10 Uber SQL Interview Questions & Solutions

Compiled from DataLemur and Common Uber Interview Topics

Question 1: Third Ride

Scenario: Analyze the behavior of users on their third-ever ride.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
user_id	integer	ID of the user.
ride_date	timestamp	Date and time of the ride.
ride_status	varchar	e.g., completed, cancelled.

Question: Write a query to find the third ride of every user who has taken at least 3 rides.

Solution:

```sql
WITH ride_rankings AS (
  SELECT
    user_id,
    ride_id,
    ride_date,
    RANK() OVER (
      PARTITION BY user_id
      ORDER BY ride_date
    ) as ride_rank
  FROM rides
  WHERE ride_status = 'completed'
)
SELECT
  user_id,
  ride_id,
  ride_date
FROM ride_rankings
WHERE ride_rank = 3;
```

Question 2: Top Drivers by Average Rating

Scenario: Identify the most reliable drivers to reward them.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
driver_id	integer	ID of the driver.
rating	integer	Rating from user (1-5).
ride_status	varchar	Status of the ride.

Question: Find the top 5 drivers with the highest average rating who have completed at least 10 rides.

Solution:

```sql
SELECT
  driver_id,
  ROUND(AVG(rating), 2) as avg_rating,
  COUNT(ride_id) as total_rides
FROM rides
WHERE ride_status = 'completed'
GROUP BY driver_id
HAVING COUNT(ride_id) >= 10
ORDER BY avg_rating DESC
LIMIT 5;
```

Question 3: Monthly Growth Rate

Scenario: The growth team wants to track business performance.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
ride_date	timestamp	Date and time of the ride.
ride_status	varchar	Status of the ride.

Question: Calculate the month-over-month percentage growth rate of completed rides.

Solution:

```sql
WITH monthly_rides AS (
  SELECT
    DATE_TRUNC('month', ride_date) as month,
    COUNT(ride_id) as ride_count
  FROM rides
  WHERE ride_status = 'completed'
  GROUP BY 1
)
SELECT
  TO_CHAR(month, 'YYYY-MM') AS year_month,
  ride_count,
  LAG(ride_count) OVER (ORDER BY month) as prev_month_rides,
  ROUND(
    (ride_count - LAG(ride_count) OVER (ORDER BY month)) * 100.0 /
    NULLIF(LAG(ride_count) OVER (ORDER BY month), 0), 2
  ) as growth_rate_pct
FROM monthly_rides
ORDER BY month;
```

Question 4: Driver Cancellation Rate

Scenario: Operations needs to flag drivers with high cancellation rates.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
driver_id	integer	ID of the driver.
ride_status	varchar	e.g., completed, cancelled_by_driver.

Question: Find all drivers with a cancellation rate (cancelled by them) higher than 10%.

Solution:

```sql
SELECT
```

```sql
  driver_id,
  COUNT(ride_id) as total_rides,
  COUNT(CASE WHEN ride_status = 'cancelled_by_driver' THEN 1 END) as cancelled_rides,
  ROUND(
    COUNT(CASE WHEN ride_status = 'cancelled_by_driver' THEN 1 END) * 100.0 /
    COUNT(ride_id),
  2) as cancellation_rate_pct
FROM rides
GROUP BY driver_id
HAVING COUNT(CASE WHEN ride_status = 'cancelled_by_driver' THEN 1 END) * 100.0 / COUNT(ride_id) > 10;
```

Question 5: User with the Most Rides in a Rolling 7-Day Period

Scenario: Identify highly active users for a marketing campaign.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
user_id	integer	ID of the user.
ride_date	date	Date of the ride.

Question: For each user, find the maximum number of rides they ever took in any 7-day rolling window.

Solution:

```sql
WITH daily_rides AS (
  SELECT
```

```
    user_id,
    ride_date,
    COUNT(ride_id) AS rides_on_day
  FROM rides
  GROUP BY user_id, ride_date
),
rolling_counts AS (
  SELECT
    user_id,
    ride_date,
    SUM(rides_on_day) OVER (
      PARTITION BY user_id
      ORDER BY ride_date
      RANGE BETWEEN INTERVAL '6 days' PRECEDING AND CURRENT ROW
    ) AS rides_in_7d
  FROM daily_rides
)
SELECT
  user_id,
  MAX(rides_in_7d) AS max_rides_in_7d
FROM rolling_counts
GROUP BY user_id
ORDER BY max_rides_in_7d DESC;
```

Question 6: Average Trip Distance by Weather Condition

Scenario: Analyze how weather affects trip behavior.

Table: trips

Column Name	Type	Description
trip_id	integer	Unique ID for the trip.
distance_miles	numeric	Distance traveled.
start_time	timestamp	Start time of the trip.

Table: weather

Column Name	Type	Description
time	timestamp	Time of weather record.
condition	varchar	e.g., Rain, Clear, Snow.

Question: Calculate the average trip distance for each weather condition.

Solution:

```sql
SELECT
  w.condition,
  ROUND(AVG(t.distance_miles), 2) AS avg_distance_miles,
  COUNT(t.trip_id) AS number_of_trips
FROM trips t
JOIN weather w
  ON DATE_TRUNC('hour', t.start_time) = DATE_TRUNC('hour', w.time)
GROUP BY w.condition
ORDER BY number_of_trips DESC;
```

Question 7: Premium vs. Economy Rides

Scenario: Finance wants to compare the revenue from different service tiers.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.
service_type	varchar	premium or economy.
fare	numeric	Amount charged for the ride.
ride_date	date	Date of the ride.

Question: Calculate the total fare amount for each service type for the current year.

Solution:

```sql
SELECT
  service_type,
  SUM(fare) AS total_fare,
  COUNT(ride_id) AS total_rides
FROM rides
WHERE EXTRACT(YEAR FROM ride_date) = EXTRACT(YEAR FROM CURRENT_DATE)
GROUP BY service_type;
```

Question 8: First Ride for Each User

Scenario: The onboarding team wants to analyze a user's first experience.

Table: rides

Column Name	Type	Description
ride_id	integer	Unique ID for the ride.

user_id	integer	ID of the user.
ride_date	timestamp	Date and time of the ride.

Question: For each user, find the details of their very first Uber ride.

Solution:

```sql
WITH first_rides AS (
  SELECT
    user_id,
    ride_id,
    ride_date,
    RANK() OVER (
      PARTITION BY user_id
      ORDER BY ride_date
    ) as ride_rank
  FROM rides
)
SELECT
  user_id,
  ride_id,
  ride_date
FROM first_rides
WHERE ride_rank = 1;
```

Question 9: Rides with Above-Average Duration

Scenario: Identify unusually long or short rides for further analysis.

Table: trips

Column Name	Type	Description
trip_id	integer	Unique ID for the trip.
duration_min	numeric	Duration of trip in minutes.

Question: Find all trips that have a duration higher than the overall average trip duration.

Solution:

```sql
SELECT
  trip_id,
  duration_min
FROM trips
WHERE duration_min > (SELECT AVG(duration_min) FROM trips)
ORDER BY duration_min DESC;
```

Question 10: Most Popular Pick-Up Locations

Scenario: Help the operations team decide where to position drivers.

Table: trips

Column Name	Type	Description
trip_id	integer	Unique ID for the trip.
pickup_location_id	integer	ID of the pickup location.
pickup_time	date	Date and time of pickup.

Question: Find the top 5 most popular pickup locations in the last month.

Solution:

```sql
SELECT
  pickup_location_id,
  COUNT(trip_id) AS number_of_trips
FROM trips
WHERE pickup_time >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY pickup_location_id
ORDER BY number_of_trips DESC
LIMIT 5;
```