```
SQL> SELECT artistname FROM artistx
  2  ;

ARTISTNAME
--------------------------------------------------------------------------------
Ariana Grande
Drake
Ed Sheeran
Post Malone
```

```
SQL> SELECT artistName FROM Rapper;

ARTISTNAME
--------------------------------------------------------------------------------
Drake
Post Malone
```

```
SQL> SELECT instrumentname FROM INstrument;

INSTRUMENTNAME
--------------------------------------------------------------------------------
Guitar
Piano
Vocal
```

```
SQL> SELECT artistname, instrumentname FROM Plays;

ARTISTNAME
--------------------------------------------------------------------------------
INSTRUMENTNAME
--------------------------------------------------------------------------------
Ariana Grande
Vocal

Ed Sheeran
Guitar

Ed Sheeran
Piano


ARTISTNAME
--------------------------------------------------------------------------------
INSTRUMENTNAME
--------------------------------------------------------------------------------
Ed Sheeran
Vocal
```

```
SQL> SELECT albumname FROM ALBUM;

ALBUMNAME
--------------------------------------------------------------------------------
Divide
Nothing Was The Same
Stoney
Take Care
Thank U, Next
```

```
SQL> SELECT albumname FROM ALBUM;

ALBUMNAME
--------------------------------------------------------------------------------
Divide
Nothing Was The Same
Stoney
Take Care
Thank U, Next

SQL> SELECT songname FROM song;

SONGNAME
--------------------------------------------------------------------------------
Shape of you
Eraser
Dive
Perfect
Castle on the Hill
Galway Girl
Happier
New Man
Imagine
Needy
NASA

SONGNAME
--------------------------------------------------------------------------------
Bloodline
Fake Smile

13 rows selected.
```

```
SQL> SELECT artistname FROM singer;

ARTISTNAME
--------------------------------------------------------------------------------
Ariana Grande
Ed Sheeran
```

```
SQL> SELECT artistname FROM rapper;

ARTISTNAME
--------------------------------------------------------------------------------
Drake
Post Malone

SQL> SELECT songID, albumName, artist FROM Creates;

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------
         1
Divide
Ed Sheeran

         2
Divide
Ed Sheeran

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------

         3
Divide
Ed Sheeran

         4
Divide

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------
Ed Sheeran
```

```
--------------------------------------------------------------------------------
         8
Divide
Ed Sheeran

         9
Thank U, Next
Ariana Grande

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------


        10
Thank U, Next
Ariana Grande

        11
Thank U, Next

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------
Ariana Grande

        12
Thank U, Next
Ariana Grande

        13

    SONGID
----------
ALBUMNAME
--------------------------------------------------------------------------------
ARTIST
--------------------------------------------------------------------------------
Thank U, Next
Ariana Grande
```

```
SQL> SELECT commentedby, description FROM Thread;

COMMENTEDBY
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
Finn
hi!

Finn
Hello.

Finn
Bonjour
```

SCHEMA:

```
CREATE SEQUENCE comment_id_seq
START WITH 1
INCREMENT BY 1;

CREATE SEQUENCE song_id_seq
START WITH 1
INCREMENT BY 1;




CREATE TABLE ArtistX (
    artistName VARCHAR(255) PRIMARY KEY,
    artistOrigin VARCHAR(255),
    artistDescription CLOB,
    monthlyListeners INT
);


CREATE TABLE Rapper (
    artistName VARCHAR(255) PRIMARY KEY,
    rapStyle VARCHAR(255),
    FOREIGN KEY (artistName) REFERENCES ArtistX(artistName) ON DELETE
CASCADE
);
```

```
CREATE TABLE Instrument (
    instrumentName VARCHAR(255) PRIMARY KEY,
    origin VARCHAR(255)
);

CREATE TABLE Plays (
    artistName VARCHAR(255) NOT NULL,
    instrumentName VARCHAR(255),
    yearsExperience INT,
    FOREIGN KEY (artistName) REFERENCES ArtistX(artistName) ON DELETE
CASCADE,
    PRIMARY KEY (artistName, instrumentName),
    FOREIGN KEY (instrumentName) REFERENCES Instrument(instrumentName)
);

CREATE TABLE Album (
    albumName VARCHAR(255),
    artist VARCHAR(255) DEFAULT NULL,
    dateCreated DATE DEFAULT NULL,
    PRIMARY KEY (albumName, artist),
    FOREIGN KEY (artist) REFERENCES ArtistX(artistName) ON DELETE CASCADE
);


CREATE TABLE Song (
    songID INT PRIMARY KEY,
    songName VARCHAR(255),
    artistName VARCHAR(255),
    albumName VARCHAR(255),
    numOfListeners INT,
    FOREIGN KEY (albumName, artistName) REFERENCES Album(albumName,
artist) ON DELETE CASCADE
);

CREATE OR REPLACE TRIGGER BeforeInsertSong
BEFORE INSERT ON Song
FOR EACH ROW
DECLARE
    album_exists NUMBER;
```

```
BEGIN

    SELECT song_id_seq.NEXTVAL
    INTO :new.songID
    FROM dual;

    SELECT COUNT(*)
    INTO album_exists
    FROM Album
    WHERE albumName = :NEW.albumName AND artist = :NEW.artistName;


    IF album_exists = 0 THEN
        INSERT INTO Album (albumName, artist)
        VALUES (:NEW.albumName, :NEW.artistName);
    END IF;
END;
/




CREATE TABLE Thread (
    commentID INT CHECK (commentID > 0) PRIMARY KEY,
    threadName VARCHAR(255) UNIQUE,
    commentedBy VARCHAR(255),
    description CLOB
);

CREATE OR REPLACE TRIGGER thread_before_insert
BEFORE INSERT ON Thread
FOR EACH ROW
BEGIN
    SELECT comment_id_seq.NEXTVAL
    INTO :new.commentID
    FROM dual;
END;
/
```

```
CREATE TABLE Singer (
    artistName VARCHAR(255) PRIMARY KEY,
    vocalRange VARCHAR(255),
    FOREIGN KEY (artistName) REFERENCES ArtistX(artistName) ON DELETE
CASCADE
);

CREATE TABLE Producer (
    artistName VARCHAR(255) PRIMARY KEY,
    FOREIGN KEY (artistName) REFERENCES ArtistX(artistName) ON DELETE
CASCADE
);


CREATE TABLE Creates (
    songID INT,
    albumName VARCHAR(255),
    artist VARCHAR(255),
    PRIMARY KEY (songID, albumName, artist),
    FOREIGN KEY (songID) REFERENCES Song(songID) ON DELETE CASCADE,
    FOREIGN KEY (albumName, artist) REFERENCES Album(albumName, artist) ON
DELETE CASCADE,
    FOREIGN KEY (artist) REFERENCES ArtistX(artistName) ON DELETE CASCADE
);
```