Q1 – Which 5 cities placed most orders (ranked highest to lowest)

SELECT a.city, COUNT(o.order\_id) as order\_count

FROM orders o

INNER JOIN address a ON o.address\_id = a.address\_id

GROUP BY a.city

ORDER BY order\_count DESC

LIMIT 5;

Q2 – Which 5 states placed most orders (ranked highest to lowest)

SELECT a.state,

COUNT(o.order\_id) as order\_count

FROM orders o

INNER JOIN address a ON o.address\_id = a.address\_id

GROUP BY a.state

ORDER BY order\_count

DESC LIMIT 5;

Q3 – What is the split between cash on delivery and prepaid

SELECT

payment\_method,

COUNT(\*) as order\_count,

ROUND(COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM orders), 2) as percentage

FROM orders

GROUP BY payment\_method;

Q4 – Which 10 products were most purchased (ranked highest to lowest)

SELECT

product\_id,

product\_name,

SUM(quantity\_ordered) as total\_purchases

FROM

order\_details

GROUP BY

product\_id, product\_name

ORDER BY

total\_purchases DESC

LIMIT 10;

Q5 – How much discount have we given in last N number of days

There is a catch here since I can see in the order table or in any table there is no date column. It is not possible that we do not record the order date, since it’s a mandatory column. So as of now I have assumed

SELECT SUM(discount) as total\_discount

FROM coupons

WHERE order\_date >= (CURRENT\_DATE - INITIAL N DAY);

Q6 – What is the revenue in last N number of days (revenue will be on the basis of selling price)

SELECT

SUM(selling\_price) as total\_revenue

FROM produts

WHERE order\_date >= (CURRENT\_DATE - INTIAL N DAY);

Q7 – If spend on marketing is assumed to b X rupees, how much profit / loss have we made in last N days

We believe there should be a table which consists of revenue table and profit loss columns as well in the following table. For now I have assumed Finance table.

SELECT

(SUM(revenue) - X - SUM(expenses)) as profit\_loss

FROM

Finance

WHERE

transaction\_date >= (CURRENT\_DATE - INITIAL N DAY);

Q8 – What is our repeat rate in last N days

SELECT

COUNT(DISTINCT users\_id) AS repeat\_customers,

COUNT(DISTINCT CASE WHEN order\_date >= (CURRENT\_DATE - INTERVAL N DAY) THEN users\_id END) AS new\_customers,

(COUNT(DISTINCT CASE WHEN order\_date >= (CURRENT\_DATE - INTERVAL N DAY) THEN user\_id END) / NULLIF(COUNT(DISTINCT users\_id), 0)) \* 100 AS repeat\_rate

FROM

users

WHERE

order\_date >= (CURRENT\_DATE - INITIAL N DAY);