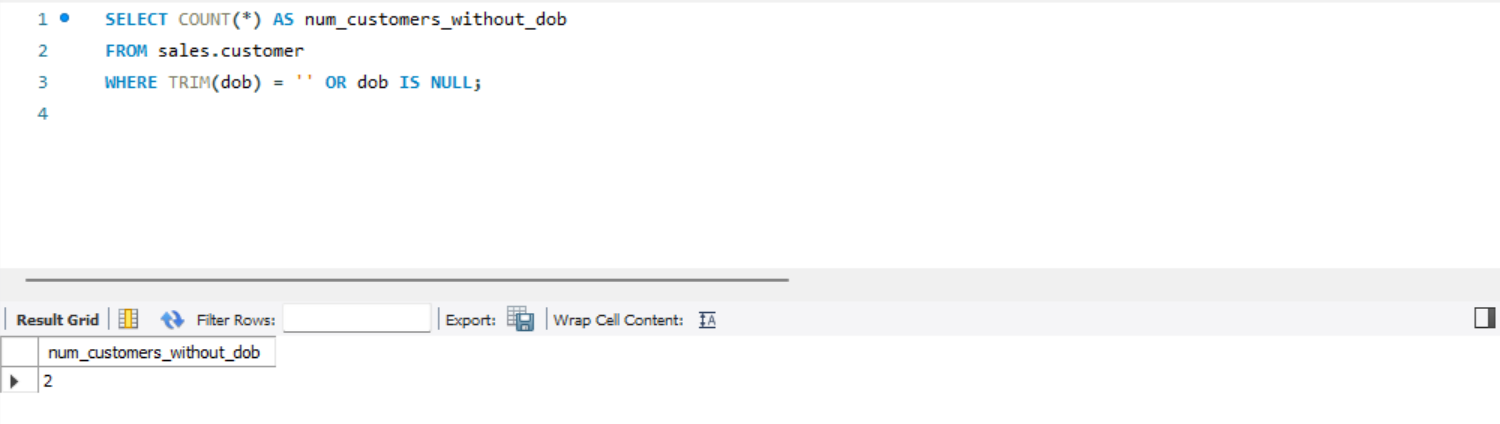
**Answer-Sheet**

**Basic Questions:**

1. How many customers do not have DOB information available?

  
- Code:  
SELECT COUNT(\*) AS num\_customers\_without\_dob

FROM sales.customer

WHERE TRIM(dob) = '' OR dob IS NULL;

2. How many customers are there in each pincode and gender combination?  
A screenshot of a computer

Description automatically generated- Code is :  
SELECT primary\_pincode, gender, COUNT(\*) AS num\_customers

FROM sales.customer

GROUP BY primary\_pincode, gender;

3. Print product name and mrp for products which have more than 50000 MRP?   
A screenshot of a computer

Description automatically generated

- Code is:

SELECT product\_name, mrp

FROM sales.product

where mrp > 5000;

4. How many delivery personal are there in each pincode?

A screenshot of a computer

Description automatically generated- Code:   
SELECT delivery\_person\_id, pincode, count(\*) as num\_delivery\_person

FROM sales.`delivery-person` group by delivery\_person\_id, pincode;

5. For each Pin code, print the count of orders, sum of total amount paid, average amount

paid, maximum amount paid, minimum amount paid for the transactions which were

paid by 'cash'. Take only 'buy' order types

- Answer is:

A screenshot of a computer

Description automatically generated  
- Code is:

SELECT

delivery\_pincode,

COUNT(order\_id) AS order\_count,

SUM(total\_amount\_paid) AS total\_amount\_paid,

AVG(total\_amount\_paid) AS avg\_amount\_paid,

MAX(total\_amount\_paid) AS max\_amount\_paid,

MIN(total\_amount\_paid) AS min\_amount\_paid

FROM sales.orders

WHERE order\_type = 'buy' AND payment\_type = 'cash'

GROUP BY delivery\_pincode;

6. For each delivery\_person\_id, print the count of orders and total amount paid for

product\_id = 12350 or 12348 and total units > 8. Sort the output by total amount paid in

descending order. Take only 'buy' order types

A screenshot of a computer

Description automatically generated- Code is:  
SELECT

delivery\_person\_id,

COUNT(order\_id) AS order\_count,

SUM(total\_amount\_paid) AS total\_amount\_paid

FROM sales.orders

WHERE order\_type = 'buy'

AND (product\_id = 12350 OR product\_id = 12348)

AND tot\_units > 8

GROUP BY delivery\_person\_id

ORDER BY total\_amount\_paid DESC;

7. Print the Full names (first name plus last name) for customers that have email on

"gmail.com"?

A screenshot of a computer

Description automatically generated

- Code is:  
SELECT concat(first\_name, last\_name) As Full\_names

from sales.customer

where email like '%@gmail.com%';

8. Which pincode has average amount paid more than 150,000? Take only 'buy' order

Types

A screenshot of a computer

Description automatically generated- Code is:

SELECT delivery\_pincode, AVG(total\_amount\_paid) AS avg\_amount\_paid

FROM sales.orders

WHERE order\_type = 'buy'

GROUP BY delivery\_pincode

HAVING avg\_amount\_paid > 150000;

9. Create following columns from order\_dim data -

 order\_date

 Order day

 Order month

 Order year

A screenshot of a computer

Description automatically generatedSELECT

order\_date,

DAY(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS order\_day,

MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS order\_month,

YEAR(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS order\_year

FROM sales.orders;

10. How many total orders were there in each month and how many of them were

returned? Add a column for return rate too.

return rate = (100.0 \* total return orders) / total buy orders

Hint: You will need to combine SUM() with CASE WHEN  
A screenshot of a computer

Description automatically generated

- Code is:

SELECT

MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS order\_month,

COUNT(\*) AS total\_orders,

SUM(CASE WHEN order\_type = 'return' THEN 1 ELSE 0 END) AS returned\_orders,

(100.0 \* SUM(CASE WHEN order\_type = 'return' THEN 1 ELSE 0 END)) / COUNT(\*) AS return\_rate

FROM sales.orders

GROUP BY order\_month;

**Question On SQL Joins:**

11. How many units have been sold by each brand? Also get total returned units for each

brand.

**\*\* at first I run a query on sales schema to rename the empty column name on the product table to product\_id  
(USE sales;**

**ALTER TABLE product**

**CHANGE COLUMN MyUnknownColumn product\_id INT;)**   
A screenshot of a computer program

Description automatically generated- Code is:   
SELECT

p.brand,

SUM(o.tot\_units) AS total\_units\_sold,

SUM(CASE WHEN o.order\_type = 'return' THEN o.tot\_units ELSE 0 END) AS total\_returned\_units

FROM

orders o

JOIN

product p ON o.product\_id = p.product\_id

GROUP BY

p.brand;

12. How many distinct customers and delivery boys are there in each state?

A screenshot of a computer

Description automatically generated  
- Code is:

SELECT

p.state,

COUNT(DISTINCT c.cust\_id) AS distinct\_customers,

COUNT(DISTINCT d.delivery\_person\_id) AS distinct\_delivery\_boys

FROM

pincode p

LEFT JOIN

customer c ON p.pincode = c.primary\_pincode

LEFT JOIN

orders o ON c.cust\_id = o.cust\_id

LEFT JOIN

`delivery-person` d ON p.pincode = d.pincode

GROUP BY

p.state;

13. For every customer, print how many total units were ordered, how many units were

ordered from their primary\_pincode and how many were ordered not from the

primary\_pincode. Also calulate the percentage of total units which were ordered from

primary\_pincode(remember to multiply the numerator by 100.0). Sort by the

percentage column in descending order.

A screenshot of a computer

Description automatically generated

- Code is:   
SELECT

c.cust\_id,

SUM(o.tot\_units) AS total\_units\_ordered,

SUM(CASE WHEN o.delivery\_pincode = c.primary\_pincode THEN o.tot\_units ELSE 0 END) AS units\_ordered\_primary\_pincode,

SUM(CASE WHEN o.delivery\_pincode != c.primary\_pincode THEN o.tot\_units ELSE 0 END) AS units\_ordered\_other\_pincode,

(100.0 \* SUM(CASE WHEN o.delivery\_pincode = c.primary\_pincode THEN o.tot\_units ELSE 0 END)) / SUM(o.tot\_units) AS percentage\_primary\_pincode

FROM

customer c

LEFT JOIN

orders o ON c.cust\_id = o.cust\_id

GROUP BY

c.cust\_id

ORDER BY

percentage\_primary\_pincode DESC;

14. For each product name, print the sum of number of units, total amount paid, total

displayed selling price, total mrp of these units, and finally the net discount from selling

price.

(i.e. 100.0 - 100.0 \* total amount paid / total displayed selling price) &

the net discount from mrp (i.e. 100.0 - 100.0 \* total amount paid / total mrp)

A screenshot of a computer

Description automatically generated-Code is:

SELECT

p.product\_name,

SUM(o.tot\_units) AS total\_units,

SUM(o.total\_amount\_paid) AS total\_amount\_paid,

SUM(o.displayed\_selling\_price\_per\_unit \* o.tot\_units) AS total\_displayed\_selling\_price,

SUM(p.mrp \* o.tot\_units) AS total\_mrp,

100.0 - 100.0 \* SUM(o.total\_amount\_paid) / SUM(o.displayed\_selling\_price\_per\_unit \* o.tot\_units) AS net\_discount\_selling\_price,

100.0 - 100.0 \* SUM(o.total\_amount\_paid) / SUM(p.mrp \* o.tot\_units) AS net\_discount\_mrp

FROM

product p

JOIN

orders o ON p.product\_id = o.product\_id

GROUP BY

p.product\_name;

**Advance Questions:**

15. For every order\_id (exclude returns), get the product name and calculate the discount

percentage from selling price. Sort by highest discount and print only those rows where

discount percentage was above 10.10%

A screenshot of a computer

Description automatically generated

- Code is:

SELECT o.order\_id, p.product\_name,

(100.0 - 100.0 \* o.total\_amount\_paid / (o.displayed\_selling\_price\_per\_unit \* o.tot\_units)) AS discount\_percentage

FROM orders o

JOIN

product p ON o.product\_id = p.product\_id

WHERE

o.order\_type != 'return'

HAVING discount\_percentage > 10.10

ORDER BY

discount\_percentage DESC;

16. Using the per unit procurement cost in product\_dim, find which product category has

made the most profit in both absolute amount and percentage

Absolute Profit = Total Amt Sold - Total Procurement Cost

Percentage Profit = 100.0 \* Total Amt Sold / Total Procurement Cost - 100.0

A screenshot of a computer

Description automatically generated

- Code is:

SELECT

p.category,

SUM(o.total\_amount\_paid) - SUM(p.procurement\_cost\_per\_unit \* o.tot\_units) AS absolute\_profit,

(100.0 \* SUM(o.total\_amount\_paid) / SUM(p.procurement\_cost\_per\_unit \* o.tot\_units)) - 100.0 AS percentage\_profit

FROM

product p

JOIN

orders o ON p.product\_id = o.product\_id

GROUP BY

p.category

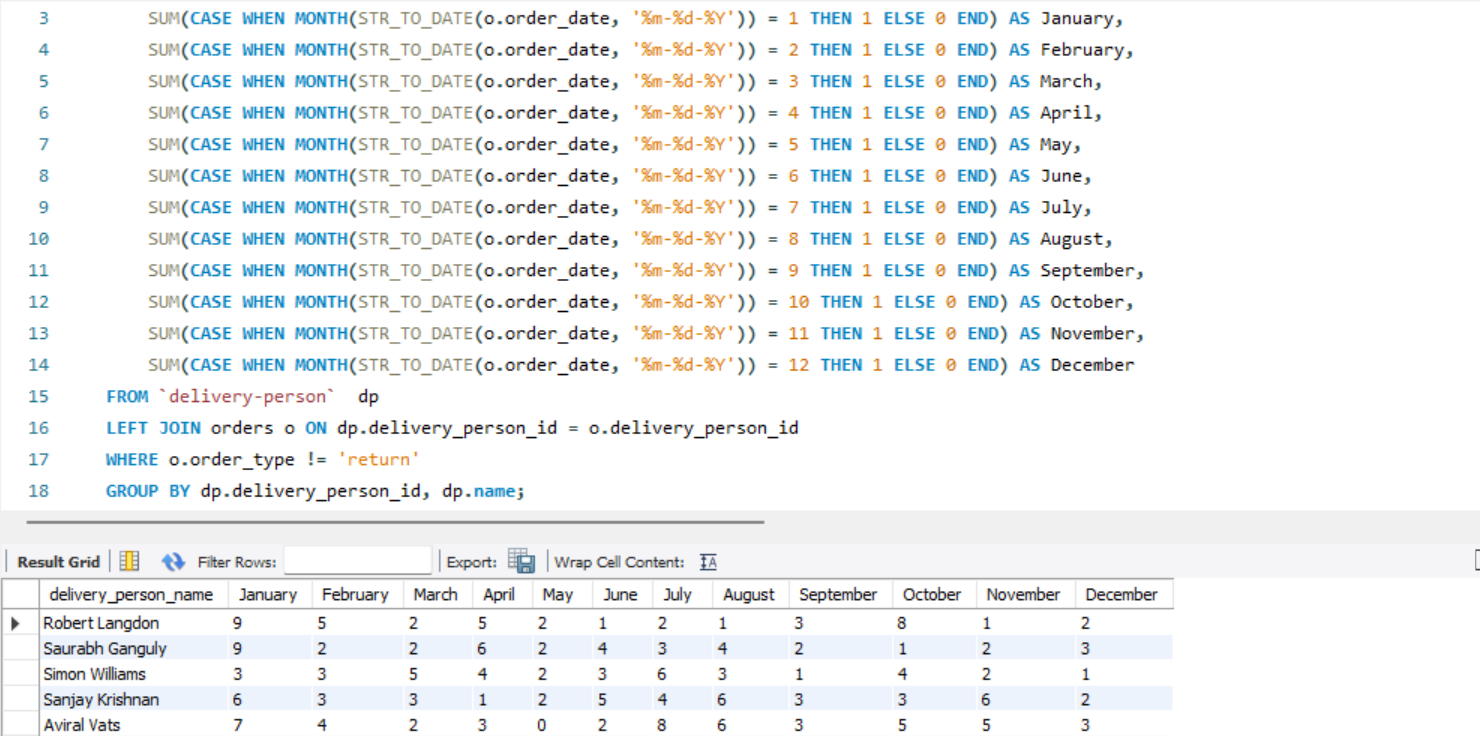
ORDER BY

absolute\_profit DESC;

17. For every delivery person(use their name), print the total number of order ids (exclude

returns) by month in separate columns i.e. there should be one row for each

delivery\_person\_id and 12 columns for every month in the year



- Code is:   
USE sales; SELECT

dp.name AS delivery\_person\_name,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 1 THEN 1 ELSE 0 END) AS January,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 2 THEN 1 ELSE 0 END) AS February,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 3 THEN 1 ELSE 0 END) AS March,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 4 THEN 1 ELSE 0 END) AS April,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 5 THEN 1 ELSE 0 END) AS May,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 6 THEN 1 ELSE 0 END) AS June,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 7 THEN 1 ELSE 0 END) AS July,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 8 THEN 1 ELSE 0 END) AS August,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 9 THEN 1 ELSE 0 END) AS September,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 10 THEN 1 ELSE 0 END) AS October,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 11 THEN 1 ELSE 0 END) AS November,

SUM(CASE WHEN MONTH(STR\_TO\_DATE(o.order\_date, '%m-%d-%Y')) = 12 THEN 1 ELSE 0 END) AS December

FROM `delivery-person` dp

LEFT JOIN orders o ON dp.delivery\_person\_id = o.delivery\_person\_id

WHERE o.order\_type != 'return'

GROUP BY dp.delivery\_person\_id, dp.name;

18. For each gender - male and female - find the absolute and percentage profit (like in

Q15) by product name

A screenshot of a computer

Description automatically generated

- Code is:

SELECT

c.gender,

p.product\_name,

SUM(o.total\_amount\_paid) - SUM(p.procurement\_cost\_per\_unit \* o.tot\_units) AS absolute\_profit,

(100.0 \* SUM(o.total\_amount\_paid) / SUM(p.procurement\_cost\_per\_unit \* o.tot\_units)) - 100.0 AS percentage\_profit

FROM

orders o

JOIN

product p ON o.product\_id = p.product\_id

JOIN

customer c ON o.cust\_id = c.cust\_id

GROUP BY

c.gender, p.product\_name

HAVING

percentage\_profit > 10.10;

19. Generally the more numbers of units you buy, the more discount seller will give you. For

'Dell AX420' is there a relationship between number of units ordered and average

discount from selling price? Take only 'buy' order types.

A screenshot of a computer

Description automatically generated

-Code is:

SELECT

o.tot\_units AS number\_of\_units,

AVG(1 - o.total\_amount\_paid / (o.displayed\_selling\_price\_per\_unit \* o.tot\_units)) AS average\_discount

FROM

orders o

JOIN

product p ON o.product\_id = p.product\_id

WHERE

p.product\_name = 'Dell AX420'

AND o.order\_type = 'buy'

GROUP BY

o.tot\_units

ORDER BY number\_of\_units;