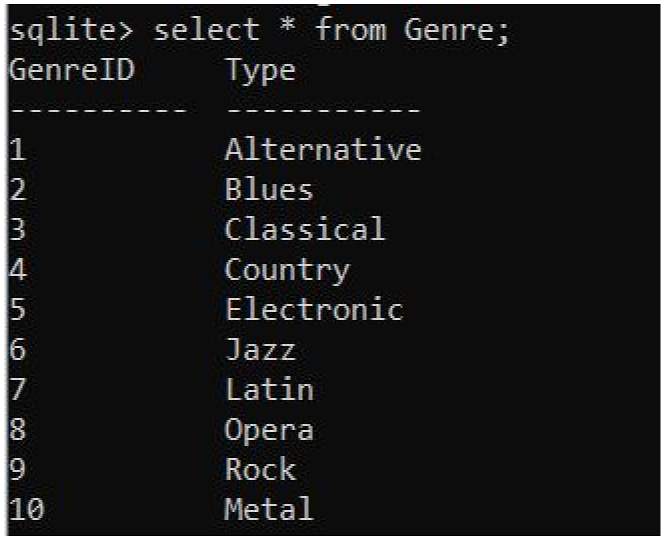
Table for each entity:

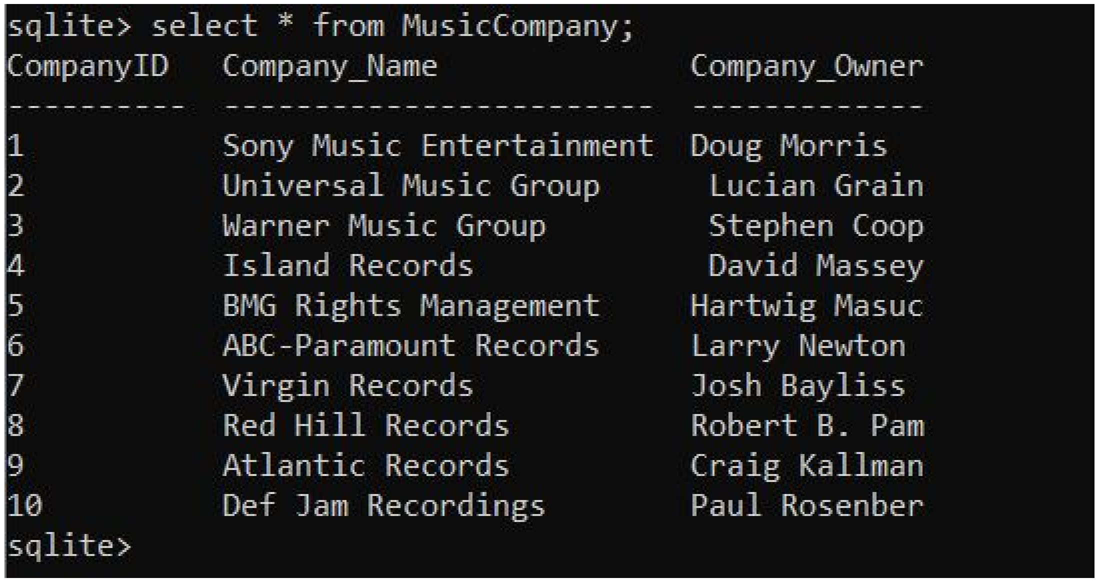
```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE Genre(
GenreID INTEGER PRIMARY KEY AUTOINCREMENT,
Type VARCHAR(10) NOT NULL,
CONSTRAINT ck_type UNIQUE (Type)
);
INSERT INTO "GENRE" VALUES(1,'Alternative');
INSERT INTO "GENRE" VALUES(2,'Blues');
INSERT INTO "GENRE" VALUES(3,'Classical');
INSERT INTO "GENRE" VALUES(4,'Country');
INSERT INTO "GENRE" VALUES(5,'Electronic');
INSERT INTO "GENRE" VALUES(6,'Jazz');
INSERT INTO "GENRE" VALUES(7,'Latin');
INSERT INTO "GENRE" VALUES(8,'Opera');
INSERT INTO "GENRE" VALUES(9,'Rock');
INSERT INTO "GENRE" VALUES(10,'Metal');
```



```
sqlite> select * from Genre;
GenreID     Type
----------  -----------
1           Alternative
2           Blues
3           Classical
4           Country
5           Electronic
6           Jazz
7           Latin
8           Opera
9           Rock
10          Metal
```

```
CREATE TABLE MusicCompany(
CompanyID INTEGER PRIMARY KEY AUTOINCREMENT,
Company_Name VARCHAR(10) NOT NULL,
Company_Owner CHAR(10) NOT NULL,
CONSTRAINT ck_Company_Name UNIQUE (Company_Name)
);
INSERT INTO "MusicCompany" VALUES(1,'Sony Music Entertainment', 'Doug Morris');
INSERT INTO "MusicCompany" VALUES(2,'Universal Music Group', 'Lucian Grainge');
INSERT INTO "MusicCompany" VALUES(3,'Warner Music Group', 'Stephen Cooper');
INSERT INTO "MusicCompany" VALUES(4,'Island Records', 'David Massey');
INSERT INTO "MusicCompany" VALUES(5,'BMG Rights Management', 'Hartwig Masuch');
INSERT INTO "MusicCompany" VALUES(6,'ABC-Paramount Records', 'Larry Newton');
INSERT INTO "MusicCompany" VALUES(7,'Virgin Records', 'Josh Bayliss');
```

INSERT INTO "MusicCompany" VALUES(8,'Red Hill Records', 'Robert B. Pamplin Jr.');
INSERT INTO "MusicCompany" VALUES(9,'Atlantic Records', 'Craig Kallman');
INSERT INTO "MusicCompany" VALUES(10,'Def Jam Recordings', 'Paul Rosenberg');

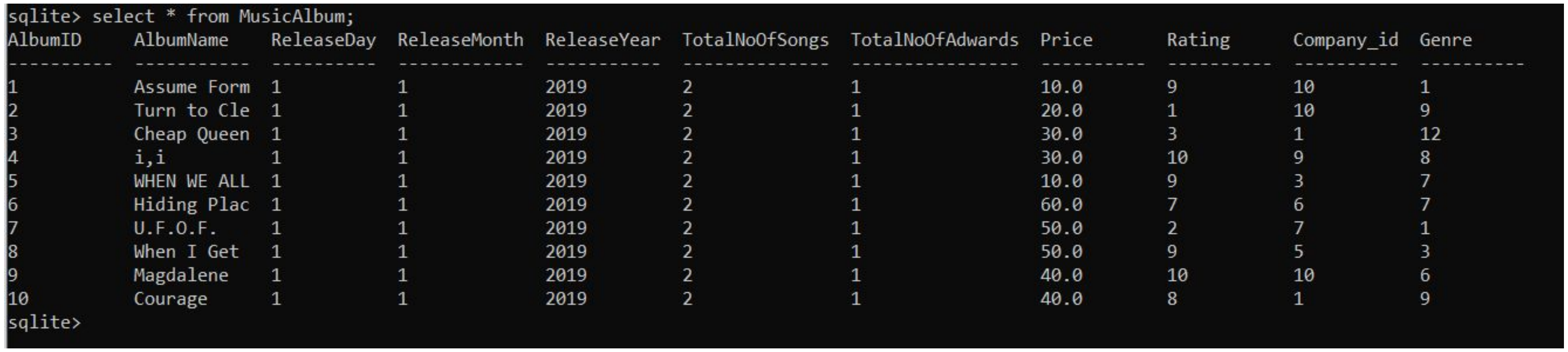


```
sqlite> select * from MusicCompany;
CompanyID   Company_Name             Company_Owner
----------  -----------------------  -------------
1           Sony Music Entertainment  Doug Morris
2           Universal Music Group      Lucian Grain
3           Warner Music Group         Stephen Coop
4           Island Records             David Massey
5           BMG Rights Management     Hartwig Masuc
6           ABC-Paramount Records     Larry Newton
7           Virgin Records            Josh Bayliss
8           Red Hill Records          Robert B. Pam
9           Atlantic Records          Craig Kallman
10          Def Jam Recordings        Paul Rosenber
sqlite>
```

CREATE TABLE MusicAlbum(
AlbumID INTEGER PRIMARY KEY AUTOINCREMENT,
AlbumName VARCHAR(255) NOT NULL,
ReleaseDay INTEGER NOT NULL,
ReleaseMonth INTEGER NOT NULL,
ReleaseYear INTEGER NOT NULL,
TotalNoOfSongs INTEGER NOT NULL,
TotalNoOfAdwards INTEGER NOT NULL,
Price REAL DEFAULT 50.00,

```
Rating INTEGER DEFAULT 0 check(Rating >=0 AND Rating<=10),
Company_id INTEGER NOT NULL,
Genre VARCHAR(10),
CONSTRAINT ck_AlbumName UNIQUE (AlbumName),
CONSTRAINT fk_Company_id  FOREIGN KEY (Company_id) REFERENCES MusicCompany(CompanyID),
CONSTRAINT fk_Genre  FOREIGN KEY (Genre) REFERENCES genre(GenreID)
);

INSERT INTO "MusicAlbum" VALUES(1,  'Assume Form',  '01',  '01',  '2019',   '2',  '1',  10.0,  9,  10,  '1');
INSERT INTO "MusicAlbum" VALUES(2,'Turn to Clear View',  '01',  '01',  '2019',   '2',  '1',    20.0,1,10,'9');
INSERT INTO "MusicAlbum" VALUES(3,'Cheap Queen',  '01',  '01',  '2019',   '2',  '1' ,30.0,3,1,'12');
INSERT INTO "MusicAlbum" VALUES(4,'i,i',    '01',  '01',  '2019',   '2',  '1'  ,30.0,10,9,'8');
INSERT INTO "MusicAlbum" VALUES(5,'WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?',    '01',  '01',
'2019',   '2',  '1',10.0,9,3,'7');
INSERT INTO "MusicAlbum" VALUES(6,  'Hiding Places',  '01',  '01',  '2019',   '2',  '1' ,60.0,7,6,'7');
INSERT INTO "MusicAlbum" VALUES(7, 'U.F.O.F.',  '01',  '01',  '2019',   '2',  '1',   50.0,2,7,'1');
INSERT INTO "MusicAlbum" VALUES(8, 'When I Get Home','01',  '01',  '2019',   '2',  '1',  50.0,   9,  5, '3');
INSERT INTO "MusicAlbum" VALUES(9,'Magdalene','01',  '01',  '2019',   '2',  '1', 40.0,10,10,'6');
INSERT INTO "MusicAlbum" VALUES(10,'Courage','01',  '01',  '2019',   '2',  '1',  40.0,8,1,'9');
```

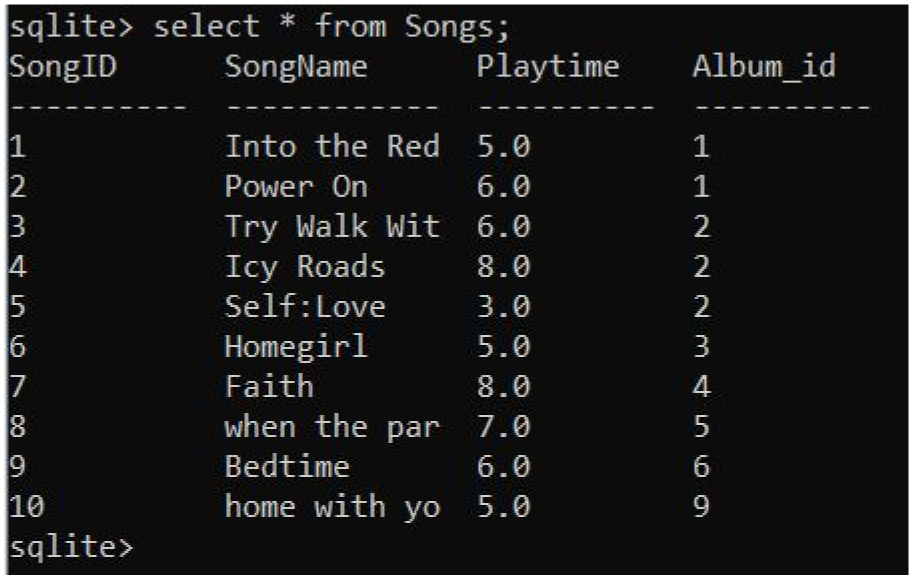


```
sqlite> select * from MusicAlbum;
AlbumID     AlbumName    ReleaseDay  ReleaseMonth  ReleaseYear  TotalNoOfSongs  TotalNoOfAdwards  Price       Rating      Company_id  Genre
----------  -----------  ----------  ------------  -----------  --------------  ----------------  ----------  ----------  ----------  ----------
1           Assume Form  1           1             2019         2               1                 10.0        9           10          1
2           Turn to Cle  1           1             2019         2               1                 20.0        1           10          9
3           Cheap Queen  1           1             2019         2               1                 30.0        3           1           12
4           i,i          1           1             2019         2               1                 30.0        10          9           8
5           WHEN WE ALL  1           1             2019         2               1                 10.0        9           3           7
6           Hiding Plac  1           1             2019         2               1                 60.0        7           6           7
7           U.F.O.F.     1           1             2019         2               1                 50.0        2           7           1
8           When I Get   1           1             2019         2               1                 50.0        9           5           3
9           Magdalene    1           1             2019         2               1                 40.0        10          10          6
10          Courage      1           1             2019         2               1                 40.0        8           1           9
sqlite>
```

```
CREATE TABLE Songs(
SongID INTEGER PRIMARY KEY AUTOINCREMENT,
SongName VARCHAR(10) NOT NULL,
Playtime REAL,
Album_id INTEGER NOT NULL,
CONSTRAINT ck_SongName  UNIQUE (SongName),
CONSTRAINT fk_Album_id FOREIGN KEY (Album_id) REFERENCES MusicAlbum(AlbumID)
);

INSERT INTO "Songs" VALUES(1,’Into the Red’,5.0,1);
INSERT INTO "Songs" VALUES(2,'Power On',6.0,1);
INSERT INTO "Songs" VALUES(3,'Try Walk With Me',6.0,2);
INSERT INTO "Songs" VALUES(4,'Icy Roads',8.0,2);
INSERT INTO "Songs" VALUES(5,'Self:Love',3.0,2);
INSERT INTO "Songs" VALUES(6,'Homegirl',5.0,3);
INSERT INTO "Songs" VALUES(7,'Faith',8.0,4);
```

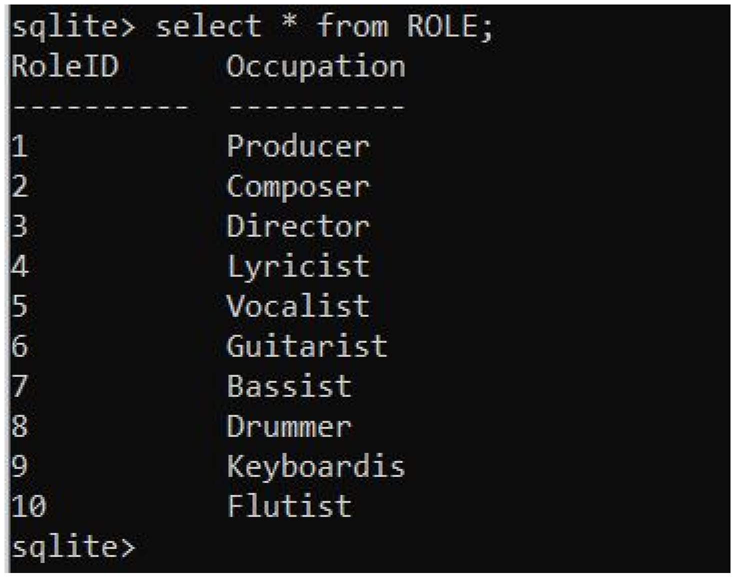
INSERT INTO "Songs" VALUES(8,'when the party over',7.0,5);
INSERT INTO "Songs" VALUES(9,'Bedtime',6.0,6);
INSERT INTO "Songs" VALUES(10,'home with you',5.0,9);



```
sqlite> select * from Songs;
SongID      SongName      Playtime    Album_id
----------  ------------  ----------  ----------
1           Into the Red  5.0         1
2           Power On      6.0         1
3           Try Walk Wit  6.0         2
4           Icy Roads     8.0         2
5           Self:Love     3.0         2
6           Homegirl      5.0         3
7           Faith         8.0         4
8           when the par  7.0         5
9           Bedtime       6.0         6
10          home with yo  5.0         9
sqlite>
```

CREATE TABLE ROLE(
RoleID INTEGER PRIMARY KEY AUTOINCREMENT,
Occupation VARCHAR(10) NOT NULL,
CONSTRAINT ck_Occupation UNIQUE (Occupation)
);
INSERT INTO "ROLE" VALUES(1,'Producer');
INSERT INTO "ROLE" VALUES(2,'Composer');

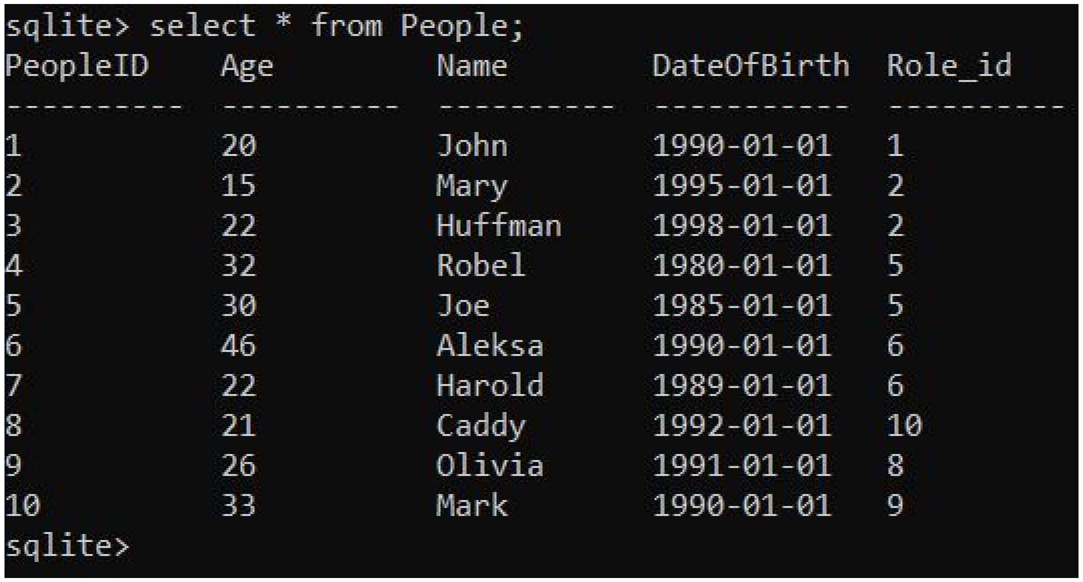
INSERT INTO "ROLE" VALUES(3,'Director');
INSERT INTO "ROLE" VALUES(4,'Lyricist');
INSERT INTO "ROLE" VALUES(5,'Vocalist');
INSERT INTO "ROLE" VALUES(6,'Guitarist');
INSERT INTO "ROLE" VALUES(7,'Bassist');
INSERT INTO "ROLE" VALUES(8,'Drummer');
INSERT INTO "ROLE" VALUES(9,'Keyboardist');
INSERT INTO "ROLE" VALUES(10,'Flutist');



```
sqlite> select * from ROLE;
RoleID      Occupation
----------  ----------
1           Producer
2           Composer
3           Director
4           Lyricist
5           Vocalist
6           Guitarist
7           Bassist
8           Drummer
9           Keyboardis
10          Flutist
sqlite>
```

CREATE TABLE People(
PeopleID INTEGER PRIMARY KEY AUTOINCREMENT,
Age INTEGER NOT NULL,

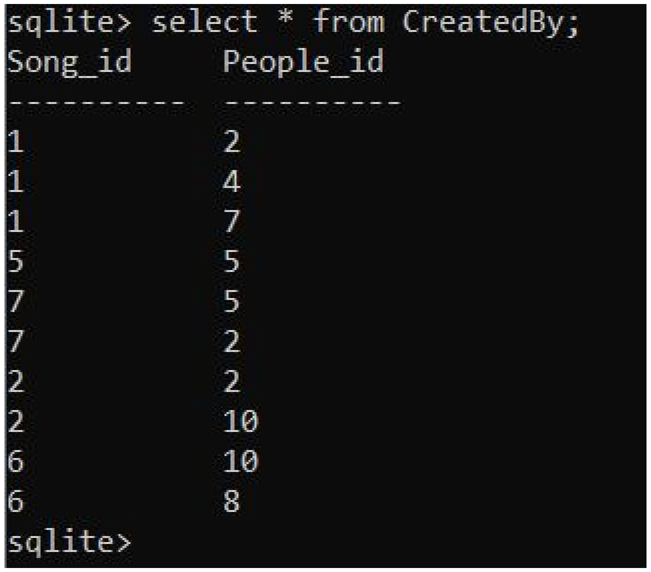
```
Name VARCHAR(10) NOT NULL,
DateOfBirth DATE NOT NULL,
Role_id INTEGER NOT NULL,
CONSTRAINT ck_Name UNIQUE (Name),
CONSTRAINT fk_Role_id FOREIGN KEY (Role_id) REFERENCES ROLE(RoleID)
);
INSERT INTO "PEOPLE" VALUES(1,20,'John','1990-01-01',1);
INSERT INTO "PEOPLE" VALUES(2,15,'Mary','1995-01-01',2);
INSERT INTO "PEOPLE" VALUES(3,22,'Huffman','1998-01-01',2);
INSERT INTO "PEOPLE" VALUES(4,32,'Robel','1980-01-01',5);
INSERT INTO "PEOPLE" VALUES(5,30,'Joe','1985-01-01',5);
INSERT INTO "PEOPLE" VALUES(6,46,'Aleksa','1990-01-01',6);
INSERT INTO "PEOPLE" VALUES(7,22,'Harold','1989-01-01',6);
INSERT INTO "PEOPLE" VALUES(8,21,'Caddy','1992-01-01',10);
INSERT INTO "PEOPLE" VALUES(9,26,'Olivia','1991-01-01',8);
INSERT INTO "PEOPLE" VALUES(10,33,'Mark','1990-01-01',9);
```



```
sqlite> select * from People;
PeopleID    Age         Name        DateOfBirth  Role_id
----------  ----------  ----------  -----------  ----------
1           20          John        1990-01-01   1
2           15          Mary        1995-01-01   2
3           22          Huffman     1998-01-01   2
4           32          Robel       1980-01-01   5
5           30          Joe         1985-01-01   5
6           46          Aleksa      1990-01-01   6
7           22          Harold      1989-01-01   6
8           21          Caddy       1992-01-01   10
9           26          Olivia      1991-01-01   8
10          33          Mark        1990-01-01   9
sqlite>
```

```
CREATE TABLE CreatedBy(
Song_id INTEGER NOT NULL,
People_id INTEGER NOT NULL
);

INSERT INTO "CreatedBy" VALUES(1,2);
INSERT INTO "CreatedBy" VALUES(1,4);
INSERT INTO "CreatedBy" VALUES(1,7);
INSERT INTO "CreatedBy" VALUES(5,5);
INSERT INTO "CreatedBy" VALUES(7,5);
INSERT INTO "CreatedBy" VALUES(7,2);
INSERT INTO "CreatedBy" VALUES(2,2);
INSERT INTO "CreatedBy" VALUES(2,10);
INSERT INTO "CreatedBy" VALUES(6,10);
    INSERT INTO "CreatedBy" VALUES(6,8);
```

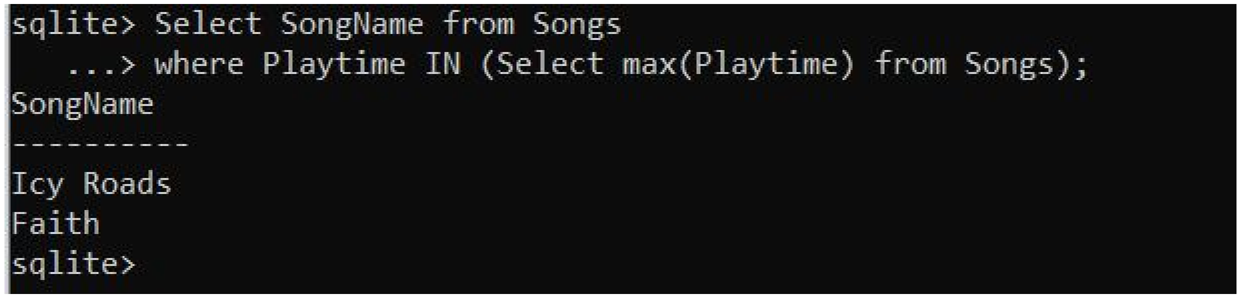


```
sqlite> select * from CreatedBy;
Song_id     People_id
----------  ----------
1           2
1           4
1           7
5           5
7           5
7           2
2           2
2           10
6           10
6           8
sqlite>
```

Queries & Description :

1.Which songs have the highest playtime?

Select SongName from Songs
where Playtime IN (Select max(Playtime) from Songs);

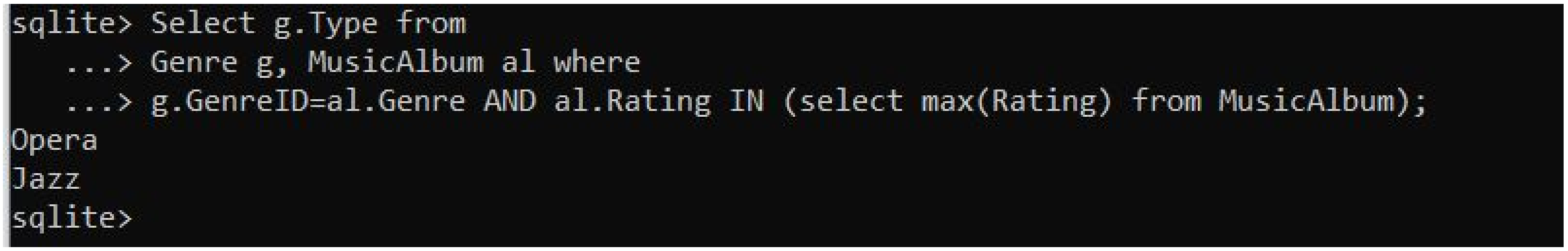


```
sqlite> Select SongName from Songs
   ...> where Playtime IN (Select max(Playtime) from Songs);
SongName
----------
Icy Roads
Faith
sqlite>
```

The query selects all the SongNames from the table Songs which has the highest playtime among all. Icy Roads and Faith satisfies the condition by having a playtime of 8 min higher than others.

2.Which genre of album has the highest rating?

Select g.Type from
Genre g, MusicAlbum al where
g.GenreID=al.Genre AND al.Rating IN (select max(Rating) from MusicAlbum);

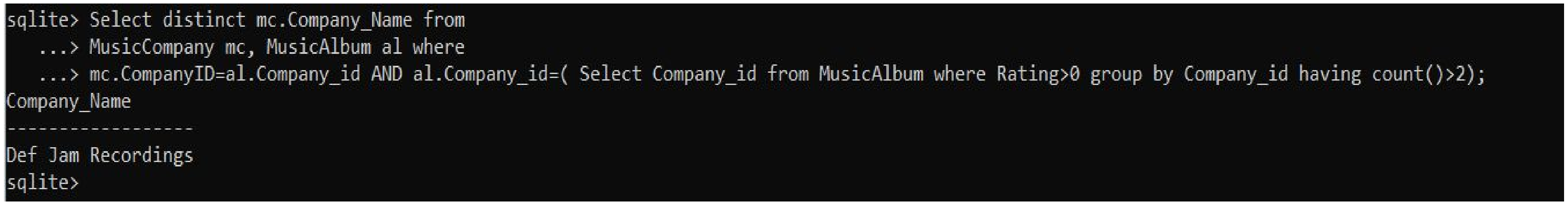


```
sqlite> Select g.Type from
   ...> Genre g, MusicAlbum al where
   ...> g.GenreID=al.Genre AND al.Rating IN (select max(Rating) from MusicAlbum);
Opera
Jazz
sqlite>
```

The query selects all the genres from the table (Genre*MusicAlbum) which has the highest rating among all. Opera and Jazz satisfies the condition by having a rating of 10, which is higher than others.

3. Which company owns more than 2 number of albums?Which company owns the number of albums released greater than 2?

Select distinct mc.Company_Name from
MusicCompany mc, MusicAlbum al where
mc.CompanyID=al.Company_id group by Company_id having count()>=2;

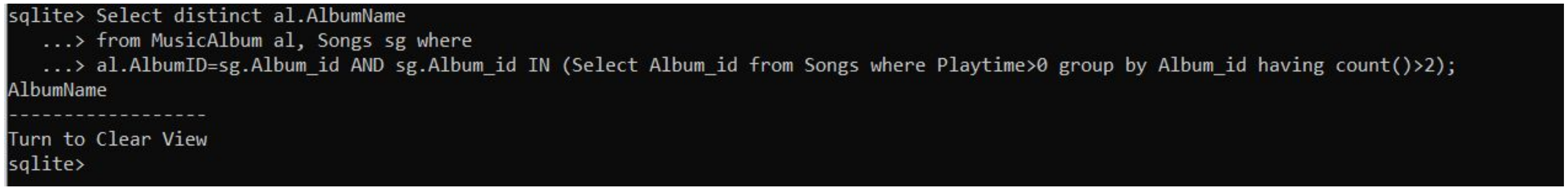


```
sqlite> Select distinct mc.Company_Name from
   ...> MusicCompany mc, MusicAlbum al where
   ...> mc.CompanyID=al.Company_id AND al.Company_id=( Select Company_id from MusicAlbum where Rating>0 group by Company_id having count()>2);
Company_Name
------------------
Def Jam Recordings
sqlite>
```

The query selects distinct (no duplicates) company names from the table (MusicCompany*MusicAlbum) which has released more than 2 albums.

4.Which albums have the number of songs greater than 2?

Select distinct al.AlbumName
from MusicAlbum al, Songs sg where
al.AlbumID=sg.Album_id group by Album_id having count()>2;

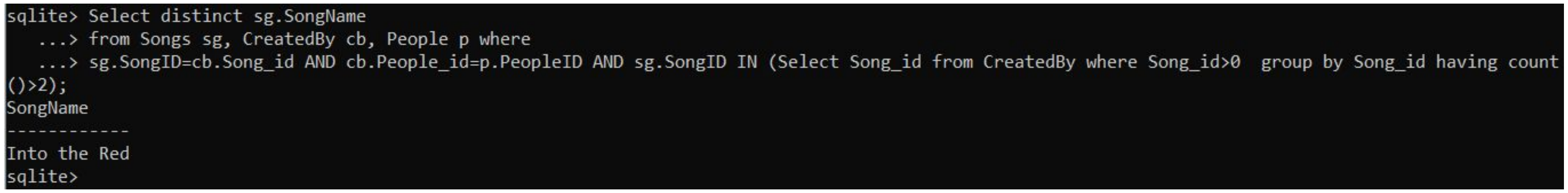


```
sqlite> Select distinct al.AlbumName
   ...> from MusicAlbum al, Songs sg where
   ...> al.AlbumID=sg.Album_id AND sg.Album_id IN (Select Album_id from Songs where Playtime>0 group by Album_id having count()>2);
AlbumName
------------------
Turn to Clear View
sqlite>
```

The query selects distinct (no duplicates) album names from the table (MusicAlbum*Songs) which has more than two songs.

5. Find the songs names which have more than 2 people working on it?

Select distinct sg.SongName
from Songs sg, CreatedBy cb, People p where
sg.SongID=cb.Song_id AND cb.People_id=p.PeopleID group by Song_id having count()>2;

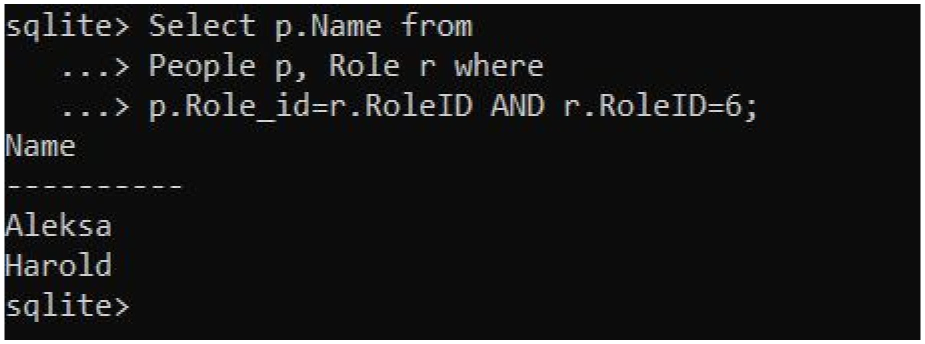


```
sqlite> Select distinct sg.SongName
   ...> from Songs sg, CreatedBy cb, People p where
   ...> sg.SongID=cb.Song_id AND cb.People_id=p.PeopleID AND sg.SongID IN (Select Song_id from CreatedBy where Song_id>0  group by Song_id having count
()>2);
SongName
------------
Into the Red
sqlite>
```

The query selects distinct (no duplicates) song names from the table (Songs*CreatedBy*People) which has more than 2 people working on it.

6. List all the people who are guitarists?

Select p.Name from
People p, Role r where
p.Role_id=r.RoleID AND r.RoleID=6;

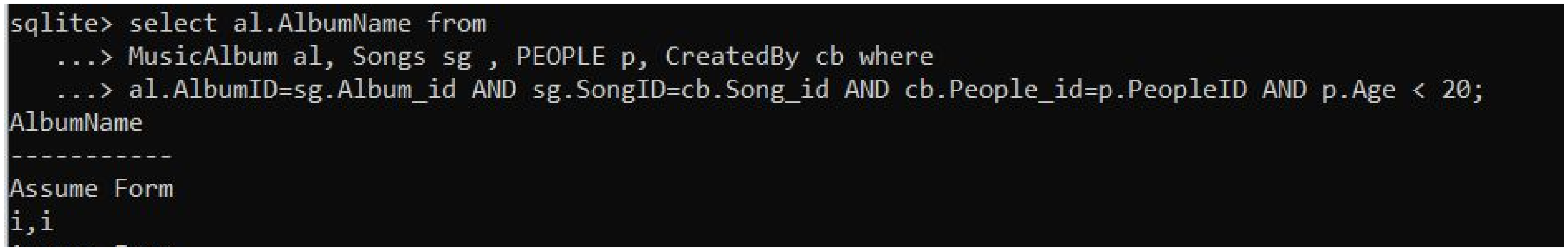


```
sqlite> Select p.Name from
   ...> People p, Role r where
   ...> p.Role_id=r.RoleID AND r.RoleID=6;
Name
----------
Aleksa
Harold
sqlite>
```

The query selects all the PeopleNames from the (People*Role) cartesian product who plays a role as a guitarist. Aleksa and Harold satisfy the condition.

7. Which Album has people of age less than 20 working in them?

select al.AlbumName from
MusicAlbum al, Songs sg , PEOPLE p, CreatedBy cb where
al.AlbumID=sg.Album_id AND sg.SongID=cb.Song_id AND cb.People_id=p.PeopleID AND p.Age < 20;

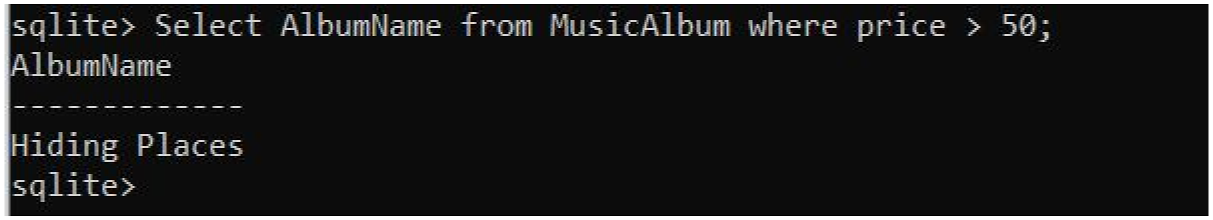


```
sqlite> select al.AlbumName from
   ...> MusicAlbum al, Songs sg , PEOPLE p, CreatedBy cb where
   ...> al.AlbumID=sg.Album_id AND sg.SongID=cb.Song_id AND cb.People_id=p.PeopleID AND p.Age < 20;
AlbumName
-----------
Assume Form
i,i
```

 The query selects all the album names that have people with age less than 20 working on them. The cartesian product can be seen here with (MusicAlbum * Songs * People * CreatedBy) from where the album names are retrieved given that the following condition is satisfied : al.AlbumID=sg.Album_id AND sg.SongID=cb.Song_id AND cb.People_id=p.PeopleID AND p.Age < 20;

8. List all the albums with price above 50?

Select AlbumName from MusicAlbum where price > 50;



```
sqlite> Select AlbumName from MusicAlbum where price > 50;
AlbumName
-------------
Hiding Places
sqlite>
```

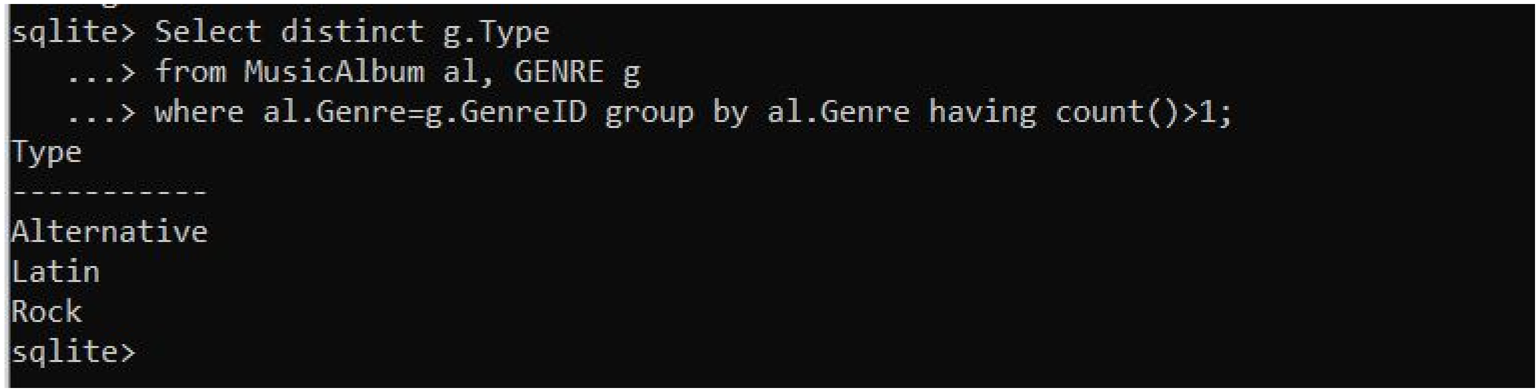
The query selects all the AlbumNames from the MusicAlbum table where the price is greater than 50. Hiding Places is the only one Album which is sold for more than 50 dollars.

9.Which Genre of Album is released more than once?

Select distinct g.Type

from MusicAlbum al, GENRE g

where al.Genre=g.GenreID group by al.Genre having count()>1;

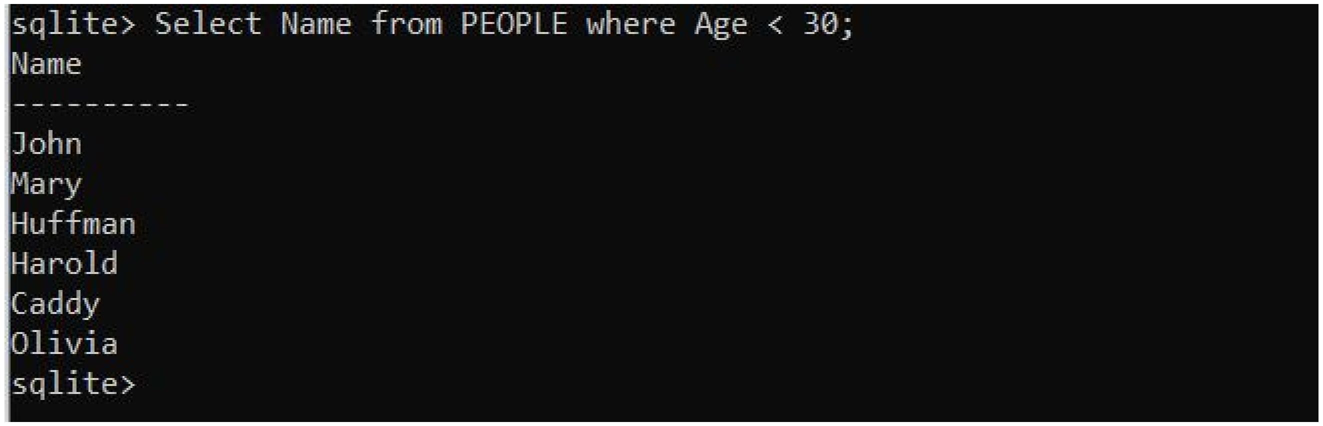


```
sqlite> Select distinct g.Type
   ...> from MusicAlbum al, GENRE g
   ...> where al.Genre=g.GenreID group by al.Genre having count()>1;
Type
-----------
Alternative
Latin
Rock
sqlite>
```

The query selects all the genres from the table (MusicAlbum*Genre) which has been released more than once. Alternative , Latin and Rock satisfies the condition

10. List all the People with age less than 30?

Select Name from PEOPLE where Age < 30;



```
sqlite> Select Name from PEOPLE where Age < 30;
Name
----------
John
Mary
Huffman
Harold
Caddy
Olivia
sqlite>
```

The query projects all the names of people from the table People where people are more than 30 years old. John, Mary, Huffman, Harold, Caddy and Olivia satisfy the condition.