**QUESTION 1**

1. Write a query to return number of unique customers from the state of Karnataka

**Columns required: state\_name, customers**

**ANS - SELECT**

count(distinct(customer\_id) customers,

state as state\_name

**FROM**

customers

**WHERE**

state = ‘Karnataka’

1. Write a query to show the name of the state with second highest number of stores available to customers

**Columns required: state\_name, stores**

**ANS – SELECT**

Count(store\_id) as stores,

State as state\_name

**FROM**

Stores

**GROUP BY**

State

**ORDER BY**

Count(store\_id) **DESC**

**LIMIT** 2 **OFFSET** 1

1. Write a query to show 10 worst performing staff based on commission earned, where commission earned is 15% of total order value (Order value to be calculated as product of quantity and list price with discount subtracted from the total)

**Columns required: staff\_id, orders\_sold, commission\_earned**

**ANS – WITH**

Tab1 **AS**

( **SELECT**

\*,

((OI.list\_price\*OI.quantity) – OI.discount) as Order\_value

**FROM**

Staff S **JOIN** Orders O

**ON** S.staff\_id = O.staff\_id

**JOIN** Order\_items OI

**ON** O.order\_id = OI.order\_id

**JOIN** Store ST

**ON** ST.store\_id = O.store\_id ),

Tab2 **AS**

**( SELECT**

**\*,**

**(**0.15\*Order\_value) as commision\_earned

FROM

Tab1),

SELECT

S.staff\_id,

sum(commision\_earned) as commision\_earned,

count(order\_id) as order\_sold

FROM

Tab2

GROUP BY

ST.state

ORDER BY

Sum(commision\_earned) ASC

LIMIT 10

1. Write a query to show the number of orders per category where there has been a delay of over 1 week between order purchased to shipped

**Columns required: category\_name, orders**

**ANS – WITH**

Tab1 **AS**

**( SELECT**

**\***

**FROM**

Orders O JOIN Orders\_items OI

ON O.order\_id = OI.order\_id

JOIN product P

ON P.product\_id = OI.product\_id

JOIN categories C

P.category\_id = c.category\_id ),

Tab2 AS

( SELECT

O.order\_id,

C.category\_name,

Datediff(O.shipped\_date,O.order\_date) as timetaken

FROM

Tab1

WHERE

Datediff(O.shipped\_date,O.order\_date) > 7 )

SELECT

Count(order\_id) as orders,

Category\_name as category

FROM

Tab2

GROUP BY

Category\_name

1. Write a query to return customer id, and the first and last name as a single string for the top 3 customers by single order value (Order value to be calculated as product of quantity and list price with discount subtracted from the total). In the same view also show the next highest purchase made by the customer

**Columns required: customer\_id, full\_name, highest\_purchase, second\_highest\_purchase**

**ANS – WITH**

Tab1 **AS**

**( SELECT**

**\*,**

((OI.list\_price\*OI.quantity) – OI.discount) as Order\_value

**FROM**

customers C JOIN orders O

ON C. customers\_id = O. customer\_id

JOIN order\_item OI

ON O.order\_id = OI.order\_id)

SELECT C.customer\_id, string\_agg( C.first\_name,C.last\_Name) as Full\_name, order\_value as highest\_purchase from Tab1 order by Order\_value desc limit 3

**QUESTION 3**

1. Find the number of employees inside the Office at “2019-06-01 15:05:00”

ANS- WITH

Cte AS

(select \* from movement\_tracker where created at = “2019-06-01 15:05:00”),

select count(distinct(employee\_id)) as employee\_inside from cte where action = “In”

1. Measure the number of hours spent by each employee inside the office since the day they started (Account for a current shift if she/he is working)

Ans- WITH

Cte1 AS

( select \* from movement\_tracker where action = ‘In’)

Cte2 AS

( select \* from movement\_tracker where action = ‘Out’)

Cte3 AS

( Select cte1.employee\_id, Datediff(cte2.created\_at – cte1.created\_at) as time\_spent

from cte1 join cte2 on cte1.employee\_id= cte2.employee\_id)

select employee\_id,

Sum(time\_spent) as total\_time

From cte3 group by employee\_id order by employee\_id

1. Measure the number of hours spent by each employee inside the office between “2019-04-01 15:00:00” and “2019-04-04 11:00:00”

ANS- WITH

Cte1 AS

( select \* from movement\_tracker where action = ‘In’ and created\_at is between “2019-04-01 15:00:00” and “2019-04-04 11:00:00”)

Cte2 AS

( select \* from movement\_tracker where action = ‘Out’ and created\_at is between “2019-04-01 15:00:00” and “2019-04-04 11:00:00”))

Cte3 AS

( Select cte1.employee\_id, Datediff(cte2.created\_at – cte1.created\_at) as time\_spent

from cte1 join cte2 on cte1.employee\_id= cte2.employee\_id)

select employee\_id,

Sum(time\_spent) as total\_time

From cte3 group by employee\_id order by employee\_id

**QUESTION 2**

**Q2)**. Given an **Order Table** with the Schema **(id, user\_id, created\_at, amount**). Write a SQL Query to extract the raw data to create a Retention **plot**. The format for the raw data and output is given.

Week Start Date is the 1st Week in which the User\_Id Placed the order, Week 0 is Unique User ids who placed their 1st Order in this week. Out of those ids, Week 1 is unique users who placed an order in 1st Week + 1, Then Week 2 is 1st Week + 2, and so on till Week 10.

ANS

SELECT user\_id,DATEPART(week, created\_at)

AS login\_week FROM order GROUP BY user\_id,

DATEPART(week, created\_at)

SELECT user\_id, min(DATEPART

(week, created\_at)) AS Week0 FROM

order GROUP BY user\_id;

Select m.user\_id,m.login\_week,n.first as first from

(SELECT user\_id,DATEPART(week, created\_at) AS login\_week

FROM order GROUP BY user\_id,DATEPART(week, created\_at)) m,

(SELECT user\_id,min(DATEPART(week, created\_at)) AS first

FROM order GROUP BY user\_id) n where m.user\_id = n.user\_id;

Select m.user\_id,m.login\_week,n.first as first,

m.login\_week-first as week\_number from

(SELECT user\_id, DATEPART(week,created\_at)

AS login\_week FROM order GROUP BY user\_id,

DATEPART(week,created\_at)) m, (SELECT user\_id,

min (DATEPART(week,created\_at)) AS first FROM order

GROUP BY user\_id) n where m.user\_id=n.user\_id;

Select first,

SUM(CASE WHEN week\_number = 0 THEN 1 ELSE 0 END) AS week\_0,

SUM(CASE WHEN week\_number = 1 THEN 1 ELSE 0 END) AS week\_1,

SUM(CASE WHEN week\_number = 2 THEN 1 ELSE 0 END) AS week\_2,

SUM(CASE WHEN week\_number = 3 THEN 1 ELSE 0 END) AS week\_3,

SUM(CASE WHEN week\_number = 4 THEN 1 ELSE 0 END) AS week\_4,

SUM(CASE WHEN week\_number = 5 THEN 1 ELSE 0 END) AS week\_5,

SUM(CASE WHEN week\_number = 6 THEN 1 ELSE 0 END) AS week\_6,

SUM(CASE WHEN week\_number = 7 THEN 1 ELSE 0 END) AS week\_7,

SUM(CASE WHEN week\_number = 8 THEN 1 ELSE 0 END) AS week\_8,

SUM(CASE WHEN week\_number = 9 THEN 1 ELSE 0 END) AS week\_9

from (select m.user\_id,m.login\_week,n.first as first,

m.login\_week-first as week\_number from (SELECT

user\_id, DATEPART(week,created\_at) AS login\_week FROM order

GROUP BY user\_id,DATEPART(week,created\_at)) m,(SELECT user\_id,

min(DATEPART(week,created\_at)) AS first FROM orders GROUP BY user\_id)

n where m.user\_id = n.user\_id) as with\_week\_number

group by first order by first;