**Import Data from CSV file**

Copy customer(customer\_id,first\_name,last\_name,email,address\_id)

FROM 'D:\Data Analyst\customer.csv'

DELIMITER ','

CSV HEADER;

Select \* From customer;

CREATE TABLE payment

(

customer\_id int8 PRIMARY KEY,

amount int8 NOT NULL,

mode varchar (50),

payment\_date date

)

select \* from payment;

SELECT UPPER(first\_name) from customer;

SELECT LOWER(first\_name) from customer;

SELECT LENGTH(first\_name) ,first\_name from customer;

SELECT SUBSTRING(first\_name,1,3) ,first\_name from customer;

SELECT CONCAT(first\_name,last\_name) ,first\_name,last\_name from customer;

SELECT REPLACE(first\_name,'Mary','Mohan') ,first\_name,last\_name from customer;

**Aggregate Functions**

SELECT COUNT(amount) From payment;

SELECT COUNT(\*) From payment;

SELECT SUM(amount) From payment;

SELECT MIN(amount) From payment;

SELECT MAX(amount) From payment;

SELECT ROUND(AVG(amount),2) From payment;

**GROUP BY AND HAVING CLAUSE**

SELECT mode, SUM(amount) AS total

FROM payment

GROUP BY mode

SELECT mode, SUM(amount) AS total

FROM payment

GROUP BY mode

ORDER BY total ASC;

SELECT mode, COUNT(amount) AS total

FROM payment

GROUP BY mode

HAVING COUNT(amount)>=3

ORDER BY total ASC;

SELECT mode, COUNT(amount) AS total

FROM payment

GROUP BY mode

HAVING COUNT(amount)>=2 AND COUNT(amount)<4

ORDER BY total ASC;

**TIMESTAMPS AND EXTRACT**

SHOW TIMEZONE

SELECT NOW()

SELECT TIMEOFDAY()

SELECT CURRENT\_TIME

SELECT CURRENT\_DATE

SELECT EXTRACT (MONTH FROM payment\_date) AS payment\_month, payment\_date

FROM payment

SELECT EXTRACT (DOW FROM payment\_date) AS payment\_dow, payment\_date

FROM payment

SELECT EXTRACT (QUARTER FROM payment\_date) AS payment\_time, payment\_date

FROM payment

**JOINS**

**Inner join**

SELECT \*

FROM customer AS c

INNER JOIN payment As p

ON c.customer\_id = p.customer\_id

SELECT c.first\_name, p.amount,p.mode

FROM customer AS c

INNER JOIN payment As p

ON c.customer\_id = p.customer\_id

**Left join**

SELECT \*

FROM customer AS c

LEFT JOIN payment As p

ON c.customer\_id = p.customer\_id

**Right join**

SELECT \*

FROM customer AS c

LEFT JOIN payment As p

ON c.customer\_id = p.customer\_id

**Full Join**

SELECT \*

FROM customer AS c

FULL OUTER JOIN payment As p

ON c.customer\_id = p.customer\_id

SELECT \*

FROM emp AS T1

JOIN emp AS T2

ON T2.empid =T1.manager\_id

SELECT T1.empname As employee\_name, T2.empname AS manager\_name

FROM emp AS T1

JOIN emp AS T2

ON T2.empid =T1.manager\_id

**SUB QUERY**

select \*

FROM payment

WHERE amount>(select avg(amount) from payment)

SELECT customer\_id,amount,mode

FROM payment

WHERE customer\_id IN (SELECT customer\_id FROM customer)

SELECT first\_name, last\_name

FROM customer c

WHERE EXISTS(

SELECT customer\_id, amount

FROM payment p

WHERE p.customer\_id = c.customer\_id

AND amount>50

)

**WINDOW FUNCTION**

**CASE EXPRESSIONS**

SELECT customer\_id, amount,

CASE

WHEN amount > 50 THEN 'Expensive product'

WHEN amount = 50 THEN 'Moderate product'

ELSE 'Inexpensive product'

END AS ProductStatus

FROM payment

SELECT customer\_id,

CASE amount

WHEN 500 THEN 'Prime Customer'

WHEN 50 THEN 'Plus Customer'

ELSE 'Regular Customer'

END AS CustomerStatus

FROM payment

**COMMON TABLE EXPRESSION (CTE)**

WITH my\_cts As (

SELECT \*, AVG(amount) OVER (ORDER BY p.customer\_id) As "Average\_price",

COUNT(address\_id) OVER (ORDER BY c.customer\_id) AS "Count"

FROM payment AS p

INNER JOIN customer As c

ON p.customer\_id = c.customer\_id

)

SELECT first\_name,last\_name,amount

FROM my\_cts

Q1: Who is the senior most employee based on job title?

select \* from employee

ORDER BY levels desc

LIMIT 1

Q2: Which countries have the most Invoices?

select COUNT(\*) as c ,billing\_country

from invoice

group by billing\_country

order by c desc

Q3: What are top 3 values of total invoice

SELECT total FROM invoice

Order by total desc

limit 3

Q4: Which city has the best customers? We would like to throw a

promotional Music Festival in the city we made the most money. Write a

query that returns one city that has the highest sum of invoice totals.

Return both the city name & sum of all invoice totals

SELECT SUM(total) as invoice\_total , billing\_city

from invoice

group by billing\_city

order by invoice\_total desc

Q5: Who is the best customer? The customer who has spent the most

money will be declared the best customer. Write a query that returns

the person who has spent the most money.

select customer.customer\_id, customer.first\_name, customer.last\_name,SUM(total) AS total\_spending

from customer

JOIN invoice ON customer.customer\_id = invoice.customer\_id

GROUP BY customer.customer\_id

ORDER BY total\_spending DESC

LIMIT 1

**Question Set 2 – Moderate**

1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A

SELECT DISTINCT email,first\_name, last\_name

FROM customer

JOIN invoice ON customer.customer\_id = invoice.customer\_id

JOIN invoice\_line ON invoice.invoice\_id = invoice\_line.invoice\_id

WHERE track\_id IN(

SELECT track\_id FROM track

JOIN genre ON track.genre\_id = genre.genre\_id

WHERE genre.name LIKE 'Rock'

)

ORDER BY email;

Method 2

SELECT DISTINCT email AS Email,first\_name AS FirstName, last\_name AS LastName, genre.name AS Name

FROM customer

JOIN invoice ON invoice.customer\_id = customer.customer\_id

JOIN invoiceline ON invoiceline.invoice\_id = invoice.invoice\_id

JOIN track ON track.track\_id = invoiceline.track\_id

JOIN genre ON genre.genre\_id = track.genre\_id

WHERE genre.name LIKE 'Rock'

ORDER BY email;

1. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands

SELECT artist.artist\_id, artist.name,COUNT(artist.artist\_id) AS number\_of\_songs

FROM track

JOIN album ON album.album\_id = track.album\_id

JOIN artist ON artist.artist\_id = album.artist\_id

JOIN genre ON genre.genre\_id = track.genre\_id

WHERE genre.name LIKE 'Rock'

GROUP BY artist.artist\_id

ORDER BY number\_of\_songs DESC

LIMIT 10;

1. Return all the track names that have a song length longer than the average song length. Return the Name and Milliseconds for each track. Order by the song length with the longest songs listed first

SELECT name,milliseconds

FROM track

WHERE milliseconds > (

SELECT AVG(milliseconds) AS avg\_track\_length

FROM track )

ORDER BY milliseconds DESC;

**Question Set 3 – Advance**

1. Find how much amount spent by each customer on artists? Write a query to return customer name, artist name and total spent

WITH best\_selling\_artist AS (

SELECT artist.artist\_id AS artist\_id, artist.name AS artist\_name, SUM(invoice\_line.unit\_price\*invoice\_line.quantity) AS total\_sales

FROM invoice\_line

JOIN track ON track.track\_id = invoice\_line.track\_id

JOIN album ON album.album\_id = track.album\_id

JOIN artist ON artist.artist\_id = album.artist\_id

GROUP BY 1

ORDER BY 3 DESC

LIMIT 1

)

SELECT c.customer\_id, c.first\_name, c.last\_name, bsa.artist\_name, SUM(il.unit\_price\*il.quantity) AS amount\_spent

FROM invoice i

JOIN customer c ON c.customer\_id = i.customer\_id

JOIN invoice\_line il ON il.invoice\_id = i.invoice\_id

JOIN track t ON t.track\_id = il.track\_id

JOIN album alb ON alb.album\_id = t.album\_id

JOIN best\_selling\_artist bsa ON bsa.artist\_id = alb.artist\_id

GROUP BY 1,2,3,4

ORDER BY 5 DESC;

1. We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared return all Genres

Method 1: Using CTE

WITH popular\_genre AS

(

SELECT COUNT(invoice\_line.quantity) AS purchases, customer.country, genre.name, genre.genre\_id,

ROW\_NUMBER() OVER(PARTITION BY customer.country ORDER BY COUNT(invoice\_line.quantity) DESC) AS RowNo

FROM invoice\_line

JOIN invoice ON invoice.invoice\_id = invoice\_line.invoice\_id

JOIN customer ON customer.customer\_id = invoice.customer\_id

JOIN track ON track.track\_id = invoice\_line.track\_id

JOIN genre ON genre.genre\_id = track.genre\_id

GROUP BY 2,3,4

ORDER BY 2 ASC, 1 DESC

)

SELECT \* FROM popular\_genre WHERE RowNo <= 1

Method 2: : Using Recursive

WITH RECURSIVE

sales\_per\_country AS(

SELECT COUNT(\*) AS purchases\_per\_genre, customer.country, genre.name, genre.genre\_id

FROM invoice\_line

JOIN invoice ON invoice.invoice\_id = invoice\_line.invoice\_id

JOIN customer ON customer.customer\_id = invoice.customer\_id

JOIN track ON track.track\_id = invoice\_line.track\_id

JOIN genre ON genre.genre\_id = track.genre\_id

GROUP BY 2,3,4

ORDER BY 2

),

max\_genre\_per\_country AS (SELECT MAX(purchases\_per\_genre) AS max\_genre\_number, country

FROM sales\_per\_country

GROUP BY 2

ORDER BY 2)

SELECT sales\_per\_country.\*

FROM sales\_per\_country

JOIN max\_genre\_per\_country ON sales\_per\_country.country = max\_genre\_per\_country.country

WHERE sales\_per\_country.purchases\_per\_genre = max\_genre\_per\_country.max\_genre\_number;

3. Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount

Method 1: using CTE

WITH Customter\_with\_country AS (

SELECT customer.customer\_id,first\_name,last\_name,billing\_country,SUM(total) AS total\_spending,

ROW\_NUMBER() OVER(PARTITION BY billing\_country ORDER BY SUM(total) DESC) AS RowNo

FROM invoice

JOIN customer ON customer.customer\_id = invoice.customer\_id

GROUP BY 1,2,3,4

ORDER BY 4 ASC,5 DESC)

SELECT \* FROM Customter\_with\_country WHERE RowNo <= 1

Method 2: Using Recursive

WITH RECURSIVE

customter\_with\_country AS (

SELECT customer.customer\_id,first\_name,last\_name,billing\_country,SUM(total) AS total\_spending

FROM invoice

JOIN customer ON customer.customer\_id = invoice.customer\_id

GROUP BY 1,2,3,4

ORDER BY 2,3 DESC),

country\_max\_spending AS(

SELECT billing\_country,MAX(total\_spending) AS max\_spending

FROM customter\_with\_country

GROUP BY billing\_country)

SELECT cc.billing\_country, cc.total\_spending, cc.first\_name, cc.last\_name, cc.customer\_id

FROM customter\_with\_country cc

JOIN country\_max\_spending ms

ON cc.billing\_country = ms.billing\_country

WHERE cc.total\_spending = ms.max\_spending

ORDER BY 1;

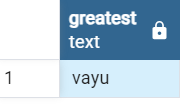
**Most Asked SQL interview Question**

select greatest(2,1,11,4)

select least(2,1,11,4)

select least('Akash','vayu','Dharti')

select greatest('Akash','vayu','Dharti')



-- **Method 1: Using greatest and Least Function**

SELECT greatest(source, destination), least(source, destination),max(distance)

FROM travel

group by greatest(source, destination), least(source, destination);

**-- Method 2: Using self-join**

WITH cte AS

(

SELECT \*, row\_number() over() AS SNo

FROM travel

)

SELECT t1.\*

FROM cte AS t1

JOIN cte AS t2

ON t1.source = t2.destination

AND t1.SNo <t2.Sno

SELECT \*

FROM travel t1

WHERE NOT EXISTS (SELECT \* FROM travel t2

WHERE t1.source = t2.destination

AND t1.destination = t2.source

AND t1.destination > t2.destination

)

**--Method 3: Using SUB Query**

SELECT \*

FROM travel t1

WHERE NOT EXISTS (SELECT \* FROM travel t2

WHERE t1.source = t2.destination

AND t1.destination = t2.source

AND t1.destination > t2.destination

)

