Name : Vaishak P R

Course : FSDA

Assignment : iNeuron\_SQL\_Project

Mail id : [pr.vaishu@gmail.com](mailto:pr.vaishu@gmail.com) (please share the feedback / suggestions if possible)

**Task 1**

CREATE DATABASE Task1

USE Task1

CREATE TABLE shopping\_history (

Product VARCHAR (30) NOT NULL,

Quantity INT NOT NULL,

Unit\_Price INT NOT NULL );

INSERT INTO shopping\_history VALUES

('Milk' , 3 , 10),

('Bread' , 7 , 3),

('Bread' , 5 , 2)

SELECT \* FROM shopping\_history

ALTER TABLE shopping\_history

ADD COLUMN Total\_Price INT AFTER Unit\_Price

UPDATE shopping\_history

SET Total\_Price = Quantity \* Unit\_Price ;

set SQL\_SAFE\_UPDATES = 0;

SELECT Product, sum(Total\_Price) FROM shopping\_history GROUP BY Product

**Task 2(1)**

CREATE DATABASE Task2

USE Task2

CREATE TABLE phones (

`name` varchar(30),

phone\_number int primary key)

INSERT INTO phones VALUES

('Jack', 1234),

('Lena', 3333),

('Mark', 9999),

('Anna', 7582)

SELECT \* FROM phones

CREATE TABLE calls(

id INT,

caller INT references phones(phone\_number),

callee INT references phones(phone\_number),

duration INT )

INSERT INTO calls VALUES

(25,1234,7582,8),

(7,9999,7582,1),

(18,9999,3333,4),

(2,7582,3333,3),

(3,3333,1234,1),

(21,3333,1234,1)

SELECT \* FROM calls

/\*Splitting Table calls into 2 small tables for 'caller' and 'callee'\*/

CREATE TABLE total\_caller

(SELECT caller, sum(duration) as total\_duration\_caller FROM calls GROUP BY caller)

CREATE TABLE total\_callee

(SELECT callee, sum(duration) as total\_duration\_callee FROM calls GROUP BY callee)

SELECT \* FROM total\_caller

SELECT \* FROM total\_callee

/\*Joining the 2 small tables \*/

CREATE TABLE total\_duration(

SELECT p.name, p.phone\_number , cr.caller , cr.total\_duration\_caller , ce.callee , ce.total\_duration\_callee from phones p

LEFT JOIN total\_caller cr on p.phone\_number = cr.caller

LEFT JOIN total\_callee ce on p.phone\_number = ce.callee)

SELECT \* FROM total\_duration

ALTER TABLE total\_duration

ADD COLUMN Total\_duration INT

UPDATE total\_duration

SET Total\_duration = total\_duration\_caller + total\_duration\_callee ;

UPDATE total\_duration

SET total\_duration\_callee = 0 WHERE total\_duration\_callee is NULL

SELECT name FROM total\_duration WHERE Total\_duration >= 10

**Task 2(2)**

CREATE DATABASE Task2\_2

USE Task2\_2

CREATE TABLE phones(

name VARCHAR(30),

phone\_number INT )

INSERT INTO phones VALUES

('John' , 6356),

('Addison' , 4315),

('Kate' , 8003),

('Ginny' , 9831)

SELECT \* FROM phones

CREATE TABLE calls(

id INT,

caller INT,

callee INT,

duration INT )

INSERT INTO calls VALUES

(65 , 8003, 9831 , 7),

(100 , 9831, 8003 , 3),

(145 , 4315, 9831 , 18)

SELECT \* FROM calls

/\*Splitting Table calls into 2 small tables for 'caller' and 'callee'\*/

CREATE TABLE total\_duration\_caller(

SELECT caller, sum(duration) as total\_duration\_caller FROM calls GROUP BY caller)

SELECT \* FROM total\_duration\_caller

CREATE TABLE total\_duration\_callee(

SELECT callee, sum(duration) as total\_duration\_callee from calls GROUP BY callee)

SELECT \* FROM total\_duration\_callee

/\*Joining the 2 small tables \*/

CREATE TABLE total\_duration(

SELECT p.name , p.phone\_number , cr.caller , cr.total\_duration\_caller , ce.callee , ce.total\_duration\_callee FROM phones p

LEFT JOIN total\_duration\_caller cr on p.phone\_number = cr.caller

LEFT JOIN total\_duration\_callee ce on p.phone\_number = ce.callee)

SELECT \* FROM total\_duration

ALTER TABLE total\_duration

ADD COLUMN Total\_duration INT

UPDATE total\_duration

SET total\_duration\_callee = 0 WHERE total\_duration\_callee IS NULL

UPDATE total\_duration

SET total\_duration\_caller = 0 WHERE total\_duration\_caller IS NULL

UPDATE total\_duration

SET Total\_duration = total\_duration\_caller + total\_duration\_callee

SELECT name FROM total\_duration where Total\_duration > 0

**Task 3(1)**

CREATE DATABASE Task3\_1

USE Task3\_1

CREATE TABLE transactions (

amount INT,

`date` DATE)

INSERT INTO transactions VALUES

(1000 , '2020-01-06'),

(-10 , '2020-01-14'),

(-75 , '2020-01-20'),

(-5 , '2020-01-25'),

(-4 , '2020-01-29'),

(2000 , '2020-03-10'),

(-75 , '2020-03-12'),

(-20 , '2020-03-15'),

(40 , '2020-03-15'),

(-50 , '2020-03-17'),

(200 , '2020-10-06'),

(-200 , '2020-10-06')

SELECT \* FROM transactions

SET SQL\_SAFE\_UPDATES = 0;

ALTER TABLE transactions

ADD COLUMN mnth INT

UPDATE transactions

SET mnth = EXTRACT(MONTH FROM date)

ALTER TABLE transactions

ADD COLUMN comments varchar(30)

UPDATE transactions

SET comments = CASE

WHEN amount > 0 THEN 'Credit'

ELSE 'Debit'

END;

/\*Using UNION ALL combining the credited amount (amount > 0) and Debited amount (amount < 0) with added charges per month\*/

CREATE TABLE transactions\_amount

SELECT mnth,

CASE

WHEN sum(amount) < -100 AND

count(amount) >= 3 THEN

sum(amount)

ELSE

sum(amount)-5

END AS amount

FROM transactions WHERE comments = 'Debit' GROUP BY mnth

UNION ALL

SELECT mnth ,sum(amount) AS amount\_credit

FROM transactions WHERE comments = 'Credit' GROUP BY mnth

SELECT \* FROM transactions\_amount

/\*In the given data only 3 months (January (1), March (3), October (10)) are mentioned ,

So considering other 9 months and charges of rest of 9 months, considering the charges -5 per month, So adding 9\*5 in the query\*/

SELECT (sum(amount) - (9\*5)) AS Balance FROM transactions\_amount

**Task 3(2)**

CREATE DATABASE Task3\_2

USE Task3\_2

CREATE TABLE transactions (

amount INT NOT NULL DEFAULT 0

`date` DATE )

INSERT INTO transactions VALUES

(1 , '2020-06-29'),

(35 , '2020-02-20'),

(-50 , '2020-02-03'),

(-1 , '2020-02-26'),

(-200 , '2020-08-01'),

(-44 , '2020-02-07'),

(-5 , '2020-02-25'),

(1 , '2020-06-29'),

(1 , '2020-06-29'),

(-100 , '2020-12-29'),

(-100 , '2020-12-30'),

(-100 , '2020-12-31')

SELECT \* FROM transactions

set SQL\_SAFE\_UPDATES = 0;

/\*Adding table to differentiate amount<0 and amount>0 \*/

ALTER TABLE transactions

ADD COLUMN comments varchar(30)

UPDATE transactions

SET comments = CASE

WHEN amount > 0 THEN 'Credit'

ELSE 'Debit'

END;

/\*Case statement to find out the total payments done including charges per month\*/

SELECT EXTRACT(MONTH FROM date) AS mnth, sum(amount < 0) as number\_payments\_done, sum(amount),

CASE

WHEN sum(amount) <= -100 AND

count(amount) >= 3 THEN

sum(amount)

ELSE

sum(amount)-5

END AS amount

FROM transactions WHERE comments = 'Debit' GROUP BY mnth

/\*Using UNION creating a table combining the payments(Debited) with adding the charges per month and credited amount\*/

CREATE TABLE transactions\_amount

SELECT EXTRACT(MONTH FROM date) AS mnth,

CASE

WHEN sum(amount) <= -100 AND

count(amount) >= 3 THEN

sum(amount)

ELSE

sum(amount)-5

END AS amount

FROM transactions WHERE comments = 'Debit' GROUP BY mnth

UNION

select EXTRACT(MONTH FROM date) AS mnth, sum(amount) AS amount FROM transactions WHERE comments = 'Credit' GROUP BY mnth

SELECT \* FROM transactions\_amount

/\*10 months are charged 5 per month, but charges of fee already included in query for August (8),

So, excluding August (8) adding (9\*5) \*/

SELECT (sum(amount)-(9\*5)) AS Balance FROM transactions\_amount

**Task3(3)**

CREATE DATABASE Task3\_3

USE Task3\_3

CREATE TABLE transactions (

amount INT ,

`date` DATE )

INSERT INTO transactions VALUES

(6000, '2020-04-03'),

(5000, '2020-04-02'),

(4000, '2020-04-01'),

(3000, '2020-03-01'),

(2000, '2020-02-01'),

(1000, '2020-01-01')

SELECT \* FROM transactions

SET SQL\_SAFE\_UPDATES = 0;

ALTER TABLE transactions

ADD COLUMN comments varchar(30)

UPDATE transactions

SET comments = CASE

WHEN amount > 0 THEN 'Credit'

ELSE 'Debit'

END;

CREATE TABLE transactions\_amount

SELECT EXTRACT(MONTH FROM date) AS mnth,

CASE

WHEN sum(amount) <= -100 AND

count(amount) >= 3 THEN

sum(amount) + 5

ELSE

sum(amount)

END AS amount

FROM transactions WHERE comments = 'Debit' GROUP BY mnth

UNION

select EXTRACT(MONTH FROM date) AS mnth, (sum(amount)-5) AS amount FROM transactions WHERE comments = 'Credit' GROUP BY mnth

SELECT \* FROM transactions\_amount

/\* only 4 months are included in the data, So by considering the charges of remaining 8 months by adding (8\*5) \*/

SELECT (SUM(amount)-(8\*5)) as Balance FROM transactions\_amount