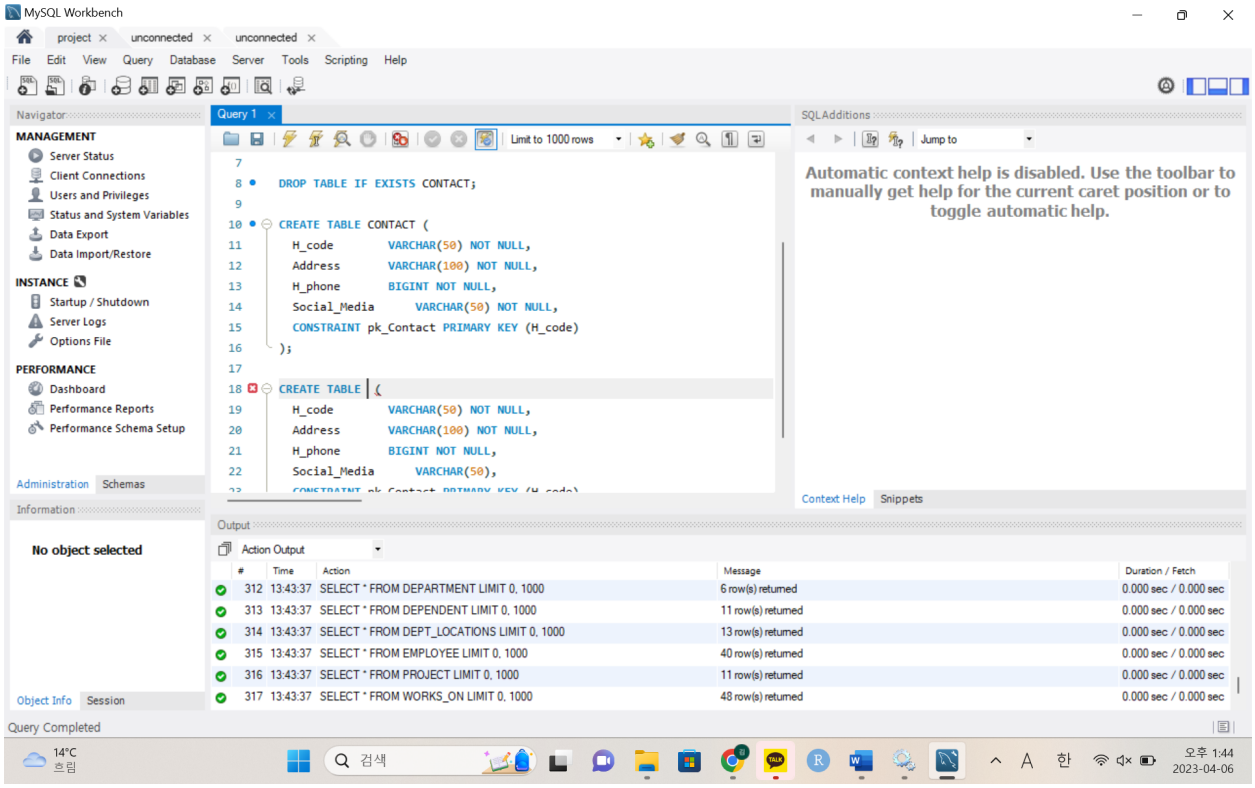
**CS 4347.004 Database Systems**
**Project Part 2 – Team 7**

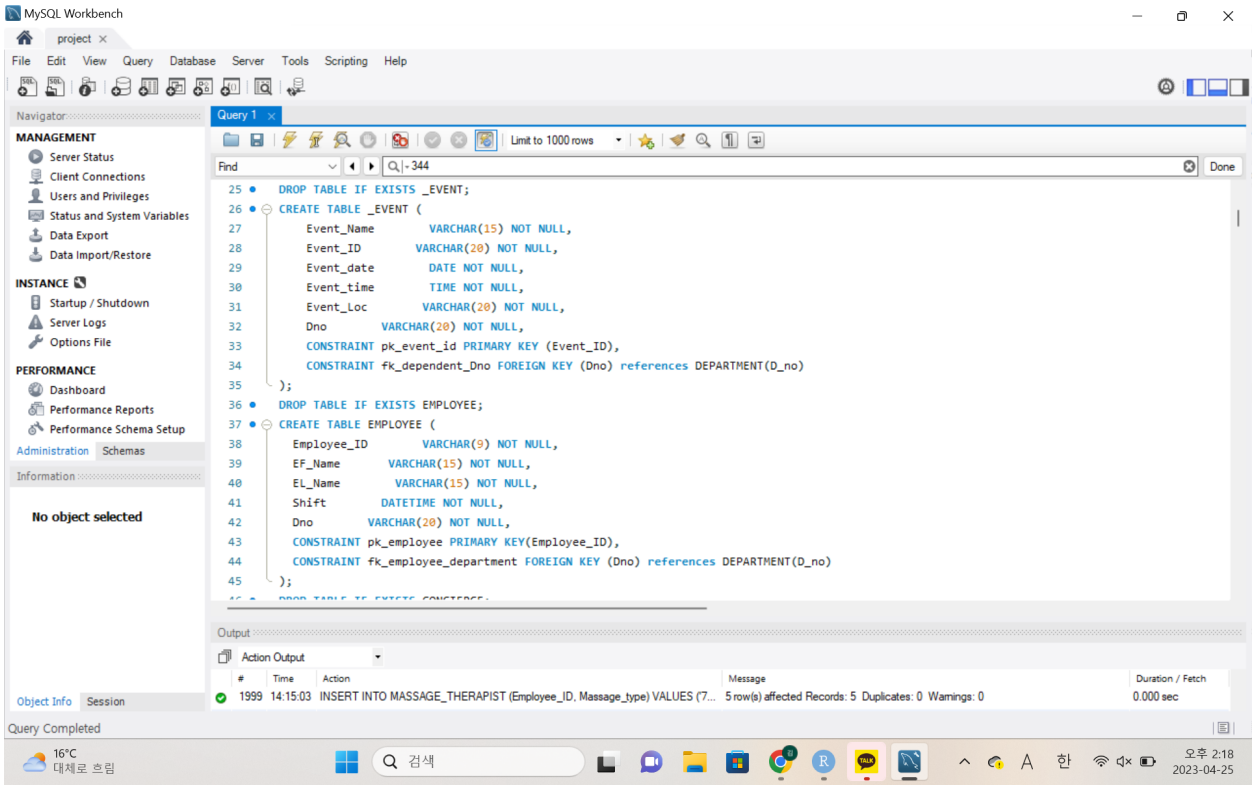
Veda Charthad – 25%
Haeun Kim – 25%
Aditi Mungale – 25%
Jihyung Park – 25%
Luu Tran – 0%

## Screenshot of the Database tool work area

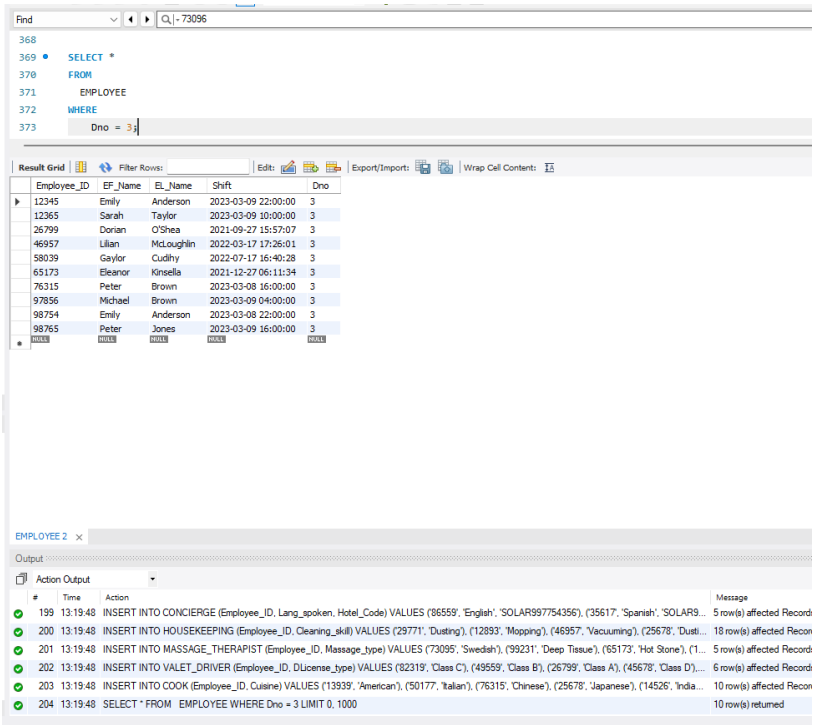



## Very specific screenshots of how the tool supports SQL:

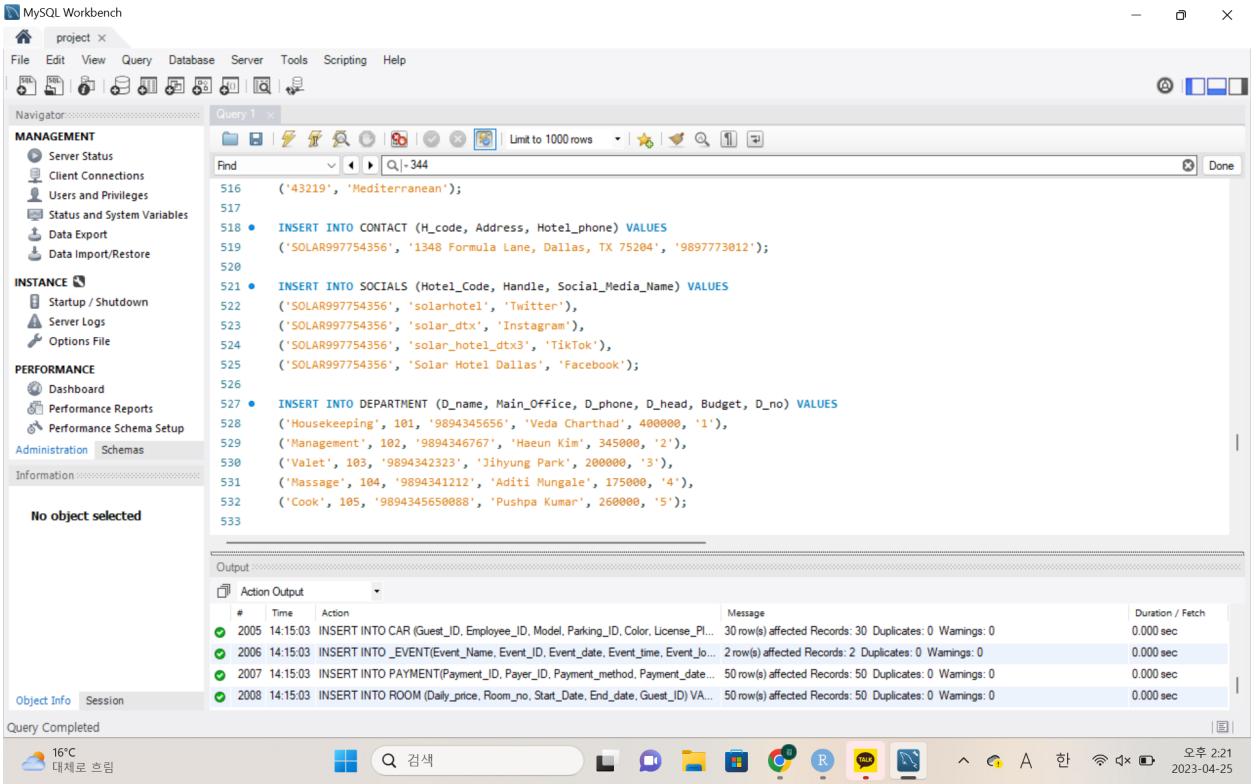
- CREATE



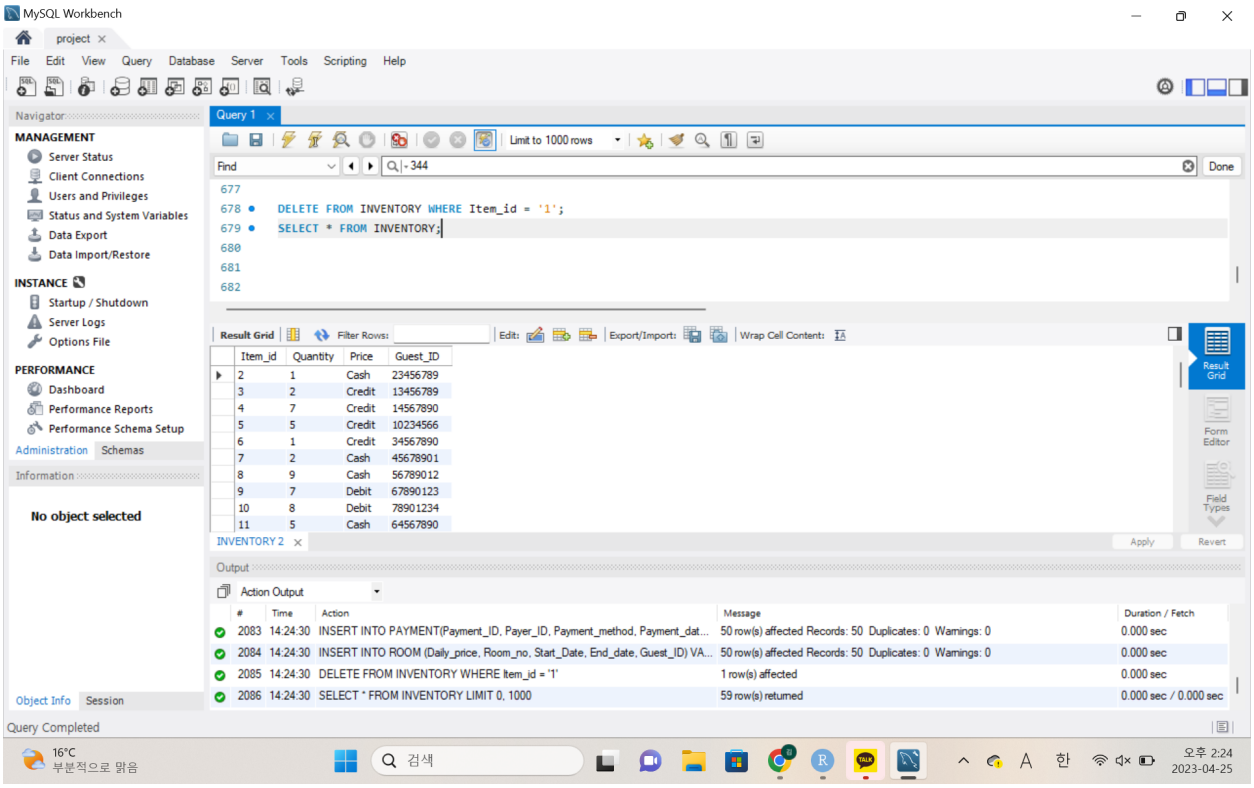

- SELECT



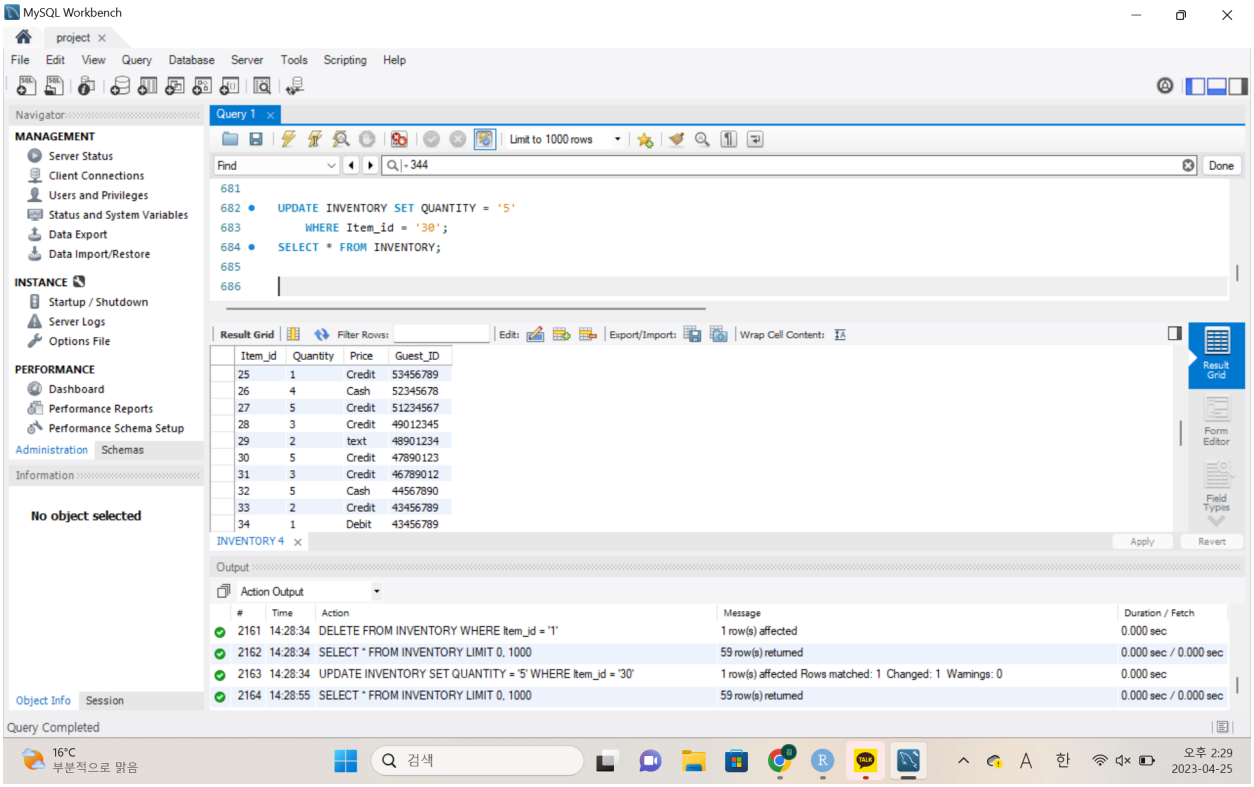

- INSERT



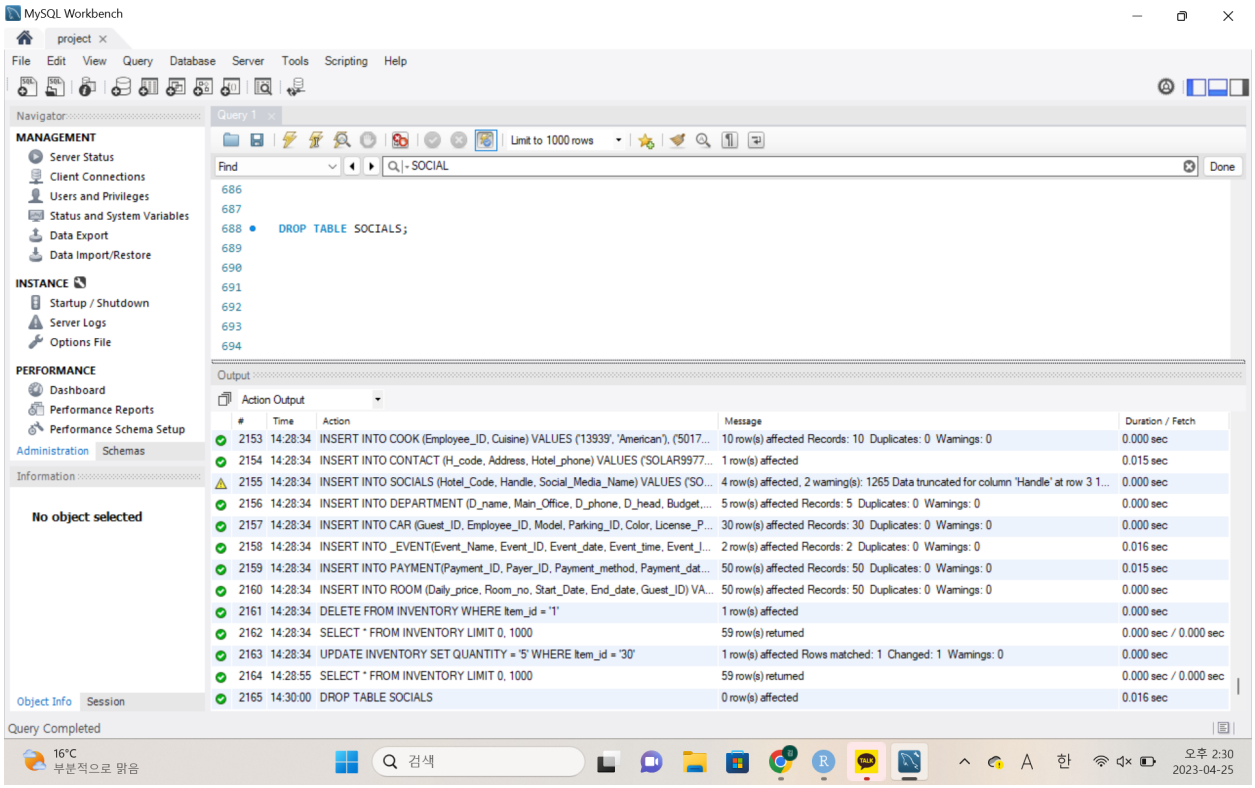

- DELETE


MySQL Workbench
Query 1
DELETE FROM INVENTORY WHERE Item_id = '1';
SELECT * FROM INVENTORY;
Result Grid
INVENTORY 2
Output
Action Output
Query Completed

- UPDATE


MySQL Workbench
Query 1
UPDATE INVENTORY SET QUANTITY = '5'
WHERE Item_id = '30';
SELECT * FROM INVENTORY;
Result Grid
INVENTORY 4
Output
Action Output
Query Completed

- DROP




**<u>Implementation Log – either generated by MySQL or handwritten notes – the log must show the dates and times of the creation of all tables, columns, keys, and updates to the database schema (include only schema related changes otherwise log file gets very large)</u>**

| # | Date | Time | Action | Message | Duration/Fetch |
|---|---|---|---|---|---|
| 1 | 04/13/2023 | 13:13:08 | Apply changes to hotel database | Changes Applied | 0.000 sec |
| 2 | 04/13/2023 | 13:13:56 | DROP TABLE IF EXISTS CONTACT;<br>CREATE TABLE CONTACT (<br>H_code VARCHAR(30) NOT NULL,<br>Address TEXT NOT NULL,<br>Hotel_phone VARCHAR(15) NOT NULL,<br>CONSTRAINT pk_hotel_code PRIMARY KEY (H_code)<br>); | 0 rows affected | 0.087 secs |
| 3 | 04/13/2023 | 13:14:16 | DROP TABLE IF EXISTS DEPARTMENT;<br>CREATE TABLE DEPARTMENT (<br>D_name TEXT NOT NULL,<br>Main_Office TEXT NOT NULL,<br>D_phone VARCHAR(15) NOT NULL,<br>D_head TEXT NOT NULL,<br>Budget MEDIUMINT NOT NULL, | 0 rows affected | 0.102 secs |

| | | | | | |
|---|---|---|---|---|---|
| | | | D_no VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_Department PRIMARY KEY (D_no)<br>); | | |
| 4 | 04/13/2023 | 13:14:24 | DROP TABLE IF EXISTS CAR;<br>CREATE TABLE CAR (<br>Guest_ID VARCHAR(20) NOT NULL,<br>Employee_ID VARCHAR(9) NOT NULL,<br>Model VARCHAR(30) NOT NULL,<br>Parking_ID VARCHAR(20) NOT NULL,<br>Color VARCHAR(15) NOT NULL,<br>License_Plate VARCHAR(10) NOT NULL,<br>CONSTRAINT pk_car PRIMARY KEY(License_Plate),<br>CONSTRAINT fk_G_id_car FOREIGN KEY (Guest_ID) references GUEST(Guest_ID),<br>CONSTRAINT fk_Emp_id_car FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.045 secs |
| 5 | 04/13/2023 | 13:14:32 | DROP TABLE IF EXISTS _EVENT;<br>CREATE TABLE _EVENT (<br>Event_Name VARCHAR(15) NOT NULL,<br>Event_ID VARCHAR(20) NOT NULL,<br>Event_date DATE NOT NULL,<br>Event_time TIME NOT NULL,<br>Event_Loc VARCHAR(20) NOT NULL,<br>Dno VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_event_id PRIMARY KEY (Event_ID),<br>CONSTRAINT fk_dependent_Dno FOREIGN KEY (Dno) references DEPARTMENT(D_no)<br>); | 0 rows affected | 0.093 secs |
| 6 | 04/13/2023 | 13:14:46 | DROP TABLE IF EXISTS EMPLOYEE;<br>CREATE TABLE EMPLOYEE (<br>Employee_ID VARCHAR(9) NOT NULL,<br>EF_Name VARCHAR(15) NOT NULL,<br>EL_Name VARCHAR(15) NOT NULL,<br>Shift DATETIME NOT NULL,<br>Dno VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_employee PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_employee_department FOREIGN KEY (Dno) references DEPARTMENT(D_no)<br>); | 0 rows affected | 0.111 secs |
| 7 | 04/13/2023 | 13:14:58 | DROP TABLE IF EXISTS CONCIERGE;<br>CREATE TABLE CONCIERGE (<br>Employee_ID VARCHAR(9) NOT NULL,<br>Lang_spoken VARCHAR(15) NOT NULL,<br>Hotel_Code VARCHAR(30) NOT NULL,<br>CONSTRAINT pk_concierge PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_concierge_employee FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_concierge_contact FOREIGN KEY (Hotel_Code) references CONTACT(H_Code)<br>); | 0 rows affected | 0.077 secs |

| | | | | | |
|---|---|---|---|---|---|
| 8 | 04/13/2023 | 13:15:06 | DROP TABLE IF EXISTS HOUSEKEEPING;<br>CREATE TABLE HOUSEKEEPING (<br>Employee_ID VARCHAR(9) NOT NULL,<br>Cleaning_skill VARCHAR(30) NOT NULL,<br>CONSTRAINT pk_housekeeping PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_housekeeping_employee FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.125 secs |
| 9 | 04/13/2023 | 13:15:24 | DROP TABLE IF EXISTS MASSAGE_THERAPIST;<br>CREATE TABLE MASSAGE_THERAPIST (<br>Employee_ID VARCHAR(9) NOT NULL,<br>Massage_type VARCHAR(30) NOT NULL,<br>CONSTRAINT pk_massage_therapist PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_massage_therapist_employee FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.127 secs |
| 10 | 04/13/2023 | 13:15:32 | DROP TABLE IF EXISTS VALET_DRIVER;<br>CREATE TABLE VALET_DRIVER (<br>Employee_ID VARCHAR(9) NOT NULL,<br>DLicense_type VARCHAR(30) NOT NULL,<br>CONSTRAINT pk_valet_driver PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_valet_driver_employee FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.094 secs |
| 11 | 04/13/2023 | 13:15:47 | DROP TABLE IF EXISTS COOK;<br>CREATE TABLE COOK (<br>Employee_ID VARCHAR(9) NOT NULL,<br>Cuisine VARCHAR(30) NOT NULL,<br>CONSTRAINT pk_cook PRIMARY KEY(Employee_ID),<br>CONSTRAINT fk_cook_employee FOREIGN KEY (Employee_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.098 secs |
| 12 | 04/13/2023 | 13:15:56 | DROP TABLE IF EXISTS GUEST;<br>CREATE TABLE GUEST (<br>Guest_ID VARCHAR(20) NOT NULL,<br>F_name VARCHAR(15) NOT NULL,<br>L_name VARCHAR(15) NOT NULL,<br>Room_no SMALLINT NOT NULL,<br>Phone BIGINT NOT NULL,<br>Credit_card TINYINT(16) NOT NULL,<br>Address VARCHAR(200) NOT NULL,<br>License_plate VARCHAR(10) NOT NULL,<br>Loyalty_pts MEDIUMINT NOT NULL,<br>Emp_ID VARCHAR(9) NOT NULL,<br>CONSTRAINT pk_guest_id PRIMARY KEY (Guest_ID),<br>CONSTRAINT fk_emp_id FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.088 secs |

| | | | | | |
|---|---|---|---|---|---|
| 13 | 04/13/2023 | 13:16:09 | DROP TABLE IF EXISTS PAYMENT;<br>CREATE TABLE PAYMENT (<br>Payment_ID VARCHAR(20) NOT NULL,<br>Payer_ID VARCHAR(20) NOT NULL,<br>Payment_method ENUM("text", "Cash", "Credit", "Debit") NOT NULL,<br>Payment_date DATE NOT NULL,<br>Payment_amt DECIMAL(19,4) NOT NULL,<br>Emp_ID VARCHAR(9) NOT NULL,<br>CONSTRAINT pk_payment_id PRIMARY KEY (Payment_ID),<br>CONSTRAINT fk_payment_payer_id FOREIGN KEY (Payer_ID) references GUEST(Guest_ID),<br>CONSTRAINT fk_payment_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID)<br>); | 0 rows affected | 0.085 secs |
| 14 | 04/13/2023 | 13:16:17 | DROP TABLE IF EXISTS FEEDBACK;<br>CREATE TABLE FEEDBACK (<br>Feedback_ID VARCHAR(20) NOT NULL,<br>Feed_Comment TEXT NOT NULL,<br>Guest_ID VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_feedback_id PRIMARY KEY (Feedback_ID),<br>CONSTRAINT fk_G_id_fd FOREIGN KEY (Guest_ID) references GUEST(Guest_ID)<br>); | 0 rows affected | 0.076 secs |
| 15 | 04/13/2023 | 13:16:21 | DROP TABLE IF EXISTS ROOM;<br>CREATE TABLE ROOM (<br>Daily_price DOUBLE NOT NULL,<br>Room_no SMALLINT NOT NULL,<br>Start_date DATE NOT NULL,<br>End_date DATE NOT NULL,<br>Guest_ID VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_Room PRIMARY KEY (Room_no),<br>CONSTRAINT fk_G_id FOREIGN KEY (Guest_ID) references GUEST(Guest_ID)<br>); | 0 rows affected | 0.113 secs |
| 16 | 04/13/2023 | 13:16:29 | DROP TABLE IF EXISTS DELIVERS;<br>CREATE TABLE DELIVERS (<br>Emp_ID VARCHAR(9) NOT NULL,<br>Room_no SMALLINT NOT NULL,<br>CONSTRAINT pk_delivers PRIMARY KEY (Emp_ID,Room_no),<br>CONSTRAINT fk_delivers_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_delivers_room FOREIGN KEY (Room_no) references ROOM(Room_no)<br>); | 0 rows affected | 0.079 secs |
| 17 | 04/13/2023 | 13:16:34 | DROP TABLE IF EXISTS ORDERS;<br>CREATE TABLE ORDERS (<br>Emp_ID VARCHAR(9) NOT NULL, | 0 rows affected | 0.042 secs |

| | | | | | |
|---|---|---|---|---|---|
| | | | G_no VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_orders PRIMARY KEY (Emp_ID, G_no),<br>CONSTRAINT fk_orders_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_orders_guests FOREIGN KEY (G_no) references GUEST(Guest_ID)<br>); | | |
| 18 | 04/13/2023 | 13:16:40 | DROP TABLE IF EXISTS CLEANS;<br>CREATE TABLE CLEANS (<br>Emp_ID VARCHAR(9) NOT NULL,<br>Room_number SMALLINT NOT NULL,<br>CONSTRAINT pk_cleans PRIMARY KEY (Emp_ID, Room_number),<br>CONSTRAINT fk_cleans_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_cleans_rooms FOREIGN KEY (Room_number) references ROOM(Room_no)<br>); | 0 rows affected | 0.065 secs |
| 19 | 04/13/2023 | 13:16:53 | DROP TABLE IF EXISTS CONVEY_REQUEST;<br>CREATE TABLE CONVEY_REQUEST (<br>Emp_ID VARCHAR(9) NOT NULL,<br>Dno VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_convey_request PRIMARY KEY (Emp_ID, Dno),<br>CONSTRAINT fk_convey_request_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_convey_request_department FOREIGN KEY (Dno) references DEPARTMENT(D_no)<br>); | 0 rows affected | 0.055 secs |
| 20 | 04/13/2023 | 13:17:00 | DROP TABLE IF EXISTS REQUEST;<br>CREATE TABLE REQUEST (<br>G_no VARCHAR(20) NOT NULL,<br>Dno VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_request PRIMARY KEY (G_no, Dno),<br>CONSTRAINT fk_request_employee FOREIGN KEY (G_no) references GUEST(GUEST_ID),<br>CONSTRAINT fk_request_department FOREIGN KEY (Dno) references DEPARTMENT(D_no)<br>); | 0 rows affected | 0.032 secs |
| 21 | 04/13/2023 | 13:17:08 | DROP TABLE IF EXISTS SOCIALS;<br>CREATE TABLE SOCIALS (<br>Hotel_Code VARCHAR(15) NOT NULL,<br>Handle VARCHAR(15) NOT NULL,<br>Social_Media_Name VARCHAR(15) NOT NULL,<br>CONSTRAINT pk_handle PRIMARY KEY(Handle),<br>CONSTRAINT fk_socials_contact FOREIGN KEY (Hotel_Code) references CONTACT(H_code)<br>); | 0 rows affected | 0.060 secs |

| | | | | | |
|---|---|---|---|---|---|
| 22 | 04/13/2023 | 13:17:17 | DROP TABLE IF EXISTS INTERACTS;<br>CREATE TABLE INTERACTS(<br>Emp_ID VARCHAR(9) NOT NULL,<br>G_no VARCHAR(20) NOT NULL,<br>CONSTRAINT pk_interacts PRIMARY KEY (Emp_ID, G_no),<br>CONSTRAINT fk_interacts_employee FOREIGN KEY (Emp_ID) references EMPLOYEE(Employee_ID),<br>CONSTRAINT fk_interacts_guest FOREIGN KEY(G_no) references GUEST(Guest_ID)<br>); | 0 rows affected | 0.052 secs |
| 23 | 04/13/2023 | 13:17:22 | DROP TABLE IF EXISTS TAKE_ACTION;<br>CREATE TABLE TAKE_ACTION (<br>Dno VARCHAR(20) NOT NULL,<br>F_id VARCHAR(20) NOT NULL,<br>CONSTRAINT fk_take_action_department FOREIGN KEY (Dno) references DEPARTMENT(D_no),<br>CONSTRAINT fk_take_action_Fid FOREIGN KEY (F_id) references FEEDBACK(Feedback_ID)<br>); | 0 rows affected | 0.144 secs |
| 24 | 04/13/2023 | 13:17:40 | DROP TABLE IF EXISTS INVENTORY;<br>CREATE TABLE INVENTORY (<br>Item_id VARCHAR(12) NOT NULL,<br>Quantity VARCHAR(20) NOT NULL,<br>Price ENUM("text", "Cash", "Credit", "Debit"),<br>Guest_ID VARCHAR(20) NOT NULL,<br>CONSTRAINT pr_inventory PRIMARY KEY (Item_id),<br>CONSTRAINT fk_G_id_inventory FOREIGN KEY (Guest_ID) references GUEST(Guest_ID)<br>); | 0 rows affected | 0.185 secs |
| 25 | 04/13/2023 | 13:17:43 | DROP TABLE IF EXISTS MAKE_PAYMENT;<br>CREATE TABLE MAKE_PAYMENT(<br>I_id VARCHAR(20) NOT NULL,<br>Pay_id VARCHAR(20) NOT NULL,<br>CONSTRAINT fk_make_payment_inventory FOREIGN KEY (I_id) references INVENTORY(Item_id),<br>CONSTRAINT fk_make_payment_payment FOREIGN KEY(Pay_id) references PAYMENT(Payment_ID)<br>); | 0 rows affected | 0.039 secs |
| 26 | 04/13/2023 | 13:17:50 | DROP TABLE IF EXISTS CAR_PLATE;<br>CREATE TABLE CAR_PLATE (<br>G_no VARCHAR(20) NOT NULL,<br>Plate_no VARCHAR(10) NOT NULL,<br>CONSTRAINT pk_carplate PRIMARY KEY (Plate_no),<br>CONSTRAINT fk_carplate_guest FOREIGN KEY (G_no) references GUEST(Guest_ID)<br>); | 0 rows affected | 0.073 secs |
| 27 | 04/13/2023 | 13:17:57 | DROP TABLE IF EXISTS BOOK;<br>CREATE TABLE BOOK (<br>G_no VARCHAR(20) NOT NULL,<br>Room_no SMALLINT NOT NULL,<br>Plate_no VARCHAR(10) NOT NULL, | 0 rows affected | 0.018 secs |

| | | | | | |
|---|---|---|---|---|---|
| | | | CONSTRAINT pk_book PRIMARY KEY (Plate_no),<br>CONSTRAINT fk_book_room FOREIGN KEY (Room_no) references ROOM(Room_no),<br>CONSTRAINT fk_book_guest FOREIGN KEY (G_no) references Guest(Guest_ID)<br>); | | |

**<u>Data Generation – Populate the database with meaningful data – data that supports the business rules – there are 2 steps to this:</u>**

1. Estimate the number of rows/tuples for each table
2. Enter the data – using scripts or other code or manually
   - Turn in a list showing each table and its planned size – be realistic - consider the relationships. For example, how many operations can be held in a year, then how many doctors, techs, and nurses working in those rooms are needed per year?
   - Turn in a list showing one row/tuple from each table with data

| TABLE NAME | # OF TUPLES | Example Tuple |
|---|---|---|
| CONTACT | 1 | H_code: SOLAR997754356<br>Address: 1348 Formula Lane, Dallas, TX 75204<br>Hotel_phone: 9897773012 |
| DEPARTMENT | 5 | D_name: Housekeeping<br>Main_Office: 101<br>D_phone: 9894345656<br>D_head: Veda Charthad<br>Budget: 400000<br>D_no: 1 |
| ROOM | 100 | Daily_price: 163.51<br>Room_no: 75<br>Start_Date: 2023-04-18<br>End_date: 2023-05-20<br>G_no: 12345678 |
| FEEDBACK | 3 | Feedback_ID: 112<br>Feed_Comment: I loved my stay at the hotel. All of the services were so nice and all of the employees were very courteous. Will definitely come again.<br>Guest_ID: 10234566 |
| CAR | 30 | Guest_ID: 25678901<br>Employee_ID: 82319<br>Model: Honda Odessy<br>Parking_ID: 1<br>Color: Black<br>License_Plate: YBG5626 |
| EVENT | 10 | |

| | | |
|---|---|---|
| | | |

| Event_Name | Event_ID | Event_Date | Event_time | Event_loc | Dno |
|---|---|---|---|---|---|
| Hotel Party | 11 | 2023-03-28 | 09:40:28 | Apple Blvrd | MANAGE6767 |

| | | |
|---|---|---|
| EMPLOYEE | 40 | |

| Employee_ID | EF_Name | EL_Name | Shift | Dno |
|---|---|---|---|---|
| 86559 | Ignacio | Jennaway | 2023-01-18 23:24:13 | 1 |

| | | |
|---|---|---|
| CONCIERGE | 5 | |

| Employee_ID | Lang_spoken | Hotel_Code |
|---|---|---|
| 86559 | English | SOLAR997754356 |

| | | |
|---|---|---|
| HOUSEKEEPING | 15 | |

| Employee_ID | Cleaning Skill |
|---|---|
| 29771 | Dusting |

| | | |
|---|---|---|
| MASSAGE_THERAPIST | 5 | |

| Employee_ID | Massage_type |
|---|---|
| 73095 | Swedish |

| | | |
|---|---|---|
| VALET_DRIVER | 5 | |

| Employee_ID | DLicense_type |
|---|---|
| 82319 | Class C |

| | | |
|---|---|---|
| COOK | 10 | |

| Employee_ID | Cuisine |
|---|---|
| 13939 | American |

| | | |
|---|---|---|
| PAYMENT | 50 | |

| Payment_ID | Payer_ID | Payment_method | Payment_date | Payment_amt | Emp_ID |
|---|---|---|---|---|---|
| 00001 | 12345678 | Card | 2023-05-20 | 300.00 | 86559 |

| | | |
|---|---|---|
| INVENTORY | 60 | |

| Item_ID | Quantity | Price | Guest_ID |
|---|---|---|---|
| 1 | 1 | 12.00 | 12345678 |

| | | |
|---|---|---|
| GUEST | 200 | |

| Guest_ID | F_name | L_name | Room_no | Phone | Credit_card | Address | Loyalty_pts | Emp_ID |
|---|---|---|---|---|---|---|---|---|
| 12345678 | JAMES | SMITH | 201 | 1234567890 | 12345678901 23456 | 2947 BAKSU ST | 604 | 86559 |

| | | |
|---|---|---|
| DELIVERS | 50 | |

| Emp_id | Room_number |
|---|---|
| 86559 | 201 |

| | | |
|---|---|---|
| ORDERS | 30 | |

| Emp_id | G_no |
|---|---|
| 35617 | 12345678 |

| | | |
|---|---|---|
| CLEAN | 100 | |

| Emp_id | Room_number |
|---|---|
| 29771 | 201 |

| | | |
|---|---|---|
| SOCIALS | 4 | |

| Hotel_Code | Handle | Social_Media_Name |
|---|---|---|
| SOLAR997754356 | solar_hotel_dtx1 | Twitter |

| | | |
|---|---|---|
| INTERACT | 200 | |

| Emp_id | G_no |
|---|---|
| 35617 | 12345678 |

| | | |
|---|---|---|
| CONVEY_REQUEST | 100 | |

| Emp_id | Dno |
|---|---|
| 35617 | 1 |

| | | |
|---|---|---|
| CAR_PLATES | 30 | |

| G_no | Plate_no |
|---|---|
| 23456789 | WEJ2938 |

| | | |
|---|---|---|
| BOOK | 50 | |

| Room_no | G_no |
|---|---|
| 201 | 123456789 |

| | | |
|---|---|---|
| TAKE_ACTION | 3 | |

| Dno | F_id |
|---|---|
| 2 | 112 |

| | | |
|---|---|---|
| MAKE_PAYMENT | 60 | |

| I_id | Pay_id |
|---|---|
| 86559 | 00001 |

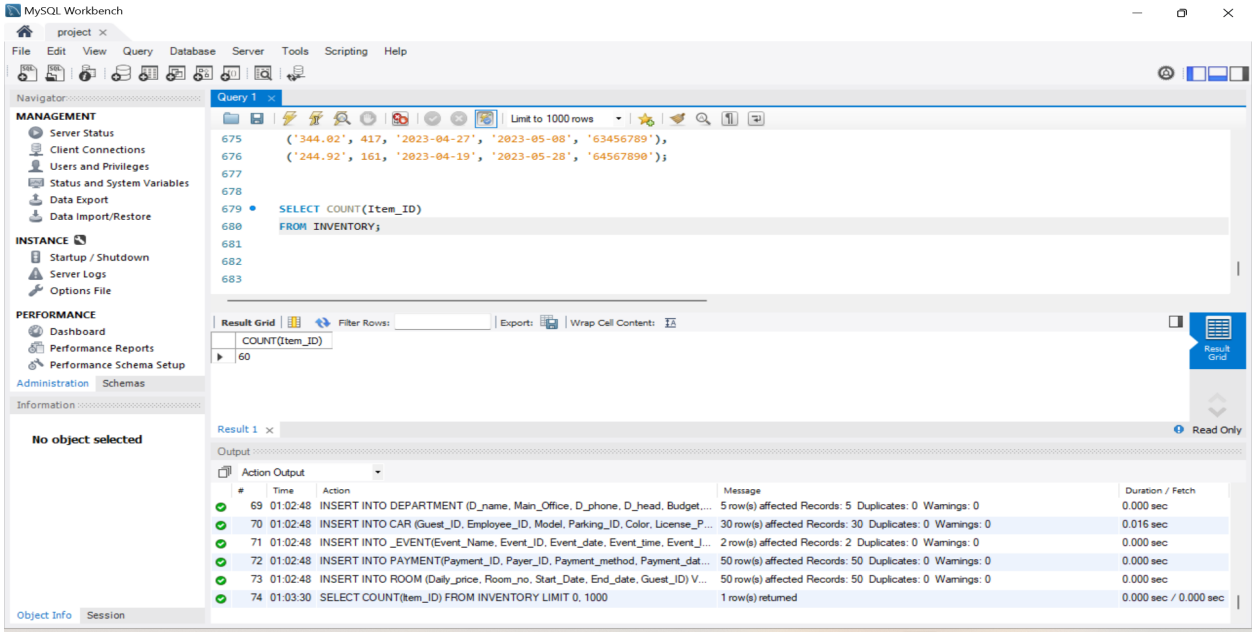
| | | |
|---|---|---|
| REQUEST | 3 | |

| G_no | Dno |
|---|---|
| 59012334 | 2 |

**<u>Query Examples – turn in queries for the following: Execute the query examples in the database tool work area and include screenshots.</u>**

<u>1. Show a count of the largest population – example - How many patients are in the hospital</u>

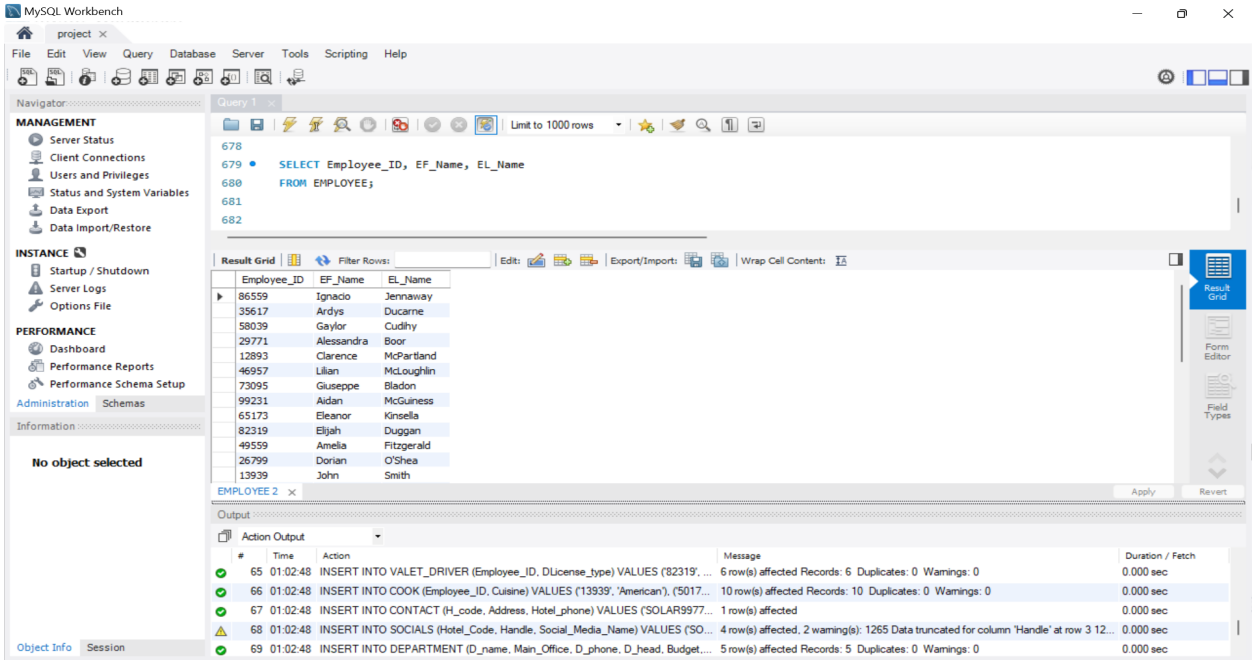
The largest population in our database comes from the Inventory Table. We are showing the count of the number of items currently in inventory.



SELECT COUNT(Item_ID)
FROM INVENTORY;

<u>2. Show a listing of a key entity in the database – example – Who are the doctors in the hospital?</u>

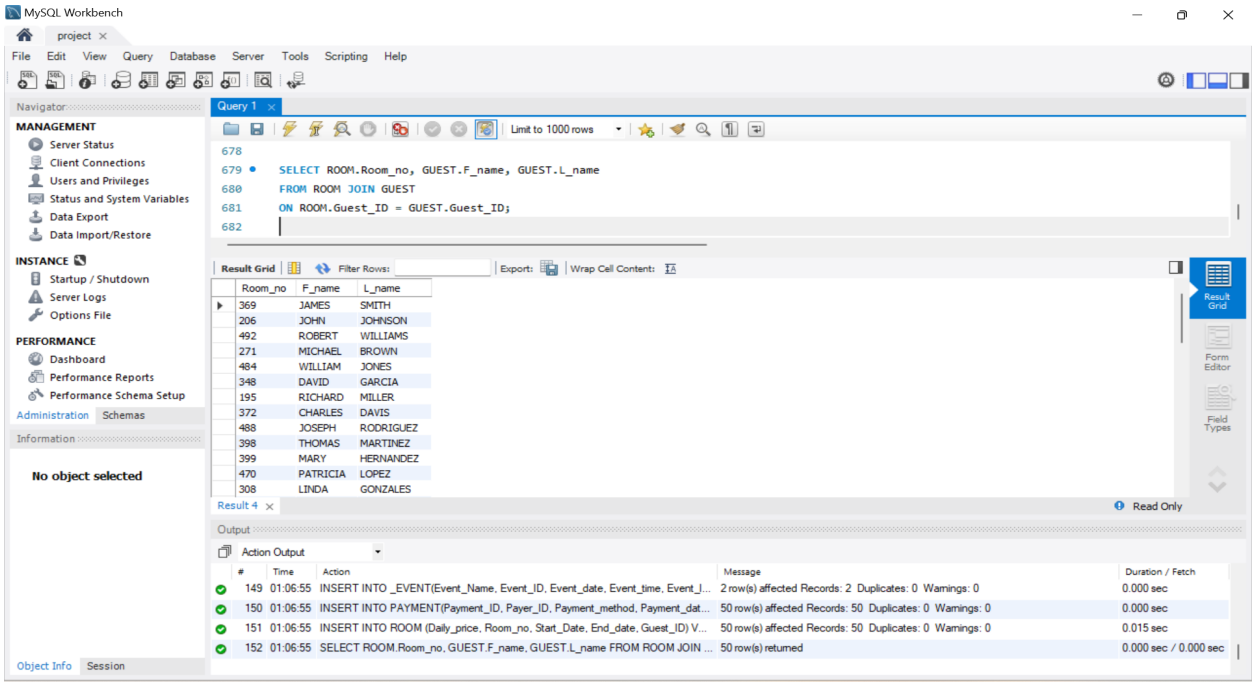
One of our key entities in the databse is our employees. We are showing the Employee ID number as well as their first and last name.




SELECT Employee_ID, EF_Name, EL_Name
FROM EMPLOYEE;

<u>3. Show a list of entities that must function together – example – Show all the doctors and technicians who work together (a join)</u>

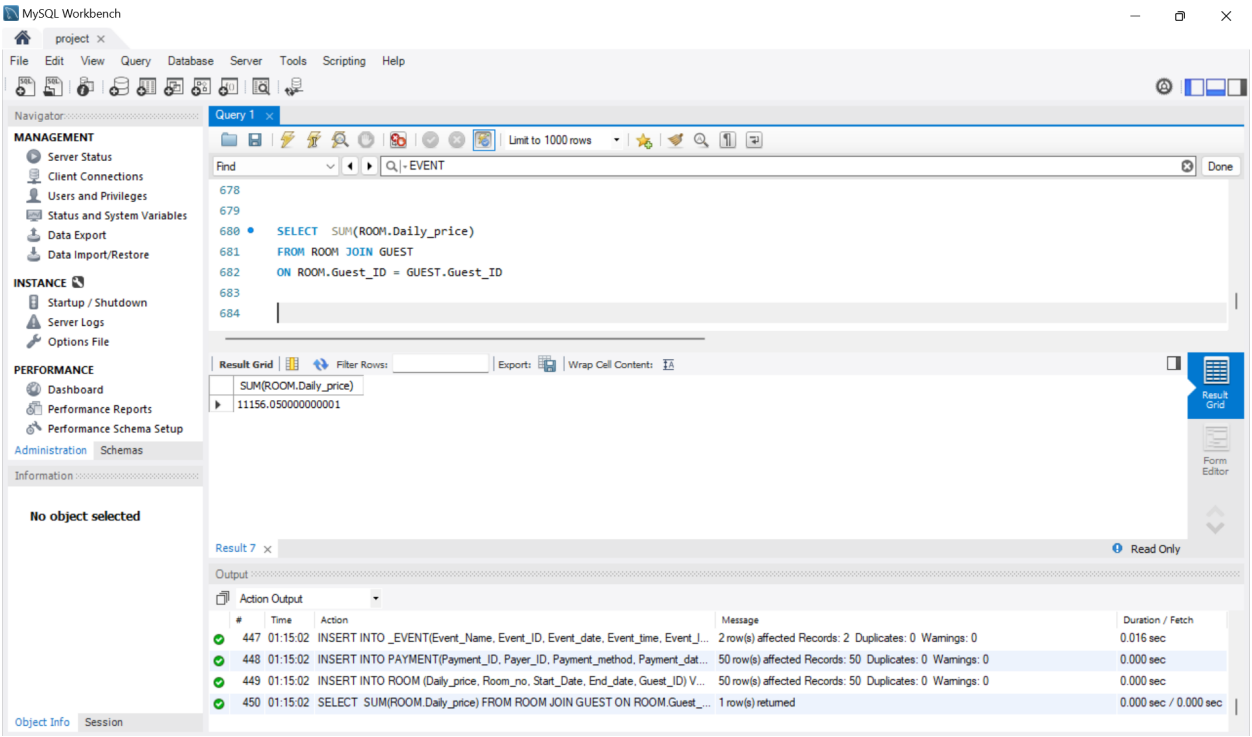
All of the rooms and the first and last names of the guests. These two entities must function together because guests stay in rooms.




SELECT ROOM.Room_no, GUEST.F_name, GUEST.L_name
FROM ROOM JOIN GUEST
ON Guest_ID = Guest_ID

<u>4. Show the cost of an occurrence, derived using aggregate functions – example – How much will a hospital stay cost a patient - consider the doctor, nurse, technician, room, and insurance</u>

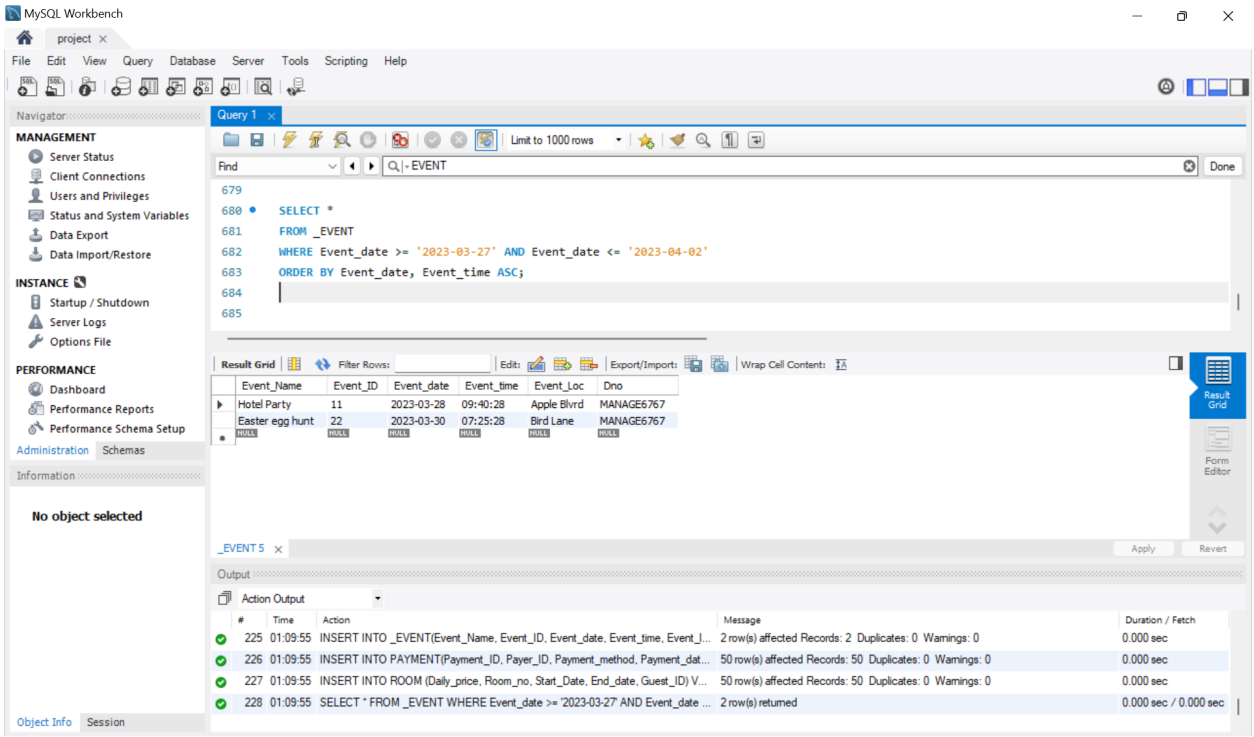
The cost of a guest's stay at the hotel.




```
SELECT  SUM(ROOM.Daily_price)
FROM ROOM JOIN GUEST
ON ROOM.Guest_ID = GUEST.Guest_ID
```

<u>5. Show a schedule for multiple occurrences, sorted by date and time – example – what is the schedule for an operating room for a specific week?</u>

The schedule of the events in the week of March 27 sorted by date and time.




SELECT *
FROM EVENT
WHERE Event_date >= ‘2023-03-27’ AND Event_date <= ‘2023-04-02’
ORDER BY Event_date, Event_time ASC