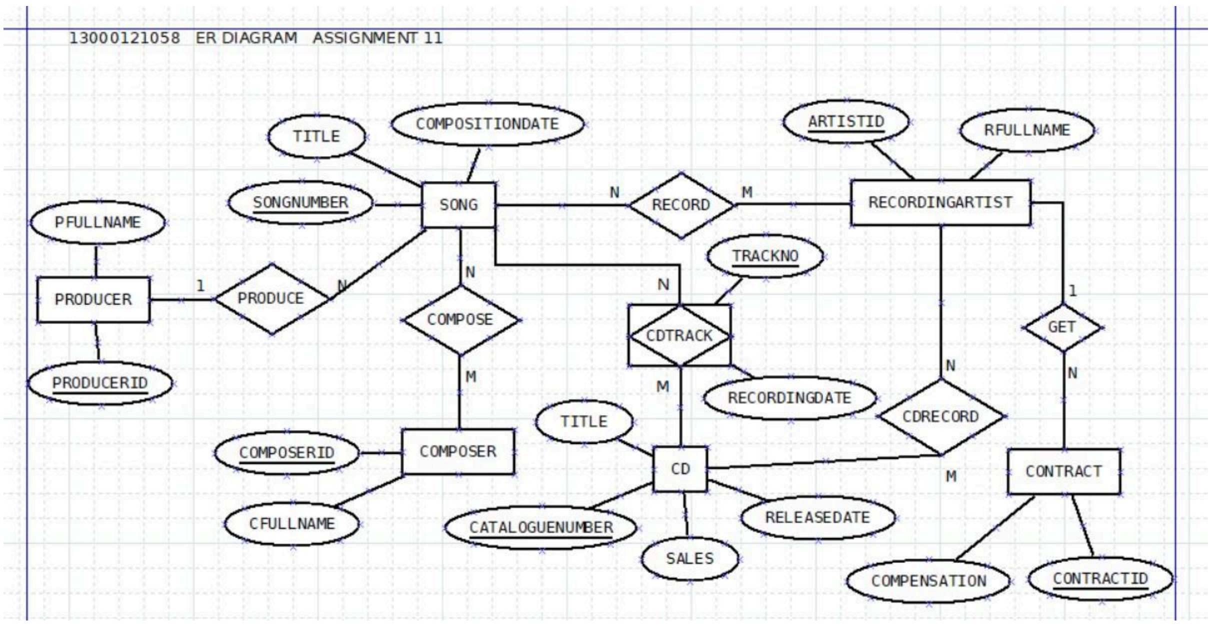
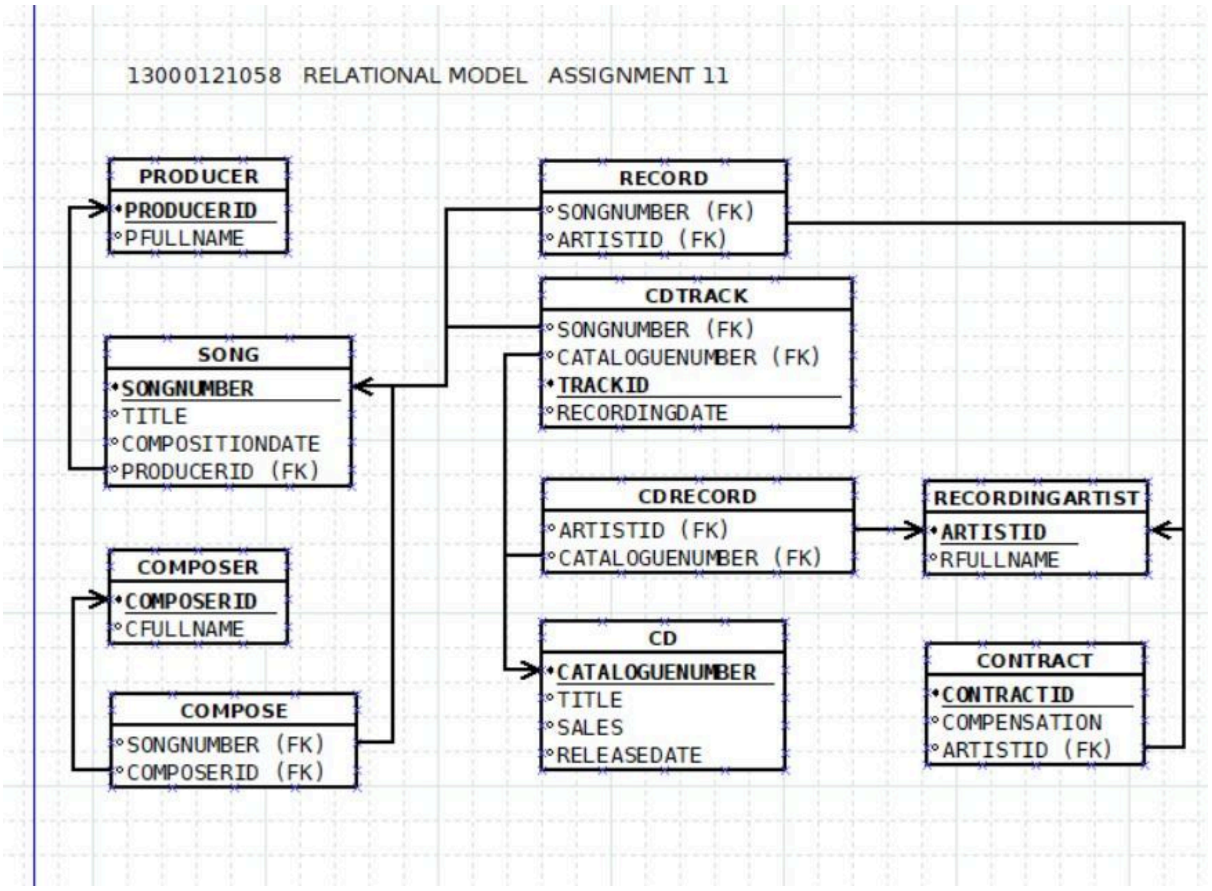
ASSIGNMENT 11

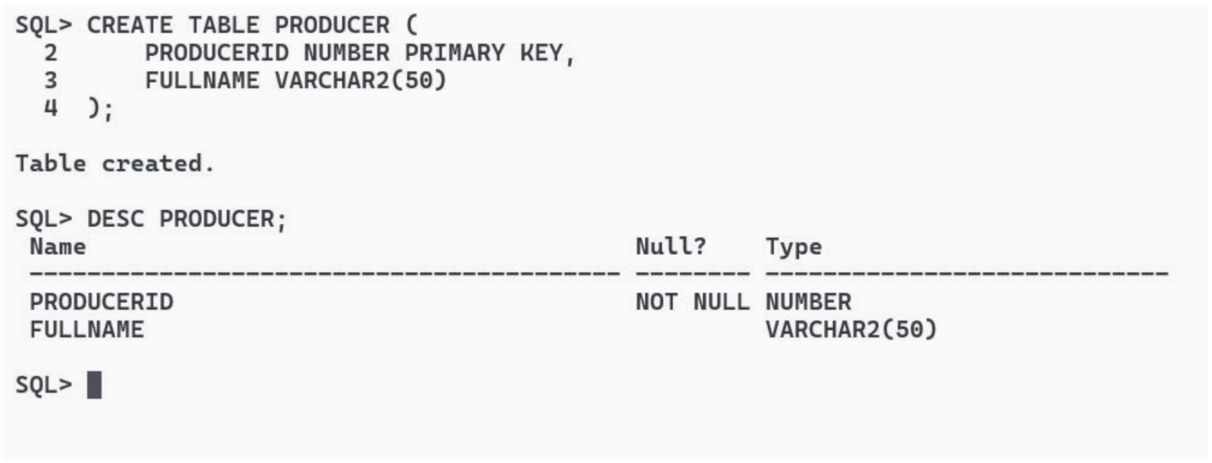
A record company wishes to use a computer database to help with its operations regarding its performers, recordings and song catalogue. A requirements analysis has elicited the following information: Songs have a unique song number, a non-unique title and a composition date. A song can be written by a number of composers; the composer's full name is required. Songs are recorded by recording artists (bands or solo performers). A song is recorded as a track of a CD. A CD has many songs on it, called tracks. CDs have a unique record catalogue number, a title and must have a producer(the full name of the producer is required). Each track must have the recording date and the track number of the CD. A song can appear on many (or no) CDs, and be recorded by many different recording artists. The same recording artist might re-record the same song on different CDs. A CD must have only 1 recording artist appearing on it. CDs can be released a number of times, and each time the release date and associated number of sales is required.

1.Use this information to design an appropriate ER and relational model.
2.Compile DDL and DML commands on the database created.
SQL:-
i>Update number of recorded albums to 4 for those artists who have recorded only 3.
ii>Find all artists who have recorded at least two albums.
iii>Find all writers who have only written one song.
PL/SQL
i>Write Procedure to insert a new Contract into the Contract relation.



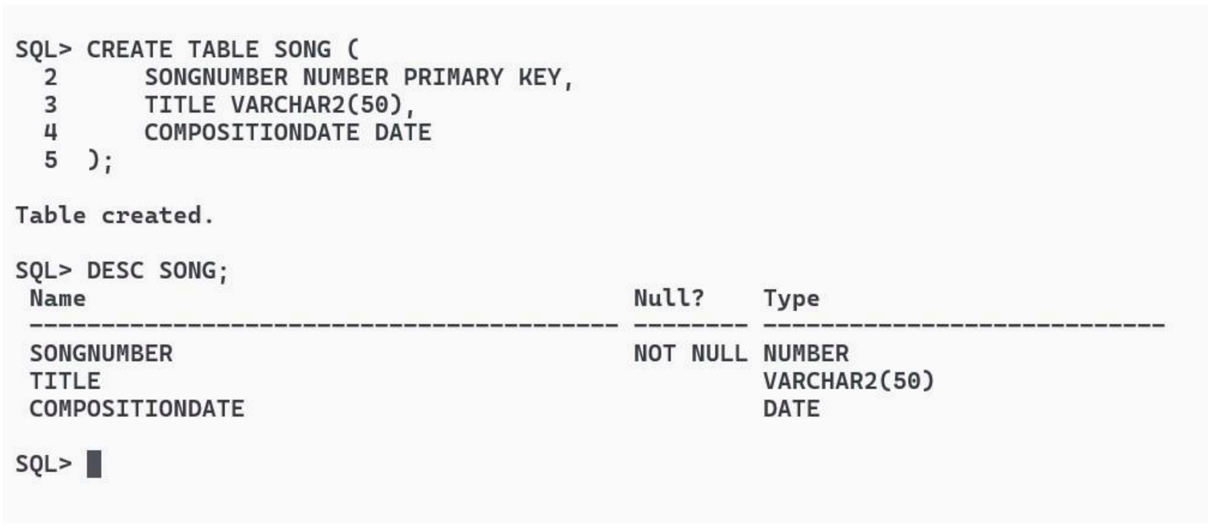


```
CREATE TABLE PRODUCER (
   PRODUCERID NUMBER PRIMARY KEY,
   FULLNAME VARCHAR2(50),
        --SONGNUMBER NUMBER,
        --CONSTRAINT PFK1 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE
);
```



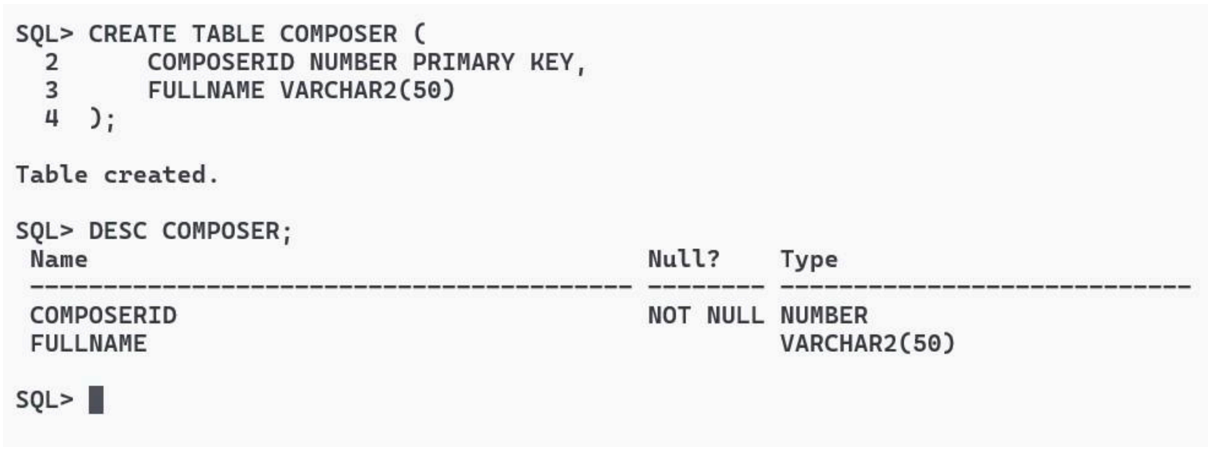
```
SQL> CREATE TABLE PRODUCER (
  2      PRODUCERID NUMBER PRIMARY KEY,
  3      FULLNAME VARCHAR2(50)
  4  );

Table created.

SQL> DESC PRODUCER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCERID                                NOT NULL NUMBER
 FULLNAME                                           VARCHAR2(50)

SQL> █
```

```
CREATE TABLE SONG (
   SONGNUMBER NUMBER PRIMARY KEY,
   TITLE VARCHAR2(50),
   COMPOSITIONDATE DATE,
       PRODUCERID NUMBER,
       CONSTRAINT SFK1 FOREIGN KEY (PRODUCERID) REFERENCES PRODUCER(PRODUCERID) ON DELETE CASCADE
);
```



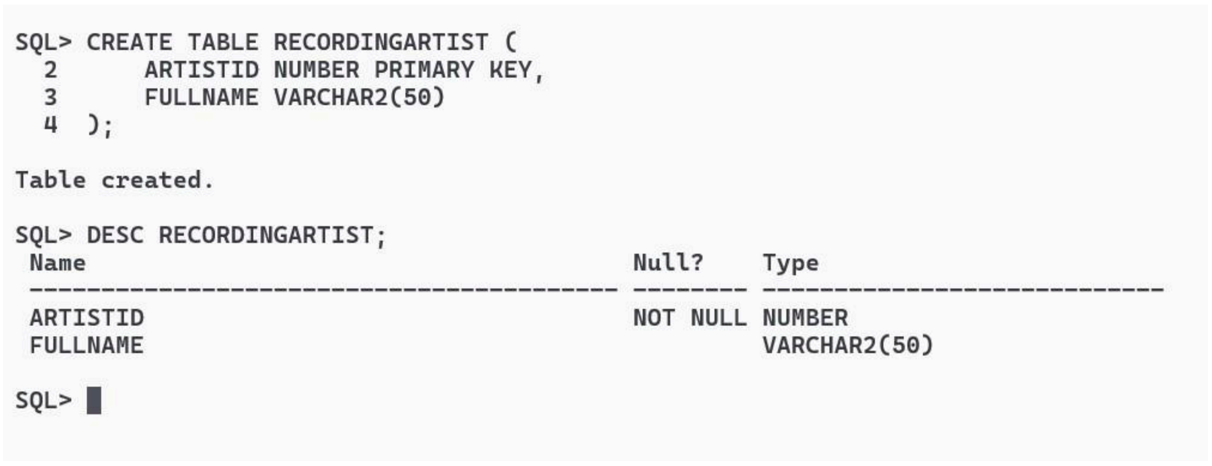
```
SQL> CREATE TABLE SONG (
  2      SONGNUMBER NUMBER PRIMARY KEY,
  3      TITLE VARCHAR2(50),
  4      COMPOSITIONDATE DATE
  5  );

Table created.

SQL> DESC SONG;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SONGNUMBER                                NOT NULL NUMBER
 TITLE                                              VARCHAR2(50)
 COMPOSITIONDATE                                    DATE

SQL> █
```

```
CREATE TABLE COMPOSER (
    COMPOSERID NUMBER PRIMARY KEY,
    FULLNAME VARCHAR2(50)
);
```



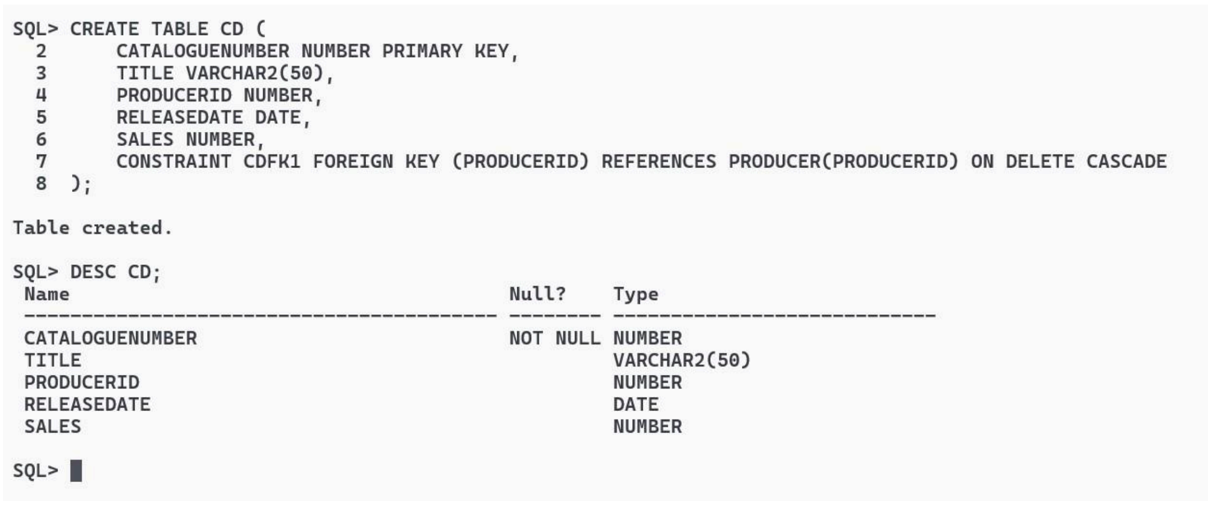
```
SQL> CREATE TABLE COMPOSER (
  2      COMPOSERID NUMBER PRIMARY KEY,
  3      FULLNAME VARCHAR2(50)
  4  );

Table created.

SQL> DESC COMPOSER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COMPOSERID                                NOT NULL NUMBER
 FULLNAME                                           VARCHAR2(50)

SQL> █
```

```
CREATE TABLE RECORDINGARTIST (
    ARTISTID NUMBER PRIMARY KEY,
    FULLNAME VARCHAR2(50)
);
```



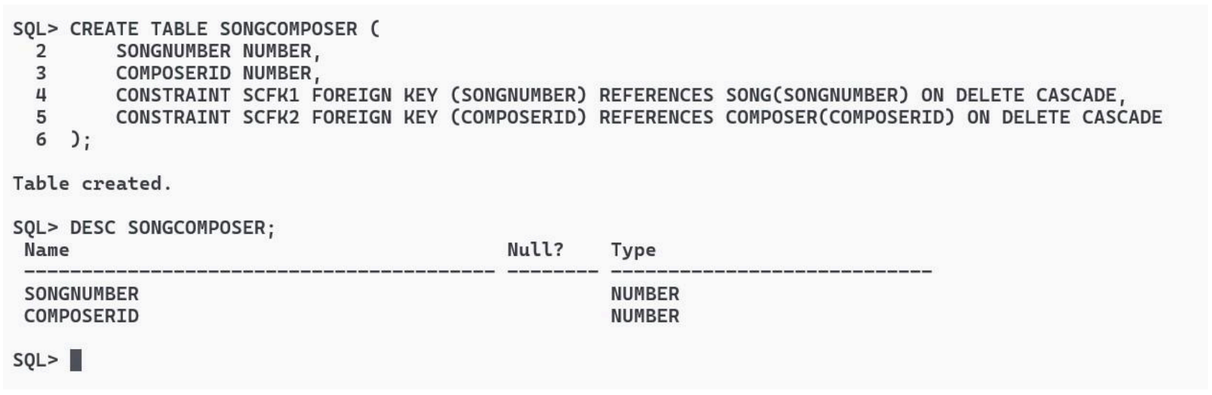
```
SQL> CREATE TABLE RECORDINGARTIST (
  2      ARTISTID NUMBER PRIMARY KEY,
  3      FULLNAME VARCHAR2(50)
  4  );

Table created.

SQL> DESC RECORDINGARTIST;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ARTISTID                                  NOT NULL NUMBER
 FULLNAME                                           VARCHAR2(50)

SQL> █
```

```
CREATE TABLE CD (
   CATALOGUENUMBER NUMBER PRIMARY KEY,
   TITLE VARCHAR2(50),
   PRODUCERID NUMBER,
   RELEASEDATE DATE,
   SALES NUMBER,
        CONSTRAINT CDFK1 FOREIGN KEY (PRODUCERID) REFERENCES
PRODUCER(PRODUCERID) ON DELETE CASCADE
);
```



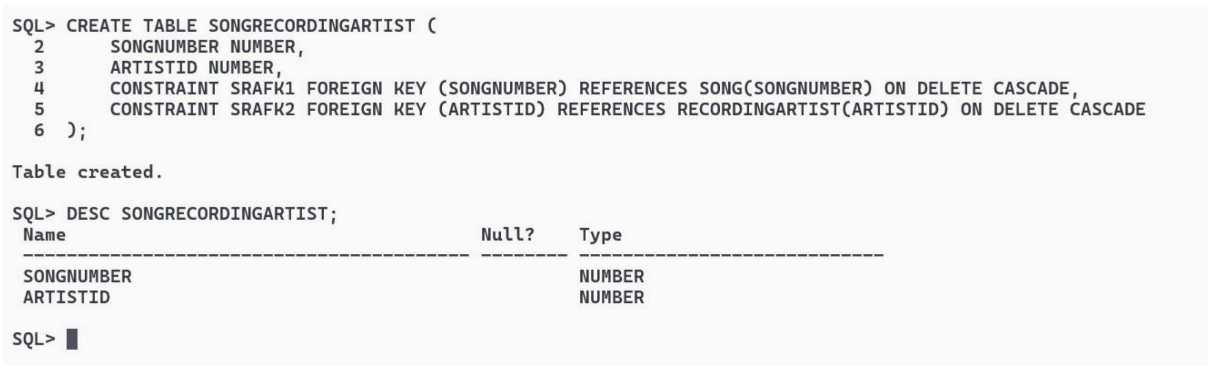
```
SQL> CREATE TABLE CD (
  2      CATALOGUENUMBER NUMBER PRIMARY KEY,
  3      TITLE VARCHAR2(50),
  4      PRODUCERID NUMBER,
  5      RELEASEDATE DATE,
  6      SALES NUMBER,
  7      CONSTRAINT CDFK1 FOREIGN KEY (PRODUCERID) REFERENCES PRODUCER(PRODUCERID) ON DELETE CASCADE
  8  );

Table created.

SQL> DESC CD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATALOGUENUMBER                           NOT NULL NUMBER
 TITLE                                              VARCHAR2(50)
 PRODUCERID                                         NUMBER
 RELEASEDATE                                        DATE
 SALES                                              NUMBER

SQL> █
```

```
CREATE TABLE SONGCOMPOSER (
   SONGNUMBER NUMBER,
   COMPOSERID NUMBER,
        CONSTRAINT SCFK1 FOREIGN KEY (SONGNUMBER) REFERENCES
SONG(SONGNUMBER) ON DELETE CASCADE,
        CONSTRAINT SCFK2 FOREIGN KEY (COMPOSERID) REFERENCES
COMPOSER(COMPOSERID) ON DELETE CASCADE
);
```



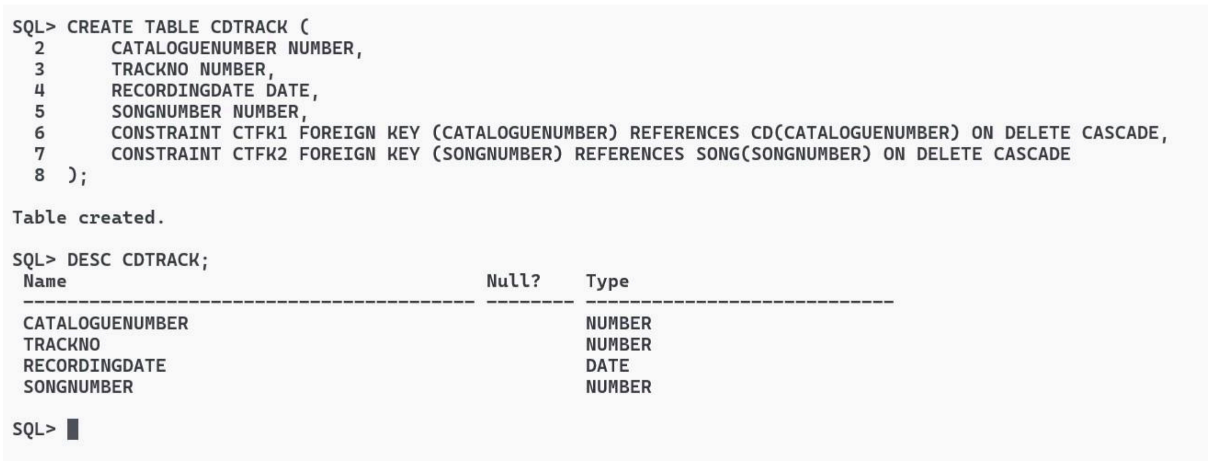
```
SQL> CREATE TABLE SONGCOMPOSER (
  2      SONGNUMBER NUMBER,
  3      COMPOSERID NUMBER,
  4      CONSTRAINT SCFK1 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE,
  5      CONSTRAINT SCFK2 FOREIGN KEY (COMPOSERID) REFERENCES COMPOSER(COMPOSERID) ON DELETE CASCADE
  6  );

Table created.

SQL> DESC SONGCOMPOSER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SONGNUMBER                                         NUMBER
 COMPOSERID                                         NUMBER

SQL>
```

CREATE TABLE SONGRECORDINGARTIST (
SONGNUMBER NUMBER,
ARTISTID NUMBER,
CONSTRAINT SRAFK1 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE,
CONSTRAINT SRAFK2 FOREIGN KEY (ARTISTID) REFERENCES RECORDINGARTIST(ARTISTID) ON DELETE CASCADE
);



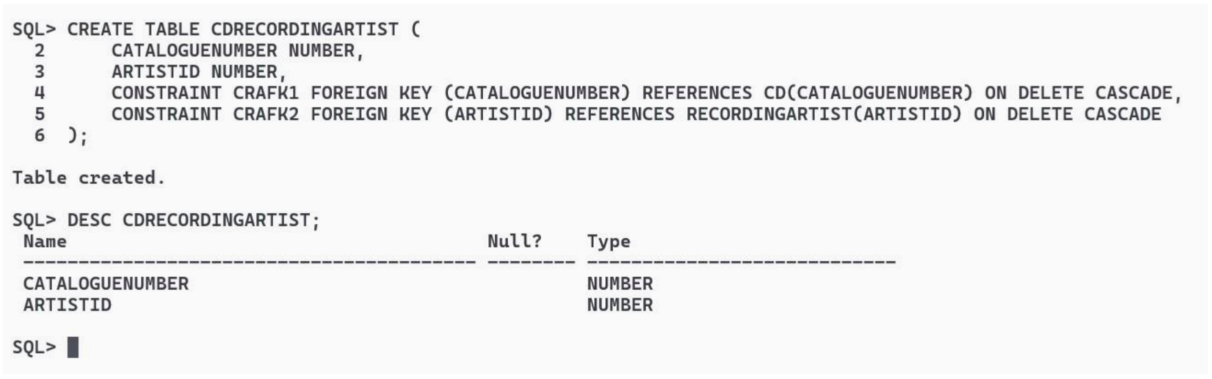
```
SQL> CREATE TABLE SONGRECORDINGARTIST (
  2      SONGNUMBER NUMBER,
  3      ARTISTID NUMBER,
  4      CONSTRAINT SRAFK1 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE,
  5      CONSTRAINT SRAFK2 FOREIGN KEY (ARTISTID) REFERENCES RECORDINGARTIST(ARTISTID) ON DELETE CASCADE
  6  );

Table created.

SQL> DESC SONGRECORDINGARTIST;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SONGNUMBER                                         NUMBER
 ARTISTID                                           NUMBER

SQL>
```

CREATE TABLE CDTRACK (
CATALOGUENUMBER NUMBER,
TRACKNO NUMBER,
RECORDINGDATE DATE,
SONGNUMBER NUMBER,
CONSTRAINT CTFK1 FOREIGN KEY (CATALOGUENUMBER) REFERENCES CD(CATALOGUENUMBER) ON DELETE CASCADE,
CONSTRAINT CTFK2 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE
);



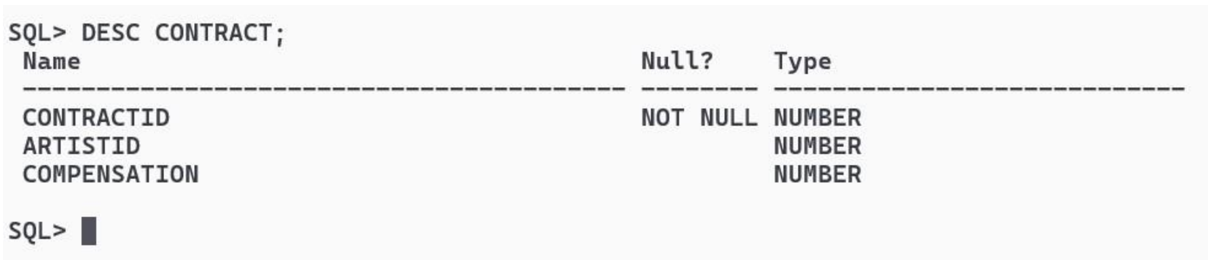
```
SQL> CREATE TABLE CDTRACK (
         CATALOGUENUMBER NUMBER,
         TRACKNO NUMBER,
         RECORDINGDATE DATE,
         SONGNUMBER NUMBER,
         CONSTRAINT CTFK1 FOREIGN KEY (CATALOGUENUMBER) REFERENCES CD(CATALOGUENUMBER) ON DELETE CASCADE,
         CONSTRAINT CTFK2 FOREIGN KEY (SONGNUMBER) REFERENCES SONG(SONGNUMBER) ON DELETE CASCADE
     );

Table created.

SQL> DESC CDTRACK;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATALOGUENUMBER                                    NUMBER
 TRACKNO                                            NUMBER
 RECORDINGDATE                                      DATE
 SONGNUMBER                                         NUMBER

SQL>
```

```
CREATE TABLE CDRECORDINGARTIST (
    CATALOGUENUMBER NUMBER,
    ARTISTID NUMBER,
    CONSTRAINT CRAFK1 FOREIGN KEY (CATALOGUENUMBER) REFERENCES CD(CATALOGUENUMBER) ON DELETE CASCADE,
    CONSTRAINT CRAFK2 FOREIGN KEY (ARTISTID) REFERENCES RECORDINGARTIST(ARTISTID) ON DELETE CASCADE
);
```



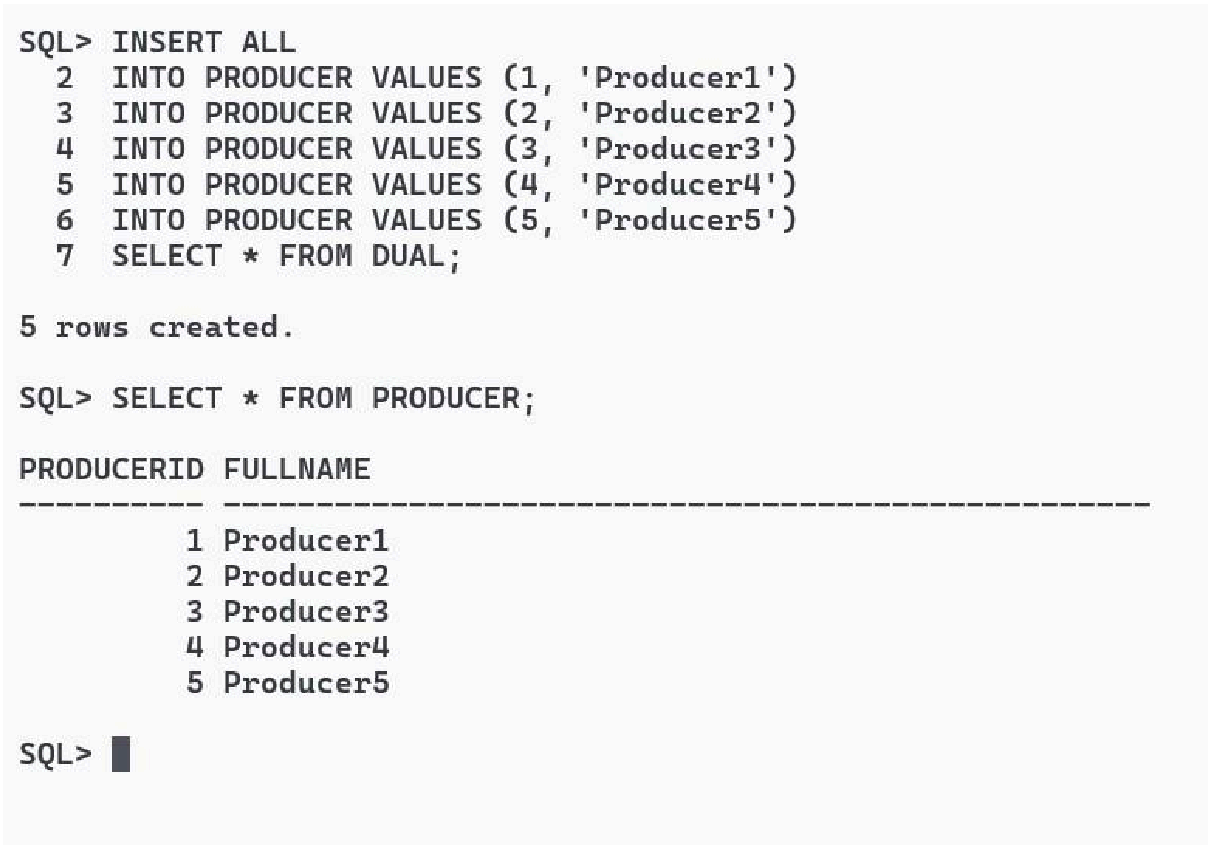
```
SQL> CREATE TABLE CDRECORDINGARTIST (
         CATALOGUENUMBER NUMBER,
         ARTISTID NUMBER,
         CONSTRAINT CRAFK1 FOREIGN KEY (CATALOGUENUMBER) REFERENCES CD(CATALOGUENUMBER) ON DELETE CASCADE,
         CONSTRAINT CRAFK2 FOREIGN KEY (ARTISTID) REFERENCES RECORDINGARTIST(ARTISTID) ON DELETE CASCADE
     );

Table created.

SQL> DESC CDRECORDINGARTIST;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CATALOGUENUMBER                                    NUMBER
 ARTISTID                                           NUMBER

SQL>
```

```
CREATE TABLE CONTRACT (
        CONTRACTID NUMBER PRIMARY KEY,
        ARTISTID NUMBER,
        COMPENSATION NUMBER,
        CONSTRAINT CNFK1 FOREIGN KEY (ARTISTID) REFERENCES RECORDINGARTIST(ARTISTID) ON DELETE CASCADE
);
```



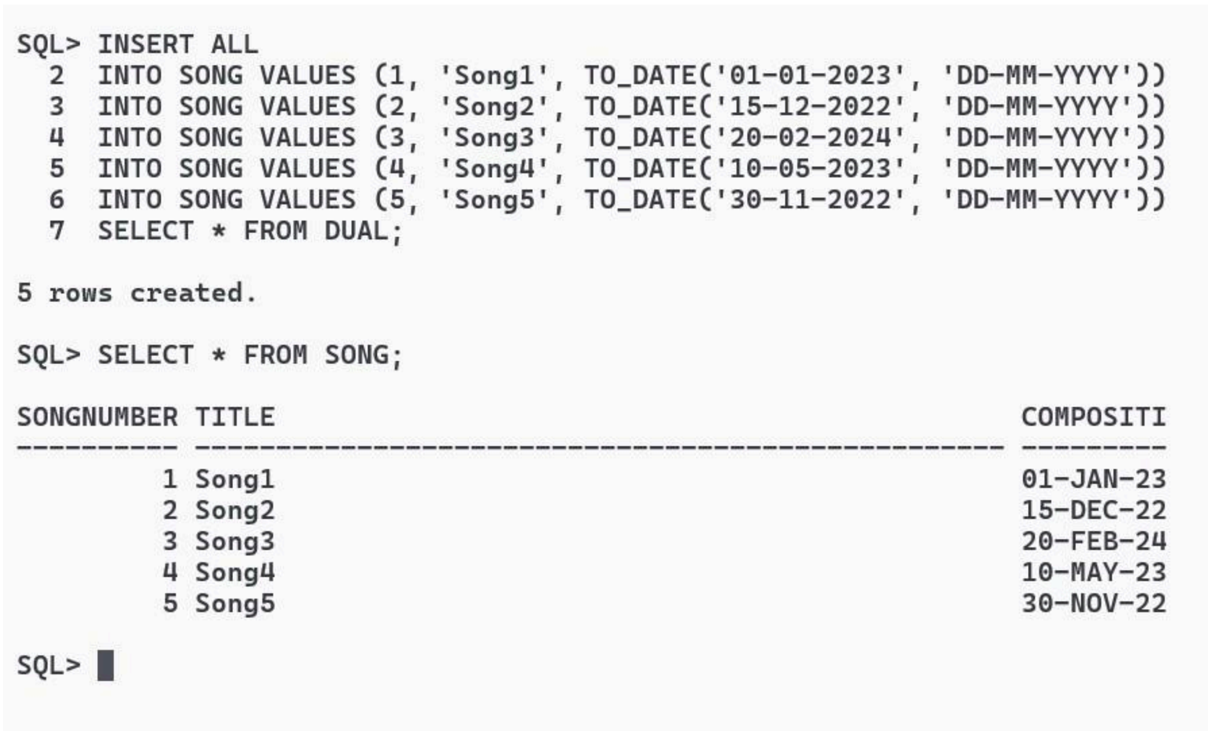
```
SQL> DESC CONTRACT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CONTRACTID                                NOT NULL NUMBER
 ARTISTID                                           NUMBER
 COMPENSATION                                       NUMBER

SQL>
```

```
INSERT ALL
INTO PRODUCER VALUES (1, 'Producer1')
INTO PRODUCER VALUES (2, 'Producer2')
INTO PRODUCER VALUES (3, 'Producer3')
INTO PRODUCER VALUES (4, 'Producer4')
INTO PRODUCER VALUES (5, 'Producer5')
SELECT * FROM DUAL;
```



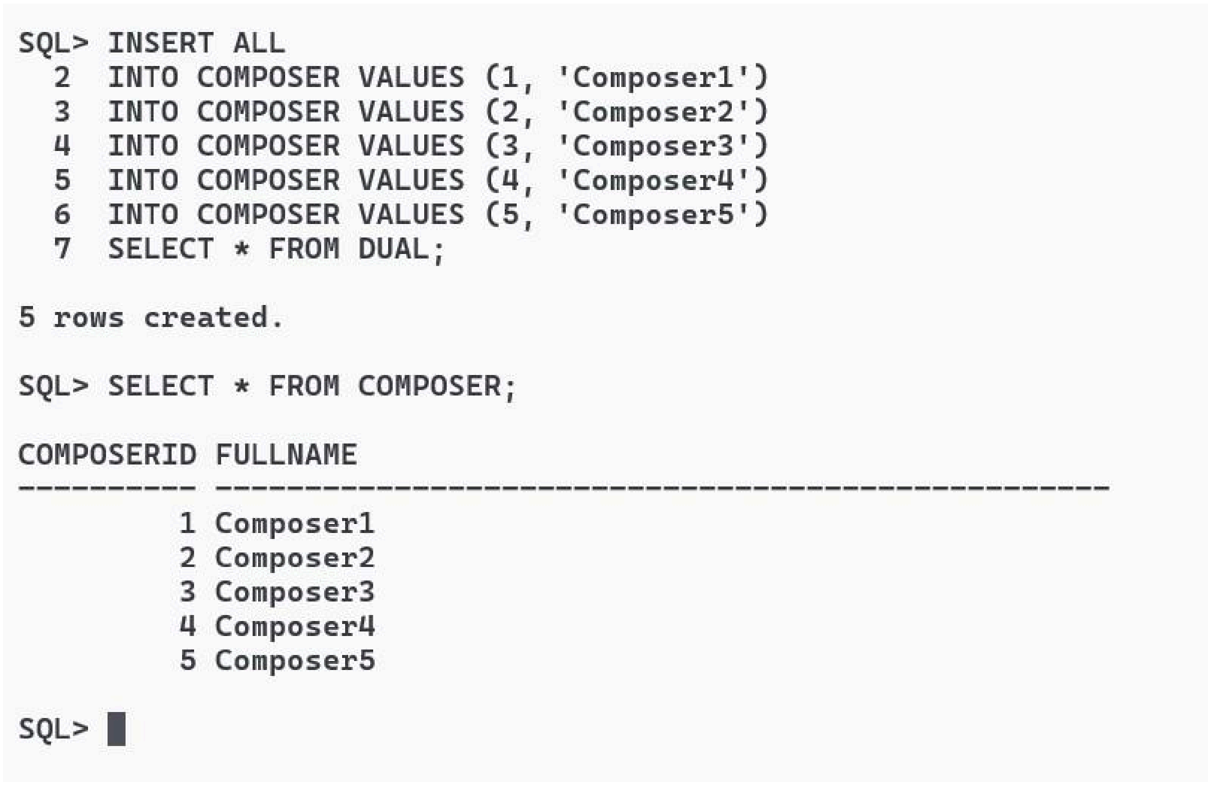
```
SQL> INSERT ALL
  2  INTO PRODUCER VALUES (1, 'Producer1')
  3  INTO PRODUCER VALUES (2, 'Producer2')
  4  INTO PRODUCER VALUES (3, 'Producer3')
  5  INTO PRODUCER VALUES (4, 'Producer4')
  6  INTO PRODUCER VALUES (5, 'Producer5')
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM PRODUCER;

PRODUCERID FULLNAME
---------- --------------------------------------------------
         1 Producer1
         2 Producer2
         3 Producer3
         4 Producer4
         5 Producer5

SQL> 
```

```
INSERT ALL
INTO SONG VALUES (1, 'Song1', TO_DATE('01-01-2023', 'DD-MM-YYYY'))
INTO SONG VALUES (2, 'Song2', TO_DATE('15-12-2022', 'DD-MM-YYYY'))
INTO SONG VALUES (3, 'Song3', TO_DATE('20-02-2024', 'DD-MM-YYYY'))
INTO SONG VALUES (4, 'Song4', TO_DATE('10-05-2023', 'DD-MM-YYYY'))
INTO SONG VALUES (5, 'Song5', TO_DATE('30-11-2022', 'DD-MM-YYYY'))
SELECT * FROM DUAL;
```



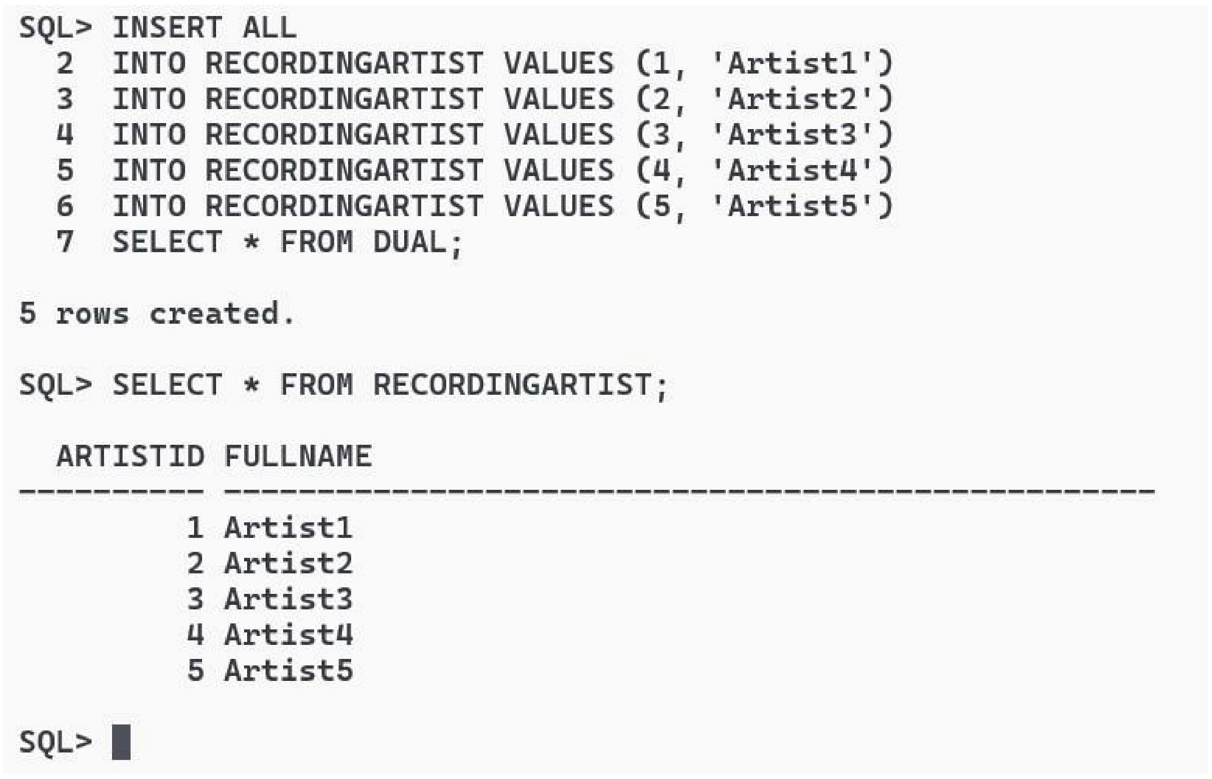
```
SQL> INSERT ALL
     INTO SONG VALUES (1, 'Song1', TO_DATE('01-01-2023', 'DD-MM-YYYY'))
     INTO SONG VALUES (2, 'Song2', TO_DATE('15-12-2022', 'DD-MM-YYYY'))
     INTO SONG VALUES (3, 'Song3', TO_DATE('20-02-2024', 'DD-MM-YYYY'))
     INTO SONG VALUES (4, 'Song4', TO_DATE('10-05-2023', 'DD-MM-YYYY'))
     INTO SONG VALUES (5, 'Song5', TO_DATE('30-11-2022', 'DD-MM-YYYY'))
     SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM SONG;

SONGNUMBER TITLE                                              COMPOSITI
---------- -------------------------------------------------- ---------
         1 Song1                                              01-JAN-23
         2 Song2                                              15-DEC-22
         3 Song3                                              20-FEB-24
         4 Song4                                              10-MAY-23
         5 Song5                                              30-NOV-22

SQL>
```

```
INSERT ALL
INTO COMPOSER VALUES (1, 'Composer1')
INTO COMPOSER VALUES (2, 'Composer2')
INTO COMPOSER VALUES (3, 'Composer3')
INTO COMPOSER VALUES (4, 'Composer4')
INTO COMPOSER VALUES (5, 'Composer5')
SELECT * FROM DUAL;
```



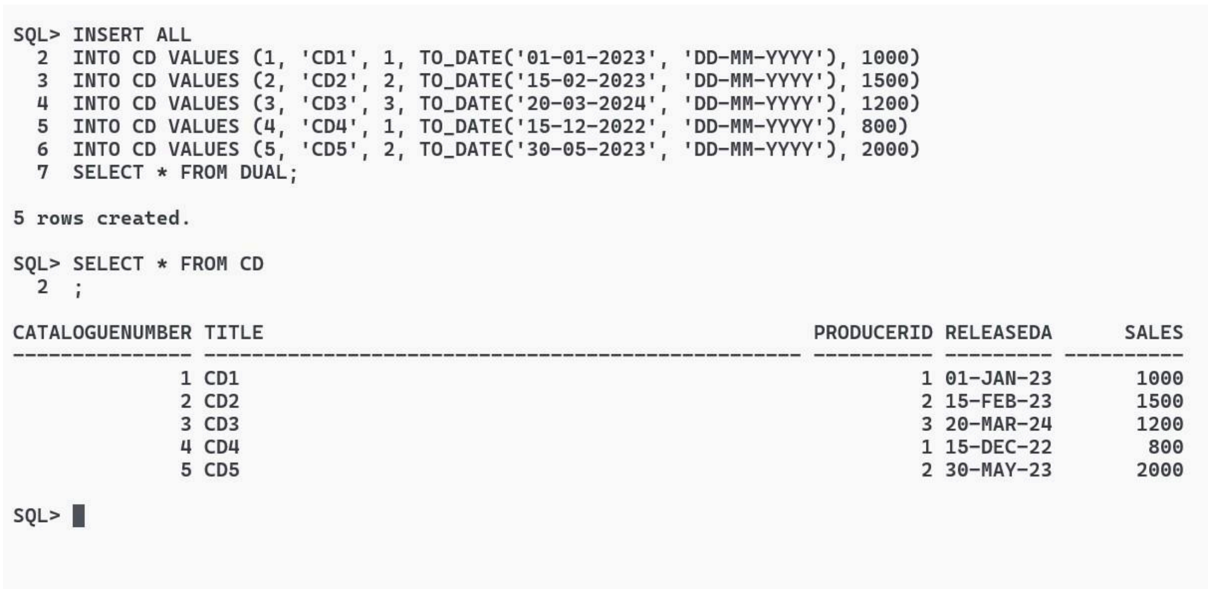
```
SQL> INSERT ALL
     INTO COMPOSER VALUES (1, 'Composer1')
     INTO COMPOSER VALUES (2, 'Composer2')
     INTO COMPOSER VALUES (3, 'Composer3')
     INTO COMPOSER VALUES (4, 'Composer4')
     INTO COMPOSER VALUES (5, 'Composer5')
     SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM COMPOSER;

COMPOSERID FULLNAME
---------- --------------------------------------------------
         1 Composer1
         2 Composer2
         3 Composer3
         4 Composer4
         5 Composer5

SQL>
```

INSERT ALL
INTO RECORDINGARTIST VALUES (1, 'Artist1')
INTO RECORDINGARTIST VALUES (2, 'Artist2')
INTO RECORDINGARTIST VALUES (3, 'Artist3')
INTO RECORDINGARTIST VALUES (4, 'Artist4')
INTO RECORDINGARTIST VALUES (5, 'Artist5')
SELECT * FROM DUAL;

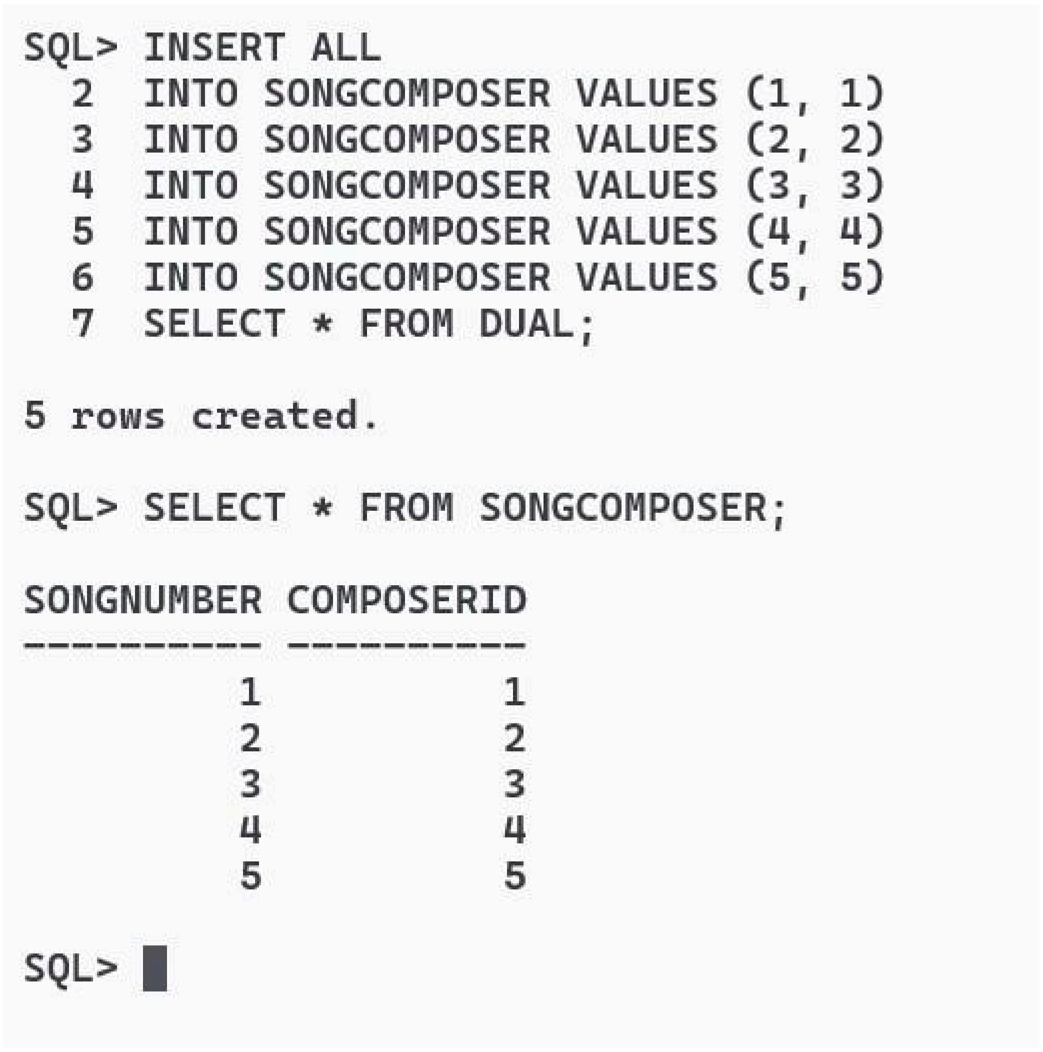


```
SQL> INSERT ALL
  2  INTO RECORDINGARTIST VALUES (1, 'Artist1')
  3  INTO RECORDINGARTIST VALUES (2, 'Artist2')
  4  INTO RECORDINGARTIST VALUES (3, 'Artist3')
  5  INTO RECORDINGARTIST VALUES (4, 'Artist4')
  6  INTO RECORDINGARTIST VALUES (5, 'Artist5')
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM RECORDINGARTIST;

  ARTISTID FULLNAME
---------- --------------------------------------------------
         1 Artist1
         2 Artist2
         3 Artist3
         4 Artist4
         5 Artist5

SQL>
```

INSERT ALL
INTO CD VALUES (1, 'CD1', 1, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 1000)
INTO CD VALUES (2, 'CD2', 2, TO_DATE('15-02-2023', 'DD-MM-YYYY'), 1500)
INTO CD VALUES (3, 'CD3', 3, TO_DATE('20-03-2024', 'DD-MM-YYYY'), 1200)
INTO CD VALUES (4, 'CD4', 1, TO_DATE('15-12-2022', 'DD-MM-YYYY'), 800)
INTO CD VALUES (5, 'CD5', 2, TO_DATE('30-05-2023', 'DD-MM-YYYY'), 2000)
SELECT * FROM DUAL;

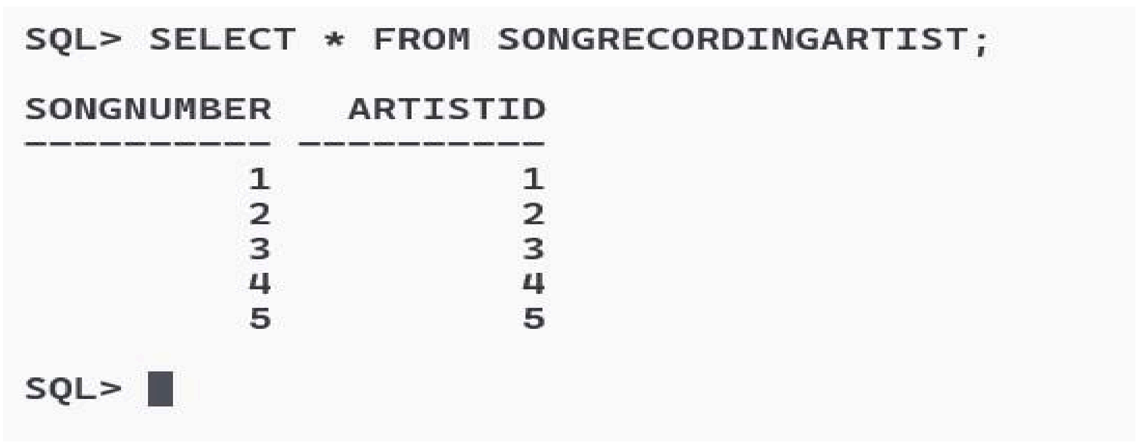

```
SQL> INSERT ALL
  2  INTO CD VALUES (1, 'CD1', 1, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 1000)
  3  INTO CD VALUES (2, 'CD2', 2, TO_DATE('15-02-2023', 'DD-MM-YYYY'), 1500)
  4  INTO CD VALUES (3, 'CD3', 3, TO_DATE('20-03-2024', 'DD-MM-YYYY'), 1200)
  5  INTO CD VALUES (4, 'CD4', 1, TO_DATE('15-12-2022', 'DD-MM-YYYY'), 800)
  6  INTO CD VALUES (5, 'CD5', 2, TO_DATE('30-05-2023', 'DD-MM-YYYY'), 2000)
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM CD
  2  ;

CATALOGUENUMBER TITLE                                              PRODUCERID RELEASEDA      SALES
--------------- -------------------------------------------------- ---------- --------- ----------
              1 CD1                                                         1 01-JAN-23       1000
              2 CD2                                                         2 15-FEB-23       1500
              3 CD3                                                         3 20-MAR-24       1200
              4 CD4                                                         1 15-DEC-22        800
              5 CD5                                                         2 30-MAY-23       2000

SQL> 
```

INSERT ALL
INTO SONGCOMPOSER VALUES (1, 1)
INTO SONGCOMPOSER VALUES (2, 2)
INTO SONGCOMPOSER VALUES (3, 3)
INTO SONGCOMPOSER VALUES (4, 4)
INTO SONGCOMPOSER VALUES (5, 5)
SELECT * FROM DUAL;



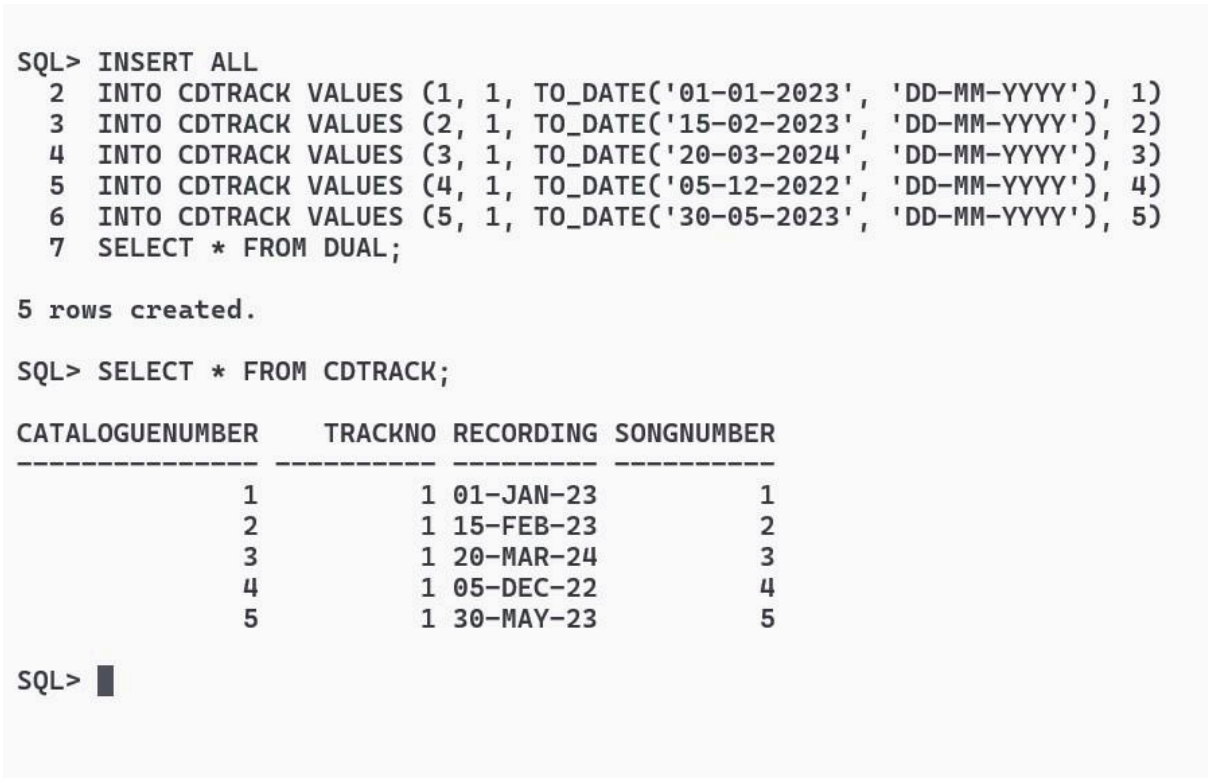
```
SQL> INSERT ALL
  2  INTO SONGCOMPOSER VALUES (1, 1)
  3  INTO SONGCOMPOSER VALUES (2, 2)
  4  INTO SONGCOMPOSER VALUES (3, 3)
  5  INTO SONGCOMPOSER VALUES (4, 4)
  6  INTO SONGCOMPOSER VALUES (5, 5)
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM SONGCOMPOSER;

SONGNUMBER COMPOSERID
---------- ----------
         1          1
         2          2
         3          3
         4          4
         5          5

SQL>
```

INSERT ALL
INTO SONGRECORDINGARTIST VALUES (1, 1)
INTO SONGRECORDINGARTIST VALUES (2, 2)
INTO SONGRECORDINGARTIST VALUES (3, 3)
INTO SONGRECORDINGARTIST VALUES (4, 4)
INTO SONGRECORDINGARTIST VALUES (5, 5)
SELECT * FROM DUAL;



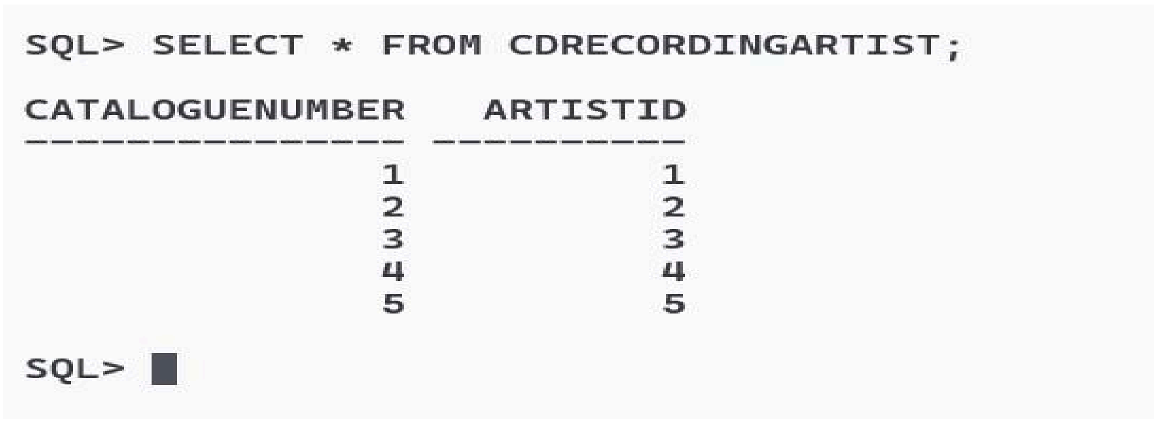
```
SQL> SELECT * FROM SONGRECORDINGARTIST;

SONGNUMBER   ARTISTID
---------- ----------
         1          1
         2          2
         3          3
         4          4
         5          5

SQL>
```

INSERT ALL
INTO CDTRACK VALUES (1, 1, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 1)
INTO CDTRACK VALUES (2, 1, TO_DATE('15-02-2023', 'DD-MM-YYYY'), 2)
INTO CDTRACK VALUES (3, 1, TO_DATE('20-03-2024', 'DD-MM-YYYY'), 3)
INTO CDTRACK VALUES (4, 1, TO_DATE('05-12-2022', 'DD-MM-YYYY'), 4)
INTO CDTRACK VALUES (5, 1, TO_DATE('30-05-2023', 'DD-MM-YYYY'), 5)
SELECT * FROM DUAL;



```
SQL> INSERT ALL
  2  INTO CDTRACK VALUES (1, 1, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 1)
  3  INTO CDTRACK VALUES (2, 1, TO_DATE('15-02-2023', 'DD-MM-YYYY'), 2)
  4  INTO CDTRACK VALUES (3, 1, TO_DATE('20-03-2024', 'DD-MM-YYYY'), 3)
  5  INTO CDTRACK VALUES (4, 1, TO_DATE('05-12-2022', 'DD-MM-YYYY'), 4)
  6  INTO CDTRACK VALUES (5, 1, TO_DATE('30-05-2023', 'DD-MM-YYYY'), 5)
  7  SELECT * FROM DUAL;

5 rows created.

SQL> SELECT * FROM CDTRACK;

CATALOGUENUMBER    TRACKNO RECORDING SONGNUMBER
--------------- ---------- --------- ----------
              1          1 01-JAN-23          1
              2          1 15-FEB-23          2
              3          1 20-MAR-24          3
              4          1 05-DEC-22          4
              5          1 30-MAY-23          5

SQL>
```

INSERT ALL
INTO CDRECORDINGARTIST VALUES (1, 1)
INTO CDRECORDINGARTIST VALUES (2, 2)
INTO CDRECORDINGARTIST VALUES (3, 3)
INTO CDRECORDINGARTIST VALUES (4, 4)
INTO CDRECORDINGARTIST VALUES (5, 5)
SELECT * FROM DUAL;

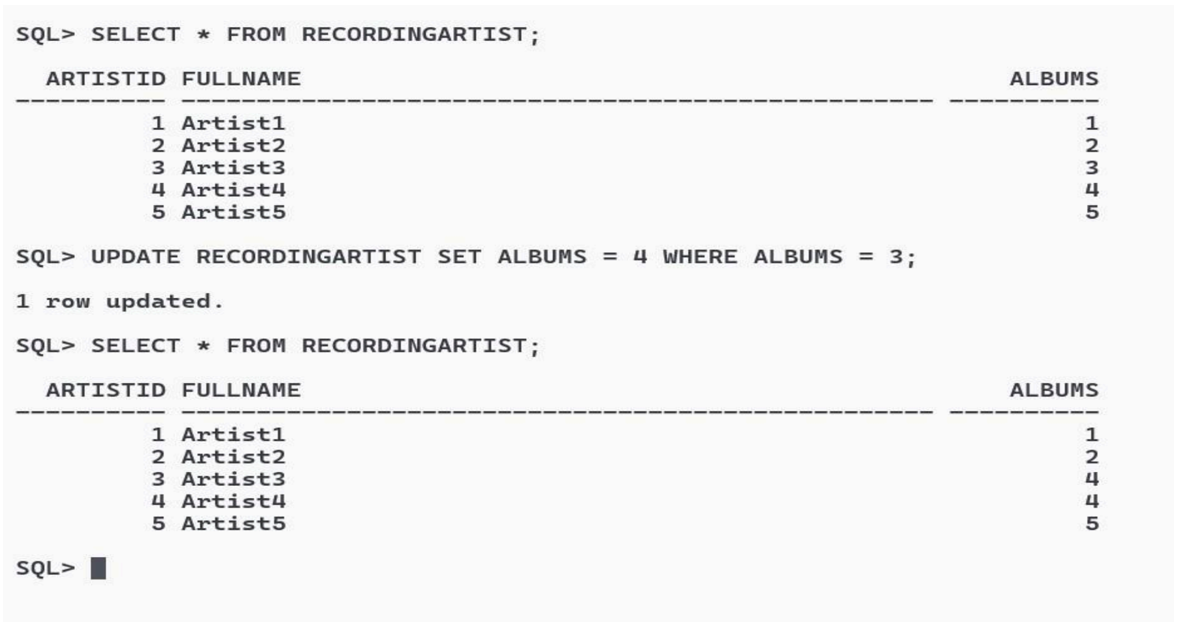


```
SQL> SELECT * FROM CDRECORDINGARTIST;

CATALOGUENUMBER   ARTISTID
--------------- ----------
              1          1
              2          2
              3          3
              4          4
              5          5

SQL>
```

SQL:-
i>Update number of recorded album to 4 for those artist who has recorded only 3.

ALTER TABLE RECORDINGARTIST ADD SALES NUMBER;
ALTER TABLE RECORDINGARTIST RENAME COLUMN SALES TO ALBUMS;
UPDATE RECORDINGARTIST SET ALBUMS = 1 WHERE ARTISTID = 1;
UPDATE RECORDINGARTIST SET ALBUMS = 2 WHERE ARTISTID = 2;
UPDATE RECORDINGARTIST SET ALBUMS = 3 WHERE ARTISTID = 3;
UPDATE RECORDINGARTIST SET ALBUMS = 4 WHERE ARTISTID = 4;
UPDATE RECORDINGARTIST SET ALBUMS = 5 WHERE ARTISTID = 5;

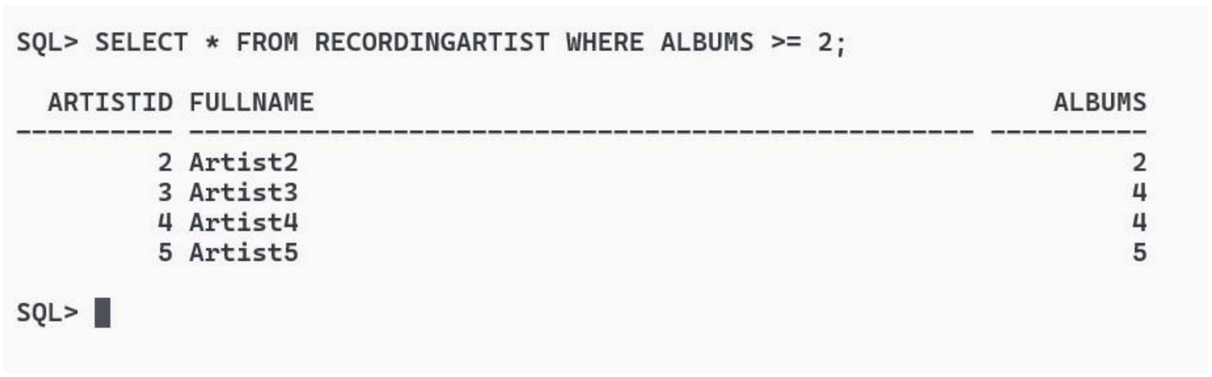
UPDATE RECORDINGARTIST SET ALBUMS = 4 WHERE ALBUMS = 3;



```
SQL> SELECT * FROM RECORDINGARTIST;

  ARTISTID FULLNAME                                               ALBUMS
---------- -------------------------------------------------- ----------
         1 Artist1                                                     1
         2 Artist2                                                     2
         3 Artist3                                                     3
         4 Artist4                                                     4
         5 Artist5                                                     5

SQL> UPDATE RECORDINGARTIST SET ALBUMS = 4 WHERE ALBUMS = 3;

1 row updated.

SQL> SELECT * FROM RECORDINGARTIST;

  ARTISTID FULLNAME                                               ALBUMS
---------- -------------------------------------------------- ----------
         1 Artist1                                                     1
         2 Artist2                                                     2
         3 Artist3                                                     4
         4 Artist4                                                     4
         5 Artist5                                                     5

SQL> 
```

ii>Find all artists who have recorded at least two albums.

SELECT * FROM RECORDINGARTIST WHERE ALBUMS >= 2;

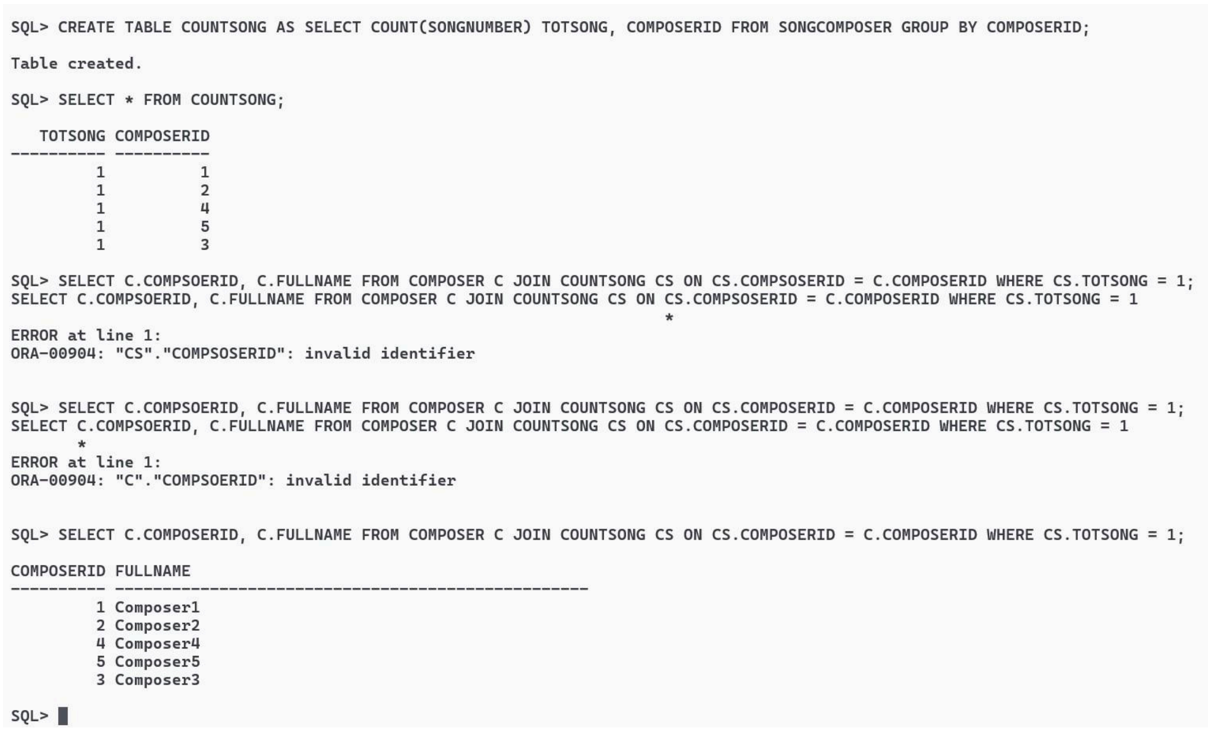


```
SQL> SELECT * FROM RECORDINGARTIST WHERE ALBUMS >= 2;

  ARTISTID FULLNAME                                               ALBUMS
---------- -------------------------------------------------- ----------
         2 Artist2                                                     2
         3 Artist3                                                     4
         4 Artist4                                                     4
         5 Artist5                                                     5

SQL> 
```

iii>Find all writers who have only written one song.

CREATE TABLE COUNTSONG AS SELECT COUNT(SONGNUMBER) TOTSONG, COMPOSERID FROM SONGCOMPOSER GROUP BY COMPOSERID;

SELECT C.COMPOSERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1;

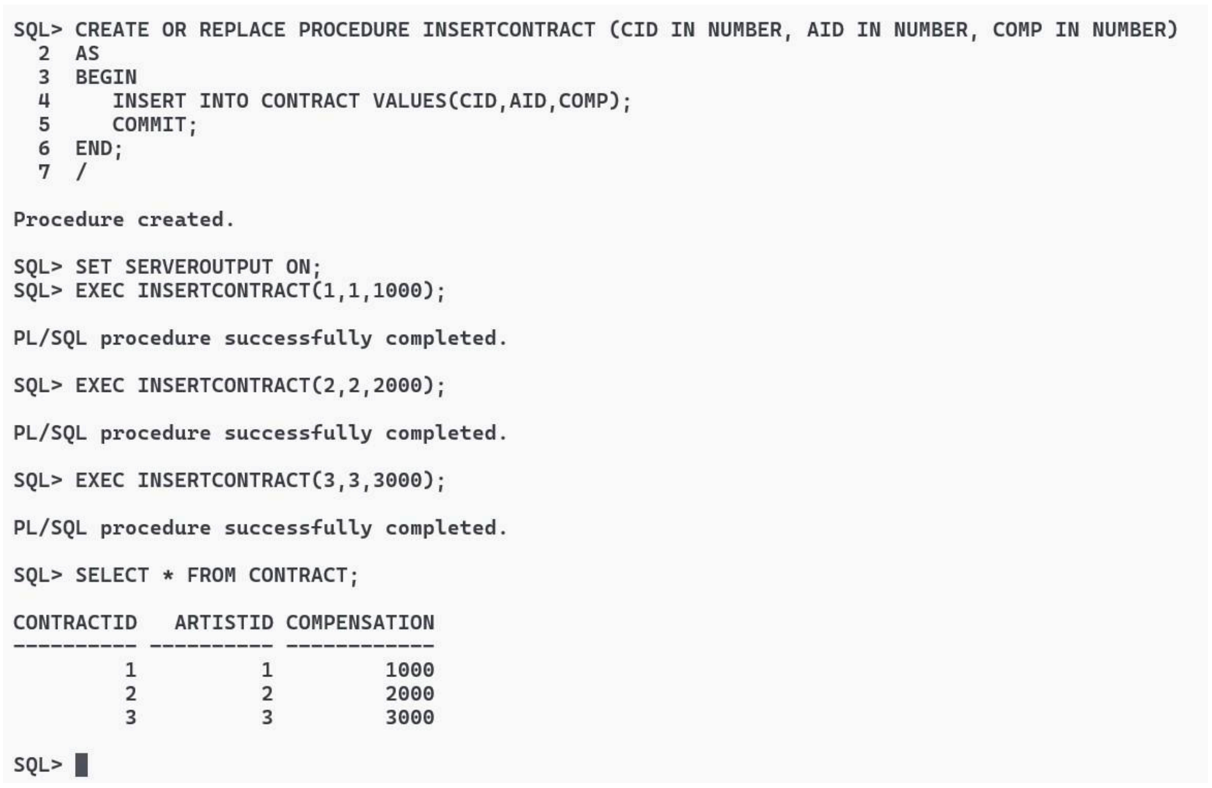


```
SQL> CREATE TABLE COUNTSONG AS SELECT COUNT(SONGNUMBER) TOTSONG, COMPOSERID FROM SONGCOMPOSER GROUP BY COMPOSERID;

Table created.

SQL> SELECT * FROM COUNTSONG;

   TOTSONG COMPOSERID
---------- ----------
         1          1
         1          2
         1          4
         1          5
         1          3

SQL> SELECT C.COMPSOERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPSOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1;
SELECT C.COMPSOERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPSOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1
                                                                       *
ERROR at line 1:
ORA-00904: "CS"."COMPSOSERID": invalid identifier


SQL> SELECT C.COMPSOERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1;
SELECT C.COMPSOERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1
       *
ERROR at line 1:
ORA-00904: "C"."COMPSOERID": invalid identifier


SQL> SELECT C.COMPOSERID, C.FULLNAME FROM COMPOSER C JOIN COUNTSONG CS ON CS.COMPOSERID = C.COMPOSERID WHERE CS.TOTSONG = 1;

COMPOSERID FULLNAME
---------- --------------------------------------------------
         1 Composer1
         2 Composer2
         4 Composer4
         5 Composer5
         3 Composer3

SQL>
```

PL/SQL

i>Write Procedure to insert a new Contract into the Contract relation.

```
CREATE OR REPLACE PROCEDURE INSERTCONTRACT (CID IN NUMBER, AID IN NUMBER, COMP IN NUMBER) AS
BEGIN
        INSERT INTO CONTRACT VALUES(CID,AID,COMP);
        COMMIT;
END;
/
```



```
SQL> CREATE OR REPLACE PROCEDURE INSERTCONTRACT (CID IN NUMBER, AID IN NUMBER, COMP IN NUMBER)
     AS
     BEGIN
        INSERT INTO CONTRACT VALUES(CID,AID,COMP);
        COMMIT;
     END;
     /

Procedure created.

SQL> SET SERVEROUTPUT ON;
SQL> EXEC INSERTCONTRACT(1,1,1000);

PL/SQL procedure successfully completed.

SQL> EXEC INSERTCONTRACT(2,2,2000);

PL/SQL procedure successfully completed.

SQL> EXEC INSERTCONTRACT(3,3,3000);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM CONTRACT;

CONTRACTID   ARTISTID COMPENSATION
---------- ---------- ------------
         1          1         1000
         2          2         2000
         3          3         3000

SQL> 
```