



# **Dhirubhai Ambani University**

## Database Management System (IT214)

### Group:5

#### Group Representative:

1) Bavarva Megh Dineshbhai (202301402)

Contact number: 8849855886

#### Group Members:

1) Patel Naman Vipulbhai (202301423)

2) Patel Het Jitendrakumar (202301421)

# QUERIES

## 1.) List all crew members with their names and assigned flight numbers:

```
SELECT E.First_name, E.Last_name, C.Role,  
F.flight_no  
FROM Crew C  
JOIN Employee E ON C.E_id = E.E_id  
JOIN Assigned A ON C.Crew_id = A.Crew_id  
JOIN Flights F ON A.flight_id = F.flight_id;
```

## 2.) Count how many flights depart from each airport:

```
SELECT departure_airport, COUNT(*) AS  
total_flights  
FROM Flights  
GROUP BY departure_airport;
```

## 3.) Count passengers by nationality:

```
SELECT Nationality, COUNT(*) AS num_passengers  
FROM Passenger  
GROUP BY Nationality;
```

## 4.) Most Popular Meal Ordered (based on bookings)

```
SELECT  
    M.meal_type,  
    COUNT(*) AS orders  
FROM Booking B  
JOIN Meals M ON B.meal_id = M.meal_id  
GROUP BY M.meal_type
```

```
ORDER BY orders DESC
LIMIT 1;
```

### 5.) Revenue per Flight and Currency Breakdown

```
SELECT
    F.flight_no,
    PY.currency,
    SUM(PY.amount) AS total_revenue
FROM Payments PY
JOIN Booking B ON PY.booking_id = B.booking_id
JOIN Flights F ON B.flight_id = F.flight_id
GROUP BY F.flight_no, PY.currency
ORDER BY total_revenue DESC;
```

### 6.) Average Delay Between Departure and Arrival per Aircraft Model.

```
SELECT
    A.Manufacturer,
    A.Model,
    ROUND(AVG(EXTRACT(EPOCH FROM (F.arrival_time -
F.departure_time)) / 3600), 2) AS
avg_duration_hours
FROM Flights F
JOIN Aircraft A ON F.Aircraft_id = A.Aircraft_id
GROUP BY A.Manufacturer, A.Model
ORDER BY avg_duration_hours DESC;
```

### 7.) Show passengers who checked in more than 2 bags:

```
SELECT P.p_id, P.First_Name, P.Last_Name
FROM Check_in C
JOIN Passenger P ON C.p_id = P.p_id
```

```
WHERE C.baggage_count > 2;
```

8.) Show bookings where loyalty ID was used in the payment:

```
SELECT PY.payment_id, B.booking_id, P.First_Name,  
FF.Membership_no  
FROM Payments PY  
JOIN Booking B ON PY.booking_id = B.booking_id  
JOIN Passenger P ON PY.p_id = P.p_id  
JOIN FrequentFlyer FF ON PY.loyalty_id =  
FF.loyalty_id;
```

9.) Find passengers who have booked more than 2 flights:

```
SELECT P.p_id, P.First_Name, P.Last_Name  
FROM Passenger P  
WHERE (  
    SELECT COUNT(*)  
    FROM Booking B  
    WHERE B.p_id = P.p_id  
) > 2;
```

10.) Show passenger names and their meal types for bookings:

```
SELECT P.First_Name, P.Last_Name, M.meal_type  
FROM Booking B  
JOIN Passenger P ON B.p_id = P.p_id  
JOIN Meals M ON B.meal_id = M.meal_id;
```

11.) List all crew members with their names and assigned flight numbers:

```

SELECT E.First_name, E.Last_name, C.Role,
F.flight_no
FROM Crew C
JOIN Employee E ON C.E_id = E.E_id
JOIN Assigned A ON C.Crew_id = A.Crew_id
JOIN Flights F ON A.flight_id = F.flight_id;

```

## 12.) Most Frequent Flight Route (Source-Destination Pair) and Number of Bookings

```

SELECT
    source,
    destination,
    COUNT(*) AS total_bookings
FROM Booking
GROUP BY source, destination
ORDER BY total_bookings DESC
LIMIT 1;

```

## 13.) CTE: Ranking Passengers by Total Spend (WITH RANKING)

```

WITH PassengerSpending AS (
    SELECT
        P.p_id,
        P.First_Name,
        P.Last_Name,
        SUM(PY.amount) AS total_spent
    FROM Passenger P
    JOIN Payments PY ON P.p_id = PY.p_id
    GROUP BY P.p_id, P.First_Name, P.Last_Name
),
RankedSpenders AS (
    SELECT *,

```

```

            RANK() OVER (ORDER BY total_spent DESC)
AS spend_rank
    FROM PassengerSpending
)
SELECT *
FROM RankedSpenders
WHERE spend_rank <= 10;

```

#### 14. Flight Occupancy Analysis

**Purpose:** Show flights with occupancy rates >80%  
along with aircraft details

```

SELECT
    f.flight_no,
    a.model AS aircraft_model,
    a.capacity,
    (a.capacity - f.available_economy_seat -
f.available_business_seat) AS occupied_seats,
    ROUND((1 - (f.available_economy_seat +
f.available_business_seat)::NUMERIC/a.capacity) *
100, 2) AS occupancy_rate
FROM flights f
JOIN aircraft a ON f.aircraft_id = a.aircraft_id
WHERE (a.capacity - f.available_economy_seat -
f.available_business_seat)::FLOAT/a.capacity > 0.8
ORDER BY occupancy_rate DESC;

```

#### 15. Premium Passenger Spending Habits

**Purpose:** Identify business class passengers who  
spent above average on meals

```

WITH business_class_meals AS (
    SELECT

```

```

        b.p_id,
        SUM(m.price_usd) AS total_meal_spending
FROM booking b
JOIN meals m ON b.meal_id = m.meal_id
WHERE b.business = TRUE
GROUP BY b.p_id
)
SELECT
    p.first_name || ' ' || p.last_name AS
passenger_name,
    bcm.total_meal_spending,
    ff.membership_tier
FROM business_class_meals bcm
JOIN passenger p ON bcm.p_id = p.p_id
LEFT JOIN frequentflyer ff ON p.p_id = ff.p_id
WHERE bcm.total_meal_spending > (SELECT
AVG(total_meal_spending) FROM
business_class_meals)
ORDER BY bcm.total_meal_spending DESC;

```

## 16. Crew Workload Analysis

**Purpose:** Find crew members working multiple flights on the same day

```

SELECT
    e.first_name || ' ' || e.last_name AS
crew_name,
    c.role,
    COUNT(DISTINCT f.flight_id) AS
flights_assigned,
    ARRAY_AGG(DISTINCT f.flight_no) AS
flight_numbers,
    f.departure_time::DATE AS operation_date
FROM assigned a

```

```

JOIN crew c ON a.crew_id = c.crew_id
JOIN employee e ON c.e_id = e.e_id
JOIN flights f ON a.flight_id = f.flight_id
GROUP BY crew_name, c.role, operation_date, e.e_id
HAVING COUNT(DISTINCT f.flight_id) > 1
ORDER BY operation_date, flights_assigned DESC;

```

## 17. International Flight Performance

**Purpose:** Compare average ratings for international vs domestic flights

```

WITH flight_categories AS (
    SELECT
        f.flight_id,
        CASE WHEN dep.country <> arr.country THEN
            'International' ELSE 'Domestic' END AS flight_type
    FROM flights f
    JOIN airports dep ON f.departure_airport =
        dep.airport_name
    JOIN airports arr ON f.arrival_airport =
        arr.airport_name
)
SELECT
    fc.flight_type,
    ROUND(AVG(r.rating), 2) AS avg_rating,
    COUNT(r.review_id) AS total_reviews,
    PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY
        r.rating) AS median_rating
FROM flight_categories fc
JOIN reviews r ON fc.flight_id = r.flight_id
GROUP BY fc.flight_type;

```

## 18. Passenger Journey Patterns



**Purpose:** Find passengers with complex itineraries  
(multiple connecting flights)

```
WITH passenger_journeys AS (  
    SELECT  
        p.p_id,  
        b1.source AS start_point,  
        b2.destination AS end_point,  
        b1.destination AS connection_point,  
        b1.booking_date AS first_flight_date,  
        b2.booking_date AS second_flight_date  
    FROM booking b1  
    JOIN booking b2 ON b1.p_id = b2.p_id  
        AND b1.destination = b2.source  
        AND b2.booking_date BETWEEN  
b1.booking_date AND b1.booking_date + INTERVAL '3  
days'  
    JOIN passenger p ON b1.p_id = p.p_id  
)  
SELECT  
    p.first_name || ' ' || p.last_name AS  
passenger_name,  
    j.start_point,  
    j.connection_point,  
    j.end_point,  
    j.first_flight_date,  
    j.second_flight_date  
FROM passenger_journeys j  
JOIN passenger p ON j.p_id = p.p_id  
WHERE j.start_point <> j.end_point;
```

## 19.) Peak Booking Time Analysis

**Purpose:** Identify hourly booking patterns and peak  
times

```

WITH booking_hours AS (
    SELECT
        EXTRACT(HOUR FROM transaction_date) AS
hour,
        COUNT(*) AS total_bookings,
        SUM(amount) AS total_revenue
    FROM payments
    GROUP BY EXTRACT(HOUR FROM transaction_date)
SELECT
    h.hour,
    COALESCE(b.total_bookings, 0) AS bookings,
    COALESCE(b.total_revenue, 0) AS revenue,
    ROUND(PERCENT_RANK() OVER (ORDER BY
b.total_bookings) * 100, 2) AS percentile_rank
FROM generate_series(0,23) h(hour)
LEFT JOIN booking_hours b ON h.hour = b.hour
ORDER BY h.hour;

```

## 20.) Crew Experience vs Flight Ratings

**Purpose:** Analyze relationship between crew experience and flight ratings

```

WITH crew_experience AS (
    SELECT
        a.flight_id,
        AVG(c.experience) AS avg_crew_experience,
        PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER
BY c.experience) AS median_experience
    FROM assigned a
    JOIN crew c ON a.crew_id = c.crew_id
    GROUP BY a.flight_id
)
SELECT
    CASE

```

```

        WHEN ce.avg_crew_experience < 2 THEN
'Novice Crew'
        WHEN ce.avg_crew_experience BETWEEN 2 AND
5 THEN 'Experienced Crew'
        ELSE 'Veteran Crew'
    END AS experience_group,
    ROUND(AVG(r.rating), 2) AS avg_rating,
    COUNT(DISTINCT r.flight_id) AS rated_flights
FROM crew_experience ce
JOIN reviews r ON ce.flight_id = r.flight_id
GROUP BY experience_group
ORDER BY avg_rating DESC;

```

## 21.) Passenger Booking Frequency Analysis

**Purpose:** Identify frequent bookers within specific time windows

```

WITH booking_intervals AS (
    SELECT
        p_id,
        booking_date,
        LAG(booking_date) OVER (PARTITION BY p_id
ORDER BY booking_date) AS prev_booking
    FROM bookings
)
SELECT
    p.first_name || ' ' || p.last_name AS
passenger,
    COUNT(*) AS total_bookings,
    AVG(bi.booking_date - bi.prev_booking) AS
avg_days_between_bookings,
    MIN(bi.booking_date - bi.prev_booking) AS
min_interval,

```

```
        MAX(bi.booking_date - bi.prev_booking) AS
max_interval
FROM booking_intervals bi
JOIN passenger p ON bi.p_id = p.p_id
WHERE bi.prev_booking IS NOT NULL
        AND bi.booking_date - bi.prev_booking <= 7
GROUP BY p.p_id
HAVING COUNT(*) > 3
ORDER BY total_bookings DESC;
```

## 22. Flight Review Summary

```
SELECT f.flight_id, f.flight_no, AVG(r.Rating) AS
avg_rating, COUNT(r.Review_id) AS review_count,
MIN(r.Rating) AS min_rating, MAX(r.Rating) AS
max_rating
FROM Flights f
JOIN Reviews r ON f.flight_id = r.flight_id
GROUP BY f.flight_id, f.flight_no
ORDER BY avg_rating DESC;
```

## 23. Employee Count per Department

```
SELECT Department, COUNT(*) AS total_employees
FROM Employee
GROUP BY Department
ORDER BY total_employees DESC;
```

## 24. Top 5 Airports with Most Departing Flights

```
SELECT departure_airport, COUNT(*) AS flight_count
FROM Flights
GROUP BY departure_airport
ORDER BY flight_count DESC
LIMIT 5;
```

## 25. Flights with Longest Distance and Aircraft Capacity

```
SELECT f.flight_id, f.flight_no, f.distance,  
a.Manufacturer, a.Model, a.Capacity  
FROM Flights f  
JOIN Aircraft a ON f.Aircraft_id = a.Aircraft_id  
ORDER BY f.distance DESC  
LIMIT 10;
```

## 26. Frequent Flyer Memberships Expiring Soon

```
SELECT ff.loyalty_id, ff.Membership_no,  
ff.Membership_tier, ff.p_id, ff.Valid_to  
FROM FrequentFlyer ff  
WHERE ff.Valid_to BETWEEN CURRENT_DATE AND  
CURRENT_DATE + INTERVAL '30 days'  
ORDER BY ff.Valid_to;
```

# Thank You

