

Name: Chitradevi Maruthavanan

CS 486/586 Introduction to DBMS

Summer 2022

Homework Assignment 2 – DDL & DML; SQL & Relational Algebra

Part One

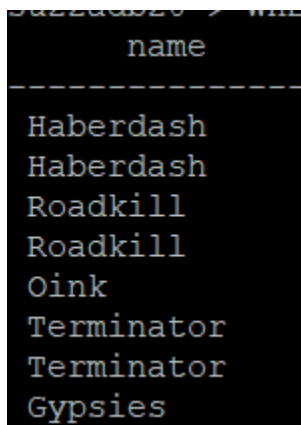
Write a single SQL statement for each of the following queries. Show the first five rows of the result for each query (or fewer, if the result is smaller) and the number of rows returned.

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 Charisma.

Answer:

```
SELECT t.name FROM team t
INNER JOIN teamrel tl ON t.team_id=tl.team_id
INNER JOIN skillrel sl ON tl.agent_id=sl.agent_id
INNER JOIN skill s ON sl.skill_id=s.skill_id
WHERE s.skill='Charisma';
```

A screenshot of a SQL query result set. The first column is labeled 'name'. The results are: Haberdash, Haberdash, Roadkill, Roadkill, Oink, Terminator, Terminator, and Gypsies. The text is displayed on a dark background with a light border around the table.

name
Haberdash
Haberdash
Roadkill
Roadkill
Oink
Terminator
Terminator
Gypsies

Number of rows returned = 19 rows

(b) List the team name for each team that weekly meets and has at least one agent who can speak French and at least one agent who speaks Malay

SELECT t.name

Answer:

```
SELECT t.name FROM team t
INNER JOIN teamrel tl ON t.team_id=tl.team_id
INNER JOIN languagerel lr ON tl.agent_id=lr.agent_id
INNER JOIN language l ON lr.lang_id=l.lang_id
WHERE l.language='French'

INTERSECT

SELECT t.name FROM team t
INNER JOIN teamrel tl ON t.team_id=tl.team_id
INNER JOIN languagerel lr ON tl.agent_id=lr.agent_id
INNER JOIN language l ON lr.lang_id=l.lang_id
WHERE l.language='Malay';
```

```
U22adb20-> WHERE
          name
-----
Roadkill
F Sharp
Vikings
ShowBiz
Blaster
Renegade
Jester
Failsafe
Widow Makers
Charley Hunter
```

Number of rows returned = 24 rows

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.)

Answer:

```
SELECT t.name, count(s.skill) AS skill FROM skill s
INNER JOIN skillrel sr ON s.skill_id=sr.skill_id
INNER JOIN teamrel tr ON sr.agent_id=tr.agent_id
INNER JOIN team t ON tr.team_id=t. team_id
GROUP BY t.name;
```

name	skill
Roadkill	26
F Sharp	32
Camaro	28
Ghost Hunters	28
Vikings	37
ShowBiz	26
Thunderbird	29
Blaster	21
Renegade	33
Failsafe	27
Jester	29
Boat Team 6	24
Widow Makers	29
Charley Hunter	28

Number of rows returned: 42 rows

- (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:

```

SELECT l.language,count(t.name) AS teams FROM language l
INNER JOIN languagerel lr ON l.lang_id=lr.lang_id
INNER JOIN teamrel tr ON lr.agent_id=tr.agent_id
INNER JOIN team t ON tr.team_id=t.team_id
GROUP BY l.language;

```

language	teams
Malay	62
Turkish	56
German	64
Cherokee	52
Chinese	46
Pashtu	55
Japanese	60
Spanish	60
Vietnamese	56
Arabic	82

No of rows returned: 20 rows

Part II Table Creation, Population, and Constraints

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](#) and posted in Week 3 in the class folder (adapted from wikipedia and bigfooty.com).

3) Create Table commands

(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.

And provide a screenshot of your created table.

Answer:

```
CREATE TABLE musicians (Artist_Name varchar(200), Birthday Date Not
null, Birth_Town varchar(200), Country_Of_Origin varchar(255),
Albums_Sold int, Studio_Albums int, live_album int,
Gender varchar(255), Primary key(artist_name));
```

artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_album	gender
-----	-----	-----	-----	-----	-----	-----	-----

(b) Insert rows for all musicians with 10 or more Studio Albums. Provide a screenshot of the updated table.

Answer:

```
INSERT INTO musicians VALUES ('David_Gilmore', '3/6/1946', 'Cambridge',
'England',230, 19, 5, 'Male');
```

```
INSERT INTO musicians VALUES ('Jimmy_Page', '1/9/1944', 'Middlesex',
'England', 201,14, 6, 'Male');
```

```
INSERT INTO musicians VALUES ('Jimi_Hendrix', '11/27/1942',
'Seattle,Washington', 'USA', 15, 3, 2,'Male');
```

```
INSERT INTO musicians VALUES ('Beyonce', '9/4/1981', 'Houston,Tx', 'USA',
121, 10, 4,'Female');
```

```
INSERT INTO musicians VALUES ('Ariana_Grande','6/26/1993',
'Boca_Raton,Florida', 'USA', 27, 5, 0, 'Female');
```

```
INSERT INTO musicians VALUES ('Drake', '10/24/1986', 'Toronto,Ontario',
'Canada', 57, 5, 0, 'Male');
```

```
INSERT INTO musicians VALUES ('Freddy_Mercury', '9/5/1946',
'Stone_Town', 'Zanzibar', 238, 15, 10, 'Male');
```

```
INSERT INTO musicians VALUES ('Neil_young', '11/12/1945',
'Toronto,Ontario', 'Canada', 101, 45, 9, 'Male');
```

```
INSERT INTO musicians VALUES ('Rihanna', '2/20/1988', 'Saint_Michael',
'Barbados', 87, 8, 0,'Female');
```

```
Insert into musicians values ('Miley_Cyrus', '11/23/1992', 'Franklin,
Tennessee', 'USA', 39, 6, 1,'Non-Binary');
```

artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_album	gender
David_Gilmore	1946-03-06	Cambridge	England	230	19	5	Male
Jimmy_Page	1944-01-09	Middlesex	England	201	14	6	Male
Jimi_Hendrix	1942-11-27	Seattle,Washington	USA	15	3	2	Male
Beyonce	1981-09-04	Houston,Tx	USA	121	10	4	Female
Ariana_Grande	1993-06-26	Boca_Raton,Florida	USA	27	5	0	Female
Drake	1986-10-24	Toronto,Ontario	Canada	57	5	0	Male
Freddy_Mercury	1946-09-05	Stone_Town	Zanzibar	238	15	10	Male
Neil_young	1945-11-12	Toronto,Ontario	Canada	101	45	9	Male
Rihanna	1988-02-20	Saint_Michael	Barbados	87	8	0	Female
Miley_Cyrus	1992-11-23	Franklin, Tennessee	USA	39	6	1	Non-Binary

(10 rows)

(c) Modify your table to add columns for Full Name. Provide a screenshot of the updated table.

Answer:

```
ALTER TABLE musicians ADD Full_Name varchar(255);
```

artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_album	gender	full_name
David_Gilmore	1946-03-06	Cambridge	England	230	19	5	Male	
Jimmy_Page	1944-01-09	Middlesex	England	201	14	6	Male	
Jimi_Hendrix	1942-11-27	Seattle,Washington	USA	15	3	2	Male	
Beyonce	1981-09-04	Houston,Tx	USA	121	10	4	Female	
Ariana_Grande	1993-06-26	Boca_Raton,Florida	USA	27	5	0	Female	
Drake	1986-10-24	Toronto,Ontario	Canada	57	5	0	Male	
Freddy_Mercury	1946-09-05	Stone_Town	Zanzibar	238	15	10	Male	
Neil_young	1945-11-12	Toronto,Ontario	Canada	101	45	9	Male	
Rihanna	1988-02-20	Saint_Michael	Barbados	87	8	0	Female	
Miley_Cyrus	1992-11-23	Franklin, Tennessee	USA	39	6	1	Non-Binary	

(10 rows)

(d) Update the existing rows in the table to add Full Name information. Provide a screenshot of the updated table.

Answer:

```
UPDATE musicians SET Full_Name = 'David Jon Gilmour' WHERE
artist_name= 'David_Gilmore';
```

```
UPDATE musicians SET full_name = 'James Patrick Page' WHERE
artist_name ='Jimmy_Page';
```

```
UPDATE musicians SET full_name = 'Johnny Allen Hendrix' WHERE  
artist_name = 'Jimi_Hendrix';
```

```
UPDATE musicians SET full_name = 'beyonce giselle knowes' WHERE  
artist_name = 'Beyonce';
```

```
UPDATE musicians SET full_name = 'Ariana Grande-Butera' WHERE  
artist_name = 'Ariana_Grande ';
```

```
UPDATE musicians SET full_name = 'Aubrey Drake Graham' WHERE  
artist_name = 'Drake';
```

```
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';
```

```
UPDATE musicians SET full_name = 'Robyn Rihanna Fenty' WHERE  
artist_name = 'Rihanna';
```

```
UPDATE musicians SET full_name = 'Miley Ray Hemsworth' WHERE  
artist_name = 'Miley_Cyrus';
```

artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_album	gender	full_name
David Gilmore	1946-03-06	Cambridge	England	230	19	5	Male	David Jon Gilmour
Jimmy_Page	1944-01-09	Middlesex	England	201	14	6	Male	James Patrick Page
Jimi_Hendrix	1942-11-27	Seattle, Washington	USA	15	3	2	Male	Johnny Allen Hendrix
Beyonce	1981-09-04	Houston, Tx	USA	121	10	4	Female	beyonce giselle knowes
Ariana_Grande	1993-06-26	Boca Raton, Florida	USA	27	5	0	Female	Ariana Grande-Butera
Freddy_Mercury	1946-09-05	Stone Town	Zanzibar	238	15	10	Male	Farrokh Bulsara
Rihanna	1988-02-20	Saint Michael	Barbados	87	8	0	Female	Robyn Rihanna Fenty
Drake	1986-10-24	Toronto, Ontario	Canada	57	5	0	Male	Aubrey Drake Graham
Miley_Cyrus	1992-11-23	Franklin, Tennessee	USA	39	6	1	Non-Binary	Miley Ray Hemsworth
Neil_young	1945-11-12	Toronto, Ontario	Canada	101	45	9	Male	Neil Percival Young
(10 rows)								

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

Answer

```
INSERT INTO musicians VALUES ('Jimmy_Page', '1/9/1944', 'Middlesex',  
'England', 201, 14, 6, 'Male');
```

When I will try to insert Jimmy Page a second time in musician table, the error will be:

```
su22adb20=> INSERT INTO musicians VALUES ('Jimmy_Page', '1/9/1944', 'Middlesex', 'England', 201,14, 6, 'Male');  
ERROR: duplicate key value violates unique constraint "musicians_pkey"  
DETAIL: Key (artist_name)=(Jimmy_Page) already exists.
```

(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

```
CREATE TABLE genre(genre varchar(250) UNIQUE);
```

```
su22adb20=> select * from genre;  
genre  
-----  
(0 rows)
```

(g) Insert rows in the second table corresponding to all possible genres. Provide a screenshot of the updated table.

Answer

```
INSERT INTO genre (genre) VALUES ('Psychedelic Rock');
```

```
INSERT INTO genre (genre) VALUES ('Hard Rock');
```

```
INSERT INTO genre (genre) VALUES ('R&B');
```

```
INSERT INTO genre (genre) VALUES ('Pop');
```

```
INSERT INTO genre (genre) VALUES ('Hip Hop');
```

```
INSERT INTO genre (genre) VALUES ('Reggae');
```

```
INSERT INTO genre (genre) VALUES ('Blues');
```


INSERT INTO genre (genre) VALUES ('Rock');

INSERT INTO genre (genre) VALUES ('Folk');

INSERT INTO genre (genre) VALUES ('Country Rock');

```
su22adb20=> select * from genre;
            genre
-----
Psychelic Rock
Hard Rock
R&B
Pop
Hip Hop
Reggae
Blues
Rock
Folk
Country Rock
(10 rows)
```

(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

```
CREATE TABLE genrerel (musicians_name varchar(255), genre varchar(255),
Foreign key (musicians_name) REFERENCES musicians(artist_name),
FOREIGN KEY(genre) REFERENCES genre(genre));
```

```
su22adb20=> select * from genrerel;
musicians_name | genre
-----+-----
(0 rows)
```

(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. Provide a screenshot of the updated table.

Answer

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('Ariana_Grande',  
'Country Rock');  
INSERT INTO genrerel (musicians_name, genre) VALUES  
('Ariana_Grande', 'Reggae');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Beyonce', 'R&B');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Beyonce', 'Pop');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Beyonce', 'Hip Hop');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimi_Hendrix', 'Folk');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimi_Hendrix', 'Pop');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Rihanna', 'Hip Hop');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Rihanna', 'R&B');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Rihanna', 'Pop');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Rihanna', 'Reggae');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Drake', 'Blues');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Neil_young', 'Rock');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Neil_young', 'Folk');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Neil_young', 'Hard  
Rock');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Neil_young', 'Country  
Rock');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Freddy_Mercury',  
'Rock');  
INSERT INTO genrerel (musicians_name, genre) VALUES ('Miley_Cyrus', 'Hip  
Hop');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimmy_Page', 'Rock');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimmy_Page', 'Blues');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimmy_Page', 'Folk');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('Jimmy_Page', 'Hard  
Rock');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('David_Gilmore', 'Rock');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('David_Gilmore',  
'Psychedelic Rock');
```

```
INSERT INTO genrerel (musicians_name, genre) VALUES ('David_Gilmore', 'Blues');
```

musicians_name	genre
Beyonce	R&B
Beyonce	Pop
Beyonce	Hip Hop
Rihanna	Hip Hop
Rihanna	R&B
Rihanna	Pop
Rihanna	Reggae
Freddy_Mercury	Rock
Ariana_Grande	Reggae
Ariana_Grande	Country Rock
Jimi_Hendrix	Folk
Jimi_Hendrix	Pop
Drake	Blues
Neil_young	Rock
Neil_young	Folk
Neil_young	Hard Rock
Freddy_Mercury	Rock
Miley_Cyrus	Hip Hop
David_Gilmore	Rock
David_Gilmore	Psychedelic Rock
David_Gilmore	Blues
Jimmy_Page	Rock
Jimmy_Page	Blues
Jimmy_Page	Folk
Jimmy_Page	Hard Rock
25 rows)	

(j) Write a query to insert Dance as a genre for Beyonce, what happens?

Answer

```
INSERT INTO genrerel (musicians_name,genre) VALUES ('Beyonce','Dance');
```

When I will try to insert Dance as a genre for Beyonce, the below error will be:

```
su22adb20=> INSERT INTO genrerel(musicians_name,genre) VALUES ('Beyonce','Dance');
ERROR: insert or update on table "genrerel" violates foreign key constraint "genrerel_genre_fkey"
DETAIL: Key (genre)=(Dance) is not present in table "genre".
su22adb20=>
```

(k) Write a query to delete the row in the first table for David Gilmore, what happens?

Answer

```
DELETE FROM musicians WHERE artist_name = 'David_Gilmore';
```

When I will try to Delete the row in the first table for David Gilmore, the error will be there:

```
ERROR: update or delete on table "musicians" violates foreign key constraint "genrerel_musicians_name_fkey" on table "genrerel"
DETAIL: Key (artist_name)=(David Gilmore) is still referenced from table "genrerel".
su22adb20=>
```

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

Answer:

```
SELECT sum(albums_sold) AS total_sales_England_Canada FROM musicians m
WHERE country_of_origin = 'England' OR country_of_origin = 'Canada';
```

```
total_sales_england_canada
-----
                           589
(1 row)
```