BIGQUERY

1. BUSIEST PICKUP LOCATIONS:

SELECT p.pickup\_latitude, p.pickup\_longitude, COUNT(\*) AS rides

FROM `uber\_data.fact\_table` f JOIN `uber\_data.pickup\_location\_dim` p ON f.pickup\_location\_id = p.pickup\_location\_id

GROUP BY p.pickup\_latitude, p.pickup\_longitude

ORDER BY rides DESC LIMIT 10

1. TOTAL TRIPS PER DAY:

SELECT d.pick\_day, d.pick\_month, d.pick\_year, COUNT(\*) AS total\_trips FROM `uber\_data.fact\_table` f JOIN `uber\_data.datetime\_dim` d ON f.datetime\_id = d.datetime\_id

GROUP BY d.pick\_year, d.pick\_month, d.pick\_day

1. TOTAL REVENUE PER DAY:

SELECT d.pick\_day, d.pick\_month, d.pick\_year, SUM(total\_amount) AS total\_revenue FROM `uber\_data.fact\_table` f JOIN `uber\_data.datetime\_dim` d ON f.datetime\_id = d.datetime\_id

GROUP BY d.pick\_day, d.pick\_month, d.pick\_year

ORDER BY d.pick\_day, d.pick\_month, d.pick\_year

1. PEAK HOURS FOR RIDES:

SELECT d.pick\_hour, COUNT(\*) AS rides FROM `uber\_data.fact\_table` f JOIN `uber\_data.datetime\_dim` d ON f.datetime\_id = d.datetime\_id

GROUP BY d.pick\_hour ORDER BY rides DESC

1. PAYMENT TYPE DISTRIBUTION:

SELECT pt.payment\_type\_name, COUNT(\*) as rides, (COUNT(\*) \* 100.0 / SUM(COUNT(\*)) OVER()) AS percentage

FROM `uber\_data.fact\_table` f JOIN `uber\_data.payment\_type\_dim` pt ON f.payment\_type\_id = pt.payment\_type\_id

GROUP BY pt.payment\_type\_name

ORDER BY rides DESC

1. NUMBER OF TRIPS PER PASSENGER COUNT:

SELECT

pc.passenger\_count,

COUNT(\*) AS ride\_count

FROM

`uber\_data.fact\_table` f JOIN

`uber\_data.passenger\_count\_dim` pc

ON f.passenger\_count\_id = pc.passenger\_count\_id

GROUP BY pc.passenger\_count

ORDER BY pc.passenger\_count

1. LONGEST AND SHORTEST TRIPS:

SELECT f.trip\_id, t.trip\_distance, p.pickup\_latitude, p.pickup\_longitude, d.dropoff\_latitude, d.dropoff\_longitude

FROM `uber\_data.fact\_table` f JOIN `uber\_data.trip\_distance\_dim` t ON f.trip\_distance\_id = t.trip\_distance\_id

JOIN `uber\_data.pickup\_location\_dim` p ON f.pickup\_location\_id = p.pickup\_location\_id

JOIN `uber\_data.dropoff\_location\_dim` d ON f.dropoff\_location\_id = d.dropoff\_location\_id

ORDER BY t.trip\_distance DESC LIMIT 10; -- for longest trips

SELECT f.trip\_id, t.trip\_distance, p.pickup\_latitude, p.pickup\_longitude, d.dropoff\_latitude, d.dropoff\_longitude

FROM `uber\_data.fact\_table` f JOIN `uber\_data.trip\_distance\_dim` t ON f.trip\_distance\_id = t.trip\_distance\_id

JOIN `uber\_data.pickup\_location\_dim` p ON f.pickup\_location\_id = p.pickup\_location\_id

JOIN `uber\_data.dropoff\_location\_dim` d ON f.dropoff\_location\_id = d.dropoff\_location\_id

ORDER BY t.trip\_distance LIMIT 10; -- for shortest trips

1. IMPACT OF TOLLS ON FARE AMOUNT:

SELECT

    CASE

        WHEN f.tolls\_amount = 0 THEN 'No Toll'

        WHEN f.tolls\_amount BETWEEN 0 AND 5 THEN 'Low Toll (0-5)'

        WHEN f.tolls\_amount BETWEEN 5 AND 15 THEN 'Medium Toll (5-15)'

        ELSE 'High Toll (15+)'

    END AS toll\_category,

    AVG(f.total\_amount) AS avg\_fare

FROM `uber\_data.fact\_table` f

GROUP BY toll\_category

ORDER BY avg\_fare DESC;

1. FARE PER TRIP DISTANCE:

SELECT

  CASE

    WHEN t.trip\_distance < 2 THEN '0-2 miles'

    WHEN t.trip\_distance BETWEEN 2 AND 5 THEN '2-5 miles'

    WHEN t.trip\_distance BETWEEN 5 AND 10 THEN '5-10 miles'

    ELSE '10+ miles'

  END AS distance\_range, AVG(fare\_amount) AS avg\_fare

FROM `uber\_data.fact\_table` f JOIN `uber\_data.trip\_distance\_dim` t ON f.trip\_distance\_id = t.trip\_distance\_id

GROUP BY distance\_range

ORDER BY avg\_fare DESC

1. DAYS WITH HIGHEST REVENUE:

SELECT d.pick\_day, d.pick\_month, d.pick\_year, SUM(f.total\_amount) AS total\_revenue

FROM `uber\_data.fact\_table` f

JOIN `uber\_data.datetime\_dim` d ON f.datetime\_id = d.datetime\_id

GROUP BY d.pick\_year, d.pick\_month, d.pick\_day

ORDER BY total\_revenue DESC

1. ANALYTICS QUERY:

CREATE OR REPLACE TABLE `uber\_data.tbl\_analytics` AS (

SELECT

f.trip\_id,

f.fare\_amount,

f.extra,

f.mta\_tax,

f.tip\_amount,

f.tolls\_amount,

f.total\_amount,

d.pick\_day,

d.pick\_month,

d.pick\_year,

pc.passenger\_count,

pt.payment\_type\_name,

td.trip\_distance

FROM

`uber\_data.fact\_table` f

JOIN `uber\_data.datetime\_dim` d ON f.datetime\_id = d.datetime\_id

JOIN `uber\_data.passenger\_count\_dim` pc ON f.passenger\_count\_id = pc.passenger\_count\_id

JOIN `uber\_data.payment\_type\_dim` pt ON f.payment\_type\_id = pt.payment\_type\_id

JOIN `uber\_data.trip\_distance\_dim` td ON f.trip\_distance\_id = td.trip\_distance\_id);