/\* FSDA SQL PROJECT ASSIGNMENTS \*/

create database MS\_PROJECT;

use MS\_PROJECT;

/\* TASK 1 \*/

-- Calculating total price from shopping history

create table shopping\_history(

product varchar(20) not null,

quantity int not null,

unit\_price int not null

);

insert into shopping\_history values

('Bread',1,15),

('Butter',2,30),

('Maggie',5,20),

('Chips',2,10),

('Bread',3,12),

('Icecream',2,25),

('Kurkure',1,10),

('Milk',6,17),

('Chocolate',1,150),

('Chips',2,30);

select \* from shopping\_history;

select product, sum(total) as total\_price

from(select product, quantity\*unit\_price as total

from shopping\_history

order by product desc) report

group by product

order by product desc;

RESULT:



-------------------------------------------

/\* TASK 2 \*/

-- Telecommunication company problem

CREATE TABLE phones (

`name` VARCHAR(20),

phone\_number INT

);

CREATE TABLE calls (

id INT,

caller INT,

callee INT,

duration INT

);

insert into phones values

('Ankan', 7745 ),

('Sarika', 8754),

('Gaurav', 9800),

('Jara', 9867),

('Dany', 6790),

('Madhupa', 9432),

('Keya', 9434),

('Ria', 7586),

('Dev', 5578),

('Swati', 9044);

select \* from phones;

insert into calls values

(25, 9800, 5578, 8),

(10, 9432, 9044, 2),

(108, 9432, 7586 , 9),

(57, 5578, 9432, 5),

(8,9044,7586,7),

(18,9044,9432,5),

(5,9800,7745,10),

(67,7586,8754,6);

select \* from calls;

with ctereport(id,caller\_callee,duration)

as(select id,caller,duration from calls

union all

select id,callee,duration from calls)

select distinct p.`name`

from phones p

join

(select caller\_callee, sum(duration) over(partition by caller\_callee) as total\_duration

from ctereport) final

on final.caller\_callee = p.phone\_number

where final.total\_duration > 10

order by p.`name`;

RESULT:



------------------------------------------------------------------

/\* TASK 3 \*/

-- BANK TRANSACTIONS PROBLEM

create table transactions(

Amount int not null,

`Date` date not null

);

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;

with ctedebit as(select distinct month(`Date`) as `month`, -sum(Amount) over(partition by month(`Date`)) as debit, count(Amount) over(partition by month(`Date`)) as debitcount

from transactions

where Amount < 0 ),

ctetran as(select distinct month(`date`) as `month`, sum(Amount) over(partition by month(`Date`)) as total

from transactions)

select sum(total) - (12-count(debitcount))\*5 as balance

from

(select ctetran.`month`, ctetran.total, ctedebit.debit , ctedebit.debitcount

from ctetran

left join ctedebit on ctetran.`month`= ctedebit.`month` and ctedebit.debit >= 100 and ctedebit.debitcount >= 3) test ;

RESULT : 

---------------------------------------------------------------------------------------------------------------------

ALL TASKS REFERENCE --------- The SQL ASSIGNMENTS FILE

