**FSDA ASSIGNMENT – 1 (SQL PROJECT)**

**Task 1:**

**Write an SQL query that for each “product” returns the total amount of money spent on it. Rows should be ordered in descending order by “product”.**

**Ans:**

USE DATABASE DEMO\_DATABASE;

**-- CREATE A TABLE**

CREATE TABLE SHOPPING\_HISTORY (

PRODUCT VARCHAR NOT NULL,

QUANTITY INTEGER NOT NULL,

UNIT\_PRICE INTEGER NOT NULL);

DESCRIBE TABLE SHOPPING\_HISTORY;

**-- INSERT VALUES INTO THE TABLE**

INSERT INTO SHOPPING\_HISTORY VALUES

('BREAD',2,10),

('BREAD',5,20),

('MILK',1,32),

('MILK',5,29),

('CURD',6,23),

('BREAD',1,9),

('EGGS',6,50),

('LEMON',3,15),

('MILK',2,60),

('BUTTER',1,62);

SELECT \* FROM SHOPPING\_HISTORY;

**-- query that for each “product” returns the total amount of money spent on it.**

SELECT PRODUCT AS TOTAL\_UNIQUE\_PRODUCT, SUM ((QUANTITY \* UNIT\_PRICE)) AS TOTAL\_PRODUCT\_PRICE

FROM SHOPPING\_HISTORY

GROUP BY PRODUCT

ORDER BY PRODUCT DESC;

**TASK 2 (1): WRITE AN SQL QUERY TO FIND ALL CLIENTS WHO TALKED FOR 10 MINS IN TOTAL**

**ANS:**

USE DATABASE DEMO\_DATABASE;

CREATE OR REPLACE TABLE PHONES(

NAME VARCHAR(20) NOT NULL UNIQUE,

PHONE\_NUMBER INTEGER NOT NULL UNIQUE);

DESCRIBE TABLE PHONES;

INSERT INTO PHONES VALUES('JACK', 1234), ('LENA',3333), ('MARK',9999), ('ANNA',7582);

SELECT \* FROM PHONES;

**-- CREATE A TABLE**

CREATE OR REPLACE TABLE CALLS(

ID INTEGER NOT NULL,

CALLER INTEGER NOT NULL,

CALLEE INTEGER NOT NULL,

DURATION INTEGER NOT NULL,

UNIQUE(ID));

DESCRIBE TABLE CALLS;

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;

**--CREATE A NEW\_TABLE SW\_CALL\_HISTORY TO FETCH THE RECORDS**

CREATE TABLE SW\_CALL\_HISTORY AS

SELECT DISTINCT NAME, SUM(DURATION) AS TOTAL\_DURATION FROM PHONES

LEFT OUTER JOIN CALLS ON PHONES.PHONE\_NUMBER = CALLS.CALLER OR PHONES.PHONE\_NUMBER = CALLS.CALLEE

GROUP BY NAME

HAVING TOTAL\_DURATION >= 10

ORDER BY NAME;

**--SQL QUERY TO FIND ALL CLIENTS WHO TALKED FOR 10 MINS IN TOTAL**

SELECT NAME FROM SW\_CALL\_HISTORY;

**TASK 2 (2):**

**ANS:**

USE DATABASE DEMO\_DATABASE;

**-- CREATE A TABLE**

CREATE OR REPLACE TABLE PHONES(

NAME VARCHAR,

PHONE\_NUMBER INTEGER);

DESCRIBE TABLE PHONES;

INSERT INTO PHONES VALUES(

'JOHN',6356),

('ADDISON',4315),

('KATE',8003),

('GINNY',9831);

SELECT \* FROM PHONES;

**-- CREATE A TABLE**

CREATE OR REPLACE TABLE CALLS(

ID INTEGER,

CALLER INTEGER,

CALLEE INTEGER,

DURATION INTEGER);

DESCRIBE TABLE CALLS;

INSERT INTO CALLS VALUES

(65,8003,9831,7),

(100,9831,8003,3),

(145,4315,9831,18);

SELECT \* FROM CALLS;

**--QUERY**

SELECT NAME FROM PHONES

RIGHT OUTER JOIN CALLS ON PHONES.PHONE\_NUMBER = CALLS.CALLER;

**TASK 3 (1):**

**ANS:**

USE DATABASE DEMO\_DATABASE;

-**-CREATE TABLE**

CREATE OR REPLACE TABLE TRANSACTIONS(

AMOUNT INTEGER NOT NULL,

DATE DATE NOT NULL);

DESCRIBE TABLE TRANSACTIONS;

**--INSERT VALUES**

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,'2020-10-10'),

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

SELECT \* FROM TRANSACTIONS;

**-- CREATE A NEW TABLE 'SW\_TRANSACTION\_DETAIL' FOR THE BREAKDOWN REPORT**

DROP TABLE IF EXISTS SW\_TRANSACTION\_DETAIL;

CREATE TABLE SW\_TRANSACTION\_DETAIL AS

SELECT DISTINCT MONTH(DATE) AS MONTH\_OF\_THE\_YEAR,

SUM(CASE WHEN AMOUNT >= 0 THEN AMOUNT ELSE 0 END) AS INCOMING\_TRANSFER,

SUM(CASE WHEN AMOUNT < 0 THEN AMOUNT ELSE 0 END) AS CREDIT\_CARD,

SUM(AMOUNT) AS TOTAL\_BALANCE,

COUNT(CASE WHEN AMOUNT < 0 THEN 1 ELSE NULL END) AS CREDIT\_CARD\_TRANSACTION

FROM TRANSACTIONS

GROUP BY 1;

SELECT \* FROM SW\_TRANSACTION\_DETAIL;

**--FROM THE TABLE WE GET TO KNOW THAT ONLY MARCH HAS 3 CREDIT\_CARD\_TRANSACTION AND CREDIT\_CARD >= -100 SO AS PER THE CHARGES 11\*5= 55 (11 MONTHS AND RS 5 EACH MONTH CHARGE)**

SELECT (SUM(TOTAL\_BALANCE) - (11\*5)) AS FINAL\_BALANCE FROM SW\_TRANSACTION\_DETAIL;

**TASK 3 (2):**

**ANS:**

USE DATABASE DEMO\_DATABASE;

**--CREATE TABLE**

CREATE OR REPLACE TABLE TRANSACTIONS(

AMOUNT INTEGER NOT NULL,

DATE DATE NOT NULL);

DESCRIBE TABLE TRANSACTIONS;

**--INSERT VALUES**

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;

**-- CREATE A NEW TABLE 'SW\_TRANSACTION\_DETAIL' FOR THE BREAKDOWN REPORT**

DROP TABLE IF EXISTS SW\_TRANSACTION\_DETAIL;

CREATE TABLE SW\_TRANSACTION\_DETAIL AS

SELECT DISTINCT MONTH(DATE) AS MONTH\_OF\_THE\_YEAR,

SUM(CASE WHEN AMOUNT >= 0 THEN AMOUNT ELSE 0 END) AS INCOMING\_TRANSFER,

SUM(CASE WHEN AMOUNT < 0 THEN AMOUNT ELSE 0 END) AS CREDIT\_CARD,

SUM(AMOUNT) AS TOTAL\_BALANCE,

COUNT(CASE WHEN AMOUNT < 0 THEN 1 ELSE NULL END) AS CREDIT\_CARD\_TRANSACTION

FROM TRANSACTIONS

GROUP BY 1;

SELECT \* FROM SW\_TRANSACTION\_DETAIL;

**--FROM THE ABOVE TABLE WE GET TO KNOW THAT ONLY DECEMBER AND FEBURARY HAS >= 3 CREDIT\_CARD\_TRANSACTION AND CREDIT\_CARD >= -100 SO AS PER THE CHARGES 10\*5= 50**

SELECT (SUM(TOTAL\_BALANCE) - (10\*5)) AS FINAL\_BALANCE FROM SW\_TRANSACTION\_DETAIL;

**TASK 3 (3):**

**ANS:**

USE DATABASE DEMO\_DATABASE;

**--CREATE TABLE**

CREATE OR REPLACE TABLE TRANSACTIONS(

AMOUNT INTEGER NOT NULL,

DATE DATE NOT NULL);

DESCRIBE TABLE TRANSACTIONS;

**--INSERT VALUES**

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;

**-- CREATE A NEW TABLE 'SW\_TRANSACTION\_DETAIL' FOR THE BREAKDOWN REPORT**

DROP TABLE IF EXISTS SW\_TRANSACTION\_DETAIL;

CREATE TABLE SW\_TRANSACTION\_DETAIL AS

SELECT DISTINCT MONTH(DATE) AS MONTH\_OF\_THE\_YEAR,

SUM(CASE WHEN AMOUNT >= 0 THEN AMOUNT ELSE 0 END) AS INCOMING\_TRANSFER,

SUM(CASE WHEN AMOUNT < 0 THEN AMOUNT ELSE 0 END) AS CREDIT\_CARD,

SUM(AMOUNT) AS TOTAL\_BALANCE,

COUNT(CASE WHEN AMOUNT < 0 THEN 1 ELSE NULL END) AS CREDIT\_CARD\_TRANSACTION

FROM TRANSACTIONS

GROUP BY 1;

SELECT \* FROM SW\_TRANSACTION\_DETAIL;

**--FROM THE ABOVE TABLE WE GET TO KNOW THAT NOT A SINGLE MONTH CREDIT CARD TRANSACTION HAPPENED SO AS PER THE CHARGES 12\*5= 60**

SELECT (SUM(TOTAL\_BALANCE) - (12\*5)) AS FINAL\_BALANCE FROM SW\_TRANSACTION\_DETAIL;