

Hotel Database – DQL & DML Tasks

DQL

1. Display all guest records.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists several databases and their structures. The central pane contains a T-SQL query window with the following code:

```
SELECT * FROM Customer;
```

The Results pane below displays the output of the query, which consists of four rows of guest information:

GstID	phone	email	name
1	91234567	ali@gmail.com	Ali Hassan
2	92345678	sara@yahoo.com	Sara Ahmed
3	93456789	omar@hotmail.com	Omar Said
4	94567890	noor@gmail.com	Noor Salm

A status bar at the bottom indicates "Query executed successfully." and shows system information like "DESKTOP-GILBFIO (15.0 RTM)" and "12/24/2025".

2. Display each guest's name, contact number, and proof ID type.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists several databases and their structures. The central pane contains a T-SQL query window with the following code:

```
SELECT name, phone  
FROM Customer;
```

The Results pane displays the output of the query, which consists of four rows of guest information:

name	phone
All Hassan	91234567
Sara Ahmed	92345678
Omar Said	93456789
Noor Salm	94567890

A status bar at the bottom indicates "Query executed successfully." and shows system information like "DESKTOP-GILBFIO (15.0 RTM)" and "12/24/2025".

3. Display all bookings with booking date, status, and total cost

```

SELECT b.BookingID, b.check_in_date, b.check_out_date, r.nightly_rate
FROM Booking b
JOIN Room r ON b.BookingID = r.BookingID;

```

BookingID	check_in_date	check_out_date	nightly_rate
1	2025-01-10	2025-01-15	45.00
2	2025-01-12	2025-01-14	70.00
3	2025-01-20	2025-01-25	120.00

Query executed successfully.

4. Display each room number and its price per night as NightlyRate.

```

SELECT RoomID, nightly_rate AS NightlyRate
FROM Room;

```

RoomID	NightlyRate
1	45.00
2	70.00
3	120.00
4	45.00
5	70.00

Query executed successfully.

5. List rooms priced above 1000 per night.

Object Explorer shows the database structure:

- BankingSystem
- college
- Company_SD
- companyDataBase
- companyDB
- companyDBase
- EDB
- libraryDB
- libraryDBase
- SQL_Aggregation_Functions_Practical_Task
- Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Temp Tables
 - Temporary Tables
 - dbo.Courses
 - dbo.Enrollments
 - dbo.Courses
 - CourseID (PK, int, not null)
 - Title (varchar(100), not null)
 - InstructorID (FK, int, not null)
 - CategoryID (FK, int, not null)
 - Price (decimal(6,2), null)
 - PublishDate (date, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- dbo.Instructors
- Views

Query window:

```
SELECT *
FROM Room
WHERE nightly_rate > 1000;
```

Results pane:

RoomID	Type	Nightly Rate	BookingID	CustID	Room Available

Message bar: Query executed successfully.

6. Display staff members working as 'Receptionist'.

Object Explorer shows the database structure:

- BankingSystem
- college
- Company_SD
- companyDataBase
- companyDB
- companyDBase
- EDB
- libraryDB
- libraryDBase
- SQL_Aggregation_Functions_Practical_Task
- Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - Temp Tables
 - Temporary Tables
 - dbo.Courses
 - dbo.Enrollments
 - dbo.Courses
 - CourseID (PK, int, not null)
 - Title (varchar(100), not null)
 - InstructorID (FK, int, not null)
 - CategoryID (FK, int, not null)
 - Price (decimal(6,2), null)
 - PublishDate (date, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- dbo.Instructors
- Views

Query window:

```
SELECT *
FROM Staff
WHERE jobtitle = 'Receptionist';
```

Results pane:

StaffID	Name	Salary	Position	BranchID
1	Fatima Said	500.00	Receptionist	1
2	Khalid Omer	600.00	Receptionist	2

Message bar: Query executed successfully.

7. Display bookings made in 2024.

Object Explorer

```

SELECT *
FROM Booking
WHERE YEAR(check_in_date) = 2024;

```

Results

BookingID	CustID	StaffID	check_in_date	check_out_date	status

Query executed successfully.

8. Display bookings ordered by total cost descending.

Object Explorer

```

SELECT b.BookingID, r.nightly_rate
FROM Booking b
JOIN Room r ON b.BookingID = r.BookingID
ORDER BY r.nightly_rate DESC;

```

Results

BookingID	nightly_rate
1	120.00
2	70.00
3	45.00

Query executed successfully.

9. Display the maximum, minimum, and average room price.

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'hotel new.sql - DESKTOP-GILBFIO.hotelDB (DESKTOP-GILBFIO\neuris (53))' is open. The query is:

```

SELECT
    MAX(nightly_rate) AS MaxPrice,
    MIN(nightly_rate) AS MinPrice,
    AVG(nightly_rate) AS AvgPrice
FROM Room;

```

The results pane shows a single row of data:

	MaxPrice	MinPrice	AvgPrice
1	120.00	45.00	70.000000

Below the results, a message says 'Query executed successfully.' The system tray at the bottom indicates it's 12:27 PM on 12/24/2025.

10. Display total number of rooms.

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'hotel new.sql - DESKTOP-GILBFIO.hotelDB (DESKTOP-GILBFIO\neuris (53))' is open. The query is:

```

SELECT COUNT(*) AS TotalRooms
FROM Room;

```

The results pane shows a single row of data:

TotalRooms
5

Below the results, a message says 'Query executed successfully.' The system tray at the bottom indicates it's 12:27 PM on 12/24/2025.

11. Display guests whose names start with 'M'.

Object Explorer

```

SELECT *
FROM Customer
WHERE name LIKE 'M%';

```

Results

CustID	phone	email	name
--------	-------	-------	------

Query executed successfully.

12. Display rooms priced between 800 and 1500.

Object Explorer

```

SELECT *
FROM Room
WHERE nightly_rate BETWEEN 800 AND 1500;

```

Results

RoomID	type	nightly_rate	BookingID	CustID	room_available
--------	------	--------------	-----------	--------	----------------

Query executed successfully.

DML

13. Insert yourself as a guest (Guest ID = 9011).

SET IDENTITY_INSERT Customer ON;

```
INSERT INTO Customer (CusID, phone, email, name)
VALUES (9011, '99999999', 'your@email.com', 'Your Name');
```

SET IDENTITY_INSERT Customer OFF;

(1 row affected)

Completion time: 2025-12-24T12:31:22.1403840+04:00

Query executed successfully.

14. Create a booking for room 205.

```
INSERT INTO Booking (CusID, StaffID, check_in_date, check_out_date)
VALUES (9011, 1, '2025-02-01', '2025-02-05');
```

(1 row affected)

Completion time: 2025-12-24T12:31:52.0050921+04:00

Query executed successfully.

15. Insert another guest with NULL contact and proof details.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. A query window titled "hotel new.sql - DESKTOP-GILBFQI\neurosa (53)" is open, displaying the following SQL code:

```

INSERT INTO Customer (phone, email, name)
VALUES (NULL, NULL, 'Unknown Guest');

```

The results pane shows the output of the query:

- (1 row affected)
- Completion time: 2025-12-24T12:32:38.439Z+04:00
- Query executed successfully.

The Object Explorer on the left shows the database structure, including tables like "Courses", "Instructors", and "Students". The status bar at the bottom indicates the system is "Ready" and shows the date and time as 12/24/2025.

16. Update your booking status to 'Confirmed'.

(Status column does NOT exist → **not applicable**)

17. Increase room prices by 10% for luxury rooms.

Object Explorer shows the database structure for 'hotelDB'.

```
UPDATE Room
SET nightly_rate = nightly_rate * 1.10
WHERE type = 'Suite';
```

(1 row affected)

Completion time: 2025-12-24T12:34:06.7574000+04:00

Query executed successfully.

18. Update booking status to 'Completed' where checkout date is before today.

Object Explorer shows the database structure for 'hotelDB'.

```
--(Status column missing → logical condition only)

SELECT *
FROM Booking
WHERE check_out_date < GETDATE();
```

BookingID	CatID	StaffID	check_in_date	check_out_date	status
1	1	2	2025-01-10	2025-01-15	NULL
2	2	3	2025-01-12	2025-01-14	NULL
3	3	1	2025-01-20	2025-01-25	NULL
4	4	9011	2025-02-01	2025-02-05	Pending

Query executed successfully.

19. Delete bookings with status 'Cancelled' (Status column does NOT exist)