CIS 251 Intro To SQL 5/19/2013
Practice Exam

Jon Raymond Melchizedek Day - 955309683

This is your practice final exam

Step 1 - Create the database and insert the data. This is a music inventory database..(This is not my collection!) You will have to keep create each table

1. Decide on primary keys and what they should be. (Create a separate unique identifier)
2. Decide if a field is required or options
3. Decide if information should be foreign key or redundant data (Do you really want redundant data?)
4. Decide on what restrictions or checks you should have on data

What you should keep in mind:

1. You are working with a relational database, create foreign keys and relationships where needed.
2. Remember that one of the great facts of relation data is that data is not repeated from table to tale!

Data Set 1: Music Categories

Music Category
Classical
Rock
Jazz
Country
New Age

Data Set 2: Recording Artists

Recording Artist	Notes
The Popular Mechanics	Classic rock sound
Crawdad Stare	Grunge Rock style.
Dog House Rules	Smooth Jazz from Chicago area.
Mary Saveley	Soft soul jase

Data Set 3: Recordings

Recording Title	Recording Artist ID	Music Category ID	Recording Label	Format	Number of Tracks	Notes
Noise in the Garage	Crawdad Stare	Rock	Goes to 11, Inc.	CD	10	1st recording to break top 10 by this artist
Look Both Ways	Crawdad Stare	Rock	Hippo Records	Cassette	12	Artists original release. Underground until Noise in garage was popular.
Outback	Crawdad Stare	Rock		CD	10	
Short Circuit	Dog House Rules	Jazz		Cassette	7	
Meditations	Mary Saveley	New Age		CD	6	
Opus 65	The Popular Mechanics	Classical		CD	5	
Sounds Better Louder	Crawdad Stare	Rock	Goes to 11, Inc.	CD	10	

Display all the statements that you use to create your database.

```
CREATE  TABLE MUSIC_CATEGORY
        (MUSIC_ID           INTEGER       NOT NULL,
         MUSIC_DESC         CHAR(20)      NOT NULL,
         PRIMARY KEY        (MUSIC_ID)                  );


CREATE  TABLE RECORDING_ARTISTS
        (ARTIST_ID          INTEGER       NOT NULL,
         ARTIST_DESC        CHAR(30)      NOT NULL,
         ARTIST_NOTES       CHAR(40)      NOT NULL,
         PRIMARY KEY        (ARTIST_ID)                 );


CREATE  TABLE RECORDINGS
        (RECORDING_ID       INTEGER       NOT NULL,
         RECORDING_TITLE    VARCHAR(20)   NOT NULL,
         ARTIST_ID          INTEGER       NOT NULL,
         MUSIC_ID           INTEGER       NOT NULL,
         RECORDING_LABEL    VARCHAR(20),
         RECORDING_FORMAT   CHAR(20)      NOT NULL,
         NUM_TRACKS         INTEGER       NOT NULL,
         RECORDING_NOTES    VARCHAR(100),
         PRIMARY KEY        (RECORDING_ID),
         FOREIGN KEY (MUSIC_ID) REFERENCES MUSIC_CATEGORY (MUSIC_ID),
         FOREIGN KEY (ARTIST_ID) REFERENCES RECORDING_ARTISTS
(ARTIST_ID) );

INSERT INTO MUSIC_CATEGORY VALUES
  (1, 'Classical'),
```

```
  (2, 'Rock'),
  (3, 'Jazz'),
  (4, 'Country'),
  (5, 'New Age');

INSERT INTO RECORDING_ARTISTS VALUES
  (1, 'The Popular Mechanics', 'Classic rock sound'),
  (2, 'Crawdad Stare', 'Grunge Rock Style.'),
  (3, 'Dog House Rules', 'Smooth Jazz from Chicago area.'),
  (4, 'Mary Saveley', 'Soft soul jase');

INSERT INTO RECORDINGS VALUES
  (1, 'Noise in the Garage', (SELECT ARTIST_ID FROM RECORDING_ARTISTS
WHERE ARTIST_DESC = 'Crawdad Stare'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Rock'),
'Goes to 11, Inc.', 'CD',
    10, '1st recording to breaktop 10 by this artist');

INSERT INTO RECORDINGS VALUES
  (2, 'Look Both Ways', (SELECT ARTIST_ID FROM RECORDING_ARTISTS WHERE
ARTIST_DESC = 'Crawdad Stare'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Rock'),
'Hippo Records', 'Cassette',
    12, 'Artists original release. Underground until Noise in garage
was popular');

INSERT INTO RECORDINGS VALUES
  (3, 'Outback', (SELECT ARTIST_ID FROM RECORDING_ARTISTS WHERE
ARTIST_DESC = 'Crawdad Stare'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Rock'),
NULL, 'CD',
    10, NULL);

INSERT INTO RECORDINGS VALUES
  (4, 'Short Circuit', (SELECT ARTIST_ID FROM RECORDING_ARTISTS WHERE
ARTIST_DESC = 'Dog House Rules'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Jazz'),
NULL, 'Cassette',
    7, NULL);

INSERT INTO RECORDINGS VALUES
  (5, 'Meditations', (SELECT ARTIST_ID FROM RECORDING_ARTISTS WHERE
ARTIST_DESC = 'Mary Saveley'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'New Age'),
NULL, 'CD',
    6, NULL);

INSERT INTO RECORDINGS VALUES
  (6, 'Opus 65', (SELECT ARTIST_ID FROM RECORDING_ARTISTS WHERE
ARTIST_DESC = 'The Popular Mechanics'),
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Classical'),
NULL, 'CD',
    5, NULL);

INSERT INTO RECORDINGS VALUES
  (7, 'Sounds Better Louder', (SELECT ARTIST_ID FROM RECORDING_ARTISTS
WHERE ARTIST_DESC = 'Crawdad Stare'),
```

```
  (SELECT MUSIC_ID FROM MUSIC_CATEGORY WHERE MUSIC_DESC = 'Rock'),
'Goes to 11, Inc.', 'CD',
    10, NULL);
```

After creating your tables and inserting your data, you may proceed to step 2.

Step 2 - Create the following SQL statements against your new database. This is a music inventory database..

You should include both your SQL statements and the results of your query.

1. Show all information available on Music Categories

```
SELECT  *
FROM    MUSIC_CATEGORY
```

2. Show all the information available on Recordings.

```
SELECT  *
FROM    RECORDINGS
```

3. Show all the information available on Recording Artists.

```
SELECT  *
FROM    RECORDING_ARTISTS
```

4. For all Recording Artists, show the artist's name, recording title, recording format and the number of tracks available on the recording.

```
SELECT    A.ARTIST_DESC, R.RECORDING_TITLE,
R.RECORDING_FORMAT, R.NUM_TRACKS
FROM      RECORDING_ARTISTS AS A LEFT JOIN RECORDINGS AS R
ON        R.ARTIST_ID = A.ARTIST_ID
ORDER BY  ARTIST_DESC
```

5. For all Music Categories, show the category and recording title.

```
SELECT    R.RECORDING_TITLE, M.MUSIC_DESC
FROM      MUSIC_CATEGORY AS M LEFT JOIN RECORDINGS AS R
ON        R.MUSIC_ID = M.MUSIC_ID
ORDER BY  MUSIC_DESC
```

6. For all Recordings, show the recording title, artist name, and music category.

```
SELECT    RECORDING_TITLE,
          (SELECT ARTIST_DESC
               FROM RECORDING_ARTISTS AS A
               WHERE R.ARTIST_ID = A.ARTIST_ID),
          (SELECT MUSIC_DESC
```

```
                FROM MUSIC_CATEGORY AS M
                WHERE R.MUSIC_ID = M.MUSIC_ID)
FROM      RECORDINGS AS R
ORDER BY  RECORDING_TITLE
```

7. You have found that you are asking for the information in question 6 often as your collection grows. Create a view entitled: "my_collection" AND show all information available from the view.

```
CREATE    VIEW MY_COLLECTION AS
SELECT    RECORDING_TITLE,
          (SELECT ARTIST_DESC FROM RECORDING_ARTISTS AS A
WHERE R.ARTIST_ID = A.ARTIST_ID),
          (SELECT MUSIC_DESC FROM MUSIC_CATEGORY AS M WHERE
R.MUSIC_ID = M.MUSIC_ID)
FROM      RECORDINGS AS R
ORDER BY  RECORDING_TITLE;

SELECT    *
FROM      MY_COLLECTION
```

8. Using your newly created view, show all the information from "my_collection" and the comments about the recordings.;

```
SELECT    *, (SELECT  R.RECORDING_NOTES
              FROM    RECORDINGS AS R
              WHERE   MC.RECORDING_TITLE = R.RECORDING_TITLE)
FROM      MY_COLLECTION AS MC
```

9. Using your newly created view, show all the information from "my_collection" and the comments about the recording artists.

```
SELECT    *, (SELECT  A.ARTIST_NOTES
              FROM    RECORDING_ARTISTS AS A
              WHERE   A.ARTIST_ID =
       (SELECT   R.ARTIST_ID
        FROM     RECORDINGS AS R
        WHERE    MC.RECORDING_TITLE = R.RECORDING_TITLE))
FROM      MY_COLLECTION AS MC
```

10. Delete your database and all information

```
DROP VIEW MY_COLLECTION;
DROP TABLE RECORDINGS;
DROP TABLE RECORDING_ARTISTS;
DROP TABLE MUSIC_CATEGORY;
```