

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:

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 שידע 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.

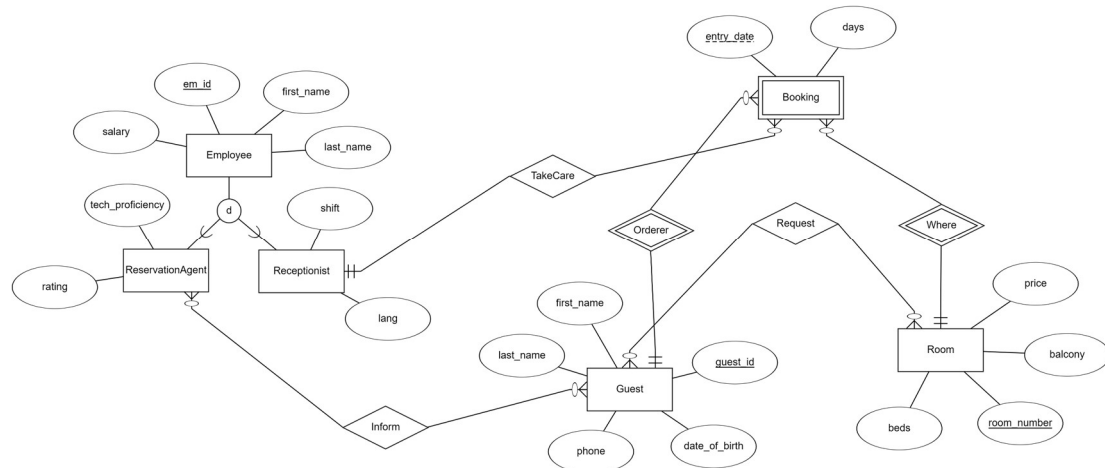
- **סוכן (ReservationAgent):** טבלת הסוכנים תכיל את כל הסוכנים של בית המלון.
 - em_id – מספר העובד.
 - rating – דירוג שקיבל הסוכן מהלקוחות.
 - tech_proficiency – התמחות טכנולוגית (רמה גבוהה, ממוצעת, נמוכה).
- **מארח (Receptionist):** טבלת המארחים תכיל את כל המארחים של בית המלון.
 - em_id – מספר העובד.
 - 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:

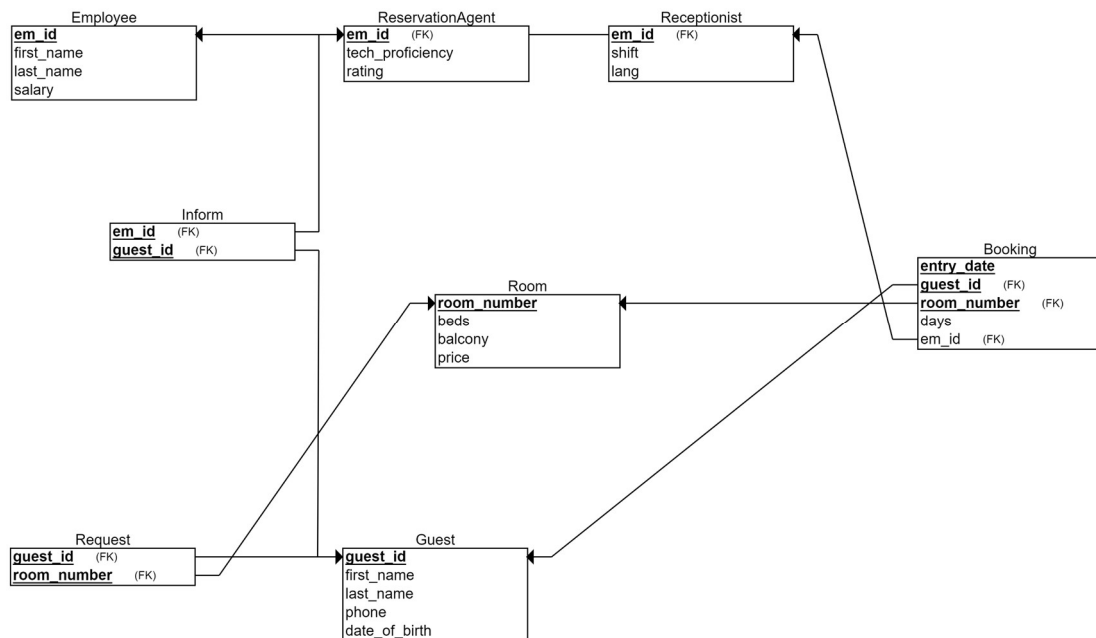
- קשר **Inform**: לא כל האורחים שרוצים להזמין חדרים במלון בקיאים בסוד העניינים. יש כאלו שזו פעם ראשונה שהם מזמינים בעצמם חדר/ים בבית מלון. לצורך כך המלון מעסיק סוכנים שתפקידם לסייע לאותם אנשים לבצע את ההזמנה עליה הם חולמים.
- קשר **TakeCare**: כאשר מתקבלת הזמנה במערכת, היא מועברת לטיפול אחד מהמארחים 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):

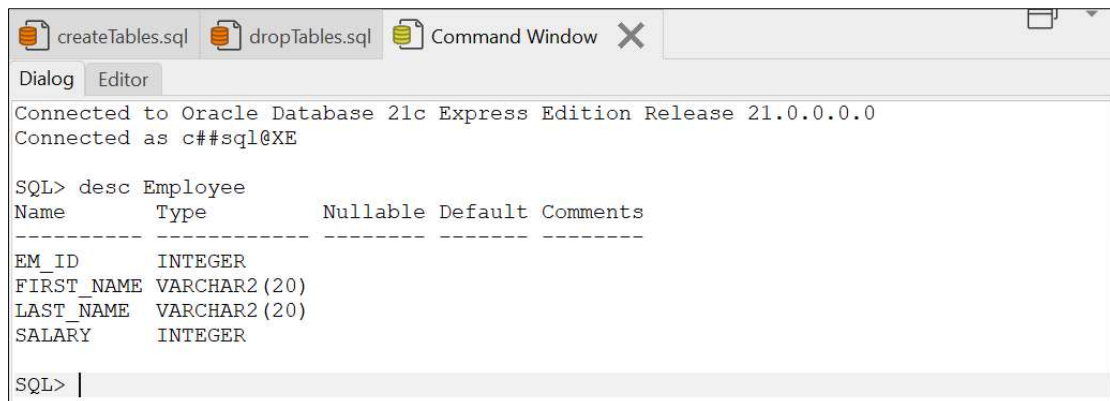
Employee, ReservationAgent, Receptionist, Room, Guest, Booking, Request, Inform

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:

The screenshot shows the SQL Developer interface with a query window titled 'Query data C##SQL.EMPLOYEE@XE'. The SQL tab is active, displaying the query: `select * from EMPLOYEE t`. Below the query editor, a toolbar contains various icons for database operations. The results pane shows a table with the following data:

| | EM_ID | FIRST_NAME | LAST_NAME | SALARY |
|---|-------|------------|-----------|--------|
| 1 | 1 | John | Doe | 50000 |
| 2 | 2 | Jane | Smith | 55000 |
| 3 | 3 | Alice | Johnson | 60000 |

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:

The screenshot shows the SQL Developer Command Window with the following text:

```
Connected to Oracle Database 21c Express Edition Release 21.0.0.0.0
Connected as c##sql@XE

SQL> desc ReservationAgent
Name                               Type                Nullable Default Comments
-----
EM_ID                             INTEGER              YES
TECH_PROFICIENCY                  VARCHAR2(15)         YES
RATING                            INTEGER              YES
```

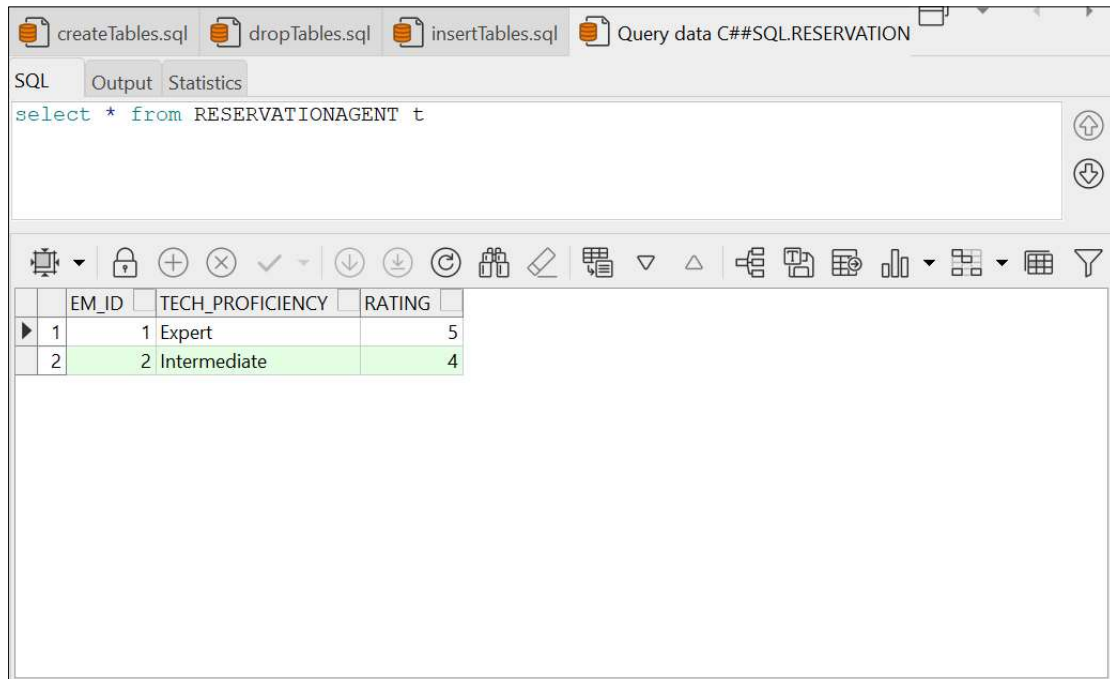
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```
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:



The screenshot shows the SQL Developer interface. The top tab is 'Query data C##SQL.RESERVATION'. The SQL editor contains the query: `select * from RESERVATIONAGENT t`. The 'Output' tab is selected, displaying a table with the following data:

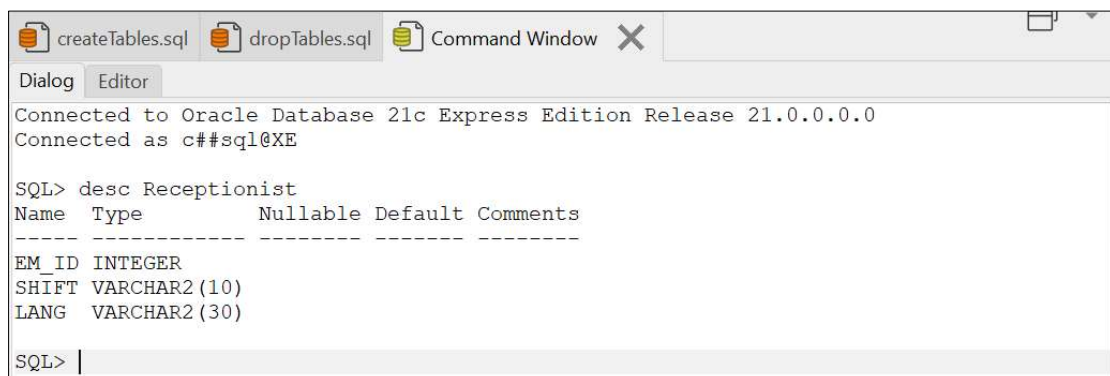
| | EM_ID | TECH_PROFICIENCY | RATING |
|---|-------|------------------|--------|
| 1 | 1 | Expert | 5 |
| 2 | 2 | Intermediate | 4 |

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:



The screenshot shows the SQL Developer Command Window. It is connected to an Oracle Database 21c Express Edition. The command `SQL> desc Receptionist` has been executed, and the output is as follows:

| Name | Type | Nullable | Default | Comments |
|-------|--------------|----------|---------|----------|
| EM_ID | INTEGER | | | |
| SHIFT | VARCHAR2(10) | | | |
| LANG | VARCHAR2(30) | | | |

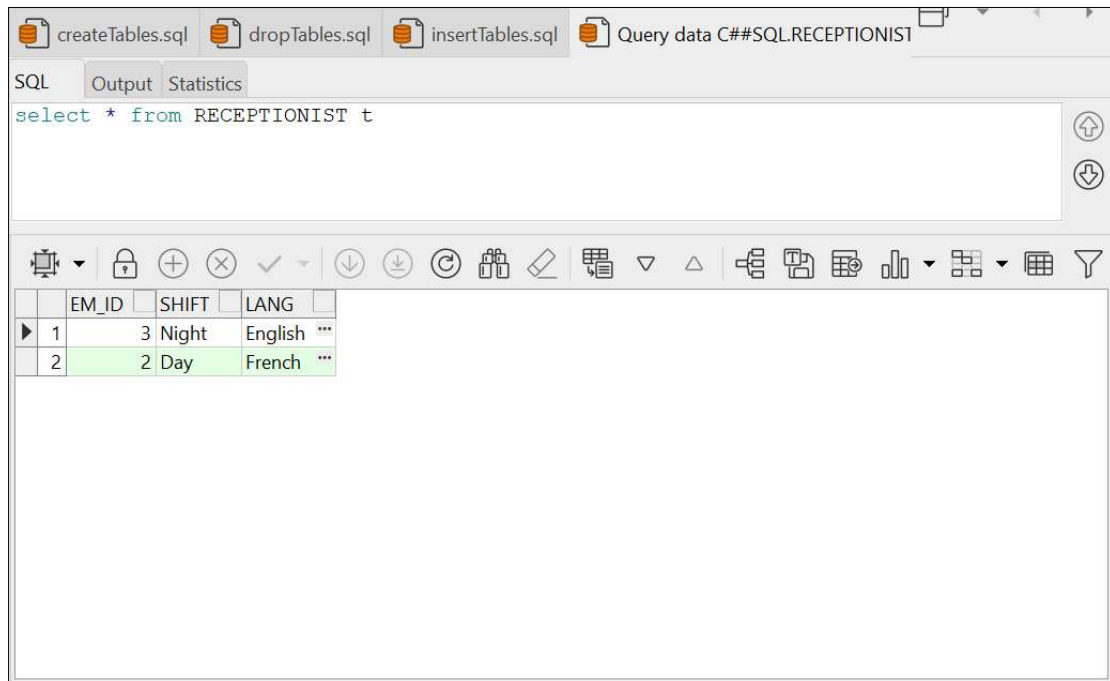
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```

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:

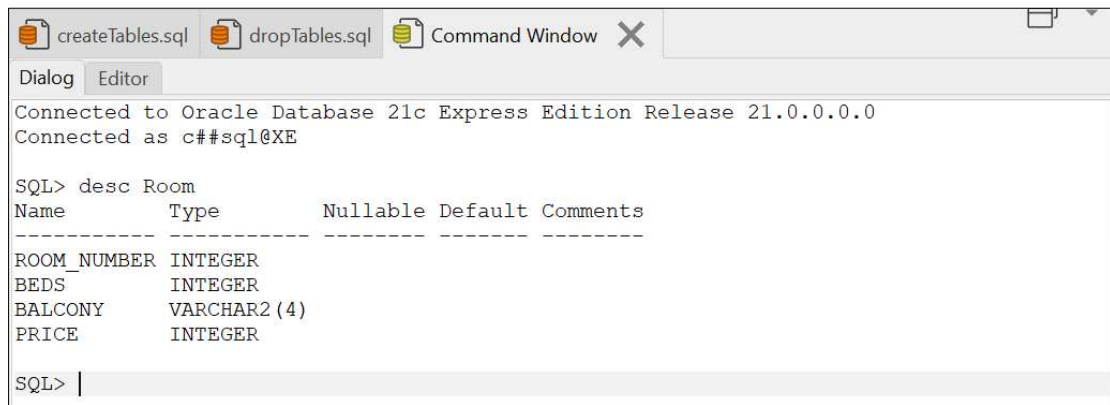
קוד ה-SQL שיוצר לנו את הטבלה הוא:

```

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)
);

```

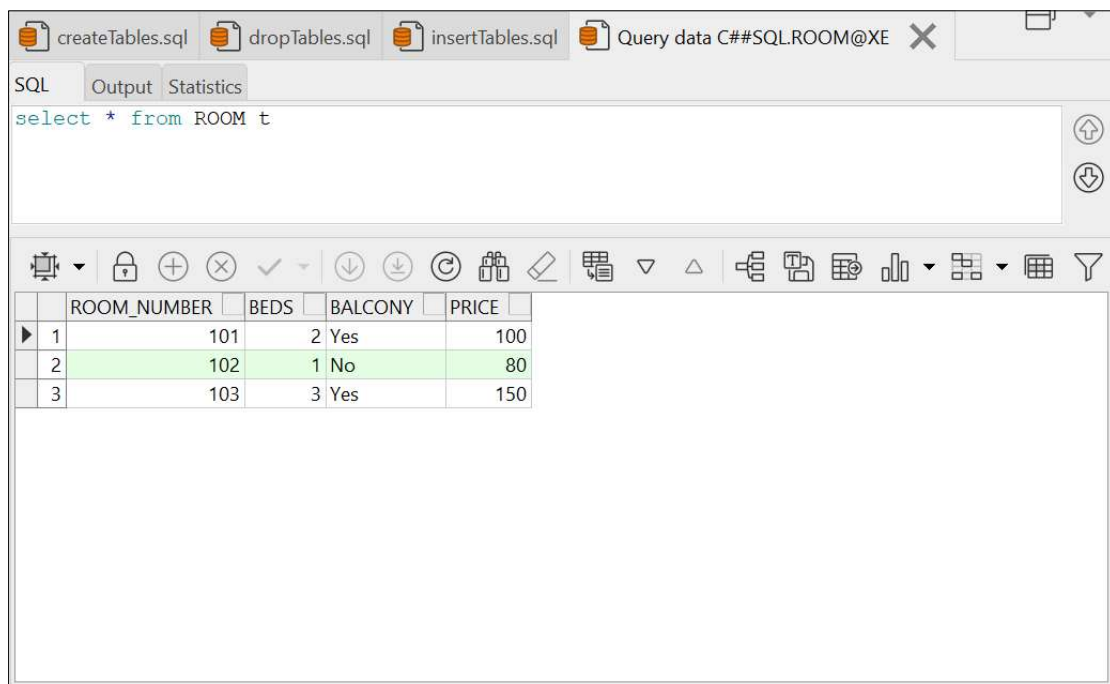
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



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:

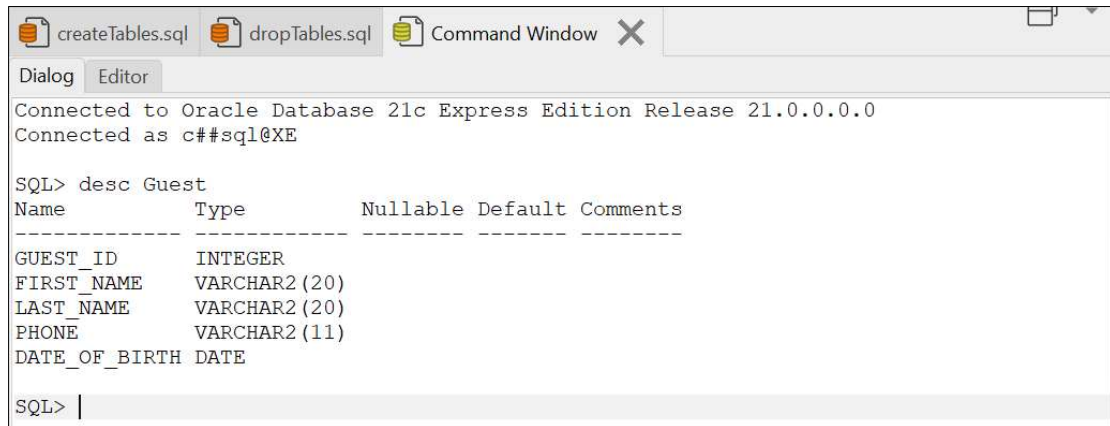
```
CREATE TABLE Guest
(
    guest_id INT NOT NULL,
    first_name VARCHAR2(20) NOT NULL,
```

```

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:



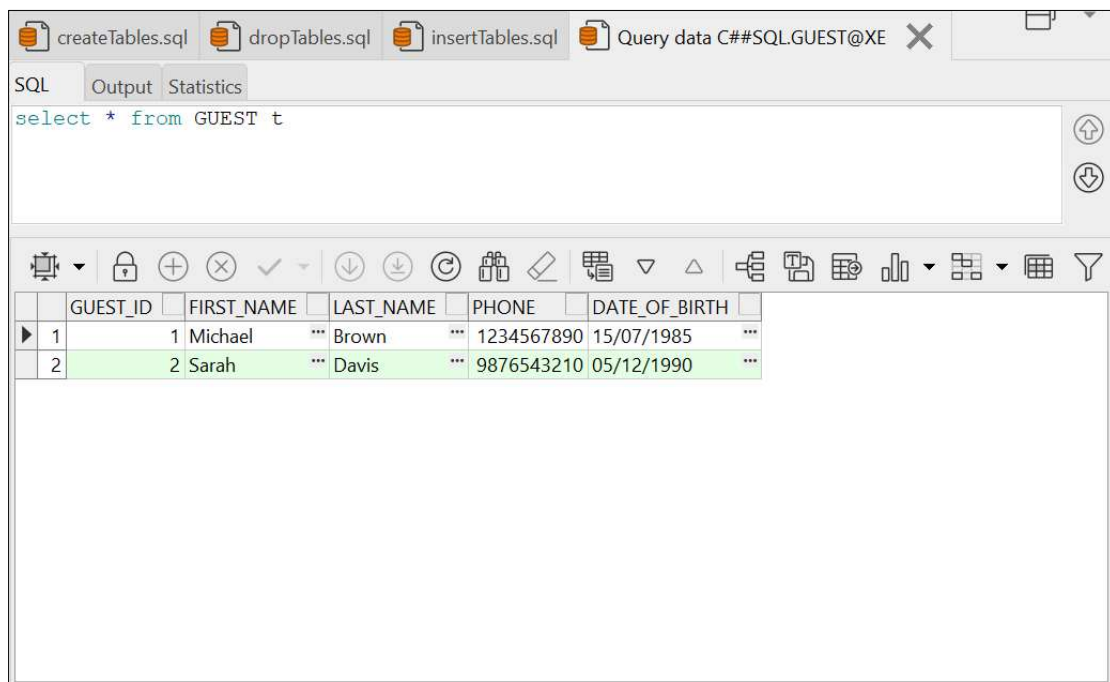
נכניס לטבלה מספר ערכים על מנת לבדוק את תקינותה של הסכמה:

```

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:

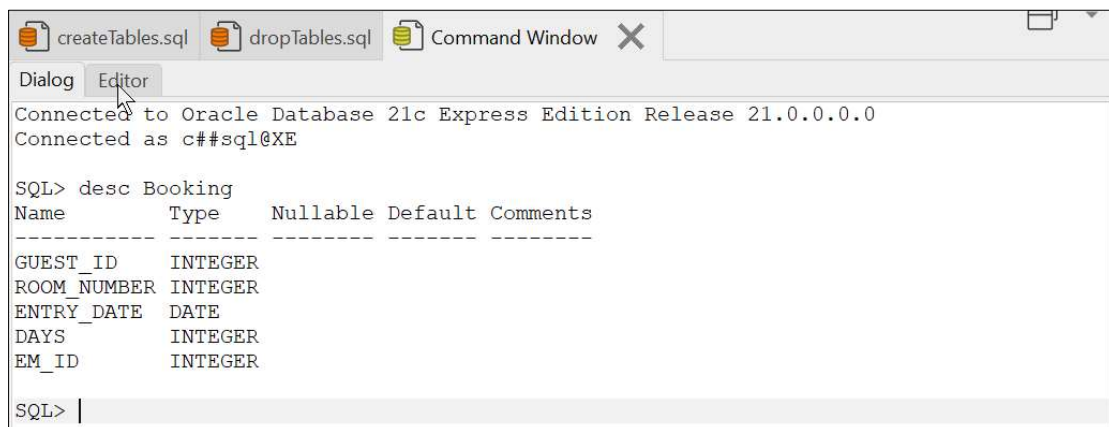


יצירת הטבלה Booking:

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)
);
```

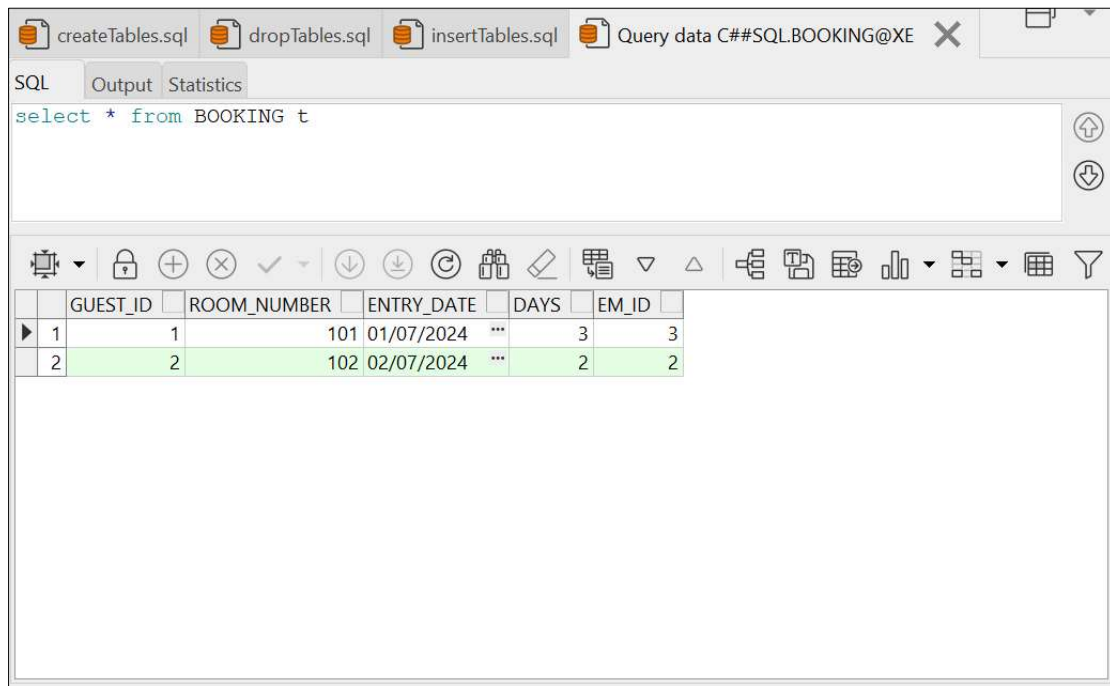
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



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:

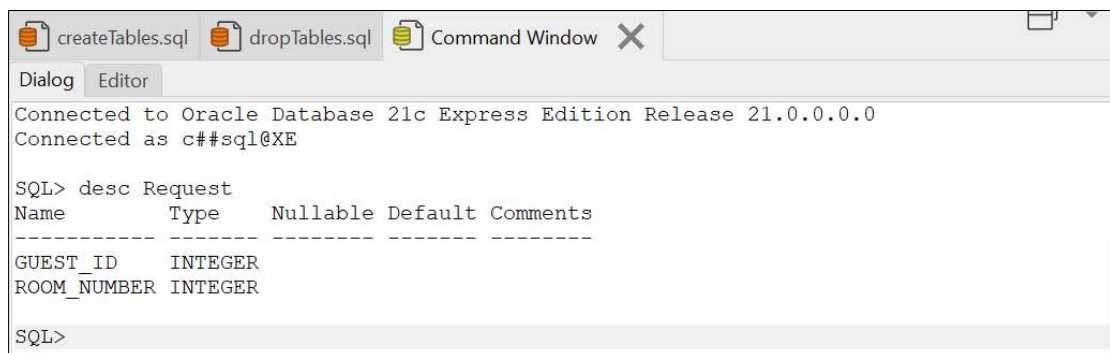


יצירת הטבלה Request:

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)
);
```

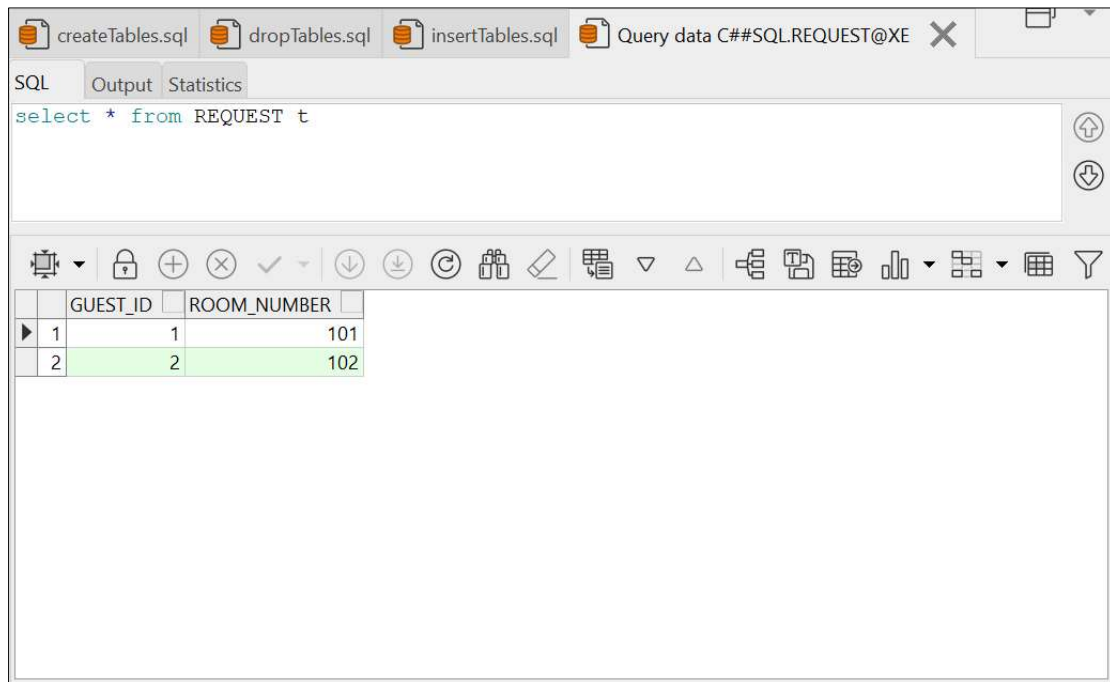
נבדוק שאכן הטבלה נוצרה בבסיס הנתונים:



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:

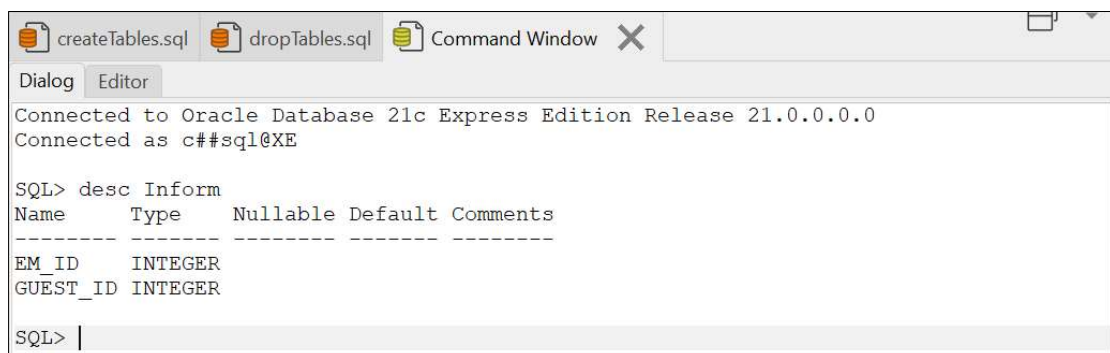


יצירת הטבלה Inform

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 Server Enterprise Manager interface. The top pane displays the SQL query: `select * from INFORM t`. The bottom pane shows the query results in a table format. The table has two columns: `EM_ID` and `GUEST_ID`. The results are as follows:

| | EM_ID | GUEST_ID |
|---|-------|----------|
| 1 | 1 | 1 |
| 2 | 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 main area is titled 'Guest'. It features a table with columns: Field Name, Type, and Options. The table contains five rows of field definitions: '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). 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 a checkbox for 'Include CREATE TABLE'. 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 עבורה 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:

| Field Name | Type | Options |
|------------------|----------------|---|
| em_id | Dataset Column | Employee, em_id, random, blank: 0% |
| tech_proficiency | Custom List | Expert, Beginner, Intermediate, random, blank: 0% |
| rating | Number | min: 1, max: 10, decimals: 0, blank: 0% |

Below the fields, there are controls for the number of rows (400), format (Excel), and an option to append to an existing dataset. At the bottom, there are 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:

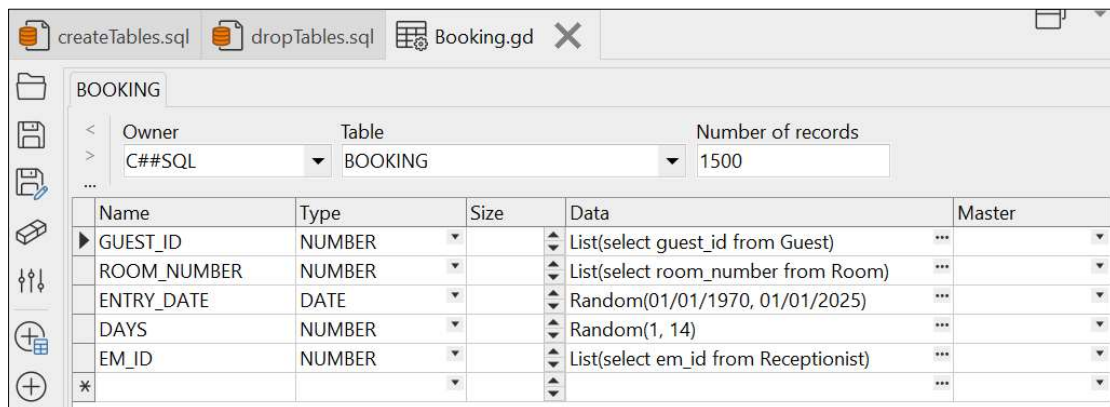
| Field Name | Type | Options |
|------------|----------------|--|
| em_id | Dataset Column | Employee, em_id, random, blank: 0% |
| shift | Custom List | Night, Morning, Evening, random, blank: 0% |
| lang | Language | blank: 0% |

Below the fields, there are controls for the number of rows (400), format (Excel), and an option to append to an existing dataset. At the bottom, there are 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.

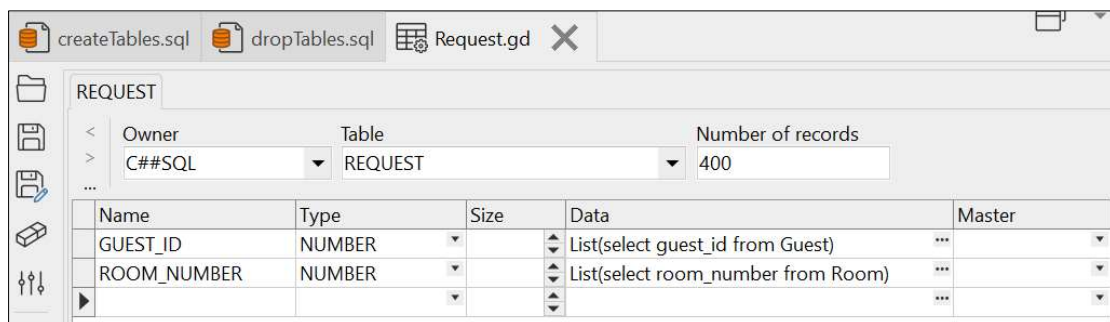
אכלוס הטבלה Booking:

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



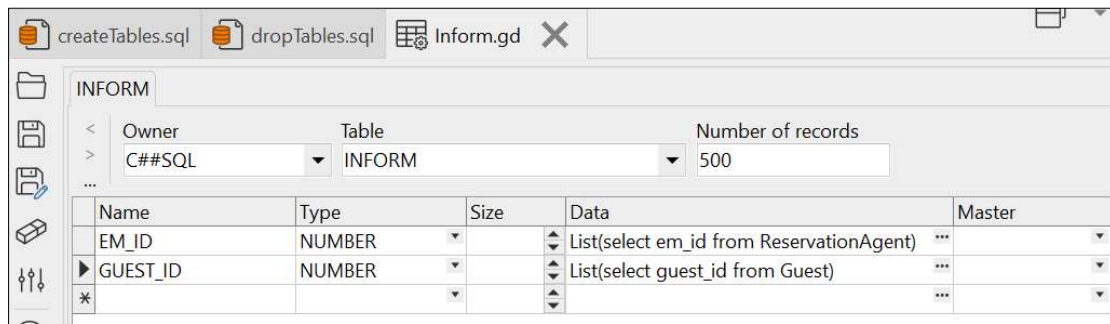
אכלוס הטבלה Request:

נשמך 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:

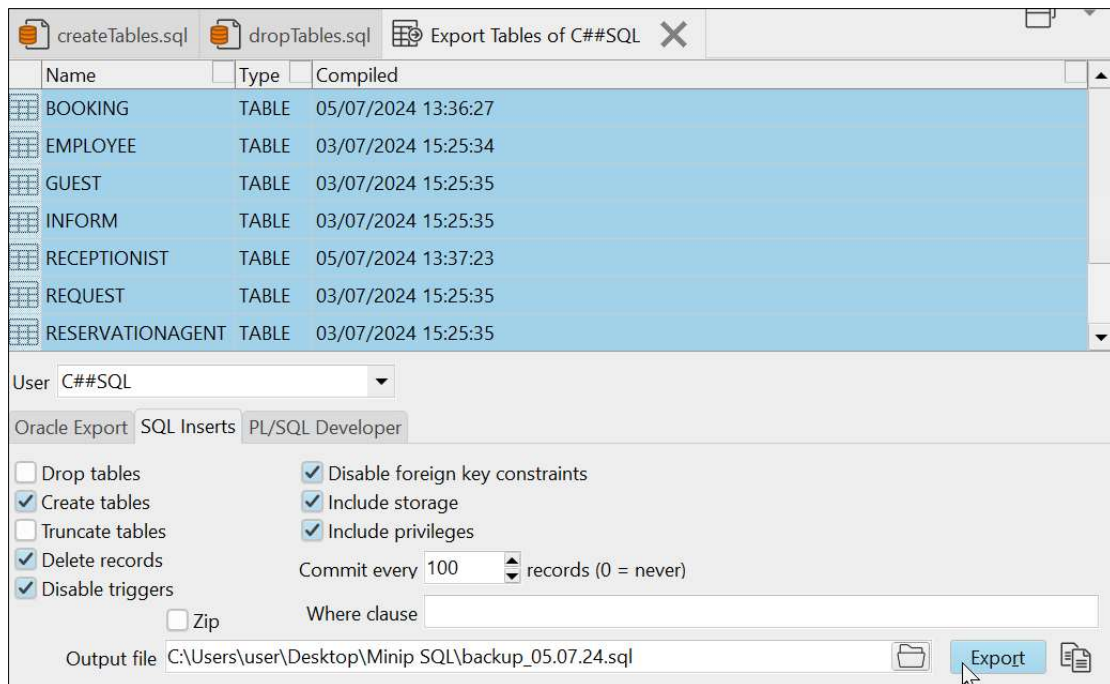
נשמך בכלי Data Generator המובנה בתוך PL/SQL. את הקובץ יצירה נשמור בשם Inform.gd כדי שנוכל להשתמש בו פעם נוספת אם נרצה.



גיבוי ושחזור בסיס הנתונים

גיבוי בסיס הנתונים:

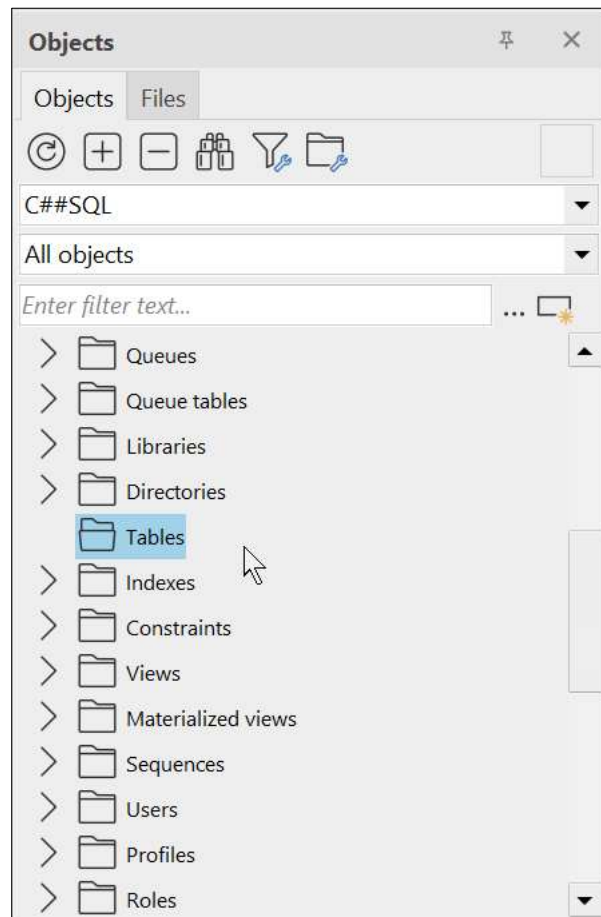
נגבה את כל נתוני יצירת ואכלוס הטבלאות שלנו אל תוך קובץ SQL ייעודי על מנת שנוכל לשחזר את הטבלאות או ליצור אותן על בסיס נתונים אחר מתי שנרצה.



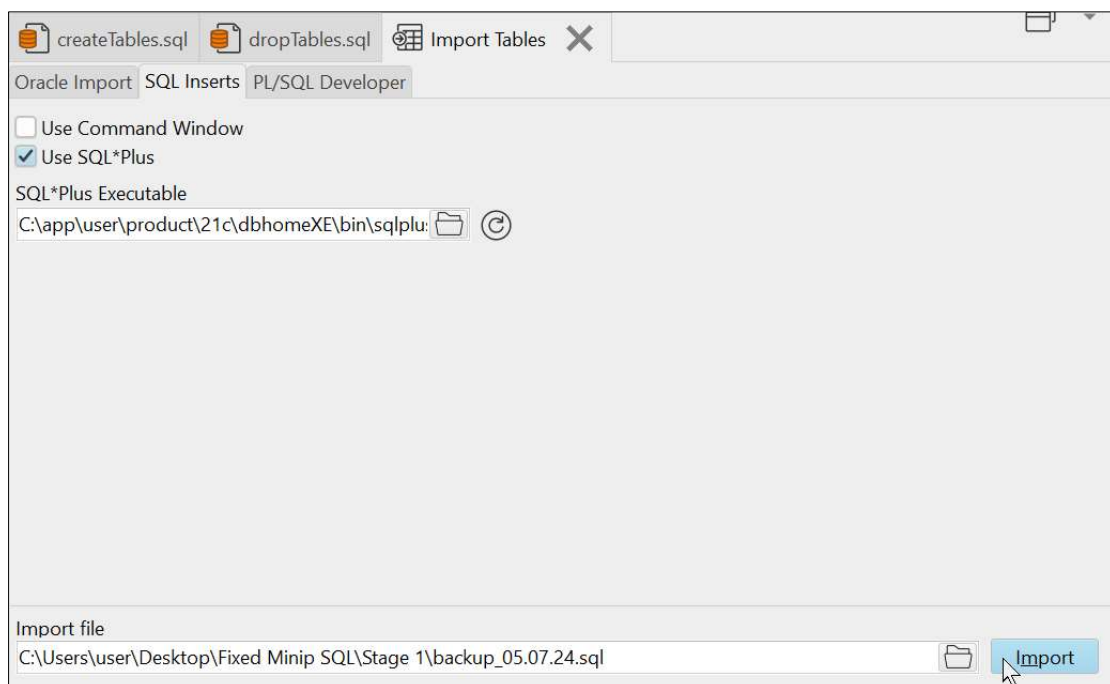
כל הקוד ליצירת בסיס הנתונים שלנו נמצא בקובץ backup_05.07.2024.sql.

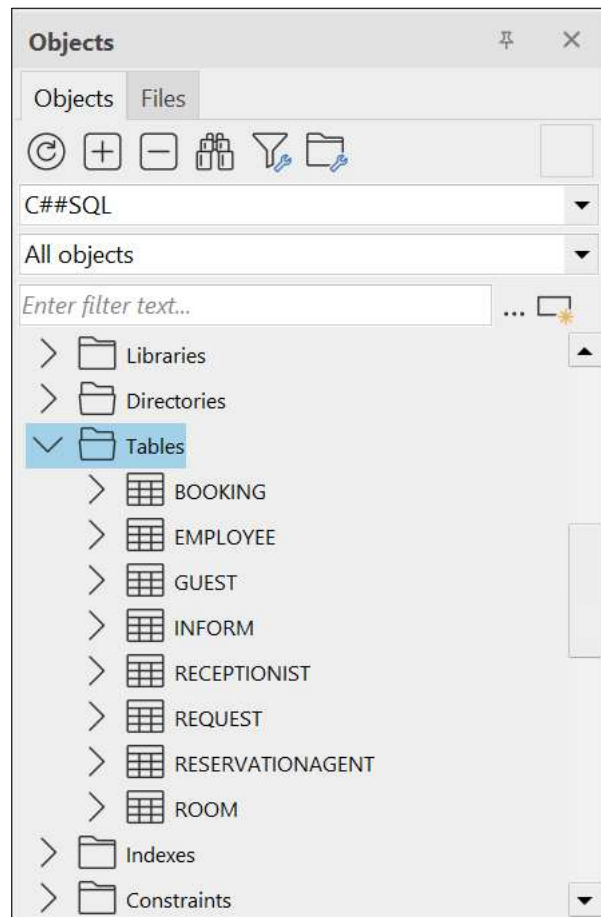
ייבוא בסיס נתונים:

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.