Final Project Deliverable COMP214

Advanced Database Concepts (Centennial College)

CrediMax

Project By: Leon Muente, Juan Diego, Sharma, Parth, Dubale, Kaleab

To: Sarajlic, Zoran

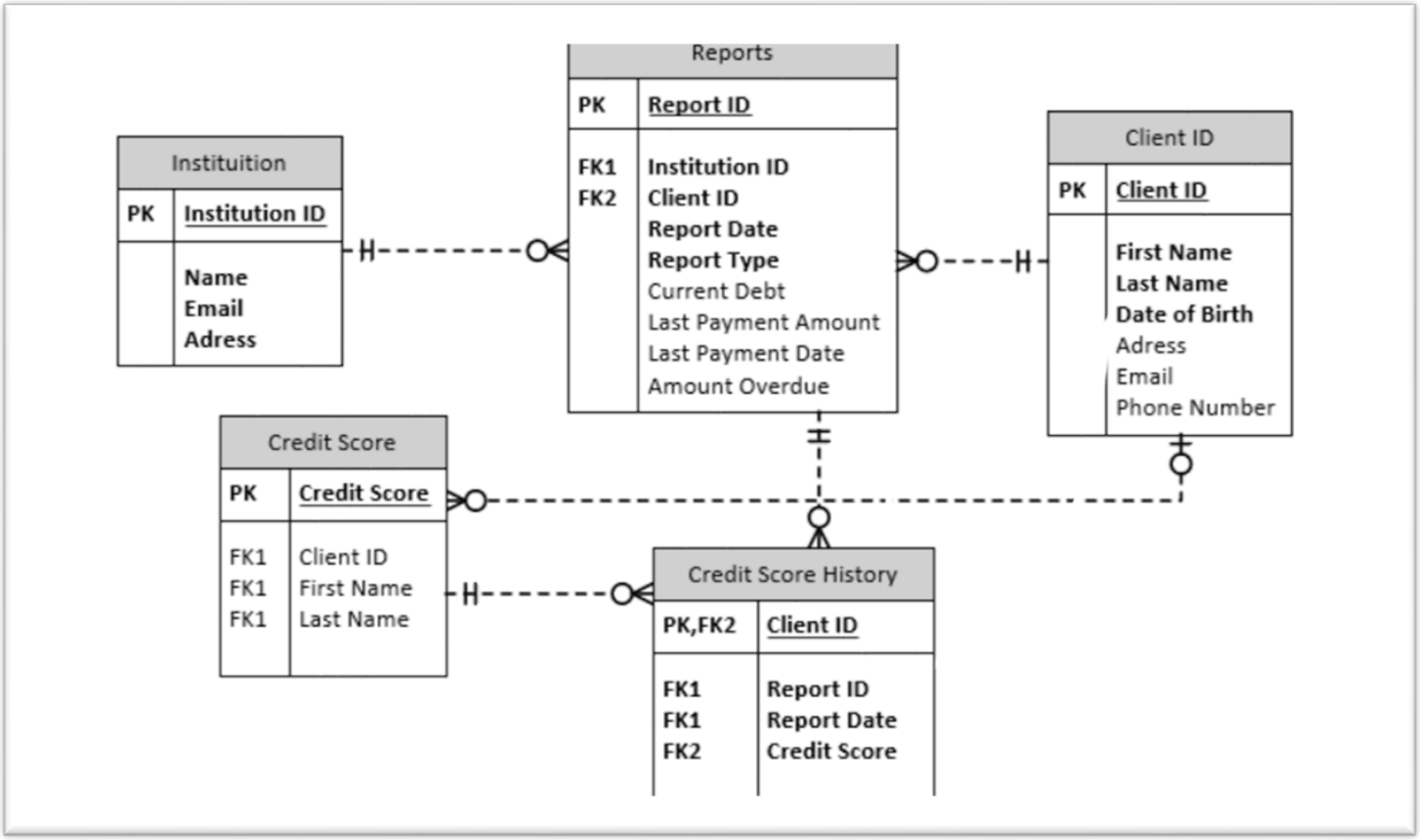
CrediMax

For this assignment, we have decided to make a credit score company offering exceptional service to cover demand at a competitive price point. Our credit score company CreditMax will use SQL and PLSQL codes to run its database system. The basics behind it are its five tables (Institution, Reports, Client ID, Credit Score, and Credit Score History).

Tables Description:

- The Institution table holds information about the institutions reporting on clients.
- The Reports holds every report made by any instuition at any given time on any given client.
- The Client ID table holds the clients personal and banking data.
- The Credit Score holds the current credit score for any given client.
- The Credit Score History table keeps all past credit scores of each client so that someone can cross-examine to see if there is any change.

ER Diagram



Instituition	
PK	Institution ID
	Name Email Adress

Reports	
PK	Report ID
FK1 FK2	Institution ID Client ID Report Date Report Type Current Debt Last Payment Amount Last Payment Date Amount Overdue

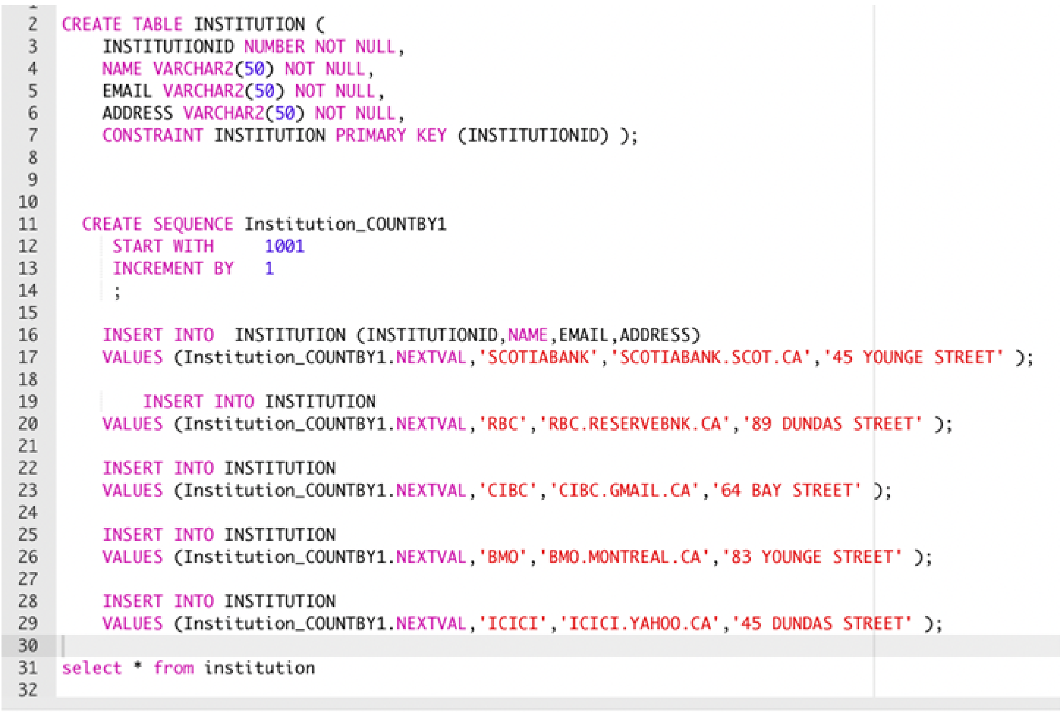
Client ID	
PK	Client ID
	First Name Last Name Date of Birth Adress Email Phone Number

Credit Score	
PK	Credit Score
FK1 FK1 FK1	Client ID First Name Last Name

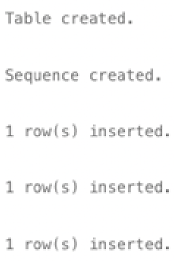
Credit Score History	
PK,FK2	Client ID
FK1 FK1 FK2	Report ID Report Date Credit Score

Scripts and results

Institution Table:

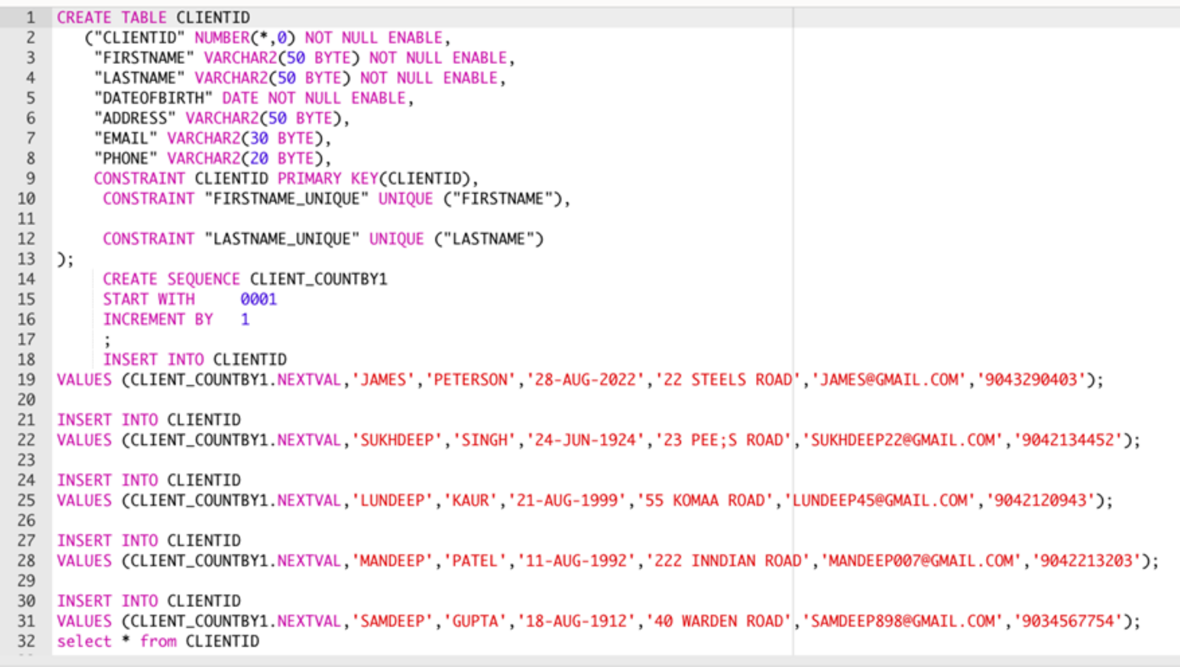


```sql
CREATE TABLE INSTITUTION (
    INSTITUTIONID NUMBER NOT NULL,
    NAME VARCHAR2(50) NOT NULL,
    EMAIL VARCHAR2(50) NOT NULL,
    ADDRESS VARCHAR2(50) NOT NULL,
    CONSTRAINT INSTITUTION PRIMARY KEY (INSTITUTIONID) );



  CREATE SEQUENCE Institution_COUNTBY1
     START WITH     1001
     INCREMENT BY   1
     ;

    INSERT INTO  INSTITUTION (INSTITUTIONID,NAME,EMAIL,ADDRESS)
    VALUES (Institution_COUNTBY1.NEXTVAL,'SCOTIABANK','SCOTIABANK.SCOT.CA','45 YOUNGE STREET' );

        INSERT INTO INSTITUTION
    VALUES (Institution_COUNTBY1.NEXTVAL,'RBC','RBC.RESERVEBNK.CA','89 DUNDAS STREET' );

    INSERT INTO INSTITUTION
    VALUES (Institution_COUNTBY1.NEXTVAL,'CIBC','CIBC.GMAIL.CA','64 BAY STREET' );

    INSERT INTO INSTITUTION
    VALUES (Institution_COUNTBY1.NEXTVAL,'BMO','BMO.MONTREAL.CA','83 YOUNGE STREET' );

    INSERT INTO INSTITUTION
    VALUES (Institution_COUNTBY1.NEXTVAL,'ICICI','ICICI.YAHOO.CA','45 DUNDAS STREET' );

select * from institution

```

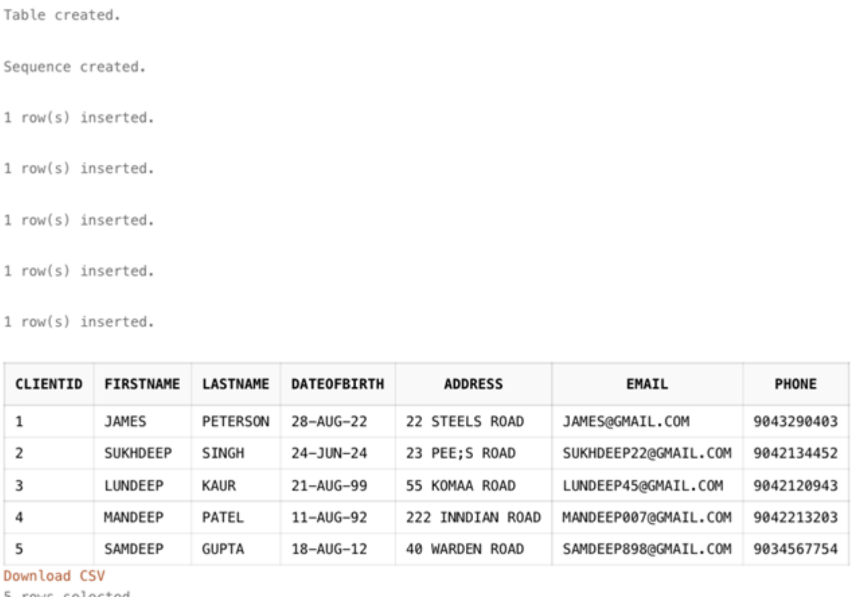


```
Table created.


Sequence created.


1 row(s) inserted.


1 row(s) inserted.


1 row(s) inserted.
```

Client ID Table:



```
CREATE TABLE CLIENTID
   ("CLIENTID" NUMBER(*,0) NOT NULL ENABLE,
    "FIRSTNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
    "LASTNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
    "DATEOFBIRTH" DATE NOT NULL ENABLE,
    "ADDRESS" VARCHAR2(50 BYTE),
    "EMAIL" VARCHAR2(30 BYTE),
    "PHONE" VARCHAR2(20 BYTE),
    CONSTRAINT CLIENTID PRIMARY KEY(CLIENTID),
     CONSTRAINT "FIRSTNAME_UNIQUE" UNIQUE ("FIRSTNAME"),

     CONSTRAINT "LASTNAME_UNIQUE" UNIQUE ("LASTNAME")
);
     CREATE SEQUENCE CLIENT_COUNTBY1
     START WITH     0001
     INCREMENT BY   1
     ;
     INSERT INTO CLIENTID
VALUES (CLIENT_COUNTBY1.NEXTVAL,'JAMES','PETERSON','28-AUG-2022','22 STEELS ROAD','JAMES@GMAIL.COM','9043290403');

INSERT INTO CLIENTID
VALUES (CLIENT_COUNTBY1.NEXTVAL,'SUKHDEEP','SINGH','24-JUN-1924','23 PEE;S ROAD','SUKHDEEP22@GMAIL.COM','9042134452');

INSERT INTO CLIENTID
VALUES (CLIENT_COUNTBY1.NEXTVAL,'LUNDEEP','KAUR','21-AUG-1999','55 KOMAA ROAD','LUNDEEP45@GMAIL.COM','9042120943');

INSERT INTO CLIENTID
VALUES (CLIENT_COUNTBY1.NEXTVAL,'MANDEEP','PATEL','11-AUG-1992','222 INNDIAN ROAD','MANDEEP007@GMAIL.COM','9042213203');

INSERT INTO CLIENTID
VALUES (CLIENT_COUNTBY1.NEXTVAL,'SAMDEEP','GUPTA','18-AUG-1912','40 WARDEN ROAD','SAMDEEP898@GMAIL.COM','9034567754');
select * from CLIENTID
```



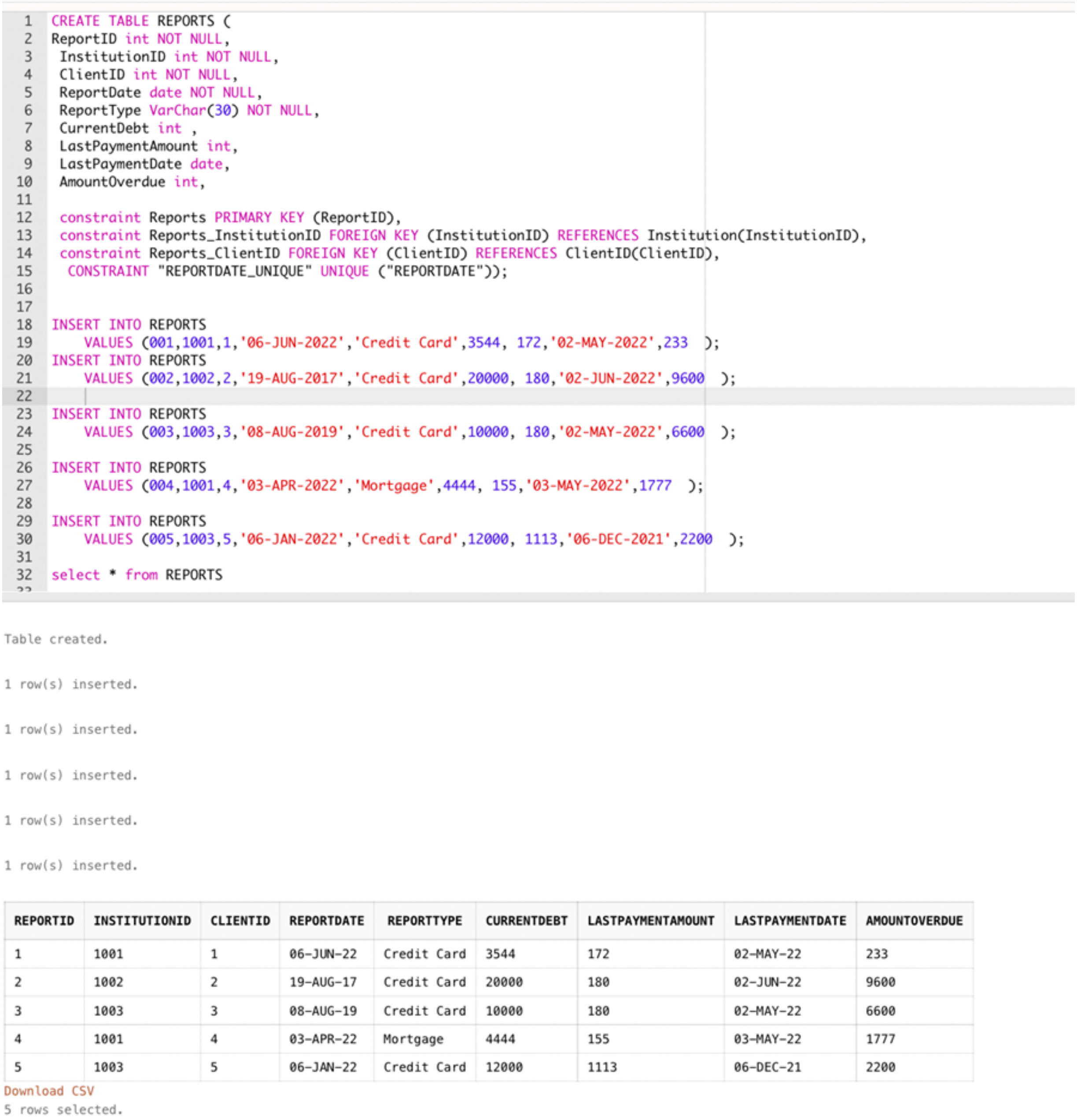
```
Table created.

Sequence created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

CLIENTID	FIRSTNAME	LASTNAME	DATEOFBIRTH	ADDRESS	EMAIL	PHONE
1	JAMES	PETERSON	28-AUG-22	22 STEELS ROAD	JAMES@GMAIL.COM	9043290403
2	SUKHDEEP	SINGH	24-JUN-24	23 PEE;S ROAD	SUKHDEEP22@GMAIL.COM	9042134452
3	LUNDEEP	KAUR	21-AUG-99	55 KOMAA ROAD	LUNDEEP45@GMAIL.COM	9042120943
4	MANDEEP	PATEL	11-AUG-92	222 INNDIAN ROAD	MANDEEP007@GMAIL.COM	9042213203
5	SAMDEEP	GUPTA	18-AUG-12	40 WARDEN ROAD	SAMDEEP898@GMAIL.COM	9034567754

Download CSV

5 rows selected

Reports Table:



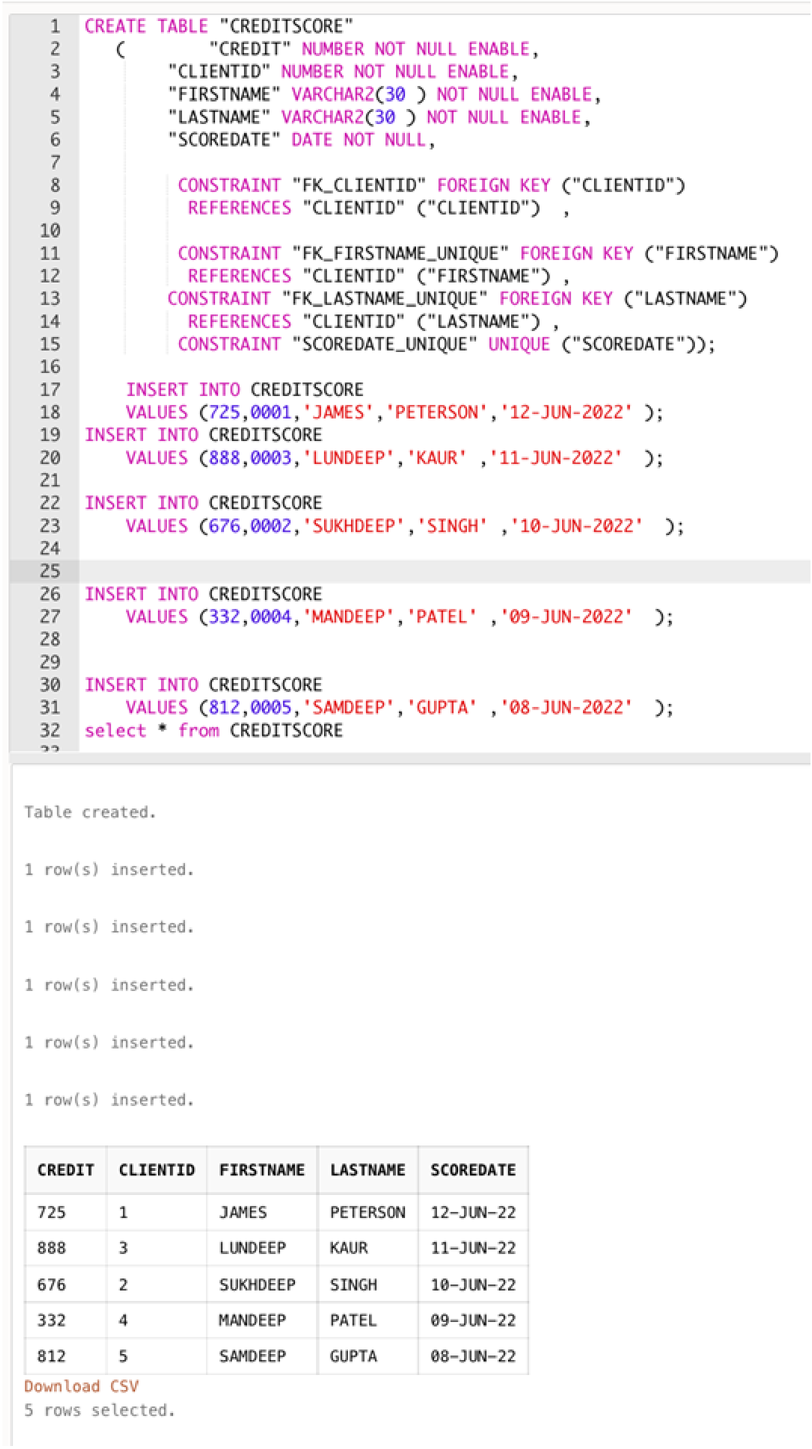
```sql
CREATE TABLE REPORTS (
ReportID int NOT NULL,
 InstitutionID int NOT NULL,
 ClientID int NOT NULL,
 ReportDate date NOT NULL,
 ReportType VarChar(30) NOT NULL,
 CurrentDebt int ,
 LastPaymentAmount int,
 LastPaymentDate date,
 AmountOverdue int,

 constraint Reports PRIMARY KEY (ReportID),
 constraint Reports_InstitutionID FOREIGN KEY (InstitutionID) REFERENCES Institution(InstitutionID),
 constraint Reports_ClientID FOREIGN KEY (ClientID) REFERENCES ClientID(ClientID),
  CONSTRAINT "REPORTDATE_UNIQUE" UNIQUE ("REPORTDATE"));


INSERT INTO REPORTS
    VALUES (001,1001,1,'06-JUN-2022','Credit Card',3544, 172,'02-MAY-2022',233  );
INSERT INTO REPORTS
    VALUES (002,1002,2,'19-AUG-2017','Credit Card',20000, 180,'02-JUN-2022',9600  );

INSERT INTO REPORTS
    VALUES (003,1003,3,'08-AUG-2019','Credit Card',10000, 180,'02-MAY-2022',6600  );

INSERT INTO REPORTS
    VALUES (004,1001,4,'03-APR-2022','Mortgage',4444, 155,'03-MAY-2022',1777  );

INSERT INTO REPORTS
    VALUES (005,1003,5,'06-JAN-2022','Credit Card',12000, 1113,'06-DEC-2021',2200  );

select * from REPORTS
```

```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

REPORTID	INSTITUTIONID	CLIENTID	REPORTDATE	REPORTTYPE	CURRENTDEBT	LASTPAYMENTAMOUNT	LASTPAYMENTDATE	AMOUNTOVERDUE
1	1001	1	06-JUN-22	Credit Card	3544	172	02-MAY-22	233
2	1002	2	19-AUG-17	Credit Card	20000	180	02-JUN-22	9600
3	1003	3	08-AUG-19	Credit Card	10000	180	02-MAY-22	6600
4	1001	4	03-APR-22	Mortgage	4444	155	03-MAY-22	1777
5	1003	5	06-JAN-22	Credit Card	12000	1113	06-DEC-21	2200

Download CSV
5 rows selected.

Credit Score Table:



```
CREATE TABLE "CREDITSCORE"
   (        "CREDIT" NUMBER NOT NULL ENABLE,
        "CLIENTID" NUMBER NOT NULL ENABLE,
        "FIRSTNAME" VARCHAR2(30 ) NOT NULL ENABLE,
        "LASTNAME" VARCHAR2(30 ) NOT NULL ENABLE,
        "SCOREDATE" DATE NOT NULL,

         CONSTRAINT "FK_CLIENTID" FOREIGN KEY ("CLIENTID")
          REFERENCES "CLIENTID" ("CLIENTID")  ,

         CONSTRAINT "FK_FIRSTNAME_UNIQUE" FOREIGN KEY ("FIRSTNAME")
          REFERENCES "CLIENTID" ("FIRSTNAME") ,
        CONSTRAINT "FK_LASTNAME_UNIQUE" FOREIGN KEY ("LASTNAME")
          REFERENCES "CLIENTID" ("LASTNAME") ,
         CONSTRAINT "SCOREDATE_UNIQUE" UNIQUE ("SCOREDATE"));

    INSERT INTO CREDITSCORE
    VALUES (725,0001,'JAMES','PETERSON','12-JUN-2022' );
INSERT INTO CREDITSCORE
    VALUES (888,0003,'LUNDEEP','KAUR' ,'11-JUN-2022'  );

INSERT INTO CREDITSCORE
    VALUES (676,0002,'SUKHDEEP','SINGH' ,'10-JUN-2022'  );


INSERT INTO CREDITSCORE
    VALUES (332,0004,'MANDEEP','PATEL' ,'09-JUN-2022'  );


INSERT INTO CREDITSCORE
    VALUES (812,0005,'SAMDEEP','GUPTA' ,'08-JUN-2022'  );
select * from CREDITSCORE
```

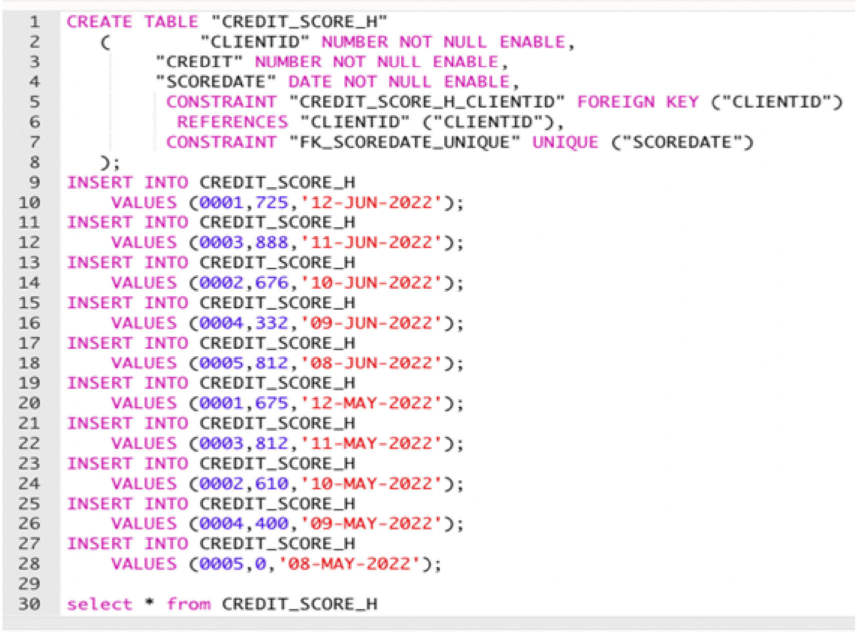
```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

CREDIT	CLIENTID	FIRSTNAME	LASTNAME	SCOREDATE
725	1	JAMES	PETERSON	12-JUN-22
888	3	LUNDEEP	KAUR	11-JUN-22
676	2	SUKHDEEP	SINGH	10-JUN-22
332	4	MANDEEP	PATEL	09-JUN-22
812	5	SAMDEEP	GUPTA	08-JUN-22

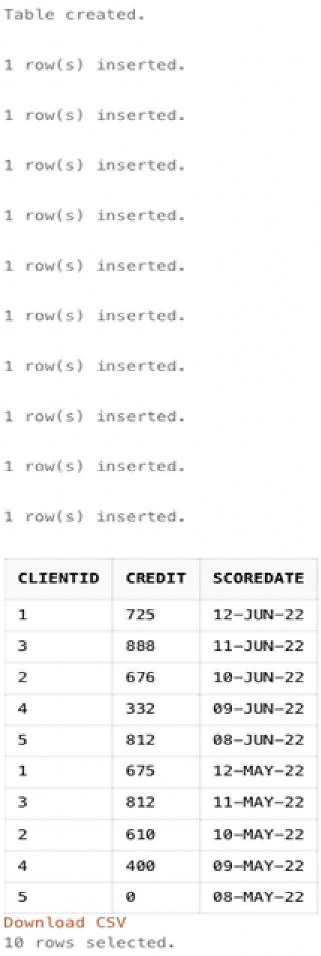
Download CSV

5 rows selected.

Credit Score History Table:



```sql
CREATE TABLE "CREDIT_SCORE_H"
   (        "CLIENTID" NUMBER NOT NULL ENABLE,
        "CREDIT" NUMBER NOT NULL ENABLE,
        "SCOREDATE" DATE NOT NULL ENABLE,
         CONSTRAINT "CREDIT_SCORE_H_CLIENTID" FOREIGN KEY ("CLIENTID")
          REFERENCES "CLIENTID" ("CLIENTID"),
         CONSTRAINT "FK_SCOREDATE_UNIQUE" UNIQUE ("SCOREDATE")
   );
INSERT INTO CREDIT_SCORE_H
    VALUES (0001,725,'12-JUN-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0003,888,'11-JUN-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0002,676,'10-JUN-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0004,332,'09-JUN-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0005,812,'08-JUN-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0001,675,'12-MAY-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0003,812,'11-MAY-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0002,610,'10-MAY-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0004,400,'09-MAY-2022');
INSERT INTO CREDIT_SCORE_H
    VALUES (0005,0,'08-MAY-2022');

select * from CREDIT_SCORE_H
```



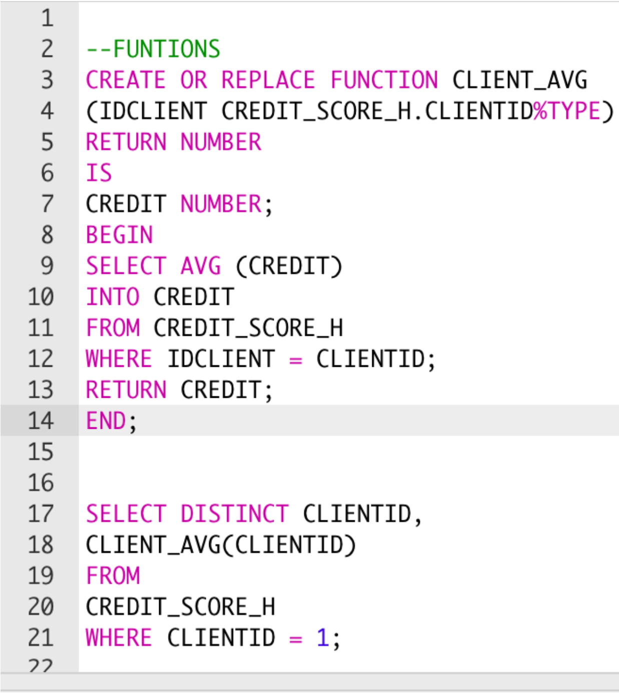
```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

CLIENTID	CREDIT	SCOREDATE
1	725	12-JUN-22
3	888	11-JUN-22
2	676	10-JUN-22
4	332	09-JUN-22
5	812	08-JUN-22
1	675	12-MAY-22
3	812	11-MAY-22
2	610	10-MAY-22
4	400	09-MAY-22
5	0	08-MAY-22

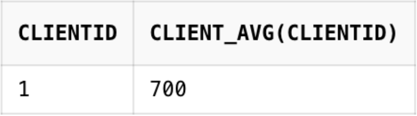
Download CSV

10 rows selected.

Average Credit Score Function:

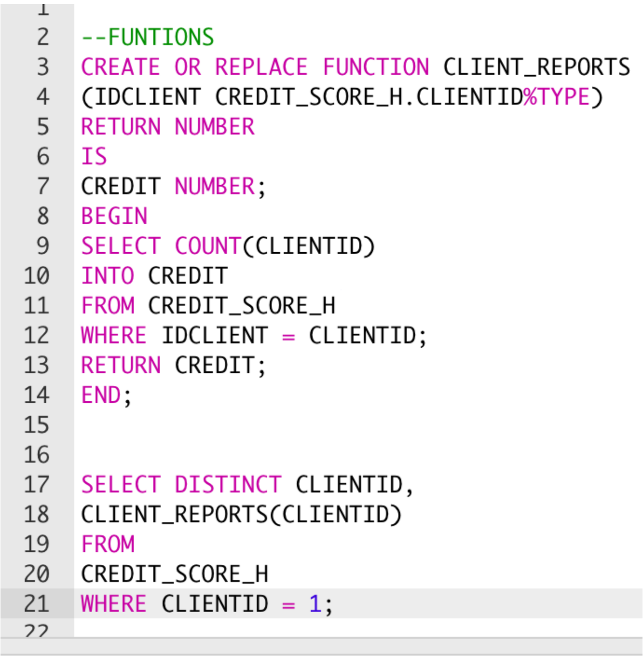


```

--FUNTIONS
CREATE OR REPLACE FUNCTION CLIENT_AVG
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
RETURN NUMBER
IS
CREDIT NUMBER;
BEGIN
SELECT AVG (CREDIT)
INTO CREDIT
FROM CREDIT_SCORE_H
WHERE IDCLIENT = CLIENTID;
RETURN CREDIT;
END;


SELECT DISTINCT CLIENTID,
CLIENT_AVG(CLIENTID)
FROM
CREDIT_SCORE_H
WHERE CLIENTID = 1;
```

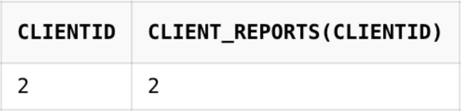


CLIENTID	CLIENT_AVG(CLIENTID)
1	700

Amount of Credit Reports Function:

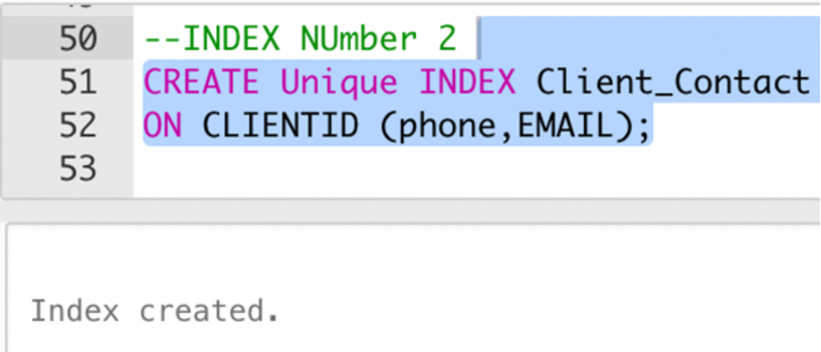


```

--FUNTIONS
CREATE OR REPLACE FUNCTION CLIENT_REPORTS
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
RETURN NUMBER
IS
CREDIT NUMBER;
BEGIN
SELECT COUNT(CLIENTID)
INTO CREDIT
FROM CREDIT_SCORE_H
WHERE IDCLIENT = CLIENTID;
RETURN CREDIT;
END;


SELECT DISTINCT CLIENTID,
CLIENT_REPORTS(CLIENTID)
FROM
CREDIT_SCORE_H
WHERE CLIENTID = 1;
```



CLIENTID	CLIENT_REPORTS(CLIENTID)
2	2

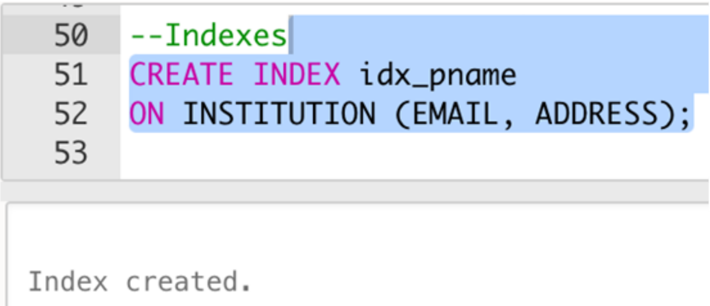
Unique Client Info Index:



```
--INDEX NUmber 2
CREATE Unique INDEX Client_Contact
ON CLIENTID (phone,EMAIL);
```

```
Index created.
```

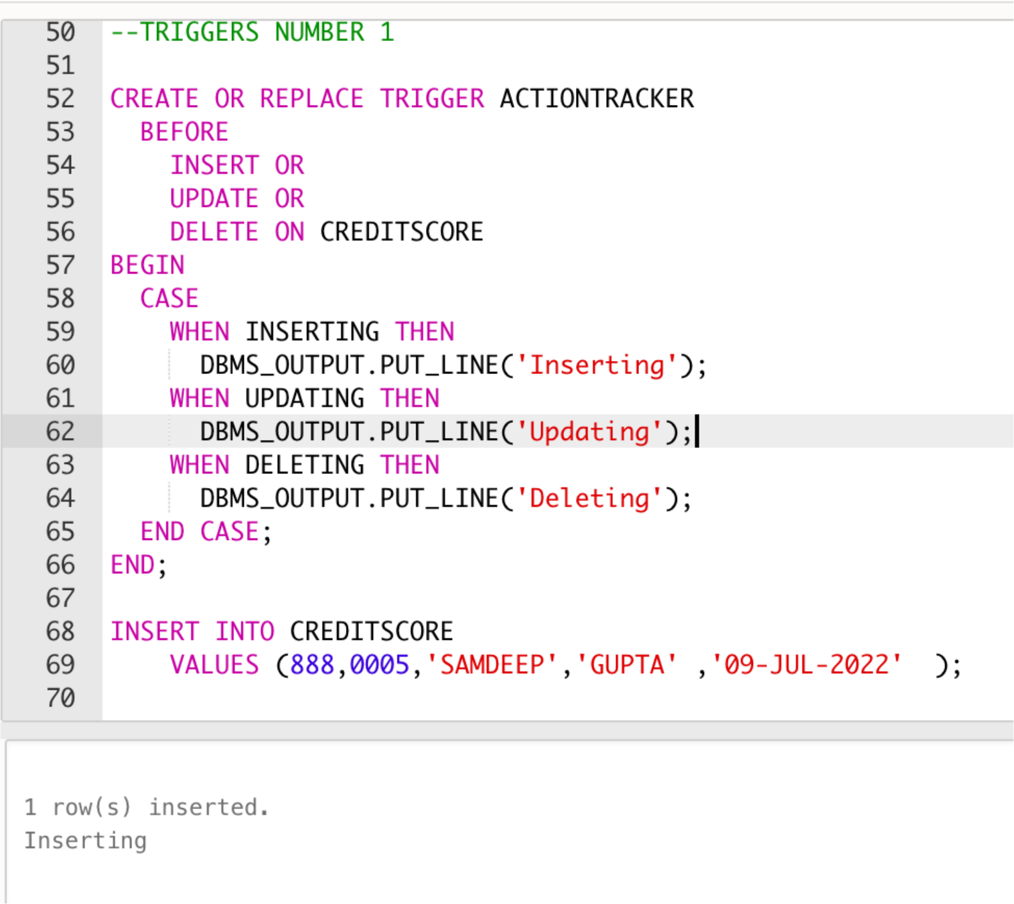
Institution Information Index:



```
--Indexes
CREATE INDEX idx_pname
ON INSTITUTION (EMAIL, ADDRESS);
```

```
Index created.
```

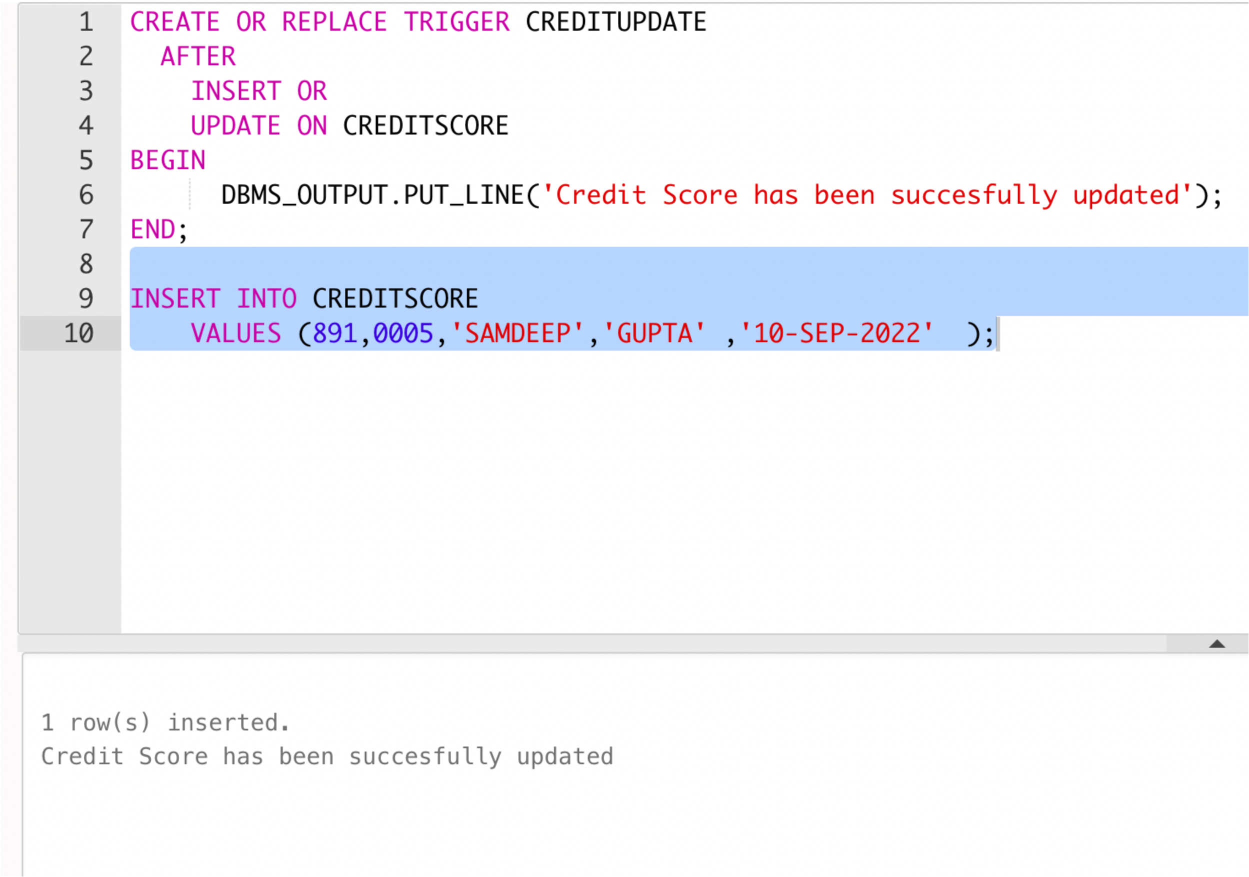
Action Tracker Trigger:



```sql
--TRIGGERS NUMBER 1

CREATE OR REPLACE TRIGGER ACTIONTRACKER
  BEFORE
    INSERT OR
    UPDATE OR
    DELETE ON CREDITSCORE
BEGIN
  CASE
    WHEN INSERTING THEN
      DBMS_OUTPUT.PUT_LINE('Inserting');
    WHEN UPDATING THEN
      DBMS_OUTPUT.PUT_LINE('Updating');
    WHEN DELETING THEN
      DBMS_OUTPUT.PUT_LINE('Deleting');
  END CASE;
END;

INSERT INTO CREDITSCORE
    VALUES (888,0005,'SAMDEEP','GUPTA' ,'09-JUL-2022'  );

```

```
1 row(s) inserted.
Inserting
```

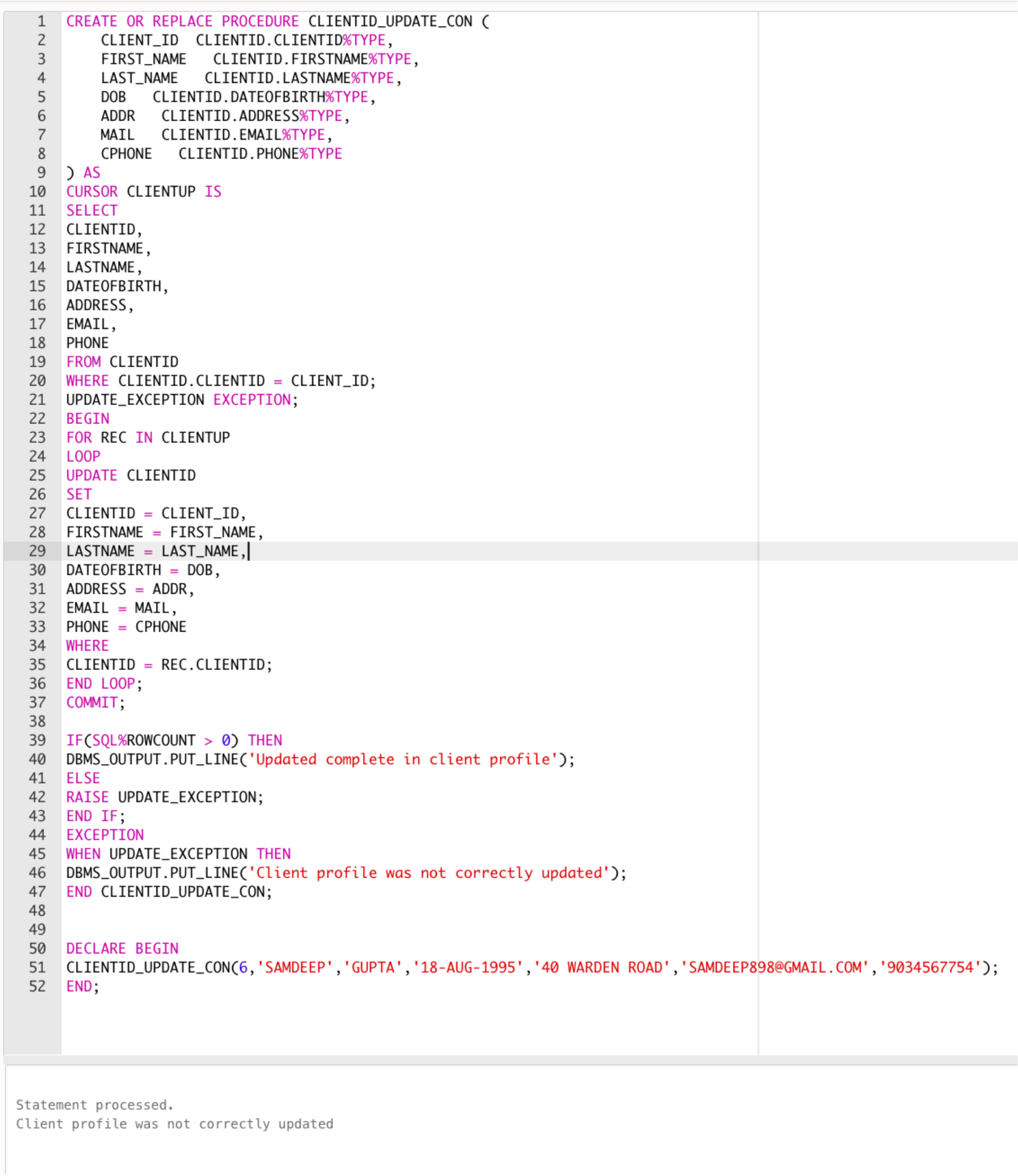
Credit Update Confirmation Trigger:



```
CREATE OR REPLACE TRIGGER CREDITUPDATE
  AFTER
    INSERT OR
    UPDATE ON CREDITSCORE
BEGIN
      DBMS_OUTPUT.PUT_LINE('Credit Score has been succesfully updated');
END;

INSERT INTO CREDITSCORE
    VALUES (891,0005,'SAMDEEP','GUPTA'  ,'10-SEP-2022'   );
```

```
1 row(s) inserted.
Credit Score has been succesfully updated
```

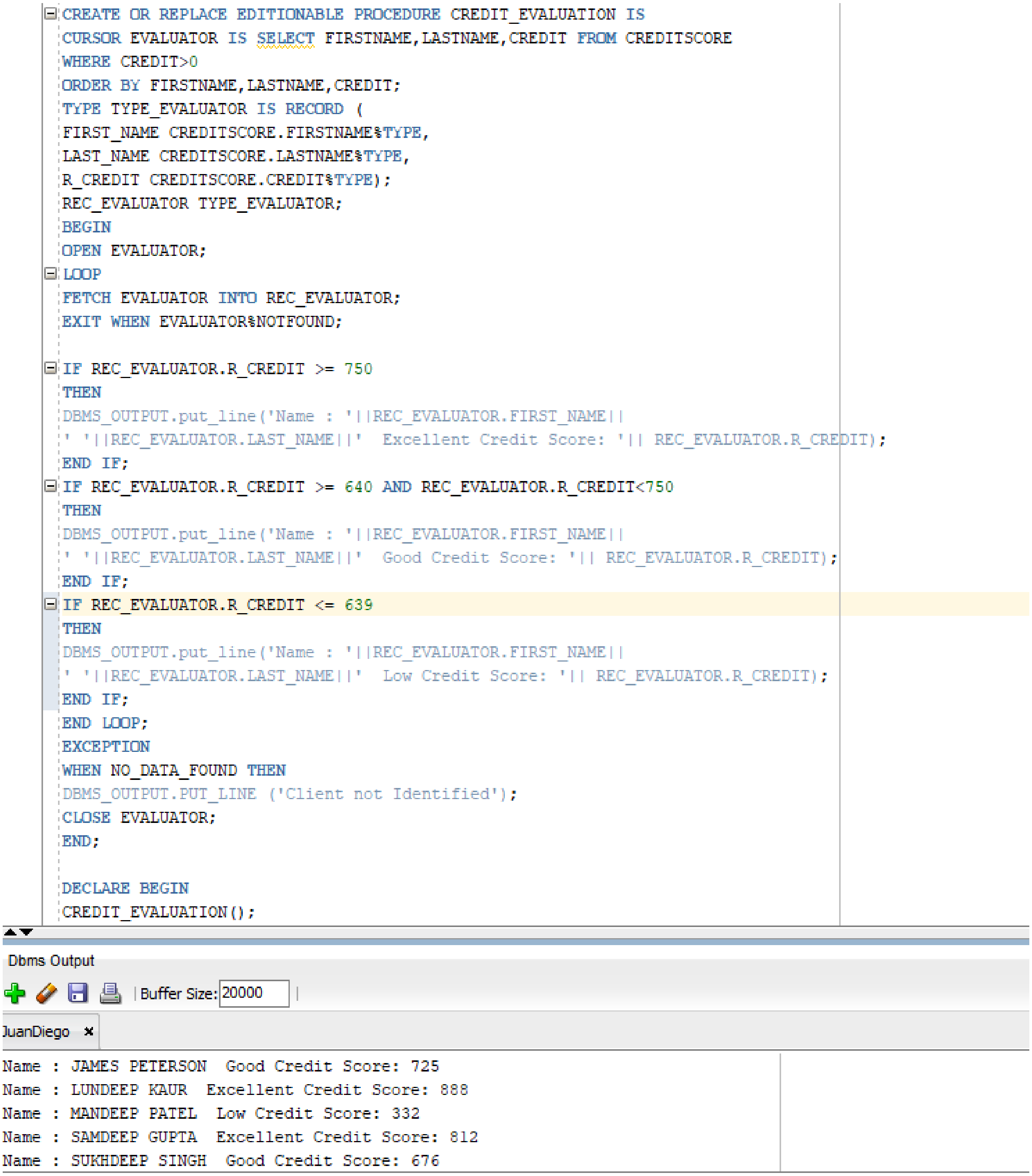
Procedures to Confirm Correct Update:



```
CREATE OR REPLACE PROCEDURE CLIENTID_UPDATE_CON (
    CLIENT_ID  CLIENTID.CLIENTID%TYPE,
    FIRST_NAME   CLIENTID.FIRSTNAME%TYPE,
    LAST_NAME   CLIENTID.LASTNAME%TYPE,
    DOB   CLIENTID.DATEOFBIRTH%TYPE,
    ADDR   CLIENTID.ADDRESS%TYPE,
    MAIL   CLIENTID.EMAIL%TYPE,
    CPHONE   CLIENTID.PHONE%TYPE
) AS
CURSOR CLIENTUP IS
SELECT
CLIENTID,
FIRSTNAME,
LASTNAME,
DATEOFBIRTH,
ADDRESS,
EMAIL,
PHONE
FROM CLIENTID
WHERE CLIENTID.CLIENTID = CLIENT_ID;
UPDATE_EXCEPTION EXCEPTION;
BEGIN
FOR REC IN CLIENTUP
LOOP
UPDATE CLIENTID
SET
CLIENTID = CLIENT_ID,
FIRSTNAME = FIRST_NAME,
LASTNAME = LAST_NAME,
DATEOFBIRTH = DOB,
ADDRESS = ADDR,
EMAIL = MAIL,
PHONE = CPHONE
WHERE
CLIENTID = REC.CLIENTID;
END LOOP;
COMMIT;

IF(SQL%ROWCOUNT > 0) THEN
DBMS_OUTPUT.PUT_LINE('Updated complete in client profile');
ELSE
RAISE UPDATE_EXCEPTION;
END IF;
EXCEPTION
WHEN UPDATE_EXCEPTION THEN
DBMS_OUTPUT.PUT_LINE('Client profile was not correctly updated');
END CLIENTID_UPDATE_CON;


DECLARE BEGIN
CLIENTID_UPDATE_CON(6,'SAMDEEP','GUPTA','18-AUG-1995','40 WARDEN ROAD','SAMDEEP898@GMAIL.COM','9034567754');
END;
```

```
Statement processed.
Client profile was not correctly updated
```

Credit Evaluation Procedure:

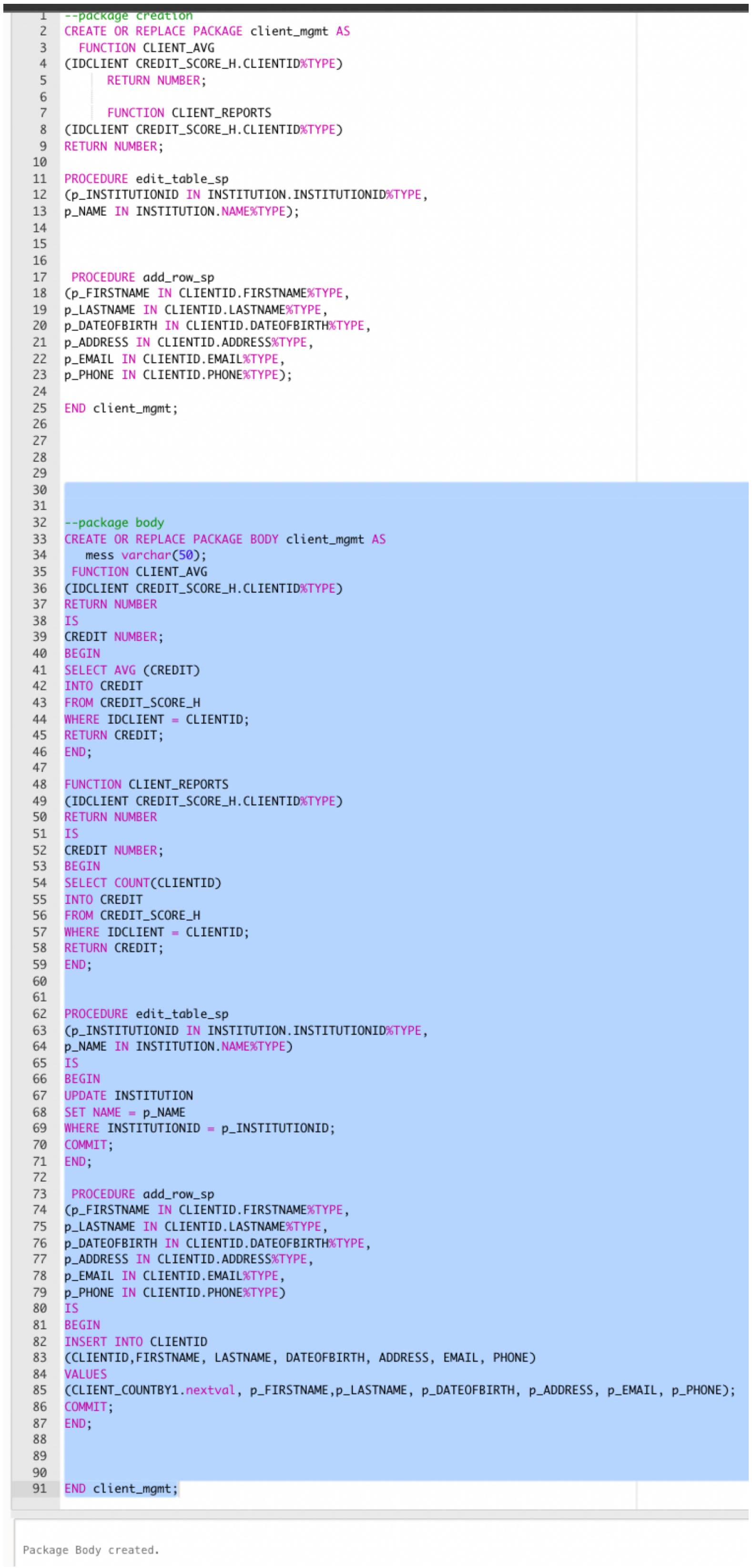


```
CREATE OR REPLACE EDITIONABLE PROCEDURE CREDIT_EVALUATION IS
CURSOR EVALUATOR IS SELECT FIRSTNAME,LASTNAME,CREDIT FROM CREDITSCORE
WHERE CREDIT>0
ORDER BY FIRSTNAME,LASTNAME,CREDIT;
TYPE TYPE_EVALUATOR IS RECORD (
FIRST_NAME CREDITSCORE.FIRSTNAME%TYPE,
LAST_NAME CREDITSCORE.LASTNAME%TYPE,
R_CREDIT CREDITSCORE.CREDIT%TYPE);
REC_EVALUATOR TYPE_EVALUATOR;
BEGIN
OPEN EVALUATOR;
LOOP
FETCH EVALUATOR INTO REC_EVALUATOR;
EXIT WHEN EVALUATOR%NOTFOUND;

IF REC_EVALUATOR.R_CREDIT >= 750
THEN
DBMS_OUTPUT.put_line('Name : '||REC_EVALUATOR.FIRST_NAME||
' '||REC_EVALUATOR.LAST_NAME||'  Excellent Credit Score: '|| REC_EVALUATOR.R_CREDIT);
END IF;
IF REC_EVALUATOR.R_CREDIT >= 640 AND REC_EVALUATOR.R_CREDIT<750
THEN
DBMS_OUTPUT.put_line('Name : '||REC_EVALUATOR.FIRST_NAME||
' '||REC_EVALUATOR.LAST_NAME||'  Good Credit Score: '|| REC_EVALUATOR.R_CREDIT);
END IF;
IF REC_EVALUATOR.R_CREDIT <= 639
THEN
DBMS_OUTPUT.put_line('Name : '||REC_EVALUATOR.FIRST_NAME||
' '||REC_EVALUATOR.LAST_NAME||'  Low Credit Score: '|| REC_EVALUATOR.R_CREDIT);
END IF;
END LOOP;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE ('Client not Identified');
CLOSE EVALUATOR;
END;

DECLARE BEGIN
CREDIT_EVALUATION();
```

Buffer Size: 20000

```
Name : JAMES PETERSON  Good Credit Score: 725
Name : LUNDEEP KAUR  Excellent Credit Score: 888
Name : MANDEEP PATEL  Low Credit Score: 332
Name : SAMDEEP GUPTA  Excellent Credit Score: 812
Name : SUKHDEEP SINGH  Good Credit Score: 676
```

Packages:



```
--package creation
CREATE OR REPLACE PACKAGE client_mgmt AS
  FUNCTION CLIENT_AVG
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
      RETURN NUMBER;

      FUNCTION CLIENT_REPORTS
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
RETURN NUMBER;

PROCEDURE edit_table_sp
(p_INSTITUTIONID IN INSTITUTION.INSTITUTIONID%TYPE,
p_NAME IN INSTITUTION.NAME%TYPE);



 PROCEDURE add_row_sp
(p_FIRSTNAME IN CLIENTID.FIRSTNAME%TYPE,
p_LASTNAME IN CLIENTID.LASTNAME%TYPE,
p_DATEOFBIRTH IN CLIENTID.DATEOFBIRTH%TYPE,
p_ADDRESS IN CLIENTID.ADDRESS%TYPE,
p_EMAIL IN CLIENTID.EMAIL%TYPE,
p_PHONE IN CLIENTID.PHONE%TYPE);

END client_mgmt;






--package body
CREATE OR REPLACE PACKAGE BODY client_mgmt AS
   mess varchar(50);
 FUNCTION CLIENT_AVG
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
RETURN NUMBER
IS
CREDIT NUMBER;
BEGIN
SELECT AVG (CREDIT)
INTO CREDIT
FROM CREDIT_SCORE_H
WHERE IDCLIENT = CLIENTID;
RETURN CREDIT;
END;

FUNCTION CLIENT_REPORTS
(IDCLIENT CREDIT_SCORE_H.CLIENTID%TYPE)
RETURN NUMBER
IS
CREDIT NUMBER;
BEGIN
SELECT COUNT(CLIENTID)
INTO CREDIT
FROM CREDIT_SCORE_H
WHERE IDCLIENT = CLIENTID;
RETURN CREDIT;
END;


PROCEDURE edit_table_sp
(p_INSTITUTIONID IN INSTITUTION.INSTITUTIONID%TYPE,
p_NAME IN INSTITUTION.NAME%TYPE)
IS
BEGIN
UPDATE INSTITUTION
SET NAME = p_NAME
WHERE INSTITUTIONID = p_INSTITUTIONID;
COMMIT;
END;

 PROCEDURE add_row_sp
(p_FIRSTNAME IN CLIENTID.FIRSTNAME%TYPE,
p_LASTNAME IN CLIENTID.LASTNAME%TYPE,
p_DATEOFBIRTH IN CLIENTID.DATEOFBIRTH%TYPE,
p_ADDRESS IN CLIENTID.ADDRESS%TYPE,
p_EMAIL IN CLIENTID.EMAIL%TYPE,
p_PHONE IN CLIENTID.PHONE%TYPE)
IS
BEGIN
INSERT INTO CLIENTID
(CLIENTID,FIRSTNAME, LASTNAME, DATEOFBIRTH, ADDRESS, EMAIL, PHONE)
VALUES
(CLIENT_COUNTBY1.nextval, p_FIRSTNAME,p_LASTNAME, p_DATEOFBIRTH, p_ADDRESS, p_EMAIL, p_PHONE);
COMMIT;
END;



END client_mgmt;
```

```
Package Body created.
```