

Day 3 – Subqueries, Set Operators & CTEs

```
CREATE DATABASE AirlinesDB;
USE AirlinesDB;

CREATE TABLE air_passenger_profile (
    profile_id      INT IDENTITY(1,1) PRIMARY KEY,
    password        VARCHAR(20) NOT NULL,
    first_name      VARCHAR(20) NOT NULL,
    last_name       VARCHAR(20) NOT NULL,
    address         VARCHAR(100),
    mobile_number   BIGINT,
    email_id        VARCHAR(30)
);

CREATE TABLE air_flight (
    flight_id       INT IDENTITY(1,1) PRIMARY KEY,
    airline_id      VARCHAR(10) NOT NULL,
    airline_name    VARCHAR(30),
    from_location   VARCHAR(20),
    to_location     VARCHAR(20),
    departure_time  TIME,
    arrival_time    TIME,
    duration_time   TIME,
    total_seats     INT
);

CREATE TABLE air_flight_details (
    flight_id       INT NOT NULL,
    flight_departure_date DATE,
    price           DECIMAL(8,2),
    available_seats INT,

    CONSTRAINT FK_flightdetails_flight
        FOREIGN KEY (flight_id)
        REFERENCES air_flight(flight_id)
);

CREATE TABLE air_credit_card_details (
    card_id         INT IDENTITY(1,1) PRIMARY KEY,
    profile_id      INT NOT NULL,
    card_number     BIGINT UNIQUE,
    card_type       VARCHAR(10),
    expiration_month INT,
    expiration_year INT,

    CONSTRAINT FK_card_passenger
        FOREIGN KEY (profile_id)
        REFERENCES air_passenger_profile(profile_id)
);
```

```

CREATE TABLE air_ticket_info (
    ticket_id          INT IDENTITY(1,1) PRIMARY KEY,
    profile_id         INT NOT NULL,
    flight_id          INT NOT NULL,
    flight_departure_date DATE,
    status             VARCHAR(15),

    CONSTRAINT FK_ticket_passenger
        FOREIGN KEY (profile_id)
        REFERENCES air_passenger_profile(profile_id),

    CONSTRAINT FK_ticket_flight
        FOREIGN KEY (flight_id)
        REFERENCES air_flight(flight_id)
);

INSERT INTO air_passenger_profile
(password, first_name, last_name, address, mobile_number, email_id)
VALUES
('p1', 'Arun', 'Kumar', 'Chennai', 9876543210, 'arun@mail.com'),
('p2', 'Bhavya', 'Reddy', 'Hyderabad', 9123456780, 'bhavya@mail.com'),
('p3', 'Charan', 'Singh', 'Bangalore', 9988776655, 'charan@mail.com'),
('p4', 'Divya', 'Sharma', 'Delhi', 9012345678, 'divya@mail.com');

INSERT INTO air_flight
(airline_id, airline_name, from_location, to_location, departure_time,
arrival_time, duration_time, total_seats)
VALUES
('ABC01', 'ABC
Airlines', 'Chennai', 'Hyderabad', '08:00', '09:30', '01:30', 180),
('ABC02', 'ABC
Airlines', 'Hyderabad', 'Delhi', '10:00', '12:30', '02:30', 160),
('ABC03', 'ABC
Airlines', 'Chennai', 'Bangalore', '14:00', '15:30', '01:30', 150),
('IN01', 'Indigo', 'Delhi', 'Mumbai', '16:00', '18:00', '02:00', 180);

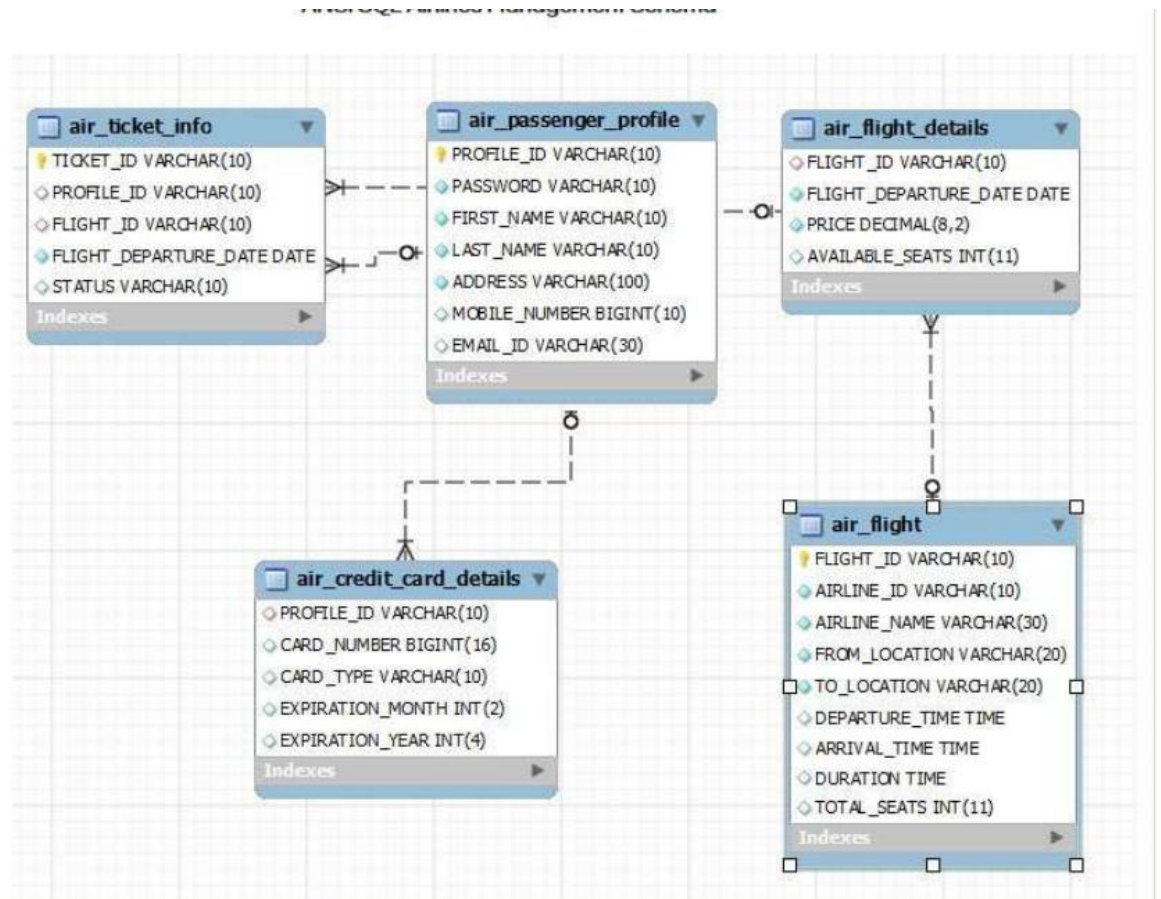
INSERT INTO air_flight_details
(flight_id, flight_departure_date, price, available_seats)
VALUES
(1, '2025-01-05', 3500, 120),
(1, '2025-02-10', 3600, 115),
(1, '2025-04-15', 3800, 110),
(2, '2025-01-20', 5200, 100),
(2, '2025-04-12', 5400, 95),
(3, '2025-02-08', 3000, 130),
(3, '2025-04-22', 3200, 125);

```

```

INSERT INTO air_ticket_info
(profile_id, flight_id, flight_departure_date, status)
VALUES
(1,1, '2025-01-05', 'BOOKED'),
(1,1, '2025-02-10', 'BOOKED'),
(2,1, '2025-04-15', 'BOOKED'),
(2,2, '2025-01-20', 'BOOKED'),
(3,2, '2025-04-12', 'BOOKED'),
(1,3, '2025-02-08', 'BOOKED'),
(4,3, '2025-04-22', 'BOOKED');

```



1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id, From_location, To_location, Month Name as "Month_Name" and average price as "Average_Price". Display the records sorted in ascending order based on flight id and then by Month Name.

```
select
    f.flight_id,
    f.from_location,
    f.to_location,
    datename(month, fd.flight_departure_date) as month_name,
    avg(fd.price) as average_price
from air_flight f
join air_flight_details fd
    on f.flight_id = fd.flight_id
where f.airline_name = 'ABC Airlines'
group by
    f.flight_id,
    f.from_location,
    f.to_location,
    datename(month, fd.flight_departure_date),
    month(fd.flight_departure_date)
order by
    f.flight_id,
    month(fd.flight_departure_date);
```

Results

Messages

	flight_id	from_location	to_location	month_name	average_price
1	1	Chennai	Hyderabad	January	3500.000000
2	1	Chennai	Hyderabad	February	3600.000000
3	1	Chennai	Hyderabad	April	3800.000000
4	2	Hyderabad	Delhi	January	5200.000000
5	2	Hyderabad	Delhi	April	5400.000000
6	3	Chennai	Bangalore	February	3000.000000
7	3	Chennai	Bangalore	April	3200.000000

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets". Display the records sorted in ascending order based on customer's first name.

Query:

```
select
    p.profile_id,
    p.first_name,
    p.address,
    count(t.ticket_id) as no_of_tickets
from air_passenger_profile p
join air_ticket_info t
    on p.profile_id = t.profile_id
join air_flight f
    on t.flight_id = f.flight_id
where f.airline_name = 'ABC Airlines'
group by
    p.profile_id,
    p.first_name,
    p.address
having count(t.ticket_id) = (
    select min(ticket_count)
    from (
        select count(t2.ticket_id) as ticket_count
        from air_ticket_info t2
        join air_flight f2
            on t2.flight_id = f2.flight_id
        where f2.airline_name = 'ABC Airlines'
        group by t2.profile_id
    ) as x
)
order by p.first_name;
```

Results		Messages		
	profile_id	first_name	address	no_of_tickets
1	3	Charan	Bangalore	1
2	4	Divya	Delhi	1

3. Write a query to display the number of flight services between locations in a month. The query should display from_location, to_location, month as “month_name” and number of flight services as “no_of_services”.

The records should be displayed in ascending order based on from_location, then by to_location, and then by month name.

```
select
    f.from_location,
    f.to_location,
    datename(month, fd.flight_departure_date) as month_name,
    count(fd.flight_departure_date) as no_of_services
from air_flight f
join air_flight_details fd
    on f.flight_id = fd.flight_id
group by
    f.from_location,
    f.to_location,
    datename(month, fd.flight_departure_date),
    month(fd.flight_departure_date)
order by
    f.from_location,
    f.to_location,
    month(fd.flight_departure_date);
```

Results		Messages		
	from_location	to_location	month_name	no_of_services
1	Chennai	Bangalore	February	1
2	Chennai	Bangalore	April	1
3	Chennai	Hyderabad	January	1
4	Chennai	Hyderabad	February	1
5	Chennai	Hyderabad	April	1
6	Hyderabad	Delhi	January	1
7	Hyderabad	Delhi	April	1

4. Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The query should display profile_id, customer's first_name, address and number of tickets booked as "no_of_tickets". Display the records in ascending order based on customer's first name.

```
select
    p.profile_id,
    p.first_name,
    p.address,
    count(t.ticket_id) as no_of_tickets
from air_passenger_profile p
join air_ticket_info t
    on p.profile_id = t.profile_id
join air_flight f
    on t.flight_id = f.flight_id
where f.airline_name = 'ABC Airlines'
group by
    p.profile_id,
    p.first_name,
    p.address
having count(t.ticket_id) = (
    select max(ticket_count)
    from (
        select count(t2.ticket_id) as ticket_count
        from air_ticket_info t2
        join air_flight f2
            on t2.flight_id = f2.flight_id
        where f2.airline_name = 'ABC Airlines'
        group by t2.profile_id
    ) x
)
order by p.first_name;
```

Results		Messages		
	profile_id	first_name	address	no_of_tickets
1	1	Arun	Chennai	3

5. Write a query to display the number of tickets booked from Chennai to Hyderabad. The query should display profile_id, first_name, last_name, flight_id, departure_date and number of tickets booked as “no_of_tickets”. Display the records sorted in ascending order based on profile id, flight id and departure date.

```
select
    p.profile_id,
    p.first_name,
    p.last_name,
    f.flight_id,
    t.flight_departure_date as departure_date,
    count(t.ticket_id) as no_of_tickets
from air_ticket_info t
join air_passenger_profile p
    on t.profile_id = p.profile_id
join air_flight f
    on t.flight_id = f.flight_id
where
    f.from_location = 'Chennai'
    and f.to_location = 'Hyderabad'
group by
    p.profile_id,
    p.first_name,
    p.last_name,
    f.flight_id,
    t.flight_departure_date
order by
    p.profile_id,
    f.flight_id,
    t.flight_departure_date;
```

Results		Messages				
	profile_id	first_name	last_name	flight_id	departure_date	no_of_tickets
1	1	Arun	Kumar	1	2025-01-05	1
2	1	Arun	Kumar	1	2025-02-10	1
3	2	Bhavya	Reddy	1	2025-04-15	1

6. Write a query to display flight id, from location, to location and ticket price of flights whose departure is in the month of april.

```
select
    f.flight_id,
    f.from_location,
    f.to_location,
    fd.price
from air_flight f
join air_flight_details fd
    on f.flight_id = fd.flight_id
where month(fd.flight_departure_date) = 4;
```

	flight_id	from_location	to_location	price
1	1	Chennai	Hyderabad	3800.00
2	2	Hyderabad	Delhi	5400.00
3	3	Chennai	Bangalore	3200.00

7. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight_id, from_location, to_location and average price as "price". Display the records sorted in ascending order based on flight id, from_location and to_location.

```
select
    f.flight_id,
    f.from_location,
    f.to_location,
    avg(fd.price) as price
from air_flight f
join air_flight_details fd
    on f.flight_id = fd.flight_id
group by
    f.flight_id,
    f.from_location,
    f.to_location
order by
    f.flight_id,
    f.from_location,
    f.to_location;
```

	flight_id	from_location	to_location	price
1	1	Chennai	Hyderabad	3633.333333
2	2	Hyderabad	Delhi	5300.000000
3	3	Chennai	Bangalore	3100.000000

8. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (first_name and last_name separated by comma) and address. Display only unique customers and sort the records in ascending order based on profile id.

```
select distinct
  p.profile_id,
  p.first_name + ',' + p.last_name as customer_name,
  p.address
from air_passenger_profile p
join air_ticket_info t
  on p.profile_id = t.profile_id
join air_flight f
  on t.flight_id = f.flight_id
where
  f.from_location = 'Chennai'
  and f.to_location = 'Hyderabad'
order by p.profile_id;
```

Results Messages

	profile_id	customer_name	address
1	1	Arun,Kumar	Chennai
2	2	Bhawa.Reddy	Hyderabad

9. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```
select
  profile_id
from air_ticket_info
group by profile_id
having count(ticket_id) = (
  select max(ticket_count)
  from (
    select count(ticket_id) as ticket_count
    from air_ticket_info
    group by profile_id
  ) x
)
order by profile_id;
```

Results

Messages

	profile_id
1	1

10. Write a query to display the total number of tickets booked in each flight in ABC Airlines. The query should display flight_id, from_location, to_location and number of tickets as “no_of_tickets”.

Display only flights with at least one ticket booked and sort the records in ascending order by flight id.

```
select
    f.flight_id,
    f.from_location,
    f.to_location,
    count(t.ticket_id) as no_of_tickets
from air_flight f
join air_ticket_info t
    on f.flight_id = t.flight_id
where f.airline_name = 'ABC Airlines'
group by
    f.flight_id,
    f.from_location,
    f.to_location
having count(t.ticket_id) >= 1
order by f.flight_id;
```

Results	Messages	Navigate to the previous error (Alt+PgUp)		
	flight_id	from_location	to_location	no_of_tickets
1	1	Chennai	Hyderabad	3
2	2	Hyderabad	Delhi	2
3	3	Chennai	Bangalore	2