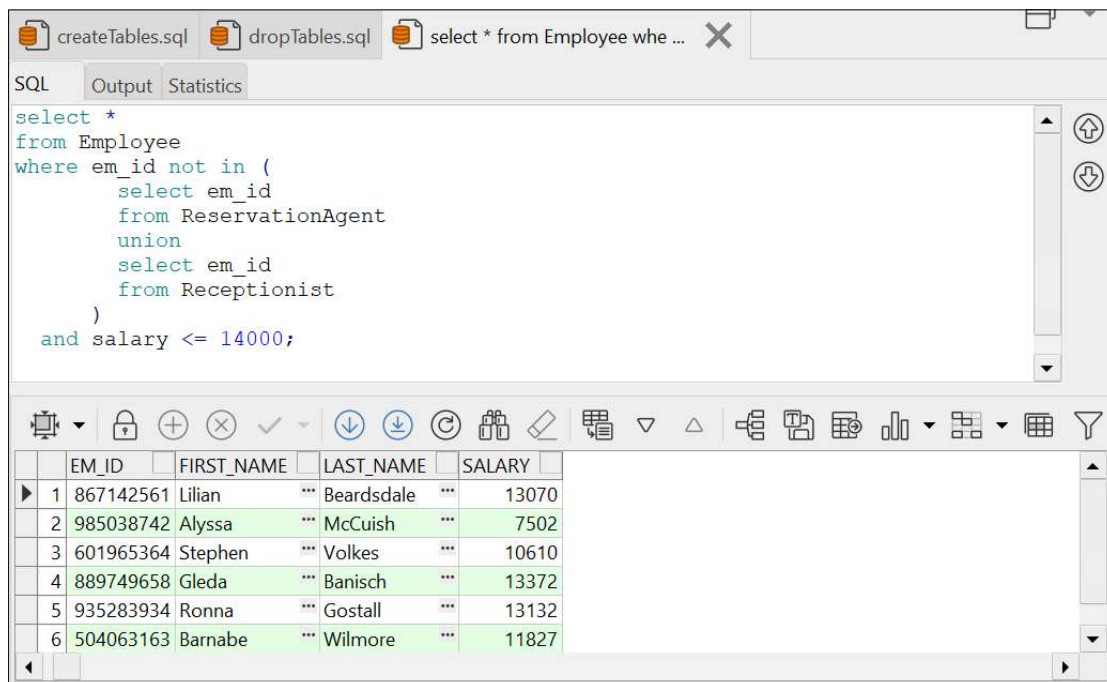
בעקבות התכנון לבניית קומה נוספת לבית המלון נדרשת המון עזרה במטלות השונות הכרוכות במשימה. בית המלון הגיע למסקנה שהעובדים הכי חיוניים בתפקוד התקין של בית המלון הם

הסוכנים, הדלפקאים וצוות ההנהלה העליון של המלון, ולכן הם יתמידו בתפקידם. כל מי שאינו מהמנויים לעיל נקרא לדגל ויסייע בכל מה שצריך.

נכתוב שאילתה שתמצא לנו את כל העובדים שנדרשים להירתם למשימה, בהנחה שהעובדים בצוות ההנהלה העליון מקבלים משכורת חודשים של מעל 14,000.

```
select *
from Employee
where em_id not in (
    select em_id
    from ReservationAgent
    union
    select em_id
    from Receptionist
)
and salary <= 14000;
```

הרצת השאילתה:



	EM_ID	FIRST_NAME	LAST_NAME	SALARY
1	867142561	Lilian	Beardsdale	13070
2	985038742	Alyssa	McCuish	7502
3	601965364	Stephen	Volkes	10610
4	889749658	Gleda	Banisch	13372
5	935283934	Ronna	Gostall	13132
6	504063163	Barnabe	Wilmore	11827

#### שאילתה 4:

בבית המלון החליטו להעניק מתנה לאורחים שבאים לפוש בחגים (למניינם ל"ע). המתנה תהיה בקבוק יין נסך עטוף בצלופן על מצע עור חזיר.

נכתוב שאילתה שתמצא לנו את כל ההזמנות לחג המולד, ראש השנה האזרחי, יום האהבה וחג הפסחא, ותמייין אותם על פי סדר התאריכים.

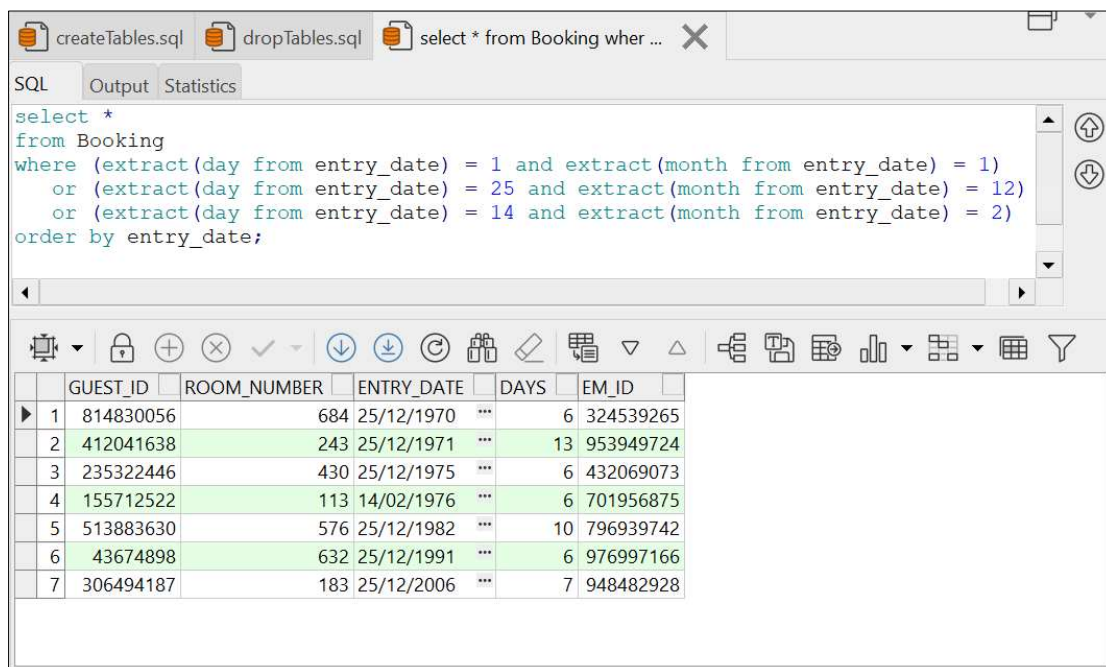
```
select *
from Booking
```

```

where (extract(day from entry_date) = 1 and extract(month from
entry_date) = 1)
    or (extract(day from entry_date) = 25 and extract(month from
entry_date) = 12)
    or (extract(day from entry_date) = 14 and extract(month from
entry_date) = 2)
order by entry_date;

```

הרצת השאילתה:



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL statement:

```

select *
from Booking
where (extract(day from entry_date) = 1 and extract(month from entry_date) = 1)
    or (extract(day from entry_date) = 25 and extract(month from entry_date) = 12)
    or (extract(day from entry_date) = 14 and extract(month from entry_date) = 2)
order by entry_date;

```

The results pane displays a table with 7 rows and 6 columns: GUEST\_ID, ROOM\_NUMBER, ENTRY\_DATE, DAYS, EM\_ID, and an unnamed column. The data is as follows:

	GUEST_ID	ROOM_NUMBER	ENTRY_DATE	DAYS	EM_ID	
1	814830056	684	25/12/1970	...	6	324539265
2	412041638	243	25/12/1971	...	13	953949724
3	235322446	430	25/12/1975	...	6	432069073
4	155712522	113	14/02/1976	...	6	701956875
5	513883630	576	25/12/1982	...	10	796939742
6	43674898	632	25/12/1991	...	6	976997166
7	306494187	183	25/12/2006	...	7	948482928

## שאילתות delete:

### שאילתה 1:

איציק עובד האחזקה טיפל היום בכל הבקשות שהתקבלו מחדר 205. הוא מעוניין להסיר את הבקשות מהמערכת על מנת שלא יטעו לחשוב שהן עוד קיימות.

נכתוב שאילתה שתסיר מטבלת הדיווחים את הרשומות שמיוחסות לחדר 205.

```

delete
from Request
where room_number = 205;

```

לפני הרצת השאילתה:



The screenshot shows a SQL IDE with three tabs: 'createTables.sql', 'dropTables.sql', and 'select \* from Request wher ...'. The 'SQL' tab is active, displaying the query: `select * from Request where room_number = 205;`. Below the query editor is a toolbar with various icons. At the bottom, a table displays the results of the query:

	GUEST_ID	ROOM_NUMBER
1	595423656	205
2	885530745	205

אחרי הרצת השאילתה:

The screenshot shows the same SQL IDE with the same query: `select * from Request where room_number = 205;`. However, the result table below the query is empty, showing only the column headers:

	GUEST_ID	ROOM_NUMBER
--	----------	-------------

## שאילתה 2:

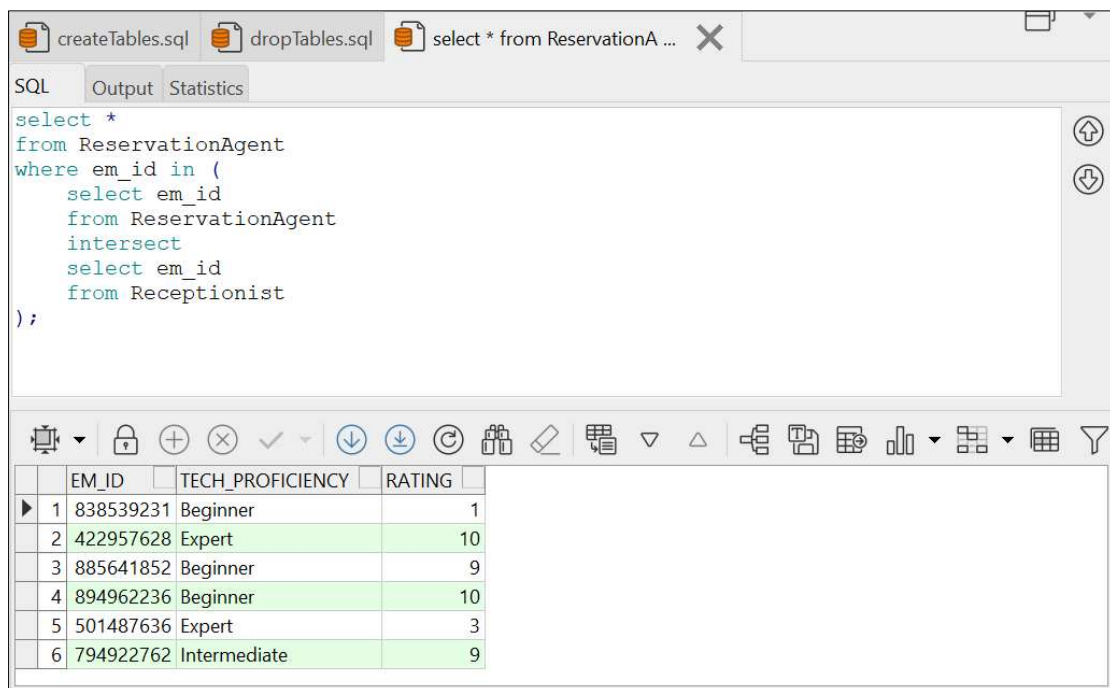
לאחרונה הגיעו מספר עובדים שטענו שהם לא מספיקים להיות עם המשפחה שלהם בערבים והידלה הקטנה בוכה בלילה והכל נופל על האישה. לאחד בירור מעמיק הנהלת בית המלון גילתה שאותם עובדים הם גם סוכנים וגם דלפקאים.

כיוון שהנהלת בית המלון מעודדת שוויון בנטל וגם הבעל צריך לקום באמצע הלילה לילדה, הוחלט פה אחד שכל עובד שהוא גם סוכן וגם דלפקאי מעתה לא יהיה סוכן כי יש יותר מדי כאלה.

נכתוב שאילתה שתמחק מבסיס הנתונים את הכפילויות האלה ותעדכן את הטבלה של הסוכנים.

```
delete
from ReservationAgent
where em_id in (
    select em_id
    from ReservationAgent
    intersect
    select em_id
    from Receptionist
);
```

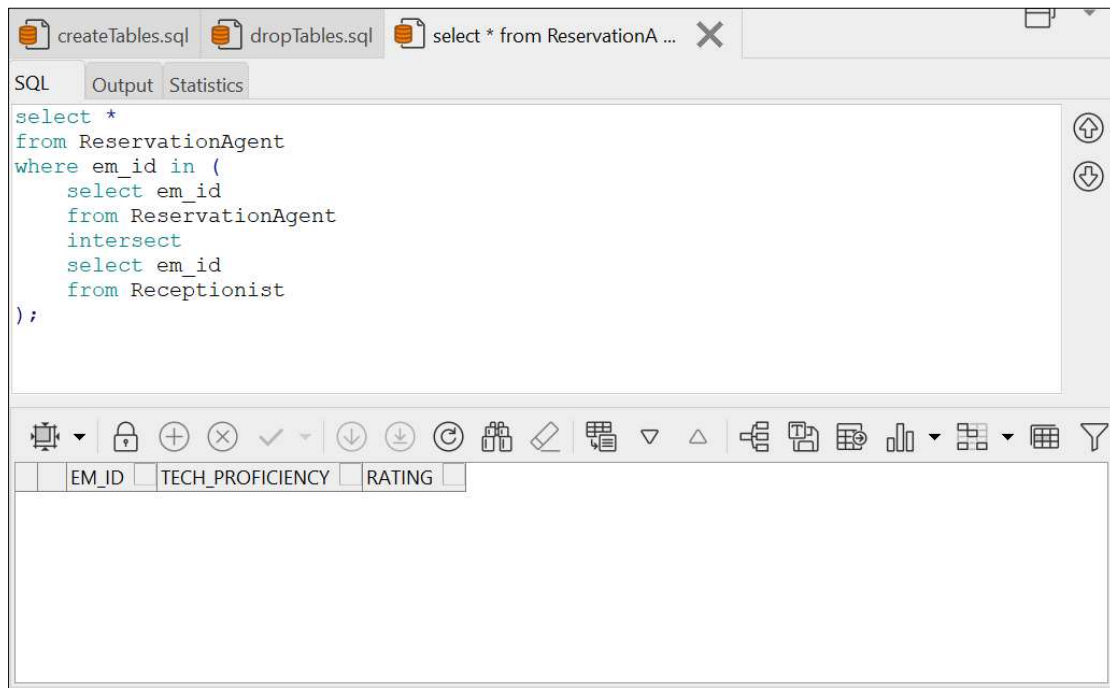
לפני הרצת השאילתה:



The screenshot shows a SQL IDE interface. The top pane displays a SQL query that selects all columns from the ReservationAgent table where the employee ID (em\_id) is in the intersection of the em\_id values from the ReservationAgent and Receptionist tables. The bottom pane shows the results of this query as a table with 6 rows and 4 columns: EM\_ID, TECH\_PROFICIENCY, and RATING. The first row is highlighted with a mouse cursor.

	EM_ID	TECH_PROFICIENCY	RATING
1	838539231	Beginner	1
2	422957628	Expert	10
3	885641852	Beginner	9
4	894962236	Beginner	10
5	501487636	Expert	3
6	794922762	Intermediate	9

אחרי הרצת השאילתה:



## שאלות update:

### שאלתה 1:

גברת קרש ומר מערוך הזמינו חדר בבית המלון לתאריך 01/09/2024. לאחר ביקור אצל רופא המשפחה נודע למר מערוך שהוא סובל מבצקת חמורה בכל חלקי מערכו, כך שיצטרך לבצע ניתוח להסרת בצק בלייזר. לרוע מזלם הניתוח וזמן ההחלמה אחריו נופלים על זמן החופשה שקבעו.

גברת קרש פנתה ליוסי הדלפקאי שמטפל בהם בבקשה לדחות את מועד ההזמנה בשבוע ימים. שם הבמה של גברת קרש הוא Trixie Punter ועליו רשמה את הזמנתה.

נכתוב שאלתה שתעדכן את התאריך לתאריך חדש שיתאים לאבי המשפחה המחלים.

```
update Booking
set entry_date = date '2024-09-30'
where guest_id = (
  select guest_id
  from Guest
  where first_name = 'Trixie' and last_name = 'Punter'
)
and entry_date > date '2024-07-07';
```

לפני הרצת השאלתה:

createTables.sql dropTables.sql select \* from Booking wher ... X

SQL Output Statistics

```

select *
from Booking
where guest_id = (
    select guest_id
    from Guest
    where first_name = 'Trixie' and last_name = 'Punter'
)
and entry_date > date '2024-07-07';

```

	GUEST_ID	ROOM_NUMBER	ENTRY_DATE	DAYS	EM_ID
▶ 1	315806874	223	01/09/2024	...	13 440327900

אחרי הרצת השאילתה:

createTables.sql dropTables.sql select \* from Booking --se ... X

SQL Output Statistics

```

select *
from Booking
where guest_id = (
    select guest_id
    from Guest
    where first_name = 'Trixie' and last_name = 'Punter'
)
and entry_date > date '2024-07-07';

```

	GUEST_ID	ROOM_NUMBER	ENTRY_DATE	DAYS	EM_ID
▶ 1	315806874	223	30/09/2024	...	13 440327900

## שאילתה 2:

לכבוד יום העובד החרוץ החליטה הנהלת בית המלון להעלות את שכרם החודשי של סוכנים אשר מספר האורחים שיצרו איתם קשר לבירור עולה על שלושה. הם יכבלו תוספת קבועה של 1000 במשכורתם החודשית.

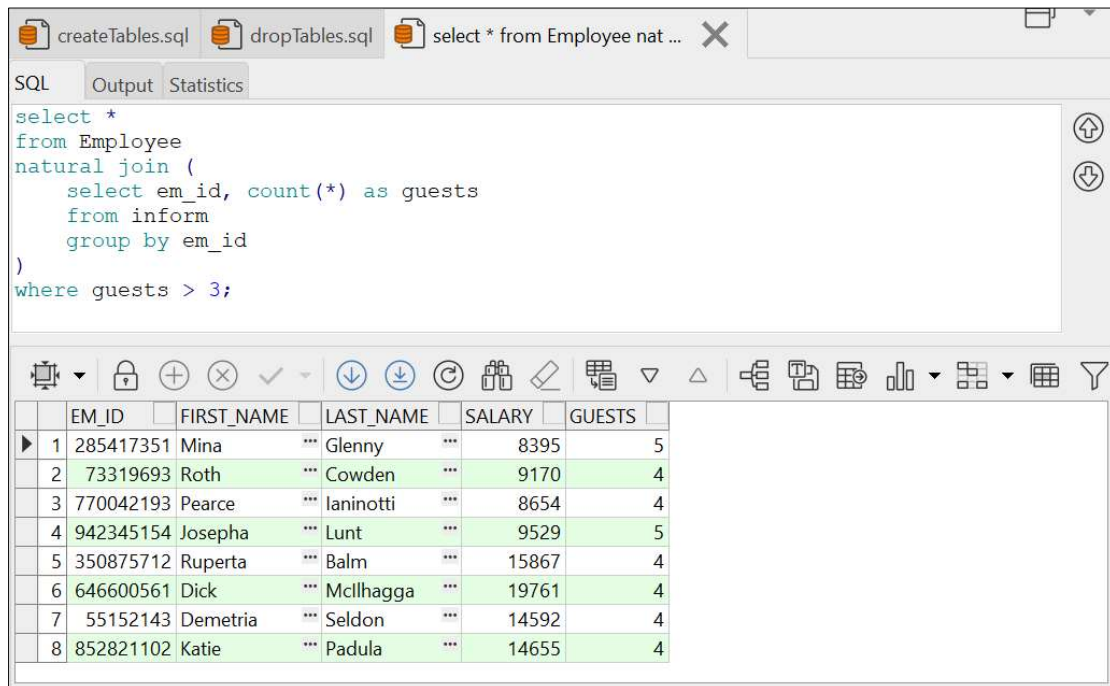
נכתוב שאילתה שתמצא לנו את אותם סוכנים ותעדכן את משכורתם בהתאם.

```

update Employee
set salary = salary + 1000
where em_id in (
    select em_id
    from (
        select em_id, count(*) as guests
        from inform
        group by em_id
    )
    where guests > 3
);

```

לפני הרצת השאילתה:



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```

select *
from Employee
natural join (
    select em_id, count(*) as guests
    from inform
    group by em_id
)
where guests > 3;

```

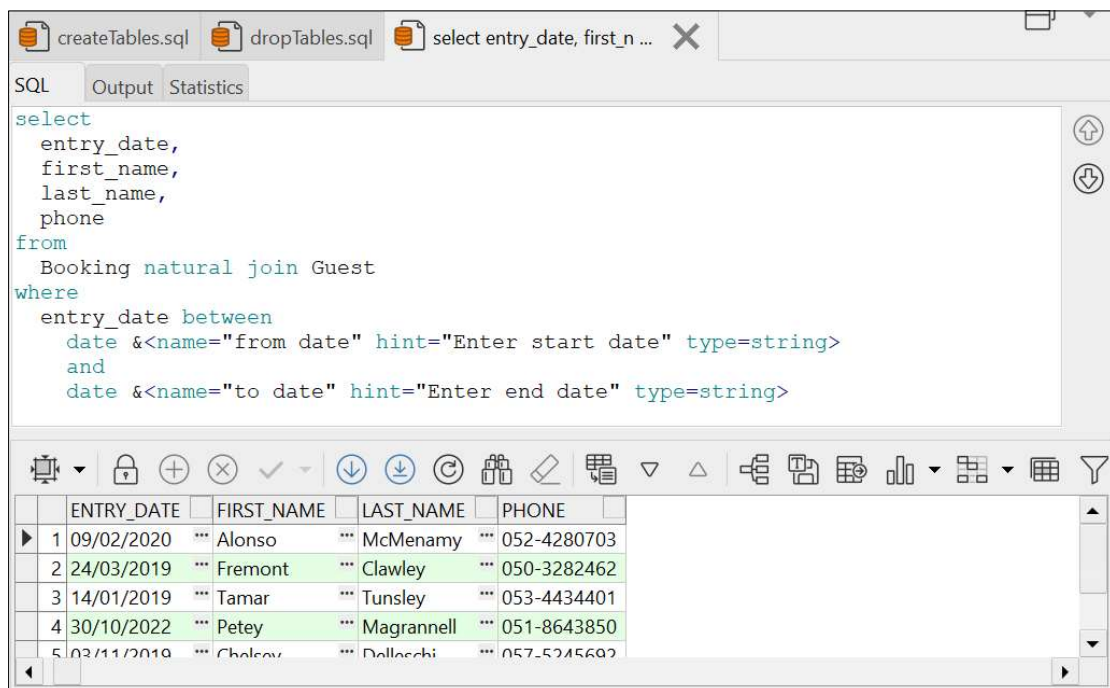
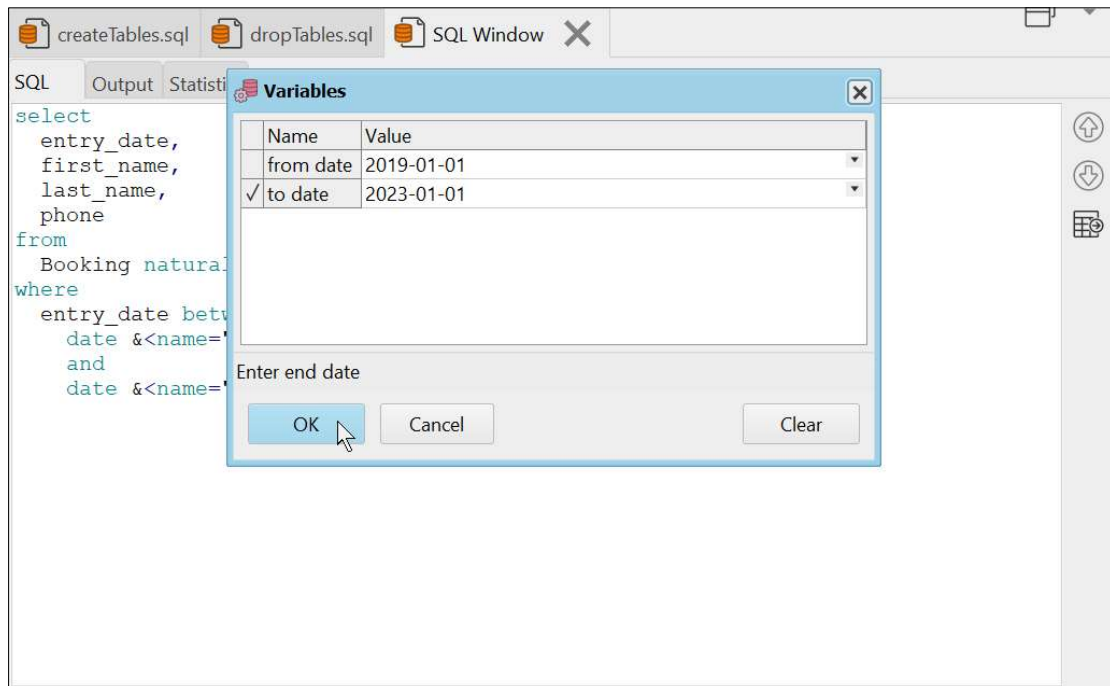
The results pane displays a table with 8 rows and 7 columns: EM\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and GUESTS. The first four columns are highlighted in green for each row.

	EM_ID	FIRST_NAME	LAST_NAME	SALARY	GUESTS
1	285417351	Mina	Glenny	8395	5
2	73319693	Roth	Cowden	9170	4
3	770042193	Pearce	Ianinotti	8654	4
4	942345154	Josepha	Lunt	9529	5
5	350875712	Ruperta	Balm	15867	4
6	646600561	Dick	McIlhagga	19761	4
7	55152143	Demetria	Seldon	14592	4
8	852821102	Katie	Padula	14655	4

אחרי הרצת השאילתה:



```
date &<name="to date" hint="Enter end date" type=string>
```

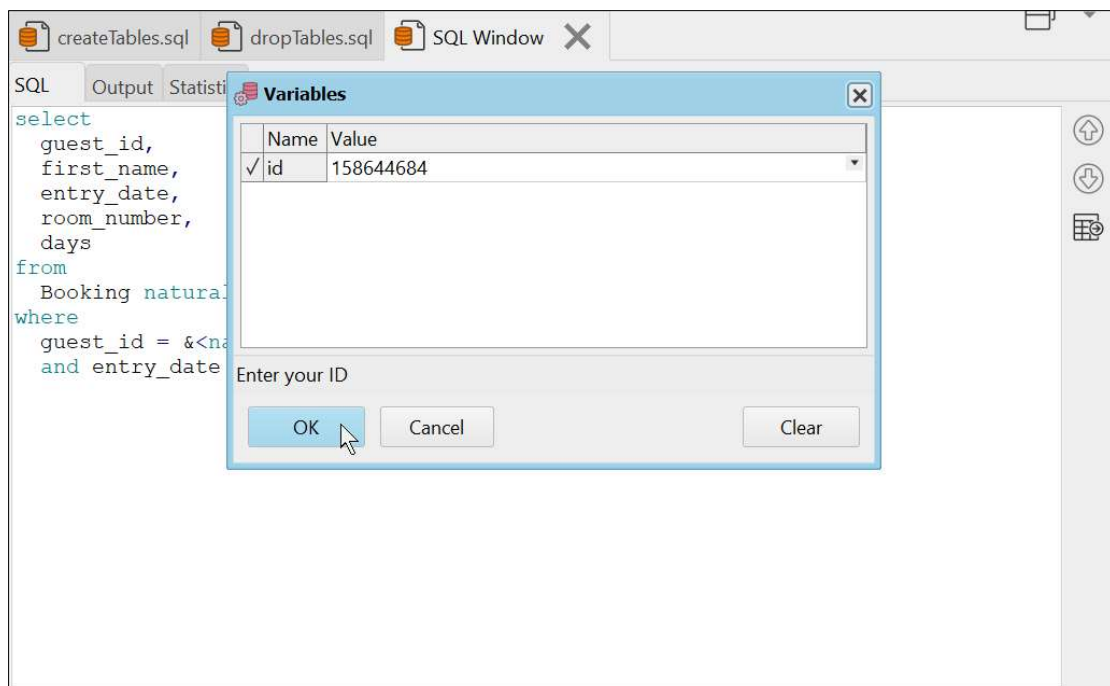


## שאלת 2:

מבין האורחים של בית המלון ישנם כאלו מאוד מפוזרים, ברמה כזאת שהם עלולים לשכוח למתי הזמינו חדר בבית המלון, אם בכלל. כדי למנוע עוגמת נפש, בית המלון החליט לפתוח ללקוחות שלו את האפשרות להתעדכן שוב בהזמנות העתידיות שלהם לבית המלון.

נכתוב שאילתה שתקבל מהמשתמש את תעודת הזהות שלו (המזמין) ותחזיר לו את פירוט ההזמנות העתידיות שלו.

```
select
  guest_id,
  first_name,
  entry_date,
  room_number,
  days
from
  Booking natural join Guest
where
  guest_id = &<name=id hint="Enter your ID">
  and entry_date > date '2024-01-01';
```





The screenshot shows a SQL IDE with three tabs: 'createTables.sql', 'dropTables.sql', and 'select guest\_id, first\_name ...'. The 'SQL' tab is active, displaying the following query:

```
select
  guest_id,
  first_name,
  entry_date,
  room_number,
  days
from
  Booking natural join Guest
where
  guest_id = &<name=id hint="Enter your ID">
  and entry_date > date '2024-01-01';
```

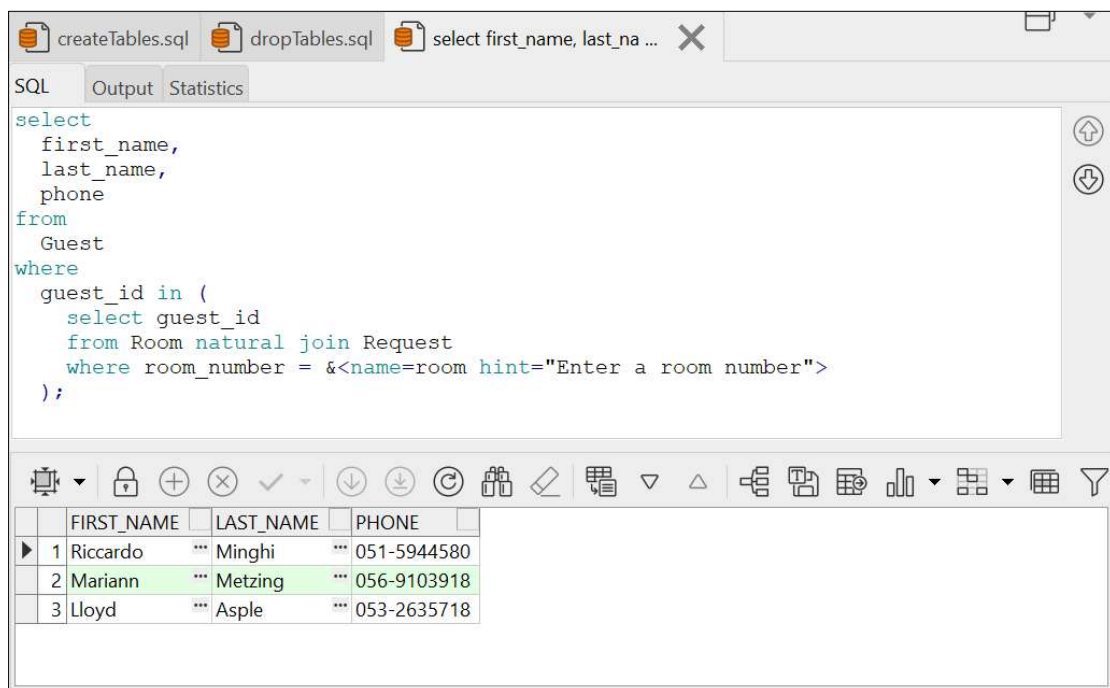
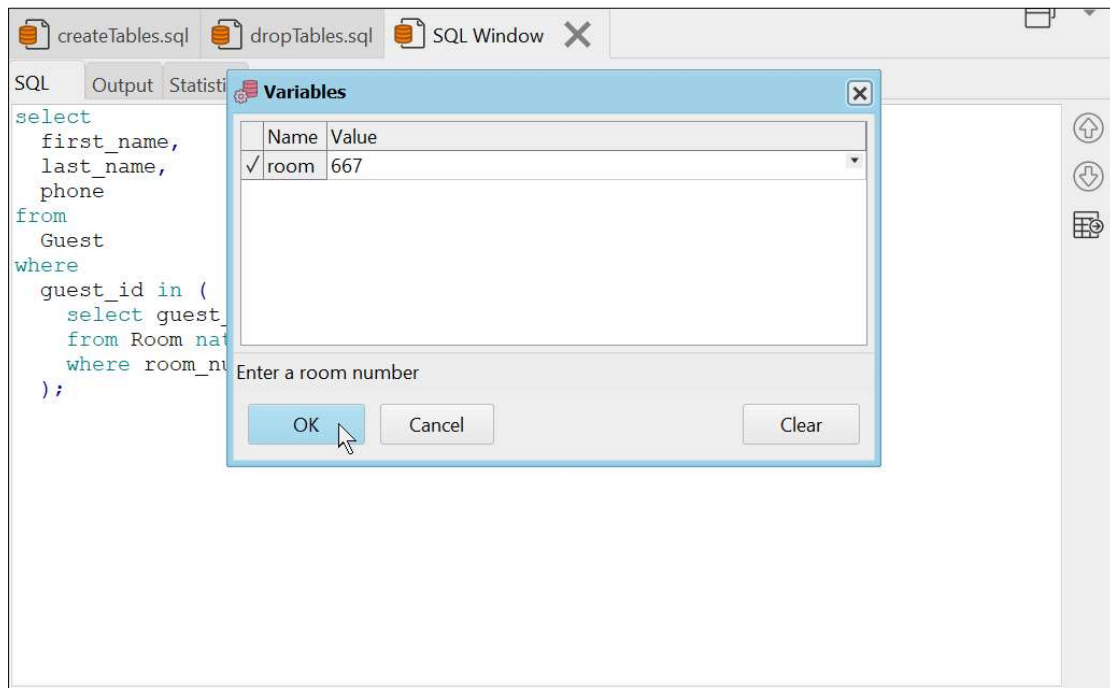
Below the query editor is a toolbar with various icons. At the bottom, a table displays the results of the query:

	GUEST_ID	FIRST_NAME	ENTRY_DATE	ROOM_NUMBER	DAYS
▶ 1	158644684	Marven	16/06/2024	592	13

### שאלתה 3:

עובדי התחזוקה של בית המלון עובדים קשה מסביב לשעון כדי לטפל בכל תקלה בהקדם האפשרי. לאורחי המלון יש את האפשרות לדווח על תקלות בחדרי המלון, כדי שצוות התחזוקה ידע על קיומן. עובד ניקיון יוכל אם ירצה לשאול עבור חדר מסוים מה התקלות שדווחו לגביו. כדי לקבל תשובה על שאלה זו יידרש העובד לפרטי הקשר של המדווחים כדי לשמוע על התקלה ממקור ראשון. נכתוב שאלתה עבור עובד בית מלון שיספק שם של חדר ויקבל את הפרטים של האנשים שדיווחו על אותו חדר.

```
select
  first_name,
  last_name,
  phone
from
  Guest
where
  guest_id in (
    select guest_id
    from Room natural join Request
    where room_number = &<name=room hint="Enter a room number">
  );
```



#### שאלתה 4:

כאשר לקוח נכנס לאתר ההזמנות של בית המלון, עליו לבחור חדר. הלקוח יודע כמה מיטות הוא מעוניין שיהיה בחדר, אך הוא לא יודע אילו חדרים עונים על הדרישה שלו.

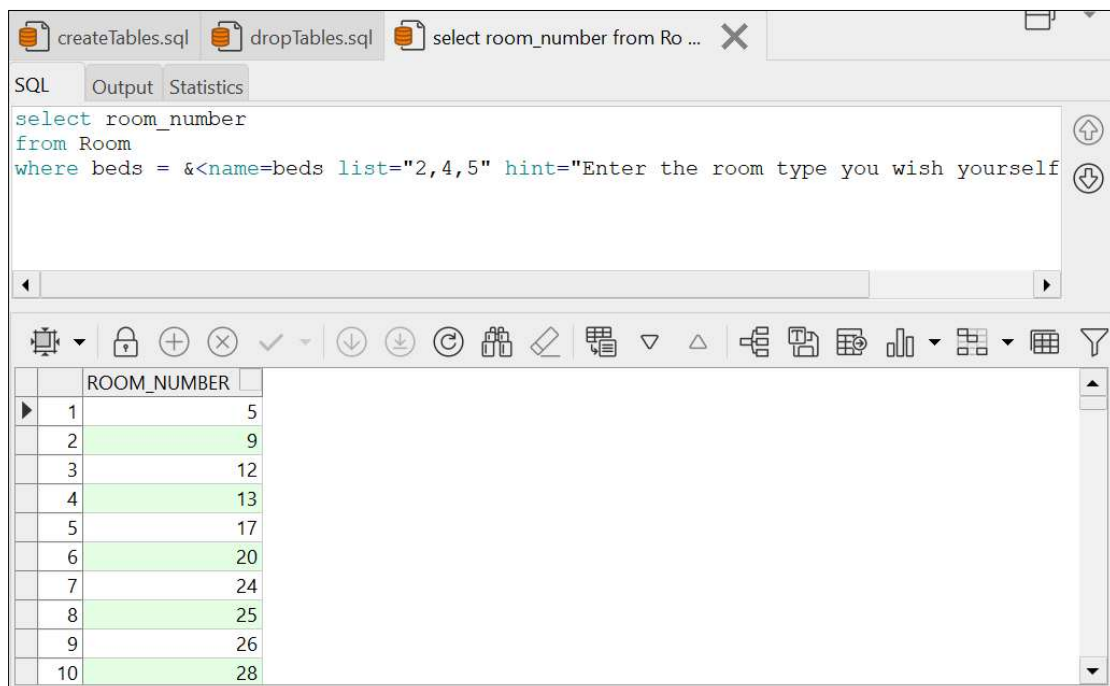
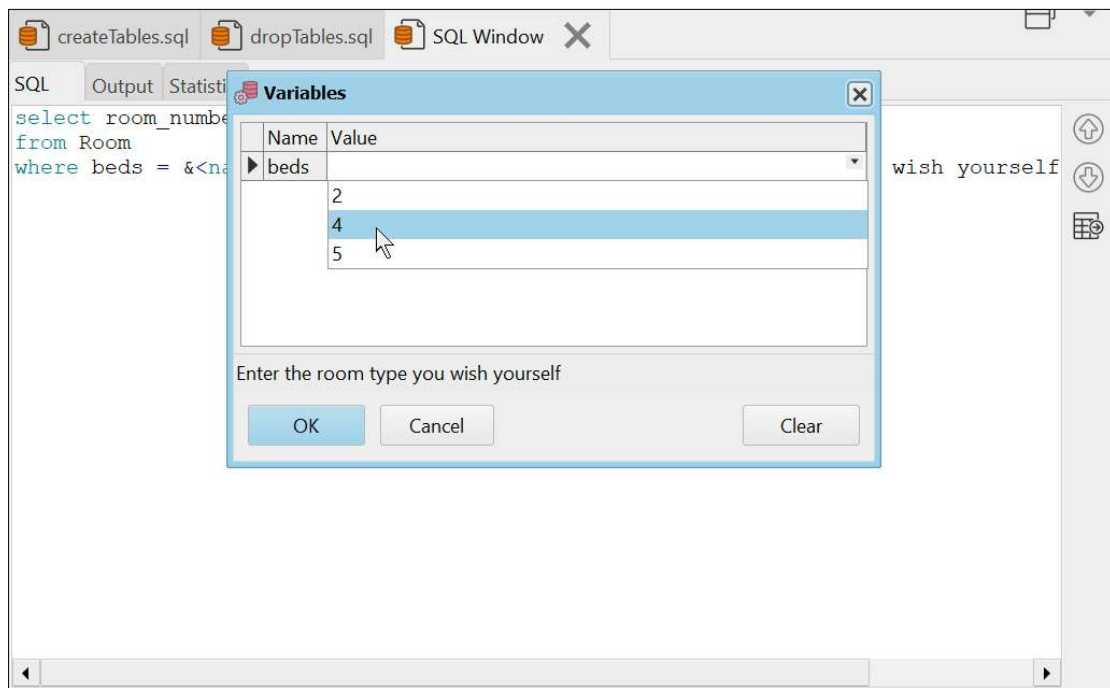
נכתוב שאילה שבה המזמין יבחר את סוג החדר שירצה (לפי מספר המיטות) מתור רשימה, ויקבל בחזרה את רשימת כל החדרים במלון מסוג זה.

```
select room_number
```

```

from Room
where beds = &<name=beds list="2,4,5" hint="Enter the room type you
wish yourself" type=integer>;

```



## הוספת אילוצים לטבלאות:

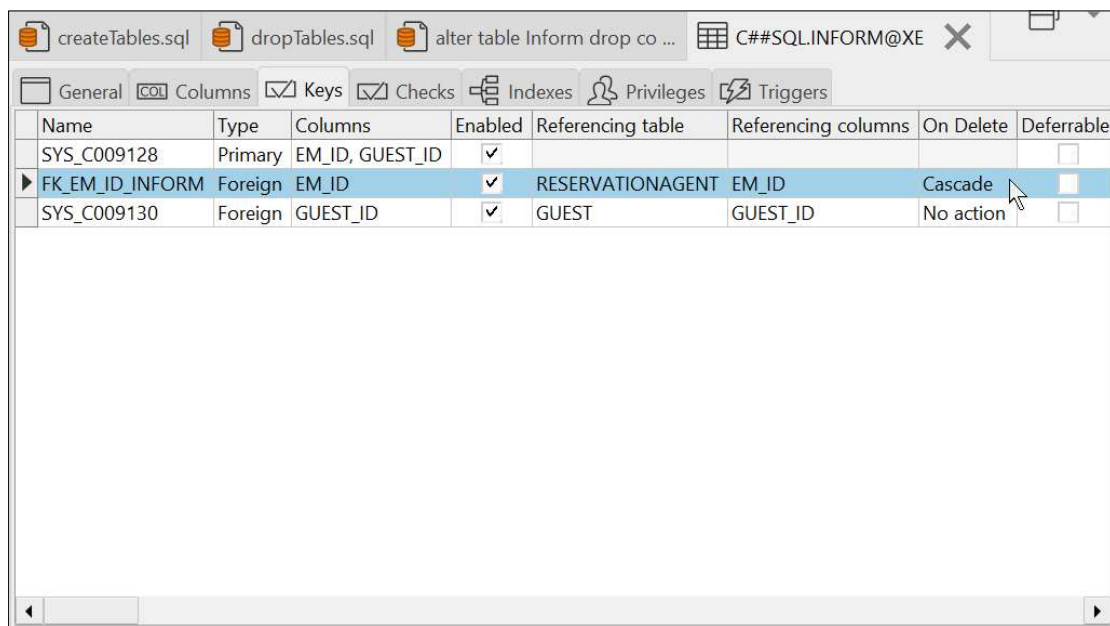
### אילוץ ראשון:

בשאלת המחיקה השנייה ניסינו למחוק מספר סוכנים וזה לא התאפשר כיוון שהם היו משויכים לרשומות Inform מסוימות. נסיף בטבלת Inform אילוץ on delete cascade לשדה em\_id, ואז כשנמחק סוכן מסוים – כל הרשומות המשייכות אליו תמחקנה גם.

```
alter table Inform
drop constraint fk_em_id_inform;

alter table Inform
add constraint fk_em_id_inform
foreign key (em_id) references ReservationAgent (em_id) on delete
cascade;
```

נראה בנתוני הטבלה Inform שבאמת נוסף האילוץ שרצינו:



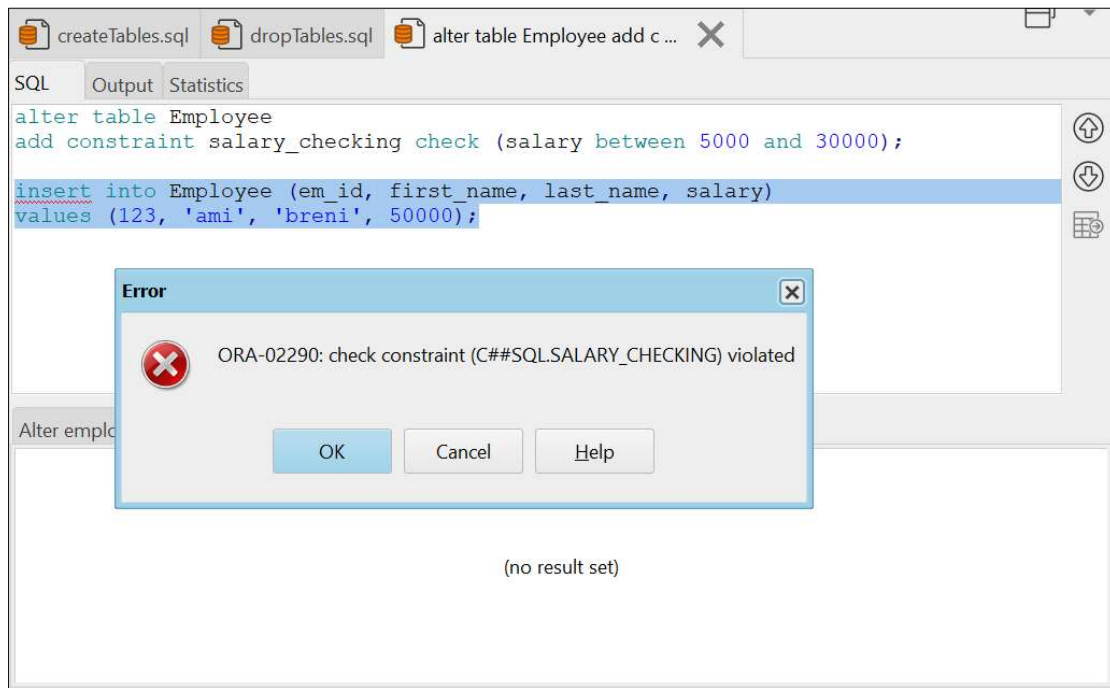
Name	Type	Columns	Enabled	Referencing table	Referencing columns	On Delete	Deferrable
SYS_C009128	Primary	EM_ID, GUEST_ID	<input checked="" type="checkbox"/>				<input type="checkbox"/>
FK_EM_ID_INFORM	Foreign	EM_ID	<input checked="" type="checkbox"/>	RESERVATIONAGENT	EM_ID	Cascade	<input type="checkbox"/>
SYS_C009130	Foreign	GUEST_ID	<input checked="" type="checkbox"/>	GUEST	GUEST_ID	No action	<input type="checkbox"/>

### אילוץ שני:

נוסיף אילוץ על הערך של salary בטבלה Employee כדי לוודא שהמשכורת שהוזנה נמצאת בטווח התקין.

```
alter table Employee
add constraint salary_check check (salary between 5000 and 30000);
```

כעת נראה שבאמת לא ניתן להוסיף רשומה שחורגת מהמגבלות:



## אילוח שלישי:

נוסיף אילוח לשדה lang שבטבלה Receptionist על מנת שערך ברירת המחדל של השפה בה יתקשר המארח תהיה אנגלית.

```
alter table Receptionist
modify lang default 'English';
```

נראה שהשאילתה הצליחה:

