**CS486/586 Introduction to Databases**

**Spring 2022 Quarter**

Assignment 2 – DDL & DML; SQL & Relational Algebra

Due: Friday, April 15th, 11:59PM on Canvas

**Submitted by: - Parth Parashar and Kirtan Patel**

**Part I – More SQL (50 points)**

Write a single SQL statement for each of the following queries. Show the first five rows of the result for each query (or fewer, if the result is smaller) and the number of rows returned.

**Question-1)** Different types of JOINs and SET operators

**(a)**  Find the team name for all teams with at least one agent who is skilled at Counterintelligence.

**(b)** List the team name for each team that has at least one agent who can speak Bengali and at least one agent who speaks Polish.

**Answer: -**

1. The query for this problem statement is given below: -

Select distinct t.name from team t join teamrel tr ON t.team\_id=tr.team\_id join skillrel sr ON tr.agent\_id=sr.agent\_id JOIN skill s ON sr.skill\_id=s.skill\_id where s.skill='Counterintelligence';

The screenshot below gives the result: -

Text

Description automatically generated

There are 21 rows in the result.

1. The query for this problem statement is given below: -

(SELECT DISTINCT t.name FROM team t JOIN teamrel tr ON t.team\_id = tr.team\_id JOIN languagerel lr ON tr.agent\_id = lr.agent\_id JOIN language l ON lr.lang\_id = l.lang\_id WHERE l.language = 'Bengali') INTERSECT (SELECT DISTINCT t.name FROM team t JOIN teamrel tr ON t.team\_id = tr.team\_id JOIN languagerel lr ON tr.agent\_id = lr.agent\_id JOIN language l ON lr.lang\_id = l.lang\_id WHERE l.language = 'Polish');

The result for the same is displayed below: -

Text

Description automatically generated with medium confidence

There are 29 rows in the result

**Question-2)**  Aggregation, Group by, Having

**(a)** Produce a list of the number of different skills that are had by members of each team. (Your result will be a list of teams and the number of skills had by members of each team.)

**(b)** For each language, list the number of different teams whose members know that language. (Your result will be a list of languages and the count of teams with at least one member who knows that language.)

**Answer: -**

1. The query for this problem statement is given below:-

select t.name, COUNT(DISTINCT s.skill\_id) from team t, skill s, teamrel tr, skillrel sr where t.team\_id=tr.team\_id AND tr.agent\_id=sr.agent\_id AND s.skill\_id=sr.skill\_id

GROUP BY t.name;

The result for this query is given below in the screenshot: -

Timeline

Description automatically generated

There are 42 rows in the result.

1. The query for this problem statement along with the screenshot of the result is given below.

Select l.language, count(t.name) from language l, team t, teamrel tr, languagerel lr where l.lang\_id=lr.lang\_id and tr.agent\_id=lr.agent\_id AND tr.team\_id=t.team\_id GROUP BY l.language;

The screenshot given below depicts the result of the given query.

Text

Description automatically generated with low confidence

There are 20 rows in this.

**Part II Table Creation, Population, and Constraints (50 pts)**

For the following exercises, you will be creating, modifying, and querying SQL tables. For each item, show the SQL you used and the resulting state (*all rows*) of your table(s) (or the error message that SQL returns). Do all these tasks using SQL statements (not a GUI). You will be using the data from the PDF file linked here: [CS486-586\_HW2\_MusicSrc.pdf](https://drive.google.com/file/d/1ekITDeFqed8Z97NR-Zn3CFA7Rfats2M2/view?usp=sharing)and posted in Week 3 in the class folder (adapted from wikipedia and bigfooty.com ).

**Question-3)** Create Table commands (various point values)

**(a)** Create a table called **Musicians** with columns for artist name, birthday, birth town, country of origin, Albums sold, studio albums, live albums, and gender

**a.** With artist name as the primary key

**b.**  Birthday is a date, and it should not allow null values.

**c.** Gender limited to "Male, Female, Non-binary"

**Answer: -** There are two ways in which we can solve this problem

1. By using Enum

The query used for getting the required result is given below: -

CREATE TYPE gender\_enum AS ENUM(‘Male’, ‘Female’, ‘Non-binary’);

CREATE TABLE musicians

(artist\_name TEXT PRIMARY KEY, birthday DATE NOT NULL, birthtown TEXT, country TEXT, albums\_sold INTEGER, studio\_albums INTEGER, live\_albums INTEGER, gender gender\_enum);

1. The second way to solve this problem is to use the CHECK constraint checker.

The query used for solving is given below: -

CREATE TABLE Musicians

(artist\_name TEXT PRIMARY KEY, birthday DATE NOT NULL, birthtown TEXT, country TEXT, albums\_sold INTEGER, studio\_albums INTEGER, live\_albums INTEGER, gender TEXT CHECK(gender IN('Male','Female','Non-binary')));

The screenshot depicting the successful creation of this table is given below: -

Text

Description automatically generated

**(b)**  Insert rows for all musicians with 10 or more Studio Albums

**Answer: -** The query along with the results is given below: -

Insert into musicians values

('David Gilmore', '3/9/1946', 'Cambridge', 'England', 230, 19, 5, 'Male'),

('Jimmy Page', '1/9/1944', 'Middlesex', 'England', 201, 14, 6, 'Male'),

('Beyonce', '9/4/1981', 'Houston, Tx', 'USA', 121, 10, 4, 'Female'),

('Freddy Mercury', '9/5/1946', 'Stone Town', 'Zanzibar', 238, 15, 10, 'Male'),

('Neil Young', '11/12/1945', 'Toronto, Ontario', 'Canada', 101, 45, 9, 'Male');

The screenshot depicting the successful insertion of the records is given below: -

Text

Description automatically generated

**(c)**  Modify your table to add columns for **Full Name**.

**Answer: -** The query along with the result in the screenshot is given below: -

ALTER TABLE musicians ADD full\_name TEXT;

Text

Description automatically generated

**(d)**  Update the existing rows in the table to add **Full Name** information.

**Answer: -** The queries along with the result in the screenshots is given below: -

🡪 UPDATE musicians SET full\_name = 'David Jon Gilmore' WHERE artist\_name='David Gilmore';

🡪UPDATE musicians SET full\_name = 'James Patrick Page' WHERE artist\_name='Jimmy Page';

🡪UPDATE musicians SET full\_name = 'Beyonce Giselle Knowles' WHERE artist\_name='Beyonce';

🡪UPDATE musicians SET full\_name = 'Farrokh Bulsara' WHERE artist\_name='Freddy Mercury';

🡪UPDATE musicians SET full\_name = 'Neil Percival Young' WHERE artist\_name='Neil Young';

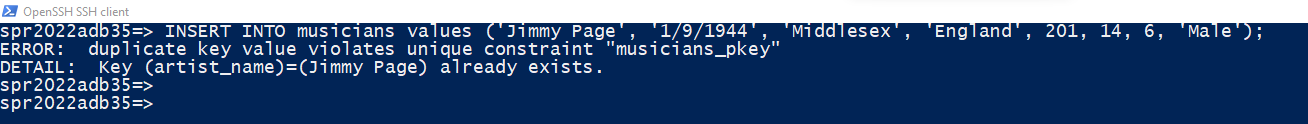
Text

Description automatically generated

**(e)** What happens if you try to insert **Jimmy Page** a second time?

**Answer: -** When we try to insert Jimmy Page tuple again, we will receive an error message conveying that we are trying to enter a duplicate primary key.

This can be verified by the screenshot given below: -



**(f)** Create a second table called **genre** to hold the list of available genres, with a single column, called genre, where genre is unique.

**Answer: -** The query for creating a table genre is given below: -

create table genre(genre TEXT UNIQUE);

Timeline

Description automatically generated

**(g)** Insert rows in the second table corresponding to all possible genres.

**Answer: -**

The query for inserting multiple rows in the genre table along with the screenshot for the insertion is given below: -

INSERT INTO genre values ('Psychedelic Rock'), ('Blues'), ('Rock'), ('Folk'), ('Hard Rock'), ('R&B'), ('Pop'), ('Hip Hop'), ('Country Rock'), ('Reggae');

Text

Description automatically generated

**(h)** Create a third table called **genrerel** to hold each musician’s genres, with columns for musician name and genre, with musician name as a foreign key to the first table, and genre a foreign key to the second table

**Answer: -**

The query for the insertion as well as screenshot depicting the successful insertion is given below: -

CREATE TABLE genrerel (artist\_name TEXT, genre TEXT, CONSTRAINT fk\_musician FOREIGN KEY(artist\_name) REFERENCES musicians(artist\_name), CONSTRAINT fk\_genre FOREIGN KEY(genre) REFERENCES genre(genre));

Text, timeline

Description automatically generated

**(i)**  Insert rows in the third table corresponding to all musicians in the first table. For musicians with multiple genres, each genre should be listed separately.

**Answer: -** The queries for the insertions as well as the screenshot depicting the successful insertion is given below: -

🡪INSERT INTO genrerel values ('David Gilmore', 'Psychedelic Rock'), ('David Gilmore', 'Blues'), ('David Gilmore', 'Rock');

🡪INSERT INTO genrerel values ('Jimmy Page', 'Rock'), ('Jimmy Page', 'Blues'), ('Jimmy Page', 'Folk'), ('Jimmy Page', 'Hard Rock');

🡪INSERT INTO genrerel values ('Beyonce', 'R&B'), ('Beyonce', 'Pop'), ('Beyonce', 'Hip Hop');

🡪INSERT INTO genrerel values ('Freddy Mercury', 'Rock');

🡪INSERT INTO genrerel values ('Neil Young', 'Rock'), ('Neil Young', 'Folk'), ('Neil Young', 'Hard Rock'), ('Neil Young', 'Country Rock');

Text

Description automatically generated with medium confidence

(j) What happens if you try to insert **Dance** as a genre for **Beyonce?**

**Answer: -** When we try to insert dance as a genre for Beyonce, it gives an error. This is because it is violating the foreign key constraint. Hence the insertion would be unsuccessful. This can also be seen from the screenshot below: -

INSERT INTO genrerel values ('Beyonce', 'Dance');

Text

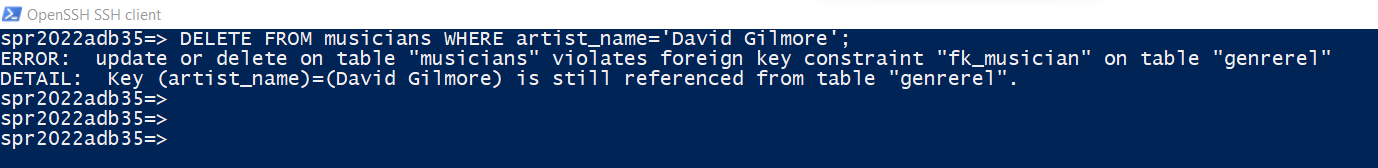
Description automatically generated

(k) What happens if you try to delete the row in the first table for **David Gilmore**?

**Answer: -** When we try to delete the tuple David Gilmore, then we will receive the error for constraint violation. This is because the referenced value from another table cannot be deleted until and unless reference is also deleted.

The query and the associated result (in the form of a screenshot) is given below: -

DELETE FROM musicians WHERE artist\_name='David Gilmore';



(l) Write a query to find the total sales amount for all Folk musicians from England and Canada.

**Answer: -** The query along with the screenshot of the result is attached below: -

SELECT SUM(m.albums\_sold) FROM musicians m, genrerel gr WHERE gr.genre = 'Folk' AND m.artist\_name = gr.artist\_name AND (m.country = 'England' OR m.country = 'Canada');

Text

Description automatically generated