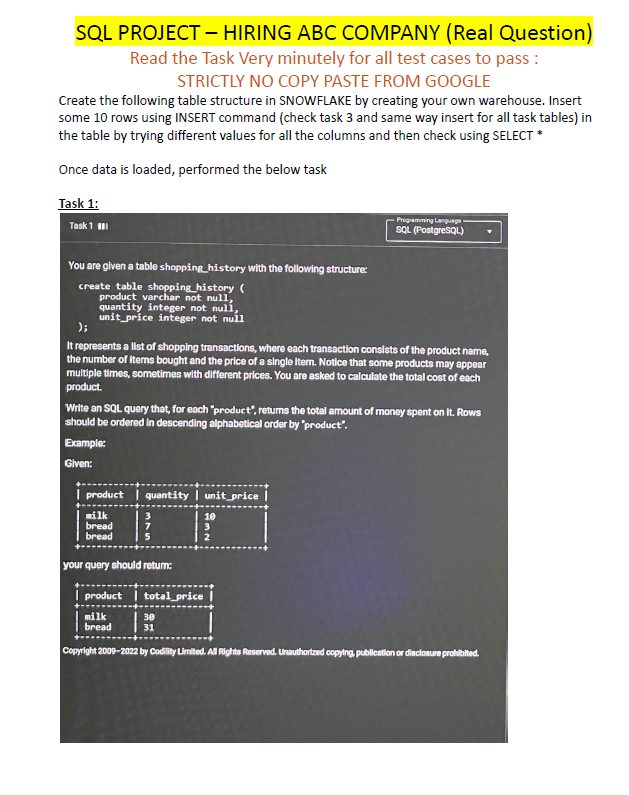
TASK #1:



**CODE:**

CREATE DATABASE HIRINGABCCOMPANY;

USE DATABASE HIRINGABCCOMPANY;

CREATE OR REPLACE TABLE SHOPPING\_HISTORY

(

PRODUCT VARCHAR NOT NULL,

QUANTITY INT NOT NULL,

UNIT\_PRICE INT NOT NULL

);

SELECT \* FROM SHOPPING\_HISTORY;

INSERT INTO SHOPPING\_HISTORY(PRODUCT, QUANTITY, UNIT\_PRICE)VALUES ('MILK', 3, 10);

INSERT INTO SHOPPING\_HISTORY(PRODUCT, QUANTITY, UNIT\_PRICE)VALUES ('BREAD', 7, 3);

INSERT INTO SHOPPING\_HISTORY(PRODUCT, QUANTITY, UNIT\_PRICE)VALUES ('BREAD', 5, 2);

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

FROM SHOPPING\_HISTORY

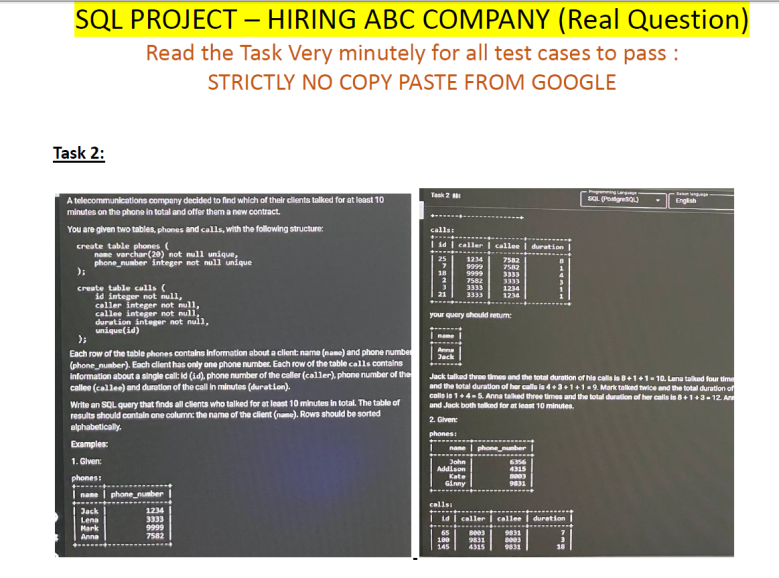
GROUP BY PRODUCT

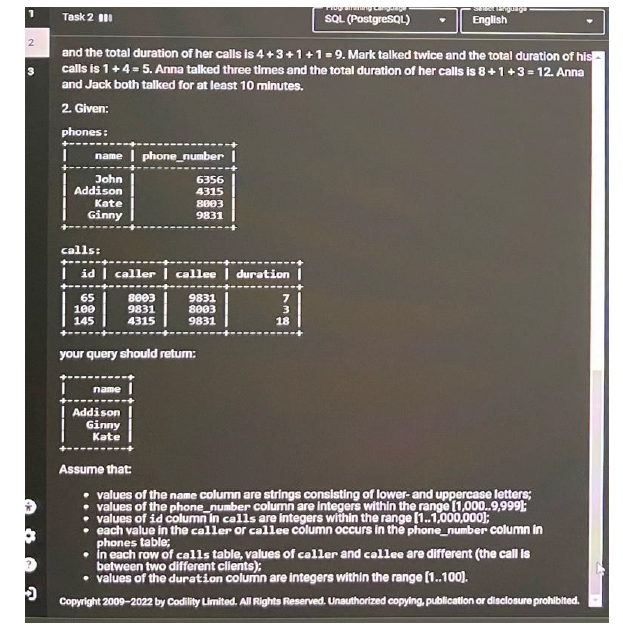
ORDER BY TOTAL\_PRICE;

**Output:**

****

TASK #2.1:





**Code:**

--TASK #2 ---

CREATE OR REPLACE DATABASE TELECOM;

USE DATABASE TELECOM;

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 UNIQUE,

CALLER INT NOT NULL,

CALLEE INT NOT NULL,

DURATION INT NOT NULL

);

SELECT \* FROM PHONES;

SELECT \* FROM CALLS;

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('JACK', 1234);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('LENA', 3333);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('MARK', 9999);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('ANNA', 7582);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (25, 1234, 7582, 8);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (7, 9999, 7582, 1);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (18, 9999, 3333, 4);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (2, 7582,3333, 3);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (3, 3333, 1234, 1);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (21, 3333, 1234, 1);

WITH TELECOM\_NAMES AS

(

SELECT a.CALLER

FROM

(

SELECT ID, CALLER, DURATION FROM CALLS

) AS a

INNER JOIN

(SELECT ID, CALLEE, DURATION FROM CALLS

) AS b ON a.ID = b.ID

WHERE

(a.DURATION + b.DURATION) >=10

UNION ALL

SELECT b.CALLEE

FROM

(

SELECT ID, CALLER, DURATION FROM CALLS

) AS a

INNER JOIN

(SELECT ID, CALLEE, DURATION FROM CALLS

) AS b ON a.ID = b.ID

WHERE

(a.DURATION + b.DURATION) >=10

)

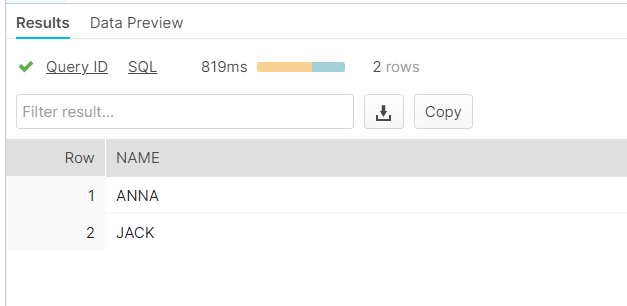
SELECT NAME FROM TELECOM\_NAMES c

INNER JOIN

PHONES p ON c.CALLER = p.PHONE\_NUMBER

ORDER BY NAME;

**Output:**



Task #2.2:

Code:

CREATE OR REPLACE DATABASE TELECOM1;

USE DATABASE TELECOM1;

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 UNIQUE,

CALLER INT NOT NULL,

CALLEE INT NOT NULL,

DURATION INT NOT NULL

);

SELECT \* FROM PHONES;

SELECT \* FROM CALLS;

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('JOHN', 6356);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('ADDISON', 4315);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('KATE', 8003);

INSERT INTO PHONES (NAME, PHONE\_NUMBER) VALUES ('GINNY', 9831);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (65, 8003, 9831, 7);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (100, 9831, 8003, 3);

INSERT INTO CALLS (ID, CALLER, CALLEE, DURATION) VALUES (145, 4315, 9831, 18);

WITH TELECOM\_NAMES1 AS

(

SELECT a.CALLER

FROM

(SELECT ID, CALLER, DURATION FROM CALLS) AS a

INNER JOIN

(SELECT ID, CALLEE, DURATION FROM CALLS) AS b

ON

a.ID = b.ID

WHERE

(a.DURATION + b.DURATION) >=10

UNION ALL

SELECT b.CALLEE

FROM

(SELECT ID, CALLER, DURATION FROM CALLS) AS a

INNER JOIN

(SELECT ID, CALLEE, DURATION FROM CALLS) AS b

ON

a.ID = b.ID

WHERE

(a.DURATION + b.DURATION) >=10

)

SELECT NAME FROM TELECOM\_NAMES1 c

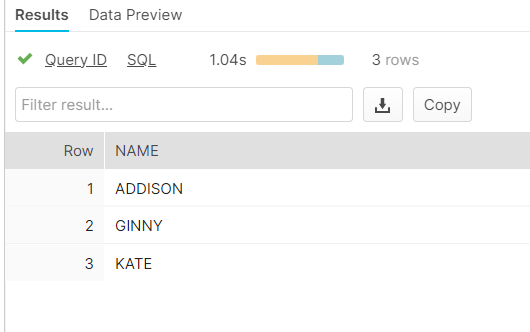
INNER JOIN

PHONES p ON c.CALLER = p.PHONE\_NUMBER

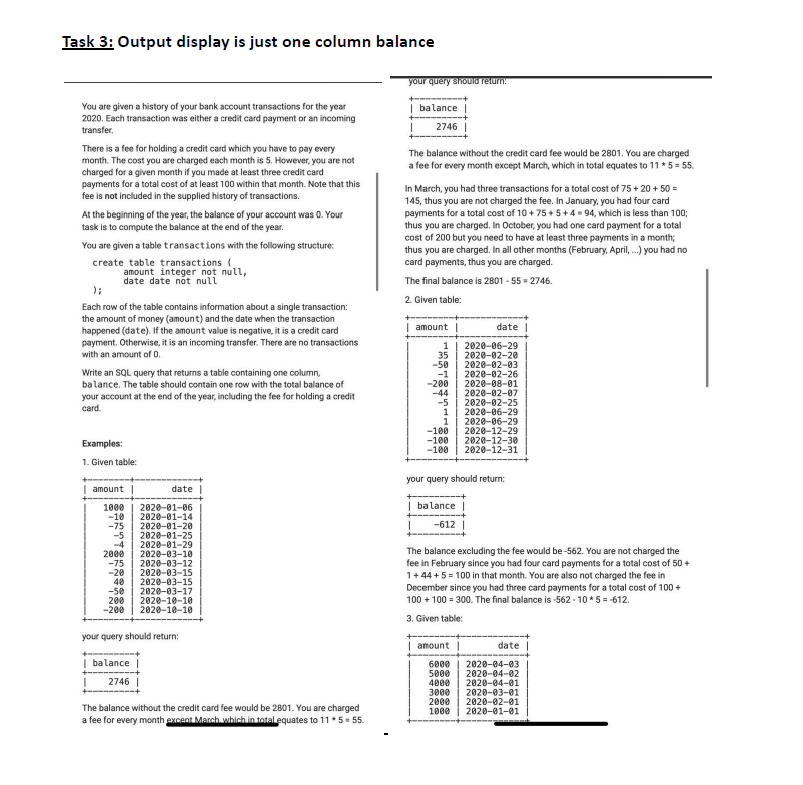
GROUP BY NAME

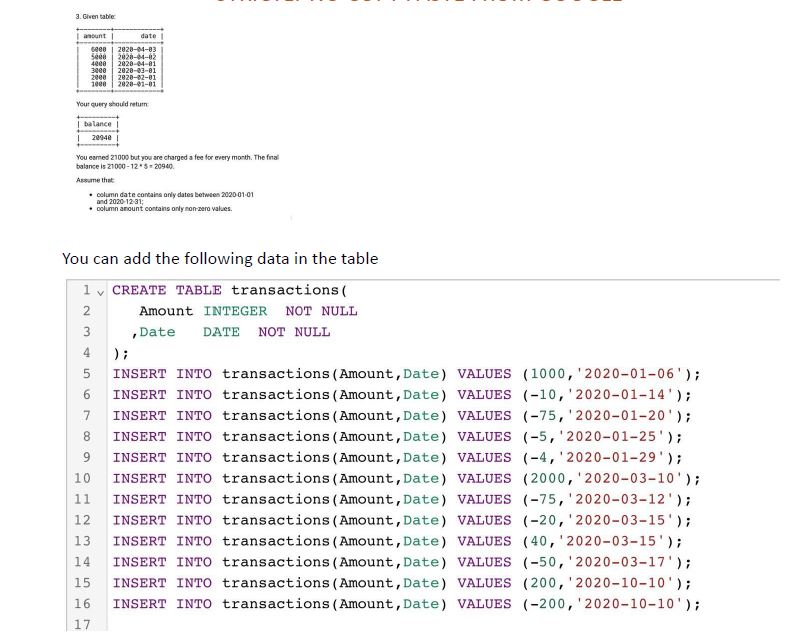
ORDER BY NAME;

**Output:**



Task #3:





**Code for Task #3.1:**

CREATE OR REPLACE DATABASE TRANSACTIONS;

USE DATABASE TRANSACTIONS;

CREATE OR REPLACE TABLE TRANSACTIONS

(

AMOUNT INT NOT NULL,

DATE DATE NOT NULL

);

SELECT \* FROM TRANSACTIONS;

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (1000, '2020-01-06');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-10, '2020-01-14');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-75, '2020-01-20');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-5, '2020-01-25');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-4, '2020-01-29');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (2000, '2020-03-10');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-75, '2020-03-12');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-20, '2020-03-15');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (40, '2020-03-15');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-50, '2020-03-17');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (200, '2020-10-10');

INSERT INTO TRANSACTIONS(AMOUNT, DATE) VALUES (-200, '2020-10-10');

SELECT SUM(AMOUNT) AS BALANCE FROM TRANSACTIONS;

SELECT MONTH(DATE) AS MONTH FROM TRANSACTIONS;

WITH TRNX AS(

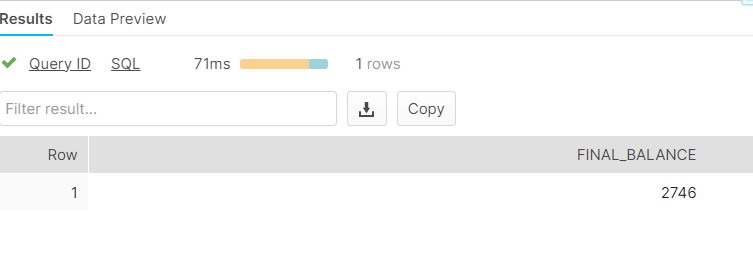
SELECT MONTH(DATE) AS MONTH FROM TRANSACTIONS WHERE AMOUNT LIKE '-%'

GROUP BY 1

HAVING SUM(AMOUNT)<=-100 AND COUNT(MONTH)>=3)

SELECT (SELECT SUM(AMOUNT)FROM TRANSACTIONS)-(60-COUNT(MONTH)\*5) AS FINAL\_BALANCE FROM TRNX ;

**Output for Task 3.1**:



Task #3.2 code:

CREATE OR REPLACE TABLE TRANSACTIONS1

(

AMOUNT INT NOT NULL,

DATE DATE NOT NULL

);

SELECT \* FROM TRANSACTIONS1;

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (1, '2020-06-29');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (35, '2020-02-20');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-50, '2020-02-03');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-1, '2020-02-26');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-200, '2020-08-01');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-44, '2020-02-07');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-5, '2020-02-25');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (1, '2020-06-29');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (1, '2020-06-29');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-100, '2020-12-29');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-100, '2020-12-30');

INSERT INTO TRANSACTIONS1(AMOUNT, DATE) VALUES (-100, '2020-12-31');

WITH TRNSX1 AS(

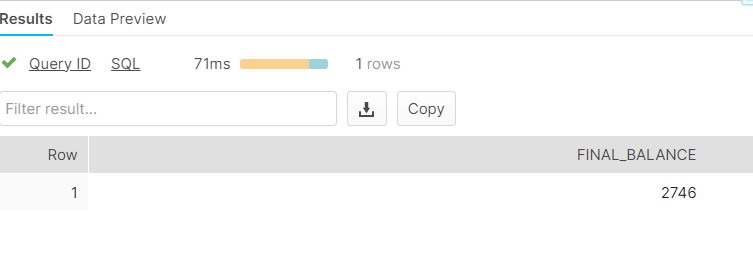
SELECT MONTH(DATE) AS MONTH FROM TRANSACTIONS1 WHERE AMOUNT LIKE '-%'

GROUP BY 1

HAVING SUM(AMOUNT)<=-100 AND COUNT(MONTH)>=3)

SELECT(SELECT SUM(AMOUNT) FROM TRANSACTIONS1) - (60-COUNT(MONTH)\*5) AS FINAL\_BALANCE1 FROM TRNSX1;

**Output screenshot for Task 3.2**:



**Code for Task #3.3**:

CREATE OR REPLACE TABLE TRANSACTIONS2

(

AMOUNT INT NOT NULL,

DATE DATE NOT NULL

);

SELECT \* FROM TRANSACTIONS2;

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (6000, '2020-04-03');

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (5000, '2020-04-02');

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (4000, '2020-04-01');

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (3000, '2020-03-01');

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (2000, '2020-02-01');

INSERT INTO TRANSACTIONS2(AMOUNT, DATE) VALUES (1000, '2020-01-01');

WITH TRNSX2 AS(

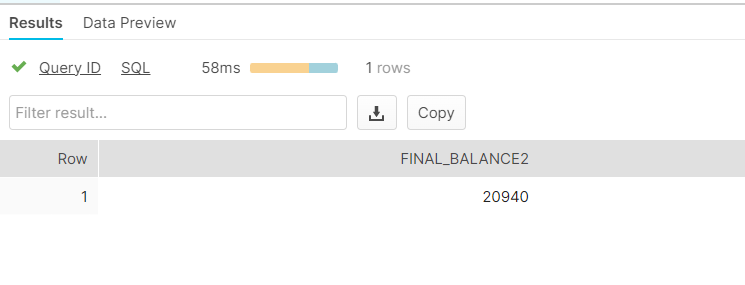
SELECT MONTH(DATE) AS MONTH FROM TRANSACTIONS2 WHERE AMOUNT LIKE '-%'

GROUP BY 1

HAVING SUM(AMOUNT)<=-100 AND COUNT(MONTH)>=3)

SELECT(SELECT SUM(AMOUNT) FROM TRANSACTIONS2) - (60-COUNT(MONTH)\*5) AS FINAL\_BALANCE2 FROM TRNSX2;

**Output for Task #3.3:**

****