6.5

Results from two SQL queries can be combined in different ways, depending on the data relationship and what you want to achieve. For instance:

- **UNION** combines rows from both queries vertically, returning a unified result set of unique rows.
- **JOIN** combines rows horizontally based on related columns, allowing you to retrieve related data from two tables.
- **INTERSECT** and **EXCEPT** are also used to combine results in a way that filters data based on similarities or differences between datasets.

**INTERSECT**: Returns only the rows that appear in both SELECT queries, effectively finding common data between the two datasets.

**EXCEPT**: Returns only the rows from the first SELECT query that are not present in the second, effectively finding unique data in the first dataset.

6.7

```
Select *
From Hotel
```

6.8

```
Select *
from Hotel
where city = ‘London’;
```

6.9

```
select guest_name, guest_address
from guest
where guest_address LIKE '%London'
order by guest_name;
```

6.10

```
select *
from room
where type = 'Double' or type = 'Family'
and price < 40
order by price asc;
```

6.11

```
select *
from booking
where date_to is null;
```

6.12

```
select count(*) as hotel_count
from hotel;
```

6.13

```
select avg(price) as room_avg_price
from room;
```

6.14

```
select sum(price) as total_revenue
from room
where type = 'Double';
```

6.15

```
select count(distinct guest_no) as total_guests_august
from booking
where date_from between '2024-08-1' and '2024-08-31';
```

6.16

```
select type, price
from ROOM
where HOTEL_NO =
                    (
                    select HOTEL_NO
                    from HOTEL
                    where HOTEL_NAME = 'Grosvenor'
                    );
```

6.17

```
SELECT GUEST_NAME, GUEST_ADDRESS
FROM GUEST
JOIN BOOKING ON GUEST.GUEST_NO = BOOKING.GUEST_NO
WHERE BOOKING.HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME
= 'Grosvenor')
  AND GETDATE() BETWEEN BOOKING.DATE_FROM AND BOOKING.DATE_TO;
```

6.18

```
SELECT ROOM.ROOM_NO, ROOM.TYPE, ROOM.PRICE, GUEST.GUEST_NAME
FROM ROOM
LEFT JOIN BOOKING ON ROOM.ROOM_NO = BOOKING.ROOM_NO
  AND ROOM.HOTEL_NO = BOOKING.HOTEL_NO
LEFT JOIN GUEST ON BOOKING.GUEST_NO = GUEST.GUEST_NO
WHERE ROOM.HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME =
'Grosvenor');
```

6.19

```
SELECT SUM(ROOM.PRICE) AS Total_Income_Today
FROM ROOM
JOIN BOOKING ON ROOM.ROOM_NO = BOOKING.ROOM_NO
WHERE BOOKING.HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME
= 'Grosvenor ')
  AND GETDATE() BETWEEN BOOKING.DATE_FROM AND BOOKING.DATE_TO;
```

6.22

```
SELECT HOTEL_NO, COUNT(*) AS Room_Count
FROM ROOM
GROUP BY HOTEL_NO;
```

6.23

```
SELECT ROOM.HOTEL_NO, COUNT(*) AS Room_Count
FROM ROOM
JOIN HOTEL ON ROOM.HOTEL_NO = HOTEL.HOTEL_NO
WHERE HOTEL.CITY = 'London'
GROUP BY ROOM.HOTEL_NO;
```

6.24

```
SELECT HOTEL_NO, AVG(Booking_Count) AS Avg_Bookings_August
FROM (
   SELECT HOTEL_NO, COUNT(*) AS Booking_Count
   FROM BOOKING
   WHERE MONTH(DATE_FROM) = 8 AND MONTH(DATE_TO) = 8
   GROUP BY HOTEL_NO, GUEST_NO
) AS AugustBookings
GROUP BY HOTEL_NO;
```

6.27

```
INSERT INTO HOTEL (HOTEL_NO, HOTEL_NAME, CITY)
VALUES
   (1, 'Grosvenor', 'London'),
   (2, 'Marriott', 'Manchester'),
   (3, 'Hilton', 'London'),
   (4, 'Holiday Inn', 'Birmingham');

INSERT INTO ROOM (ROOM_NO, HOTEL_NO, TYPE, PRICE)
VALUES
   (101, 1, 'Single', 120.00),
   (102, 1, 'Double', 180.00),
   (103, 2, 'Suite', 300.00),
   (104, 3, 'Family', 250.00),
   (105, 4, 'Double', 150.00);
```

```
INSERT INTO GUEST (GUEST_NO, GUEST_NAME, GUEST_ADDRESS)
VALUES
  (201, 'Alice Smith', '123 Elm St, London'),
  (202, 'John Doe', '456 Oak St, Manchester'),
  (203, 'Emily Jones', '789 Pine St, London'),
  (204, 'Michael Brown', '321 Maple Ave, Birmingham');

INSERT INTO BOOKING (HOTEL_NO, GUEST_NO, DATE_FROM, DATE_TO, ROOM_NO)
VALUES
  (1, 201, '2024-10-01', '2024-10-05', 101),
  (2, 202, '2024-10-10', '2024-10-12', 103),
  (1, 203, '2024-10-15', '2024-10-20', 102),
  (3, 204, '2024-10-25', '2024-10-30', 104);
```

6.28

```
UPDATE ROOM
SET PRICE = PRICE * 1.10;
```

6.20

```
SELECT ROOM.ROOM_NO, ROOM.TYPE, ROOM.PRICE
FROM ROOM
LEFT JOIN BOOKING ON ROOM.ROOM_NO = BOOKING.ROOM_NO
 AND ROOM.HOTEL_NO = BOOKING.HOTEL_NO
WHERE ROOM.HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME =
'Grosvenor')
 AND (BOOKING.DATE_FROM IS NULL OR GETDATE() NOT BETWEEN
BOOKING.DATE_FROM AND BOOKING.DATE_TO);
```

6.21

```
SELECT SUM(PRICE) AS Lost_Income
FROM ROOM
WHERE HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME =
'Grosvenor')
  AND ROOM_NO NOT IN (
    SELECT ROOM_NO
    FROM BOOKING
    WHERE HOTEL_NO = (SELECT HOTEL_NO FROM HOTEL WHERE HOTEL_NAME =
'Grosvenor ')
      AND GETDATE() BETWEEN DATE_FROM AND DATE_TO
  );
```

6.25

```
SELECT HOTEL.HOTEL_NO, ROOM.TYPE, COUNT(ROOM.TYPE) AS Booking_Count
FROM ROOM
JOIN BOOKING ON ROOM.ROOM_NO = BOOKING.ROOM_NO
JOIN HOTEL ON ROOM.HOTEL_NO = HOTEL.HOTEL_NO
WHERE HOTEL.CITY = 'London'
GROUP BY HOTEL.HOTEL_NO, ROOM.TYPE
HAVING COUNT(ROOM.TYPE) = (
  SELECT MAX(Type_Count)
  FROM (
    SELECT COUNT(ROOM.TYPE) AS Type_Count
    FROM ROOM
    JOIN BOOKING ON ROOM.ROOM_NO = BOOKING.ROOM_NO
    WHERE ROOM.HOTEL_NO = HOTEL.HOTEL_NO
```

```
      GROUP BY ROOM.TYPE
   ) AS TypeCounts
);
```

6.26

```
SELECT HOTEL_NO, SUM(PRICE) AS Lost_Income
FROM ROOM
WHERE ROOM_NO NOT IN (
   SELECT ROOM_NO
   FROM BOOKING
   WHERE CURDATE() BETWEEN DATE_FROM AND DATE_TO
)
GROUP BY HOTEL_NO;
```