COMP3611 A3

Mengyang Zhang

T00696467

Q1.

DROP TABLE OWNS;

CREATE TABLE OWNS(

ClientNbr INT NOT NULL,

AccountNbr INT NOT NULL,

PRIMARY KEY(AccountNbr, ClientNbr),

FOREIGN KEY(ClientNbr) REFERENCES CLIENT,

FOREIGN KEY(AccountNbr) REFERENCES ACCOUNT

);

DROP TABLE CLIENT;

CREATE TABLE CLIENT(

ClientNbr INT NOT NULL,

FName VARCHAR(20),

LName VARCHAR(20),

Street VARCHAR(20),

City VARCHAR(20),

Prov VARCHAR(20),

PostCd VARCHAR(20),

Phone INT,

Email VARCHAR(20),

PRIMARY KEY(ClientNbr)

);

DROP TABLE ACCOUNT;

CREATE TABLE ACCOUNT(

AccountNbr INT NOT NULL,

Balance DECIMAL NOT NULL,

PRIMARY KEY(AccountNbr)

);

DROP TABLE TX\_TYPE;

CREATE TABLE TX\_TYPE(

Code VARCHAR(1),

Description VARCHAR(20),

PRIMARY KEY(Code)

);

DROP TABLE BRANCH;

CREATE TABLE BRANCH(

Nbr INT,

Name VARCHAR(20),

PRIMARY KEY(Nbr)

);

DROP TABLE MERCHANT;

CREATE TABLE MERCHANT(

Nbr INT,

Name VARCHAR(20),

PRIMARY KEY(Nbr)

);

DROP TABLE TRANSACTION;

CREATE TABLE TRANSACTION(

TxNbr INT,

TxCode VARCHAR(1),

AccountNbr INT,

Amount DECIMAL,

DT TIMESTAMP,

RefNbr INT,

Balance DECIMAL,

PRIMARY KEY(TxNbr),

FOREIGN KEY(AccountNbr) REFERENCES ACCOUNT,

FOREIGN KEY(TxCode) REFERENCES TX\_TYPE

);

INSERT INTO TRANSACTION VALUES (1, 'X', 1000001, 123.45, TO\_DATE('2019-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 101 ,100);

INSERT INTO TRANSACTION VALUES (2, 'D', 1000000, 234.56, TO\_DATE('2019-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 101 ,100);

INSERT INTO TRANSACTION VALUES (3, 'D', 1000001, 345.67, TO\_DATE('2019-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 111,100);

INSERT INTO TRANSACTION VALUES (4, 'D', 1000001, 100, TO\_DATE('2019-05-01 10:00:00','YYYY-MM-DD HH24:MI:SS'), 101,100);

INSERT INTO TRANSACTION VALUES (5, 'D', 1000001, 200, TO\_DATE('2019-05-11 11:00:00','YYYY-MM-DD HH24:MI:SS'), 101,300);

INSERT INTO TRANSACTION VALUES (6, 'D', 1000001, 300, TO\_DATE('2019-05-21 12:00:00','YYYY-MM-DD HH24:MI:SS'), 101,600);

INSERT INTO TRANSACTION VALUES (7, 'W', 1000001, 50, TO\_DATE('2019-05-29 10:00:00','YYYY-MM-DD HH24:MI:SS'), 102,550);

INSERT INTO TRANSACTION VALUES (8, 'W', 1000001, 75, TO\_DATE('2019-05-29 11:00:00','YYYY-MM-DD HH24:MI:SS'), 103,475);

INSERT INTO TRANSACTION VALUES (9, 'D', 1000001, 123.45, TO\_DATE('2019-06-15 13:00:00','YYYY-MM-DD HH24:MI:SS'), 101,598.45);

INSERT INTO TRANSACTION VALUES (10, 'D', 1000002, 1000.00, TO\_DATE('2019-05-15 09:00:00','YYYY-MM-DD HH24:MI:SS'), 104,1000.00);

INSERT INTO TRANSACTION VALUES (11, 'W', 1000002, 456.78, TO\_DATE('2019-05-15 09:05:00','YYYY-MM-DD HH24:MI:SS'), 104,543.22);

INSERT INTO TRANSACTION VALUES (12, 'D', 1000003, 456.78, TO\_DATE('2019-05-15 09:10:00','YYYY-MM-DD HH24:MI:SS'), 104,456.78);

INSERT INTO TRANSACTION VALUES (13, 'W', 1000003, 500.00, TO\_Date('2019-05-18 14:00:00','YYYY-MM-DD HH24:MI:SS'), 104 ,-43.22);

INSERT INTO TRANSACTION VALUES (14, 'D', 1000003, 100.00, TO\_Date('2019-05-20 13:00:00','YYYY-MM-DD HH24:MI:SS'), 104 ,56.78);

INSERT INTO TRANSACTION VALUES (15, 'P', 1000003, 65.78, TO\_Date('2019-05-20 14:50:00','YYYY-MM-DD HH24:MI:SS'), 304 ,-9);

INSERT INTO TRANSACTION VALUES (16, 'B', 1000002, 100.00, TO\_Date('2019-05-21 09:00:00','YYYY-MM-DD HH24:MI:SS'), 301 ,443.22);

INSERT INTO TRANSACTION VALUES (17, 'P', 1000002, 200.00, TO\_Date('2019-05-21 10:00:00','YYYY-MM-DD HH24:MI:SS'), 302 ,243.22);

INSERT INTO TRANSACTION VALUES (18, 'R', 1000002, 50.00, TO\_Date('2019-05-26 12:34:00','YYYY-MM-DD HH24:MI:SS'), 301 ,293.22);

INSERT INTO TRANSACTION VALUES (20, 'D', 1000004, 2000.00, TO\_Date('2019-06-01 13:00:00','YYYY-MM-DD HH24:MI:SS'), 101 ,2000);

INSERT INTO TRANSACTION VALUES (21, 'D', 1000005, 2000.00, TO\_DATE('2019-06-01 13:00:00','YYYY-MM-DD HH24:MI:SS'), 101 ,2000);

INSERT INTO TRANSACTION VALUES (22, 'D', 1000005, 2000.00, TO\_DATE('2019-06-01 14:00:00','YYYY-MM-DD HH24:MI:SS'), 102 ,4000);

INSERT INTO TRANSACTION VALUES (23, 'B', 1000005, 3456.78, TO\_DATE('2019-06-10 12:00:00','YYYY-MM-DD HH24:MI:SS'), 301 ,543.22);

INSERT INTO TRANSACTION VALUES (24, 'B', 1000005, 432.10, TO\_DATE('2019-06-15 14:30:00','YYYY-MM-DD HH24:MI:SS'), 302 ,111.12);

INSERT INTO TRANSACTION VALUES (25, 'B', 1000005, 100.00, TO\_DATE('2019-06-20 15:55:00','YYYY-MM-DD HH24:MI:SS'), 303 ,11.12);

INSERT INTO TRANSACTION VALUES (26, 'B', 1000005, 80.00, TO\_DATE('2019-06-25 16:56:00','YYYY-MM-DD HH24:MI:SS'), 304 ,-68.88);

INSERT INTO OWNS VALUES(10002, 1000001);

INSERT INTO OWNS VALUES(10003, 1000002);

INSERT INTO OWNS VALUES(10003, 1000003);

INSERT INTO OWNS VALUES(10004, 1000002);

INSERT INTO OWNS VALUES(10004, 1000003);

INSERT INTO OWNS VALUES(10005, 1000004);

INSERT INTO OWNS VALUES(10005, 1000005);

INSERT INTO OWNS VALUES(10006, 1000006);

INSERT INTO OWNS VALUES(10007, 1000001);

INSERT INTO OWNS VALUES(10001, 1000007);

INSERT INTO ACCOUNT VALUES (1000001, 0);

INSERT INTO ACCOUNT VALUES (1000002, 0);

INSERT INTO ACCOUNT VALUES (1000003, 0);

INSERT INTO ACCOUNT VALUES (1000004, 0);

INSERT INTO ACCOUNT VALUES (1000005, 0);

INSERT INTO ACCOUNT VALUES (1000006, 0);

INSERT INTO CLIENT VALUES(10001, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO CLIENT VALUES(10002, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO CLIENT VALUES(10003, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO CLIENT VALUES(10004, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO CLIENT VALUES(10005, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO CLIENT VALUES(10006, 'M', 'Z','2009 JOHNS RD', 'VICTORIA', 'BC', 'V8R3H7',20485938, '2738579@gmail.com');

INSERT INTO TX\_TYPE VALUES('D', 'Deposit');

INSERT INTO TX\_TYPE VALUES('W', 'Withdrawal');

INSERT INTO TX\_TYPE VALUES('B', 'Bill Payment');

INSERT INTO TX\_TYPE VALUES('P', 'Purchase');

INSERT INTO TX\_TYPE VALUES('R', 'Return');

INSERT INTO BRANCH VALUES(101, 'M');

INSERT INTO BRANCH VALUES(102, 'M');

INSERT INTO BRANCH VALUES(103, 'M');

INSERT INTO BRANCH VALUES(104, 'M');

INSERT INTO MERCHANT VALUES(301, 'M');

INSERT INTO MERCHANT VALUES(302, 'M');

INSERT INTO MERCHANT VALUES(303, 'M');

INSERT INTO MERCHANT VALUES(304, 'M');

Q2.

Join of Transaction to Type description (1 mark)

CREATE VIEW Transaction\_description

AS SELECT \*

FROM TRANSACTION,TX\_TYPE

WHERE TxCode=Code;

Join of Client to Account via the Owns table, containing the client’s number, name, the account number, and balance. (1 mark) (Hint: Use an outer join as one client has no accounts.)

CREATE VIEW transaction\_detail

AS(

SELECT \*

FROM TRANSACTION,BRANCH

WHERE TRANSACTION.RefNbr = branch.nbr

UNION

SELECT \*

FROM TRANSACTION,MERCHANT

WHERE TRANSACTION.RefNbr = merchant.nbr);

Join of Deposit and Withdraw transactions to Bank Branch UNION with join of Bill Payment and Debit Purchase, or Return transactions to Merchant (i.e., all transactions with appropriate reference name.) (2 marks)

CREATE VIEW Client\_Account

AS(

SELECT CLIENT.\*,ACCOUNT.\*

FROM (CLIENT LEFT OUTER JOIN OWNS ON client.clientnbr=owns.clientnbr LEFT OUTER JOIN ACCOUNT ON owns.accountnbr=account.accountnbr));

Q3.

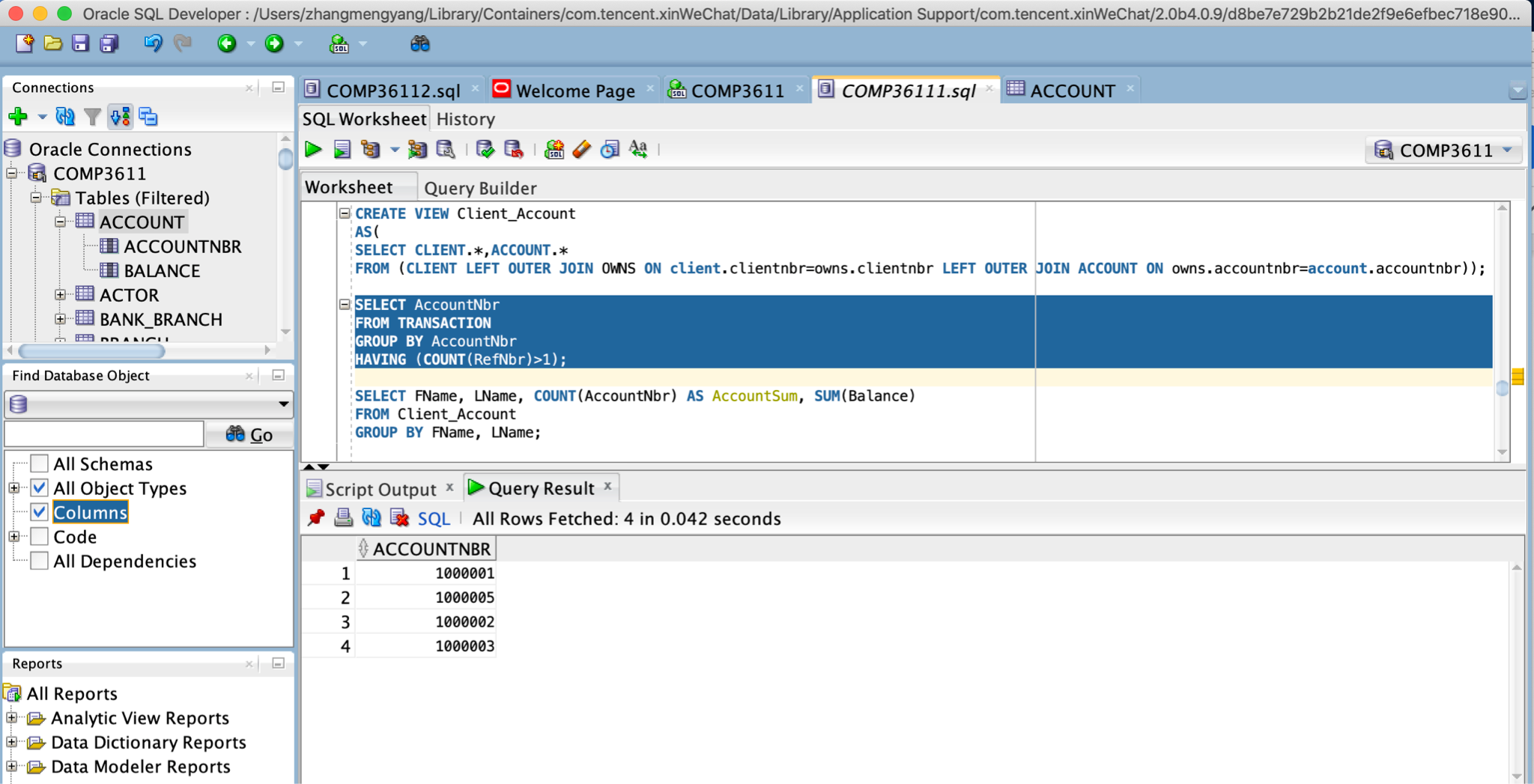
List of only the Accounts that have multiple Clients associated (1 mark)

SELECT AccountNbr

FROM TRANSACTION

GROUP BY AccountNbr

HAVING (COUNT(RefNbr)>1);



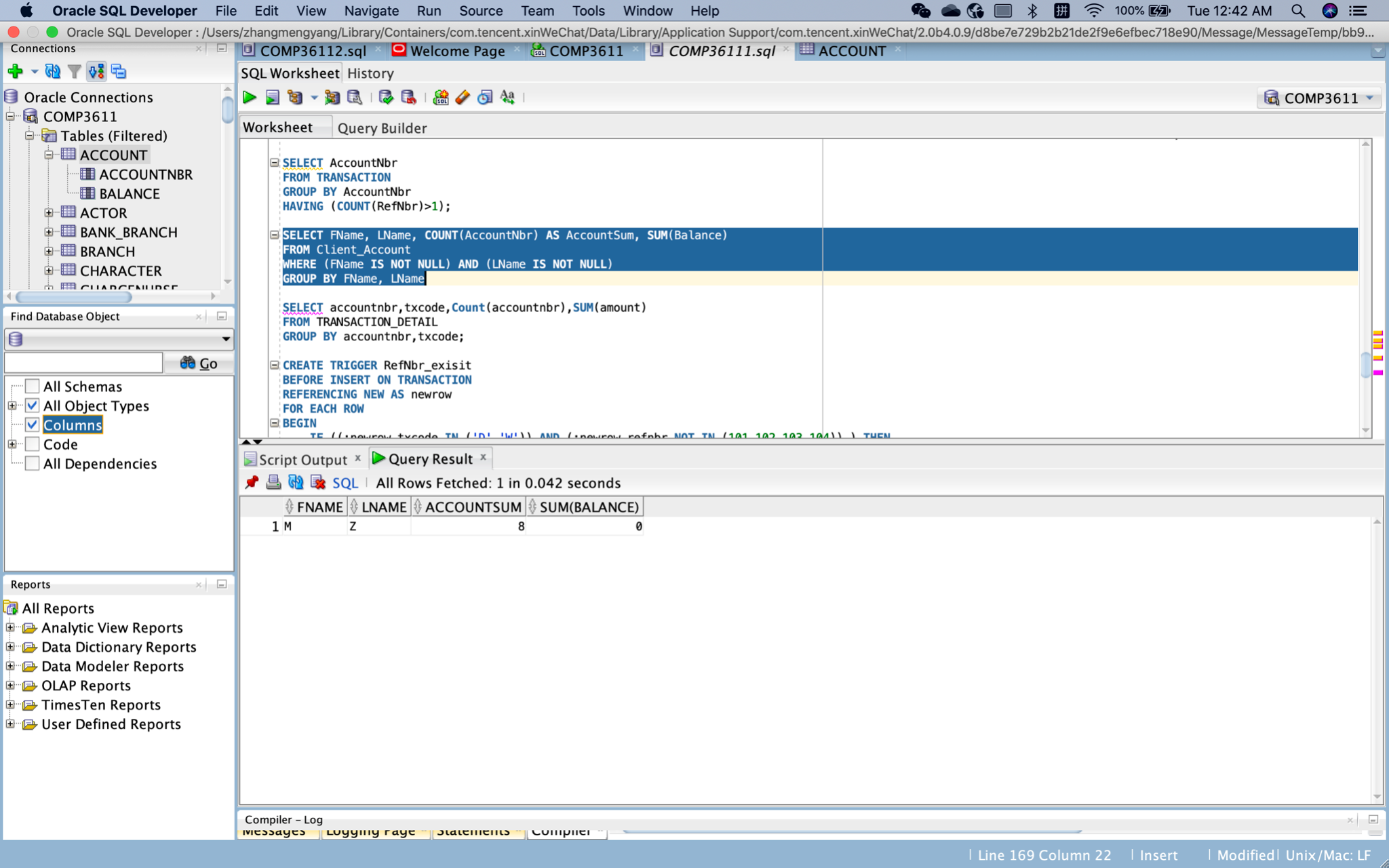
Provide an alphabetic list by last name of all Clients showing their full name (e.g., Bob Barlow), with the number of Accounts they hold and the total balance of those Accounts (1 mark)

SELECT FName, LName, COUNT(AccountNbr) AS AccountSum, SUM(Balance)

FROM Client\_Account

WHERE (FName IS NOT NULL) AND (LName IS NOT NULL)

GROUP BY FName, LName;

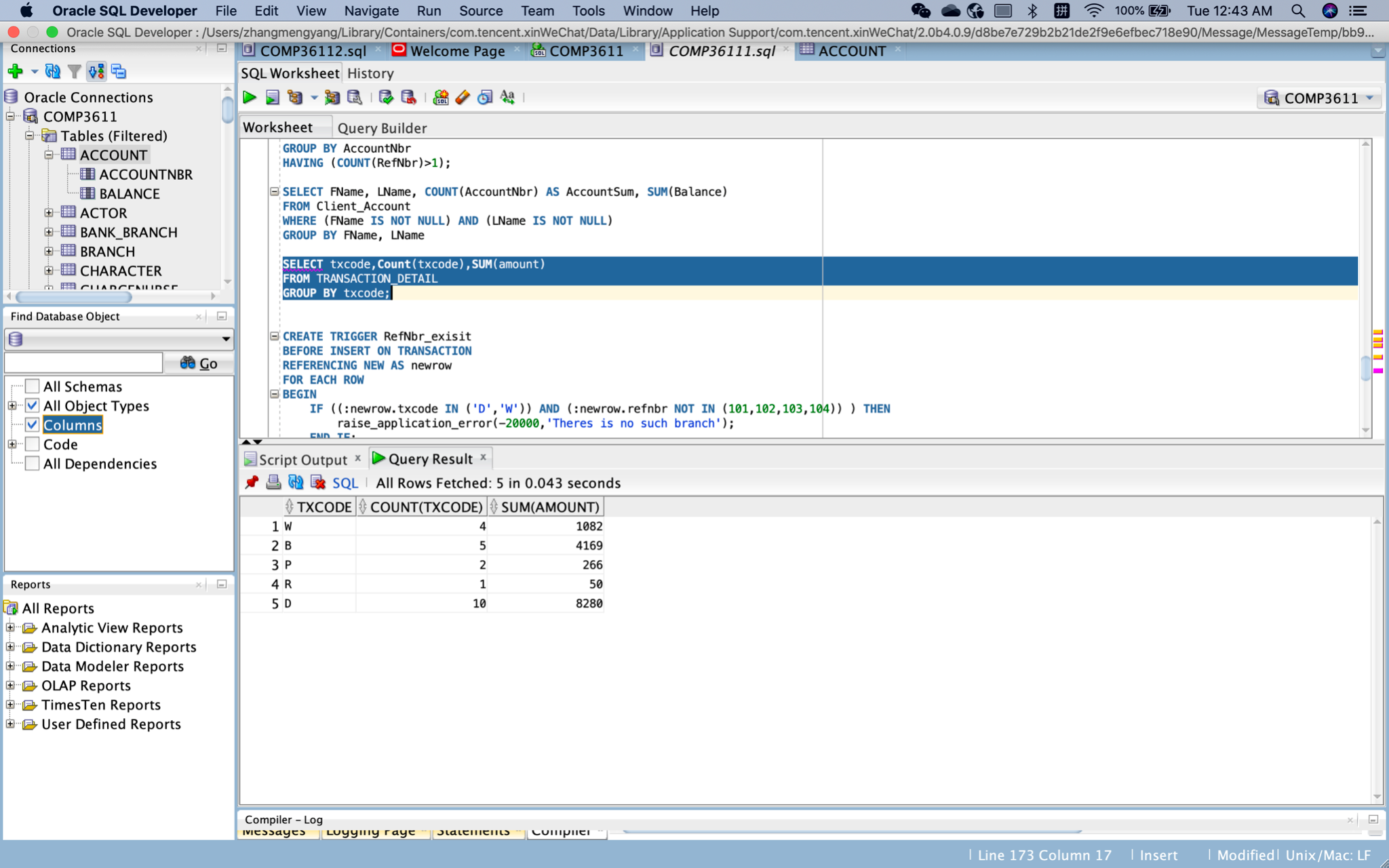


Provide acount and total amount of Transactions for each Type description (1 mark)

SELECT txcode,Count(txcode),SUM(amount)

FROM TRANSACTION\_DETAIL

GROUP BY txcode;



List of each Account showing the first Transaction date, type, and amount. (Hint: This is a correlated sub-query.) (2 marks)

SELECT accountnbr,dt,txcode,amount

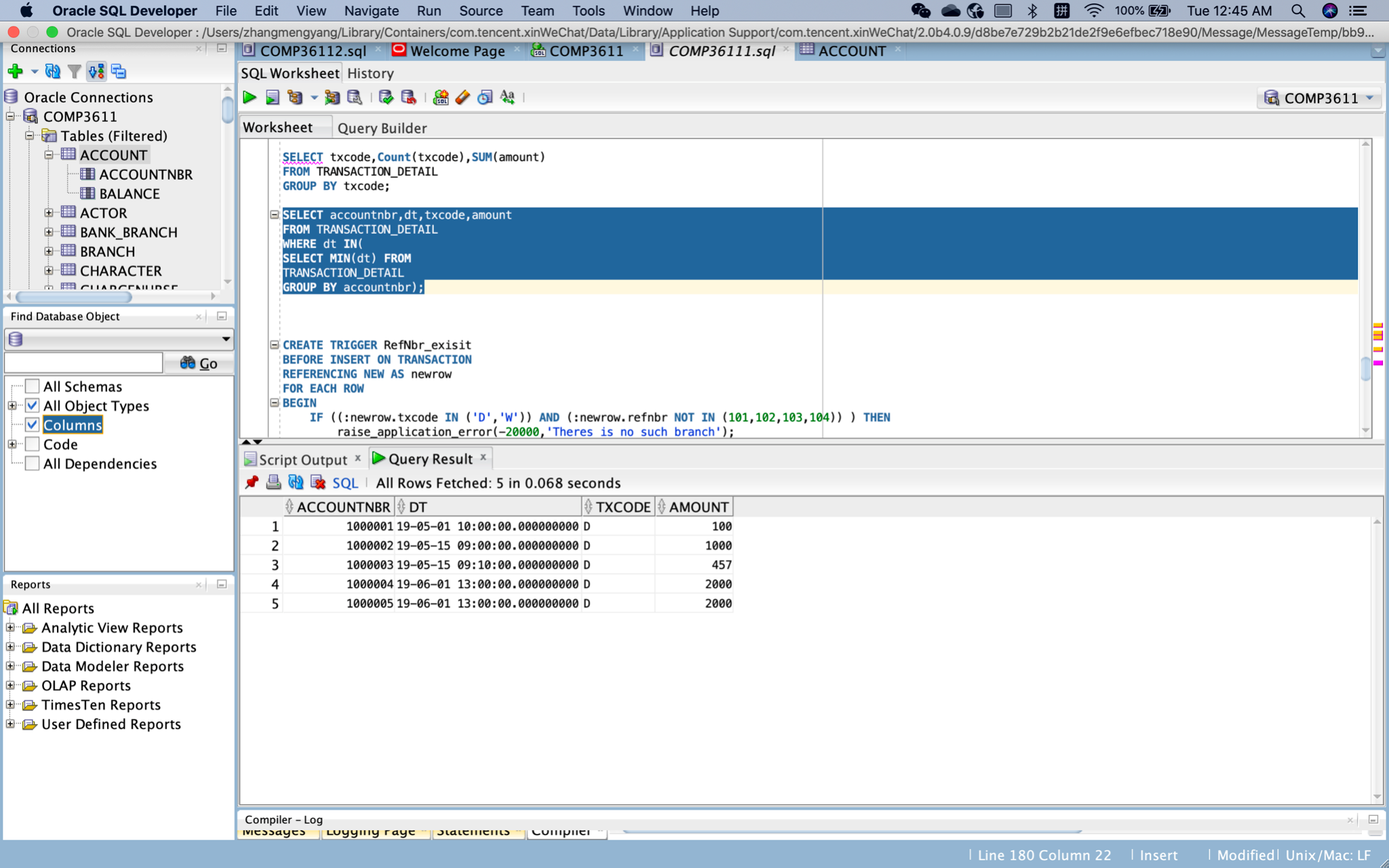
FROM TRANSACTION\_DETAIL

WHERE dt IN(

SELECT MIN(dt) FROM

TRANSACTION\_DETAIL

GROUP BY accountnbr);

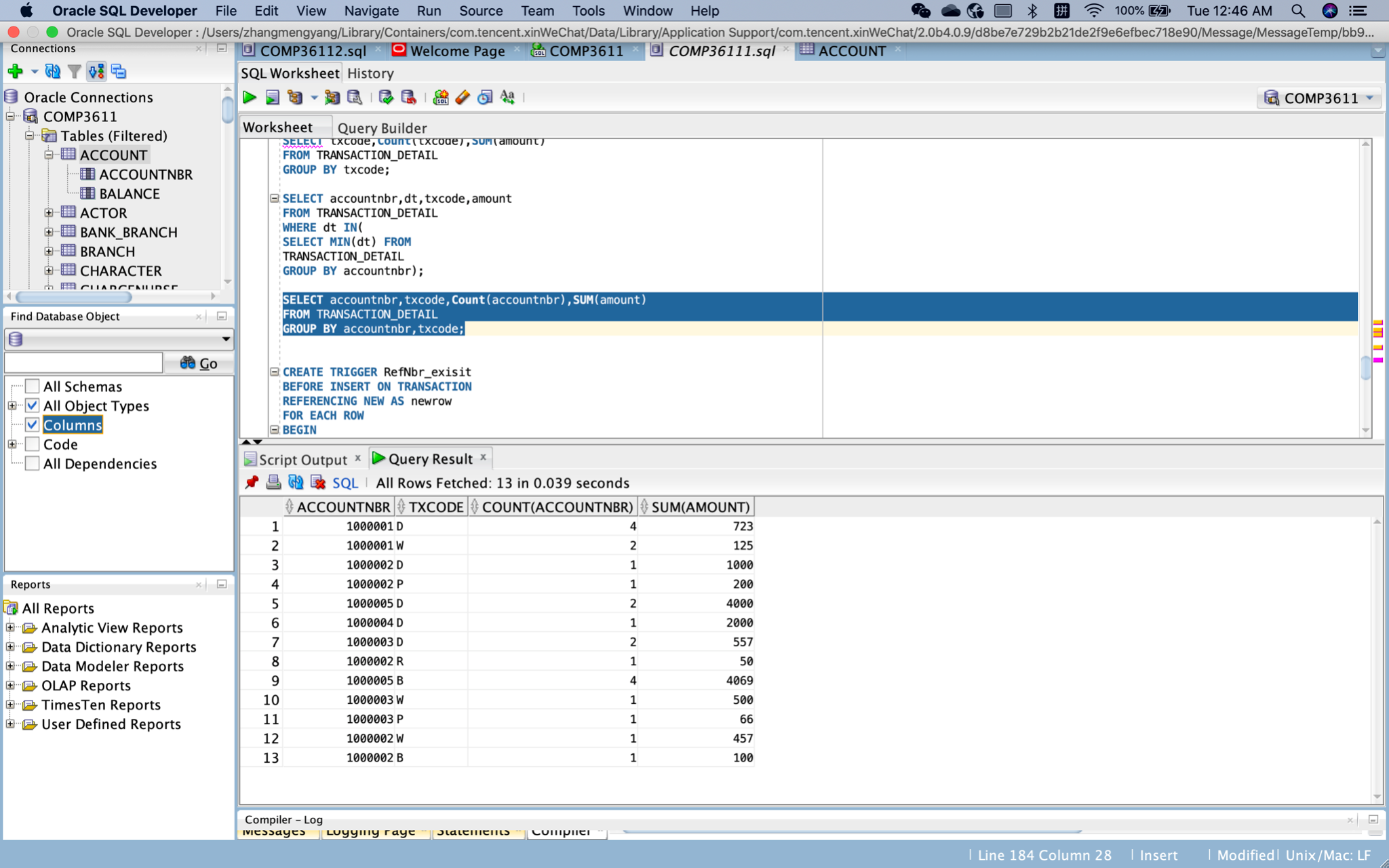


Count and Total Amount for each Transaction Type within each Account (Hint: This is an extended GROUP BY.) (2 marks)

SELECT accountnbr,txcode,Count(accountnbr),SUM(amount)

FROM TRANSACTION\_DETAIL

GROUP BY accountnbr,txcode;



Q4.

Trigger to enforce the referential integrity for the Transaction Ref\_Nbr: (3 marks)

CREATE TRIGGER RefNbr\_exisit

BEFORE INSERT ON TRANSACTION

REFERENCING NEW AS newrow

FOR EACH ROW

BEGIN

IF ((:newrow.txcode IN ('D','W')) AND (:newrow.refnbr NOT IN (101,102,103,104)) ) THEN

raise\_application\_error(-20000,'Theres is no such branch');

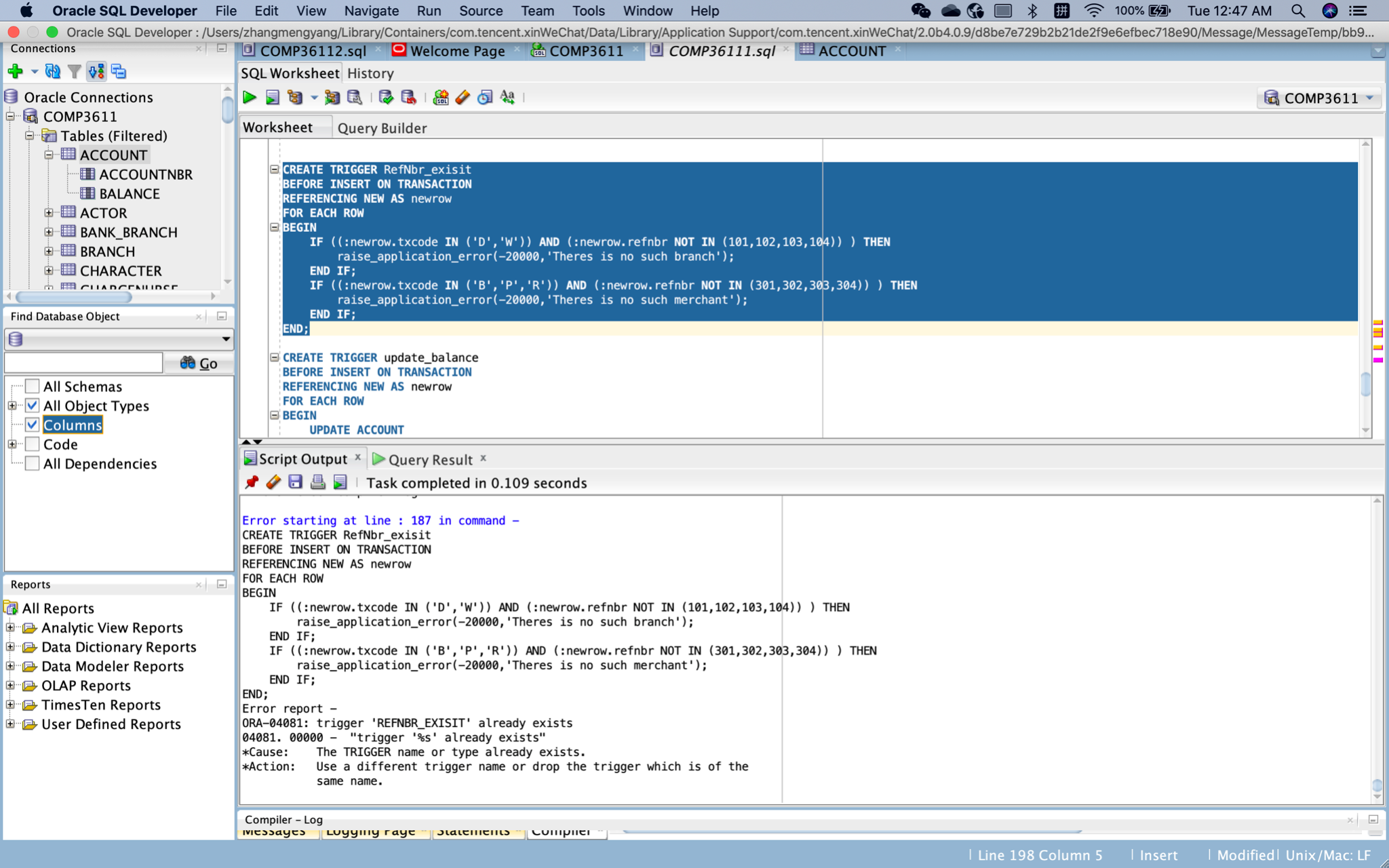
END IF;

IF ((:newrow.txcode IN ('B','P','R')) AND (:newrow.refnbr NOT IN (301,302,303,304)) ) THEN

raise\_application\_error(-20000,'Theres is no such merchant');

END IF;

END;



IfI try INSERT INTO TRANSACTION VALUES (30, 'D', 1000001, 100, TO\_DATE('2020-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 105 ,100); It reports an error “There is no such branch because 105 is not a valid branch number.

If I try INSERT INTO TRANSACTION VALUES (30, 'P', 1000001, 100, TO\_DATE('2020-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 401 ,100); It also reports an error “there is no such merchant” because 401 is not a valid merchant number.

Trigger to update the Account balance for each new transaction entered (assume that a transaction will never be updated or deleted). (3 marks)

CREATE TRIGGER update\_balance

BEFORE INSERT ON TRANSACTION

REFERENCING NEW AS newrow

FOR EACH ROW

BEGIN

UPDATE ACCOUNT

SET balance = :newrow.balence

WHERE accountnbr = :newrow.accountnbr;

END;

The balance of account 1000001 was 598 and the balance of account 1000002 was 293. After INSERT INTO TRANSACTION VALUES (27, 'D', 1000001, 100, TO\_DATE('2020-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 102 ,778);

INSERT INTO TRANSACTION VALUES (28, 'D', 1000002, 100, TO\_DATE('2020-05-01 12:00:00','YYYY-MM-DD HH24:MI:SS'), 102 ,983);

The balances become778 and 983 respectively.

A procedure that displays a nicely formatted audit statement for a given account number (as a parameter). This will show each transaction in date / time sequence along with the running balance. (4 marks)

CREATE OR REPLACE PROCEDURE show\_transaction(account\_nbr IN NUMBER) IS

vdt TRANSACTION.dt%TYPE;

vbalance TRANSACTION.balence%TYPE;

CURSOR cursor IS

SELECT dt,balence

FROM TRANSACTION

WHERE accountnbr = account\_nbr

ORDER BY dt;

BEGIN

OPEN cursor;

LOOP

FETCH cursor INTO vdt,vbalance;

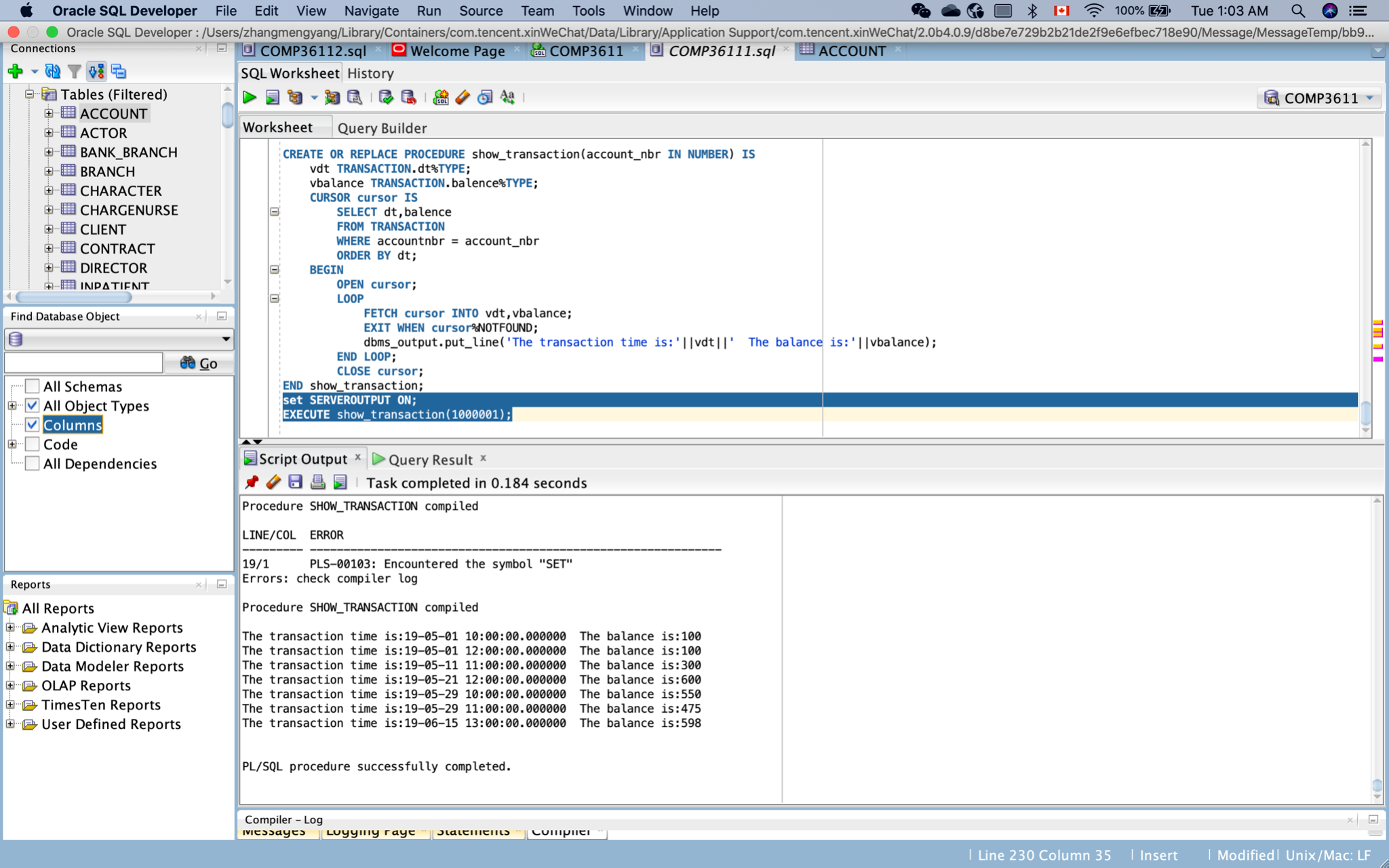
EXIT WHEN cursor%NOTFOUND;

dbms\_output.put\_line('The transaction time is:'||vdt||' The balance is:'||vbalance);

END LOOP;

CLOSE cursor;

END show\_transaction;



When I run EXECUTE show\_transaction(1000001). The transaction and balance detail of account 1000001 will be shown.