iNeuron_SQL_Project

Task: 1

create database ineuron_assignments;

use ineuron_assignments;

CREATE or replace table shopping_history

(Product varchar not null,

  Quantity integer not null,

  Unit_price integer not null);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES ('bread',1,20);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('bread',2,20);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('milk',2,26);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('bread',5,15);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('milk',1,26);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('milk',3,24);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('milk',1,26);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('bread',3,18);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('bread',1,30);

INSERT INTO shopping_history(product,quantity,unit_price) VALUES('milk',4,25);

SELECT * FROM shopping_history;

SELECT Product,SUM(Quantity*Unit_price) as Total_Price FROM shopping_history GROUP BY 1 ORDER BY 1 DESC;

Results	Data Preview

| ✔ Query ID | SQL | 275ms | | 2 rows |

Row	PRODUCT	TOTAL_PRICE
1	milk	276
2	bread	219

Task: 2.1

CREATE OR REPLACE Table phones

(

  name varchar(20) not null unique,

```
    phone_number int not null unique
);

CREATE OR REPLACE Table calls (
    id int not null,
    caller int not null,
    callee int not null,
    duration int not null,
    unique(id)
);

INSERT INTO phones VALUES('Jack',1234),('Lena',3333),('Mark',9999),('Anna',7582);
SELECT * FROM phones;

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;

SELECT p.name FROM phones p, calls c
WHERE p.phone_number=c.caller OR p.phone_number=c.callee
GROUP BY 1 HAVING SUM(c.duration)>=10 ORDER BY 1;
```

Results   Data Preview

✔ Query ID   SQL   218ms   2 rows

Filter result...

Row	NAME
1	Anna
2	Jack

Task: 2.2

```
CREATE OR REPLACE Table phones1
(
    name varchar(20) not null unique,
    phone_number int not null unique
);

CREATE OR REPLACE Table calls1 (
```

```
    id int not null,

    caller int not null,

    callee int not null,

    duration int not null,

    unique(id)

);
```

INSERT INTO phones1 VALUES('John',6356), ('Addison',4315), ('Kate',8003), ('Ginny',9831);

INSERT INTO calls1 VALUES (65,8003,9831,7), (100,9831,8003,3), (145,4315,9831,18);

SELECT p.name FROM phones1 p, calls1 c

WHERE p.phone_number=c.caller OR p.phone_number=c.callee

GROUP BY 1 HAVING SUM(c.duration)>=10 ORDER BY 1;

Results Data Preview

✔ Query ID   SQL   73ms   3 rows

Filter result...   ⬇   Copy

Row	NAME
1	Addison
2	Ginny
3	Kate

==

Task: 3.1

CREATE OR REPLACE 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'),

(200,'2020-10-10'), (-200,'2020-10-10'), (-50,'2020-03-17');

SELECT * FROM transactions;

SELECT a.sum_amount - 5*(12-b.fee_excempted) balance FROM

(SELECT SUM(amount) as sum_amount, 'a' name FROM transactions) a LEFT OUTER JOIN

(SELECT COUNT(*) fee_excempted, 'a' name FROM

       (SELECT month(date) FROM transactions WHERE amount <0 GROUP BY month(date) HAVING
       (COUNT(amount) >=3 AND SUM(amount) <=-100))

) as b ON a.name = b.name;

Row	BALANCE
1	2746

Query ID   SQL   44ms   1 rows

Task: 3.2

CREATE OR REPLACE Table transactions1

(amount int not null,

date date not null);

INSERT INTO transactions1 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 transactions1;

SELECT a.sum_amount - 5*(12-b.fee_excempted) balance FROM

(SELECT sum(amount) as sum_amount, 'a' name FROM transactions1) a LEFT OUTER JOIN

(SELECT count(*) fee_excempted, 'a' name FROM

       (SELECT month(date) FROM transactions1 WHERE amount <0

       GROUP BY month(date) HAVING (COUNT(amount) >=3 AND SUM(amount) <=-100))

) as b ON a.name = b.name;

Query ID   SQL   345ms   1 rows

Row	BALANCE
1	-612

-- Task- 3.3

```sql
CREATE OR REPLACE Table transactions2 (

amount int not null,

date date not null);


INSERT INTO transactions2 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 transactions2;


SELECT a.sum_amount - 5*(12-b.fee_excempted) balance FROM

(SELECT sum(amount) as sum_amount, 'a' name FROM  transactions2) a  LEFT OUTER JOIN

(SELECT count(*) fee_excempted, 'a' name FROM

(SELECT month(date)

FROM transactions2

WHERE amount <0

GROUP BY month(date)

HAVING (COUNT(amount) >=3 AND SUM(amount) <=-100))) as b

ON a.name = b.name;
```

Results   Data Preview

✔ Query ID   SQL   291ms ▭▭▭   1 rows

Filter result...   ⬇   Copy

Row	BALANCE
1	20940