Homework 2 - DDL & DML; SQL & Relational Algebra

Sai Teja Garlapati (PSU-ID:985031204)
Anvitha Pasumarthi(PSU-ID:956236565)

1)
a)
πfirst,last(σsc_level<3(agent⋈clearance_id=sc_id securityclearance))

b)
πagent.agent_id, agent.first, agent.last(σsalary>50800 and sc_level=’Secret’ (agent⋈clearance_id=sc_id securityclearance))

c)
πfirst,last(σcountry=’Spain’ and skill=’Communications’ (agent⋈agent_id=agent_id skillrel⋈skill_id=skill_id Skill))

2)a)
```
SELECT name FROM team
INNER JOIN teamrel ON teamrel.team_id = team.team_id
INNER JOIN skillrel ON skillrel.agent_id = teamrel.agent_id
INNER JOIN skill ON skillrel.skill_id = skill.skill_id AND skill.skill = 'Biologist' GROUP By name;
```

Output:

name
Leadphut
Oink
Camaro
Ghost Hunters
Vikings
Cha Cha Cha
Thunderbird
Cyclone
Renegade
Jester
Failsafe
Giraffe
Blackout
BumbleBee
Widow Makers
Boat Team 3
Blue Dagger
SqueakyClean
Terminator

19 row(s)

Total runtime: 5.942 ms

SQL executed.

b)

```
 (SELECT T.name
FROM team T inner join teamrel TR on T.team_id=TR.team_id
Inner join languagerel LR on TR.agent_id=LR.agent_id
Inner join language L on LR.lang_id=L.lang_id
WHERE L.language='German')
intersect
(SELECT T.name
FROM team T inner join teamrel TR on T.team_id=TR.team_id
Inner join languagerel LR on TR.agent_id=LR.agent_id
Inner join language L on LR.lang_id=L.lang_id
WHERE L.language='Arabic' ) ;
```

Output :

name
F Sharp
Camaro
Ghost Hunters
Vikings
ShowBiz
Thunderbird
Renegade
Jester
Boat Team 6
Widow Makers
Charley Hunter
Boat Team 3
Blunt
Leadphut
Boat Team 7
Oink
Timebomb
Beasties
Boat Team 1
Wired
SpecialForces
Rimspeed
Cha Cha Cha
Gypsies
Haberdash
Giraffe
BumbleBee
Scorpion
Boat Team 4
Rolaids
Blue Dagger

31 row(s)

Total runtime: 6.207 ms

SQL executed.

3)
a) select t.name,count(sr.skill_id) as Number_of_skills from team t
INNER JOIN teamrel tr on t.team_id=tr.team_id
INNER JOIN agent a ON tr.agent_id=a.agent_id
INNER JOIN skillrel sr ON a.agent_id=sr.agent_id group by t.name;
Output:

name	number_of_skills
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
Boat Team 3	23
Boat Team 2	31
SqueakyClean	22
Spoiler	2
Leadphut	22
Blunt	30
Boat Team 7	20
Oink	26
Timebomb	18
Beasties	33
Boat Team 1	28
Swing Voters	21
Wired	12
SpecialForces	29
Rimspeed	31
Cha Cha Cha	32
Cyclone	31
Gypsies	26
Haberdash	29
Giraffe	23
BumbleBee	24
Blackout	18
Scorpion	20
Boat Team 4	26
Rolaids	29
Blue Dagger	23
Terminator	37
FlyOnTheWall	29

42 row(s)

Total runtime: 5.589 ms

SQL executed.

b)
select count(distinct tr.team_id) as Number_Of_Teams, l.language from teamrel tr INNER JOIN agent a ON tr.agent_id=a.agent_id INNER JOIN languagerel lr ON a.agent_id=lr.agent_id INNER JOIN language l ON lr.lang_id=l.lang_id group by l.language;
Output :

number_of_teams	language
38	Arabic
31	Bengali
33	Cherokee
28	Chinese
6	English
34	Farsi
32	French
34	German
29	Hebrew
34	Hindi
31	Japanese
32	Korean
31	Malay
29	Pashtu
38	Polish
34	Portuguese
34	Russian
33	Spanish
34	Turkish
32	Vietnamese

20 row(s)

Total runtime: 6.337 ms

SQL executed.

4)
a) CREATE TYPE gender AS ENUM ('Male', 'Female', 'Non-Binary');

CREATE TABLE Musicians (artist_name
birthday
birth_town country_of_origin albums_sold studio_albums live_albums

VARCHAR(150), Date NOT NULL, VARCHAR(100),

VARCHAR(100), INTEGER, INTEGER, INTEGER,

gender
PRIMARY KEY (artist_name));

Output :

Column	Type	Not Null	Default	Constraints	Actions				Comment
artist_name	character varying(150)	NOT NULL			Browse	Alter	Privileges	Drop	
birthday	date	NOT NULL			Browse	Alter	Privileges	Drop	
birth_town	character varying(100)				Browse	Alter	Privileges	Drop	
country_of_origin	character varying(100)				Browse	Alter	Privileges	Drop	
albums_sold	integer				Browse	Alter	Privileges	Drop	
studio_albums	integer				Browse	Alter	Privileges	Drop	
live_albums	integer				Browse	Alter	Privileges	Drop	
gender	gender				Browse	Alter	Privileges	Drop	

b)

INSERT INTO Musicians (artist_name , birthday , birth_town , country_of_origin , albums_sold , studio_albums , live_albums , gender)

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

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

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

('David Gilmore', '3/6/1946', 'Cambridge', 'England', 230, 19, 5,'Male'),

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

Output :

artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_albums	gender
Freddy Mercury	1946-09-05	Stone Town	Zanzibar	238	15	10	Male
David Gilmore	1946-03-06	Cambridge	England	230	19	5	Male
Beyonce	1981-09-04	Houston,Tx	USA	121	10	4	Female
Neil Young	1945-11-12	Toronto, Ontario	Canada	101	45	9	Male
Jimmy Page	1944-01-09	Middlesex	England	201	14	6	Male

5 row(s)

Total runtime: 1.153 ms

SQL executed.

c)
ALTER Table Musicians Add full_name VARCHAR(255)

Output :

Column	Type	Not Null	Default	Constraints
artist_name	character varying(150)	NOT NULL		
birthday	date	NOT NULL		
birth_town	character varying(100)			
country_of_origin	character varying(100)			
albums_sold	integer			
studio_albums	integer			
live_albums	integer			
gender	gender			
full_name	character varying(255)			

d)
UPDATE Musicians SET Full_name = 'David Jon Gilmour' WHERE artist_name = 'David Gilmore';
UPDATE Musicians SET Full_name = 'Neil Percival Young' WHERE artist_name = 'Neil Young';
UPDATE Musicians SET Full_name = 'Beyoncé Giselle Knowles' WHERE artist_name = 'Beyonce';
UPDATE Musicians SET Full_name = 'James Patrick Page' WHERE artist_name = 'Jimmy Page';
UPDATE Musicians SET Full_name = 'Farrokh Bulsara' WHERE artist_name = 'Freddy Mercury';

Output :

Actions	artist_name	birthday	birth_town	country_of_origin	albums_sold	studio_albums	live_albums	gender	full_name
Edit Delete	David Gilmore	1946-03-06	Cambridge	England	230	19	5	Male	David Jon Gilmour
Edit Delete	Neil Young	1945-11-12	Toronto, Ontario	Canada	101	45	9	Male	Neil Percival Young
Edit Delete	Beyonce	1981-09-04	Houston,Tx	USA	121	10	4	Female	Beyoncé Giselle Knowles
Edit Delete	Jimmy Page	1944-01-09	Middlesex	England	201	14	6	Male	James Patrick Page
Edit Delete	Freddy Mercury	1946-09-05	Stone Town	Zanzibar	238	15	10	Male	Farrokh Bulsara

5 row(s)

e)
INSERT INTO Musicians (artist_name , birthday , birth_town , country_of_origin , albums_sold , studio_albums , live_albums , gender)

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

Output :

SQL error:

```
ERROR:  syntax error at or near "'Jimmy Page'"
LINE 2: ('Jimmy Page', '1/9/1944', 'Middlesex', 'England', 201, 14, ...
         ^
```

In statement:
INSERT INTO Musicians (artist_name , birthday , birth_town , country_of_origin , albums_sold , studio_albums , live_albums , gender)
('Jimmy Page', '1/9/1944', 'Middlesex', 'England', 201, 14, 6,'Male')

Total runtime: 0.733 ms

SQL executed.

f)
CREATE TABLE Genre (genre VARCHAR(100) UNIQUE);

Output :

Column	Type	Not Null	Default	Constraints	Actions				Comment
genre	character varying(100)			1	Browse	Alter	Privileges	Drop	

Browse | Select | Insert | Empty | Drop | Add column | Alter

g)
INSERT INTO Genre VALUES ('Psychedelic Rock');
INSERT INTO Genre VALUES ('R&B');
INSERT INTO Genre VALUES ('Pop');
INSERT INTO Genre VALUES ('Hard Rock');
INSERT INTO Genre VALUES ('Blues');
INSERT INTO Genre VALUES ('Rock');
INSERT INTO Genre VALUES ('Country Rock');
INSERT INTO Genre VALUES ('Reggae');
INSERT INTO Genre VALUES ('Folk');
INSERT INTO Genre VALUES ('Hip Hop');

Output :

Actions		genre
Edit	Delete	Psychedelic Rock
Edit	Delete	R&B
Edit	Delete	Pop
Edit	Delete	Hard Rock
Edit	Delete	Blues
Edit	Delete	Rock
Edit	Delete	Country Rock
Edit	Delete	Reggae
Edit	Delete	Folk
Edit	Delete	Hip Hop

10 row(s)

h)

CREATE Table genrerel (artist_name VARCHAR(150), genre VARCHAR(100),
FOREIGN KEY (artist_name) REFERENCES Musicians(artist_name),
FOREIGN KEY (genre) REFERENCES genre(genre));

Output :

Column	Type	Not Null	Default	Constraints	Actions				Comment
artist_name	character varying(150)				Browse	Alter	Privileges	Drop	
genre	character varying(100)				Browse	Alter	Privileges	Drop	

i)

INSERT INTO genrerel (artist_name, genre) VALUES ('David Gilmore', 'Psychedelic Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('David Gilmore', 'Blues');
INSERT INTO genrerel (artist_name, genre) VALUES ('David Gilmore', 'Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Jimmy Page', 'Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Jimmy Page', 'Blues');
INSERT INTO genrerel (artist_name, genre) VALUES ('Jimmy Page', 'Folk');
INSERT INTO genrerel (artist_name, genre) VALUES ('Jimmy Page', 'Hard Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Beyonce', 'R&B');
INSERT INTO genrerel (artist_name, genre) VALUES ('Beyonce', 'Pop');
INSERT INTO genrerel (artist_name, genre) VALUES ('Beyonce', 'Hip Hop');
INSERT INTO genrerel (artist_name, genre) VALUES ('Freddy Mercury', 'Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Neil Young', 'Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Neil Young', 'Folk');
INSERT INTO genrerel (artist_name, genre) VALUES ('Neil Young', 'Hard Rock');
INSERT INTO genrerel (artist_name, genre) VALUES ('Neil Young', 'Country Rock');

Output :

```
SELECT * FROM "fall2021db40"."genrerel";
```

Submit

artist_name	genre
David Gilmore	Psychedelic Rock
David Gilmore	Blues
David Gilmore	Rock
Jimmy Page	Rock
Jimmy Page	Blues
Jimmy Page	Folk
Jimmy Page	Hard Rock
Beyonce	R&B
Beyonce	Pop
Beyonce	Hip Hop
Freddy Mercury	Rock
Neil Young	Rock
Neil Young	Folk
Neil Young	Hard Rock
Neil Young	Country Rock

15 row(s)

j) INSERT INTO genrerel (artist_name, genre) VALUES ('Beyonce', 'Dance');

Output :

SQL error:

```
ERROR:  insert or update on table "genrerel" violates foreign key constraint "genrerel_genre_fkey"
DETAIL:  Key (genre)=(Dance) is not present in table "genre".
```

In statement:
INSERT INTO genrerel (artist_name, genre) VALUES ('Beyonce', 'Dance');

Total runtime: 2.468 ms

SQL executed.

Edit SQL

k) DELETE FROM Musicians WHERE artist_name = ‘David Gilmore’ ;

Output :

SQL error:

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

In statement:
DELETE FROM Musicians WHERE artist_name = 'David Gilmore' ;

Total runtime: 2.637 ms

SQL executed.

Here while trying to delete artist_name David Gilmore from the table, it violates the foreign key constraint as shown above.

l) SELECT SUM(albums_sold) FROM Musicians m JOIN genrerel gr ON (m.artist_name = gr.artist_name AND genre = 'Folk' AND country_of_origin IN ('England', 'Canada'));

Output :

sum
604

1 row(s)

Total runtime: 3.110 ms

SQL executed.

m) SELECT artist_name, MAX (studio_albums + live_albums) AS combination
FROM Musicians GROUP by artist_name;

Output :

artist_name	combination
Jimmy Page	20
Neil Young	54
David Gilmore	24
Beyonce	14
Freddy Mercury	25

5 row(s)

Total runtime: 2.251 ms

SQL executed.