**Course Project Work - Big Teeth Reality TV**

By Group 5

Jenish Dhaduk, Ashlea PIN-JUI Huang, Htein Lynn Thar, Diem Nguyen

December 12, 2019

**Distribution Of the project work among group members:**

Each group member was responsible for their form. There were four forms and each group member was assigned a Form. First, we started by creating a table that has all the necessary entities for each form. Then we went to performing normalization step by step.

*Diem Nguyen (989353845)*

Created a list of normalized tables after performing normalization to the main table for APPLICATION FORM. The list contained 8 tables. She turned out to be an excellent co-partner to work on ERD as she provides some detailed insights into creating ERD. Also, provide her expertise on database operations in achieving the team’s goal.

*Jenish Dhaduk (989354198)*

Created a list of normalized tables after performing normalization to the main table for VOTING FORM. The list contained 4 tables.

He turned out to be an excellent co-partner to took charge of the database. When we had all the tables ready. He helped to fill the database with all tables and also populated the data into Database with the help of Htein Lynn Thar.

*Ashlea PIN-JUI Huang (989357938)*

Created a list of normalized tables after performing normalization to the main table for EVENTS FORM. The list 9 contained tables.

Ashlea did a great job working on ERD and creating list of normalized tables. While creating queries as well she turned out to be an excellent co-partner.

*Htein Lynn Thar (989357674)*

Created a list of normalized tables after performing normalization to the main table for BACKGROUND CHECK FORM. The list 4 contained tables.

Htein Lynn Thar worked with Jenish Dhaduk on database operations. He actually did really well with inserting relevant data to the database.

Apart from the assigned work all team members also collaborated together whenever it was necessary. For establishing relationships among tables. We all got together as it was required to understand how to connect each table that were created separately.

Preparing for the presentation, and many such instances that required a collaborative approach, All team members gave equal justice to their role in achieving the completion of the project.

**Exercises**

**1.create a normalized list of tables for each of the above forms.**

**LIST OF TABLES FOR EACH FORM AFTER NORMALIZATION**

The following forms are the main forms will be using in the project including application form, background check form, events form, and voting form. After reviewing carefully the forms, we indicate needed tables as well as identify the primary keys and foreign keys for each table.

**APPLICATION FORM**

1. PRODUCER
2. DIRECTOR
3. CONTESTANT
4. MEDICATIONS
5. JOBS
6. ADDRESS
7. RATINGS
8. CONTESTANT\_GENDER

**BACKGROUND CHECK FORM**

1. EVALUATION
2. EMPLOYER
3. EDUCATION
4. JUDICIAL

**EVENTS FORM**

1. EPISODE
2. EVENTS
3. DANGER\_LEVEL
4. EVENT\_ESTIMATED\_TIME
5. DIRECTION\_SETUP
6. SCENE\_TIME
7. TEAMS
8. TASKS
9. POINTS

**VOTING FORM**

1. LOCATION
2. VOTES
3. METHODS
4. METHODS\_ID

**SQL QUERIES FOR CREATING TABLES**

We then query to create tables so that we can establish our database and get ready for the next steps. The total of tables is 25 including some main tables such as Producers, Directors, Contestants, Ratings. Points and so on. We carefully indicate primary keys and foreign keys to ensure the relationships between tables are accurate which supportively ensure the good data flow between tables so that our next step to dig into the data and normalize them will include no obstacles. We cautiously select the entities for each table so that nothing is missing in our dataset or not so many extra things added to the dataset which might adversely be too much time consuming for the group and might cause confusion for the end users as the project is implemented.

**8 TABLES FOR FORM APPLICANT**

CREATE TABLE PRODUCERS (

producer\_id int,

name varchar (50),

PRIMARY KEY (producer\_id)

);

CREATE TABLE DIRECTORS(

director\_id int,

name varchar (50),

PRIMARY KEY (director\_id)

);

CREATE TABLE CONTESTANT(

contestant\_id int,

applicant\_id int,

name varchar(50),

Email varchar(100),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE CONTESTANT\_GENDER(

contestant\_id int,

gender varchar(50),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE MEDICATIONS(

contestant\_id int,

medicine varchar(50),

reason varchar(100),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE JOBS(

contestant\_id int,

job\_title varchar(50),

start\_date datetime,

end\_date datetime,

description varchar(500),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE ADDRESS(

contestant\_id int,

apartment\_number int,

block varchar(50),

street varchar(100),

city varchar(50),

state varchar(50),

postalcode int,

country varchar(50),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE RATINGS(

applicant\_id int,

producer\_rating int,

director\_rating int,

PRIMARY KEY (applicant\_id)

);

**4 TABLES FOR FORM BACKGROUND CHECK**

CREATE TABLE EVALUATION(

contestant\_id int,

national\_id varchar(50),

apperance\_rating int,

strength\_rating int,

religion varchar(50),

PRIMARY KEY (contestant\_id)

);

CREATE TABLE EMPLOYER(

contestant\_id int,

employer\_name varchar(50),

employer\_phone int,

employer\_id int,

comments varchar(100),

PRIMARY KEY (employer\_id)

);

CREATE TABLE EDUCATION(

contestant\_id int,

highest\_qualification varchar(50),

comments varchar(100),

institution varchar(50),

institution\_contact\_number int,

PRIMARY KEY (contestant\_id)

);

CREATE TABLE JUDICIAL(

contestant\_id int,

judicial\_id int,

record\_date datetime,

category varchar(100),

outcome varchar(50),

description varchar(500),

PRIMARY KEY (judicial\_id)

);

**9 TABLES FOR FORM EVENTS**

CREATE TABLE EPISODES(

episode\_id int,

date\_aired datetime,

name varchar(100),

PRIMARY KEY (episode\_id)

);

CREATE TABLE EVENTS(

event\_id int,

episode\_id int,

director\_id int,

producer\_id int,

team\_event bit,

name varchar(50),

PRIMARY KEY (event\_id)

);

CREATE TABLE TEAMS(

event\_id int,

contestant\_id int,

team\_id int,

points int

PRIMARY KEY (event\_id,contestant\_id)

);

CREATE TABLE TASKS(

event\_id int,

contestant\_id int,

task\_id int,

completion bit,

time int

PRIMARY KEY (event\_id,contestant\_id)

);

CREATE TABLE POINTS(

event\_id int,

contestant\_id int,

points int

PRIMARY KEY (event\_id,contestant\_id)

);

CREATE TABLE DANGER\_LEVEL(

event\_id int,

danger\_level int,

PRIMARY KEY (event\_id)

);

CREATE TABLE ESTIMATED\_TIME(

event\_id int,

time\_length int,

PRIMARY KEY (event\_id)

);

CREATE TABLE DIRECTION\_SETUP(

action\_scene\_id int,

event\_id int,

description varchar(100),

camera\_setting varchar(100),

PRIMARY KEY (action\_scene\_id)

);

CREATE TABLE SCENE\_TIME(

action\_scene\_id int,

time int,

PRIMARY KEY (action\_scene\_id)

);

**4 TABLES FOR FORM VOTING**

CREATE TABLE LOCATION(

voting\_id int,

region varchar(50),

country varchar(50),

state varchar(100),

city varchar(100),

postal\_code int,

PRIMARY KEY (voting\_id)

);

CREATE TABLE VOTES(

voting\_id int,

event\_id int,

episode\_id int,

contestant\_id int,

PRIMARY KEY (voting\_id)

);

CREATE TABLE METHODS(

voting\_id int,

method\_id int,

PRIMARY KEY (voting\_id)

);

CREATE TABLE METHODS\_ID(

method\_id int,

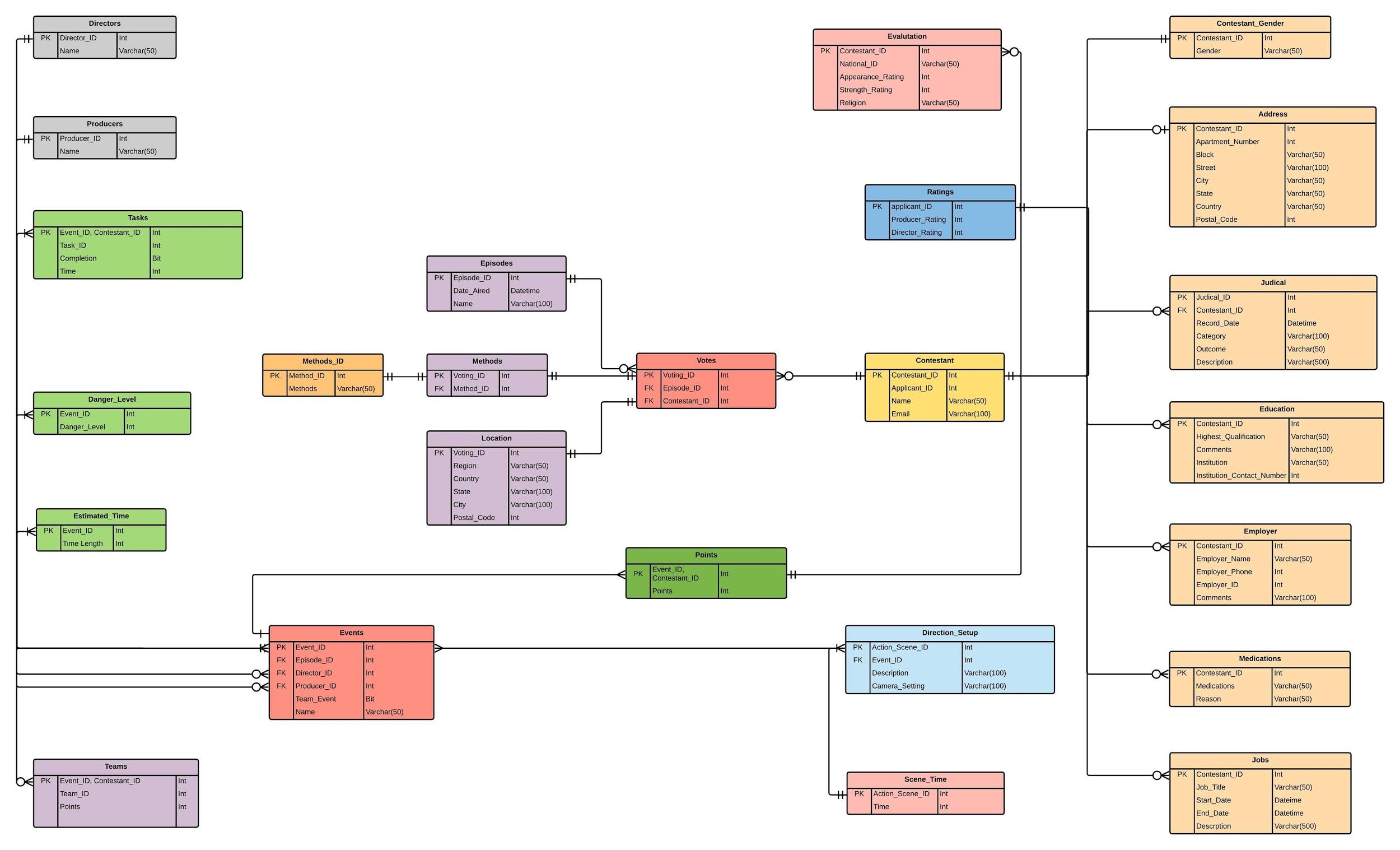
methods varchar(50),

PRIMARY KEY (method\_id)

);

**2. Create an ERD which shows these tables and fields. You may use PowerPoint or Visio or some other data modeling tool.**

The coherent relationship of 25 tables is shown in the entity-relationship diagram (ERD) below. Each of the relationships is built and connected consistently with one or other tables in the dataset. On the right-hand side of the ERD is showing all the information needed from a contestant who will be sending the application to participate in some events, challenges or tasks hosted by the producer and/ or the director so that they can then be one of the promising candidates to join the Big Teeth TV show. Note that the color of the tables is set with the purpose of grouping the related tables together so that the ERD will be easier to follow; especially for the end-users to understand the flow of the project when they are first introduced to it.

****

**3. Enter sample data to populate the tables to test your design. You do not need to create data entry forms**

As tables and entity relationships are well prepared in the previous steps, we now will perform the next interesting step which is populating the data into existing tables. Please note the data is populated randomly.

**Populating the data**

INSERT INTO DANGER\_LEVEL

(event\_id,danger\_level)

VALUES

(1,7),

(2,9),

(3,8),

(4,10),

(5,9),

(6,6);

INSERT INTO SCENE\_TIME

(action\_scene\_id,time)

VALUES

(1,15),

(2,25),

(3,10),

(4,12),

(5,9),

(6,20),

(7,15),

(8,13),

(9,14);

INSERT INTO VOTES

(voting\_id,episode\_id,event\_id,contestant\_id)

VALUES

(1,1,1,2),

(2,1,2,1),

(3,1,2,3),

(4,1,1,4),

(5,1,2,3),

(6,1,2,3),

(7,1,2,3),

(8,1,2,1),

(9,1,2,3),

(10,2,3,4),

(11,2,3,4),

(12,2,4,1),

(13,2,3,4),

(14,2,4,1),

(15,2,4,1),

(16,2,4,1);

INSERT INTO CONTESTANT

(contestant\_id,applicant\_id,name,Email)

VALUES

(1,2,'ASHLEY','ash456@gmail.com'),

(2,3,'Lynn','lynn666@gmail.com'),

(3,4,'Jenish','jen745@gmail.com'),

(4,5,'Diem','diem444@gmail.com'),

(5,7,'Moxshil','mox4555@gmail.com'),

(6,8,'Jet','jet888@gmail.com'),

(7,11,'Jack','jack7525@gmail.com'),

(8,12,'Mark','Mark8856@gmail.com'),

(9,13,'Steve','Steve7856@gmail.com');

INSERT INTO LOCATION

(voting\_id,region,country,state,city,postal\_code)

VALUES

(11,'australia','new zealand','state','yangon',94112)

(12,'asia','Burma','yangon','yangon',94112),

(13,'europe','germany','bavaria','munich',80638),

(14,'asia','Taiwan','Taiwan Province','taipie',10460),

(15,'africa','Nigeria','Lagos state','lagos',100242),

(16,'north america', 'USA','Texas','austin',78652),

(17,'australia','Australia','sydney','New south wales',2026),

(18,'asia','India','mumbai','Maharashtra',400026),

(19,'asia','China','Jinan','Shandong',250014),

(20,'north america','Canada','torronto','Ontario',91710),

(11,'europe','england','','london',25798),

(12,'Asia','singapore','','singapore',10113),

(13,'europe','spain','','barcelona',87009),

(14,'europe','netherlands','','amsterdand',45389),

(15,'north america','USA','','boston',11111),

(16,'north america','USA','','new york',89866),

(17,'asia','hong kong','','hong kong',22456),

(18,'north america','USA','','Chicago',35689),

(19,'asia','india','','delhi',22570),

(20,'europe','france','','paris',44970);

INSERT INTO TEAM\_MEMBER\_POINTS (event\_id,contestant\_id,team\_id,points)

VALUES

(2,1,1,8),

(2,2,1,7),

(2,3,1,9),

(2,4,2,-2),

(2,5,2,0),

(2,6,2,3),

(4,4,3,3),

(4,3,3,4),

(4,2,3,6),

(4,1,4,7),

(4,5,4,8),

(4,6,4,9);

INSERT INTO TASKS (event\_id,contestant\_id,task\_id,completion,time)

VALUES

(1,1,1,0,45),

(1,2,2,1,30),

(1,3,3,1,25),

(1,4,4,1,35),

(1,5,5,0,50),

(1,6,6,0,55),

(2,1,7,1,130),

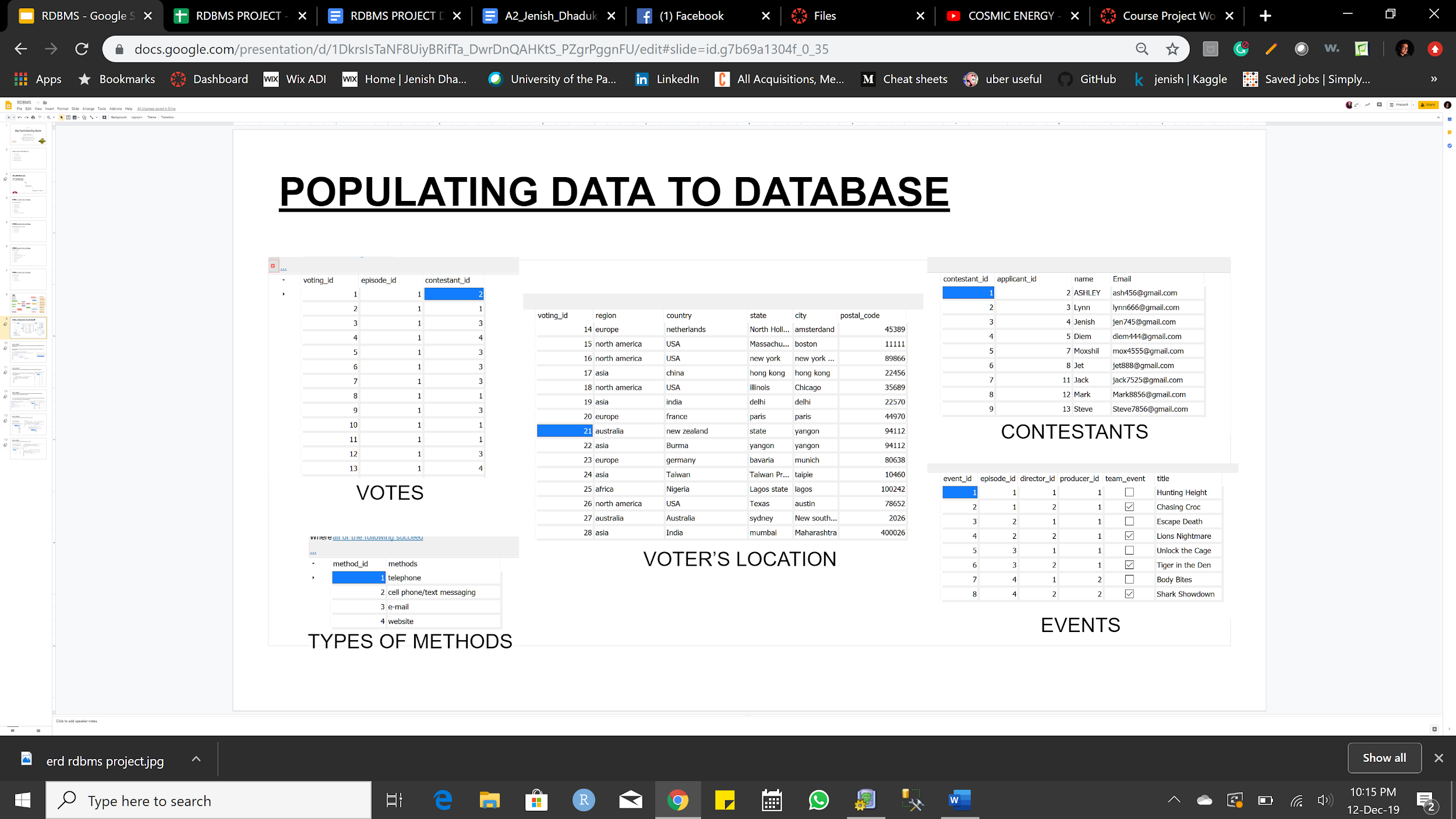
(2,2,7,1,130),

(2,3,7,1,130),

(2,4,8,0,150),

(2,5,8,0,150),

(2,6,8,0,150);

****

**Big Teeth TV Show Database**

**Create queries to answer the following questions:**

***1. Calculate the average producer and director ratings for a specific event's team members (you pick the event title).***

***QUERY***

SELECT AVG(r.producer\_rating)as producer\_rating\_average\_of\_team,

AVG(r.director\_rating) as director\_rating\_average\_of\_team

FROM RATINGS r

JOIN CONTESTANT c

ON c.applicant\_id = r.applicant\_id

WHERE

c.contestant\_id IN(SELECT contestant\_id

FROM TEAMS

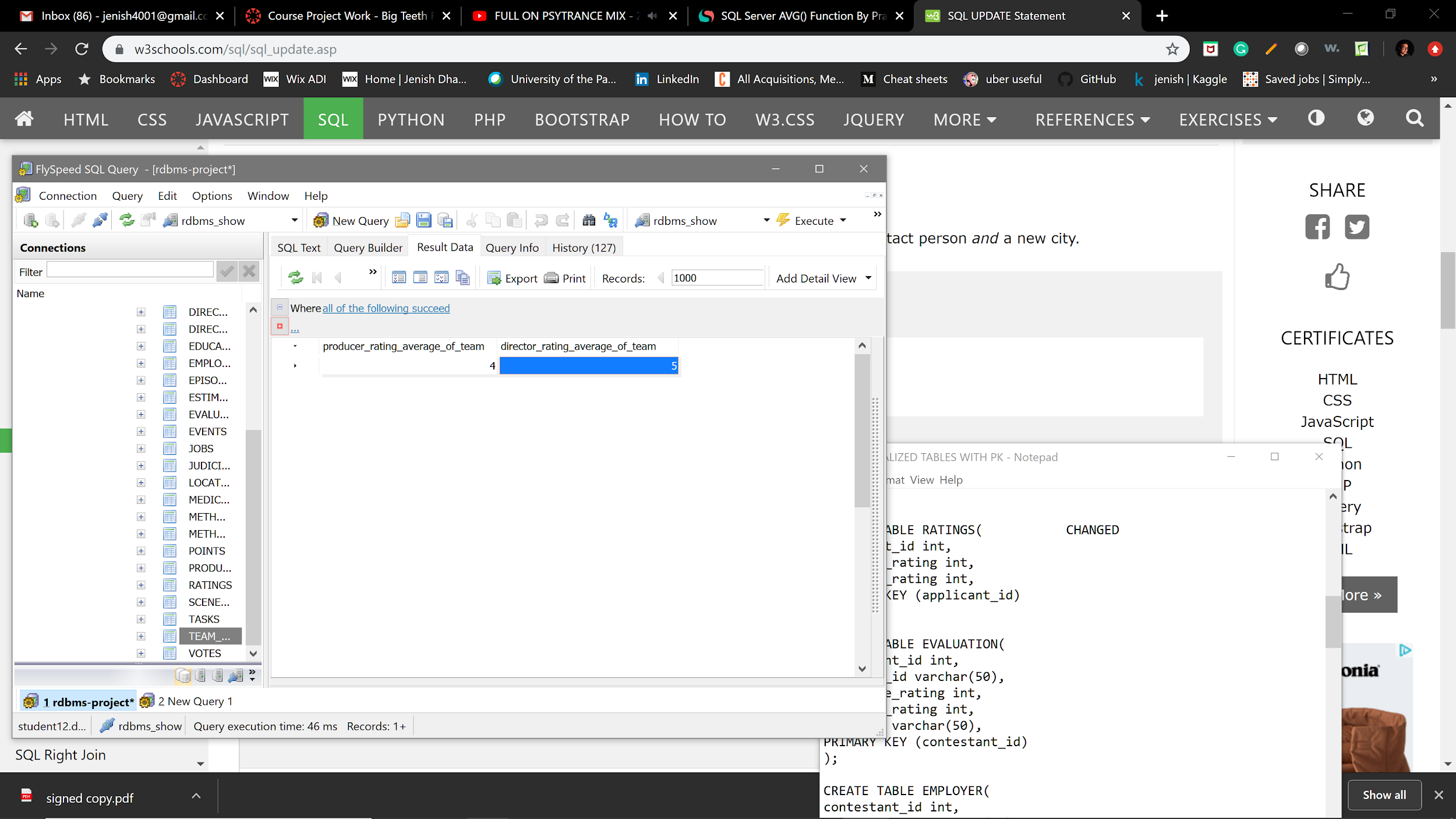
WHERE team\_id=1

AND event\_id = (SELECT event\_id

FROM EVENTS e

WHERE title = 'ChasingCroc'))

Answer: the average producer ratings is 4 while it is 5 for the average director ratings.

******

***2. Determine which actions will take the longest. Rank all actions from longest to shortest.***

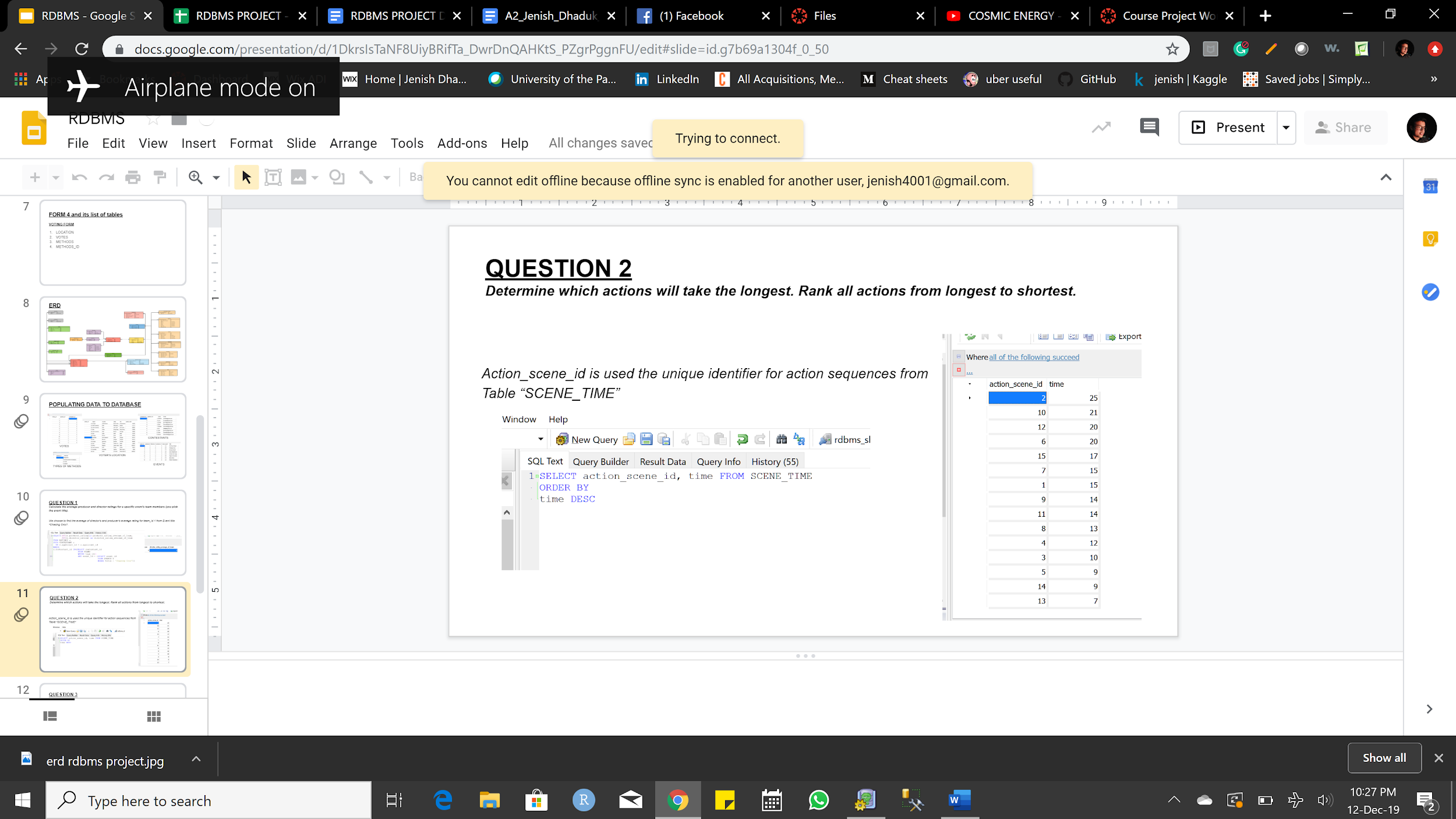
***QUERY***

SELECT \* FROM SCENE\_TIME

ORDER BY

time DESC

Answer: action scene with the ID of 2 takes the longest time which is 25.



***3. List each contestant's total votes by region and method for a specific episode (you pick the episode title). Rank this from highest to lowest.***

***QUERY***

SELECT v.contestant\_id,mid.methods,l.region,COUNT(v.voting\_id) as vote\_counts

FROM VOTES v

JOIN LOCATION l

ON l.voting\_id = v.voting\_id

JOIN METHODS m

ON m.voting\_id = l.voting\_id

JOIN METHODS\_ID mid

ON mid.method\_id = m.method\_id

WHERE v.episode\_id = (SELECT episode\_id FROM EPISODES WHERE name = 'FUN BEGINS')

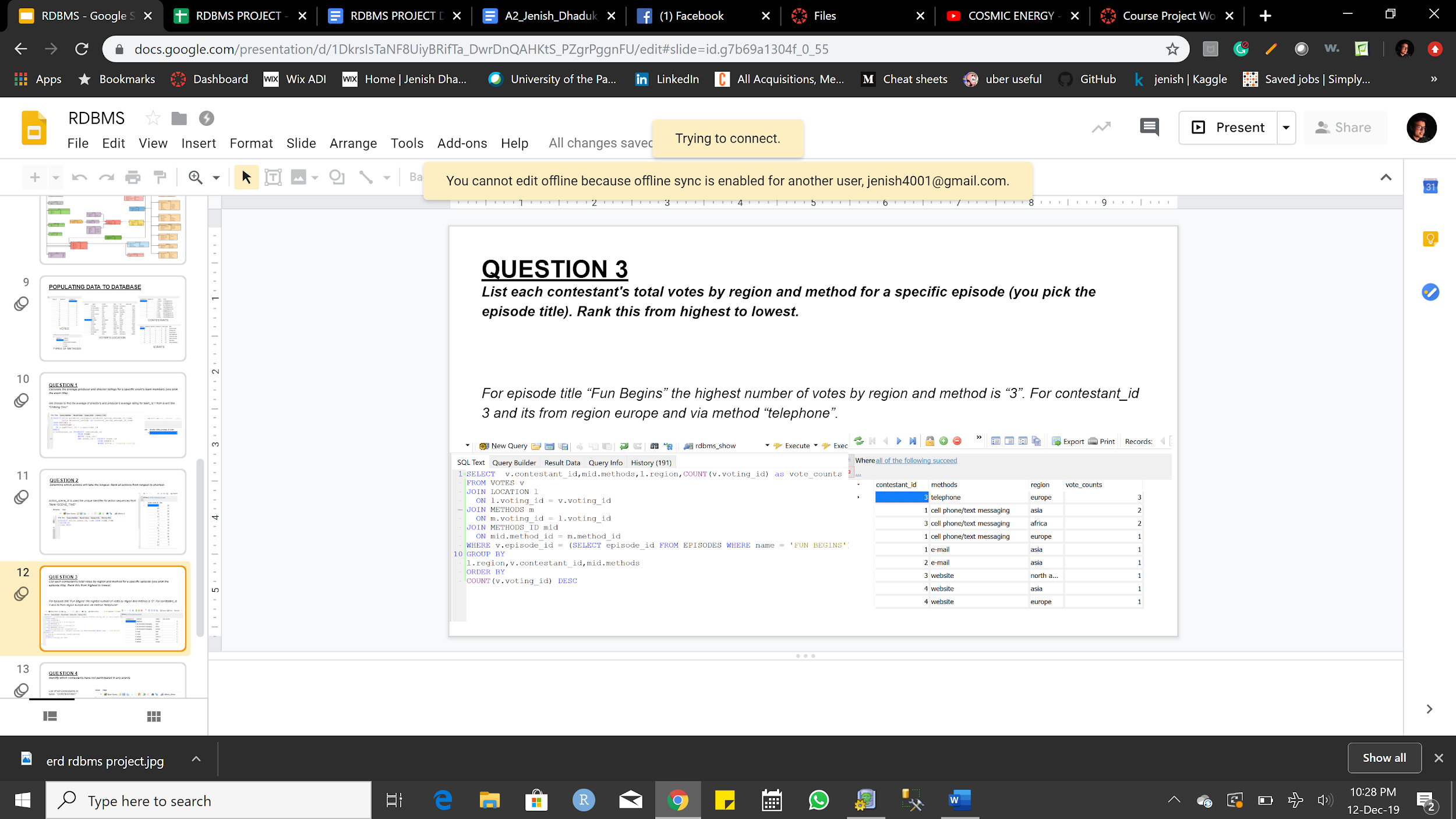
GROUP BY

l.region,v.contestant\_id,mid.methods

ORDER BY

COUNT(v.voting\_id) DESC

Answer: Below is the list of 9 contestants with their votes by region and method for the episode names *Fun Begins.*

******

***4. Identify which contestants have not participated in any events.***

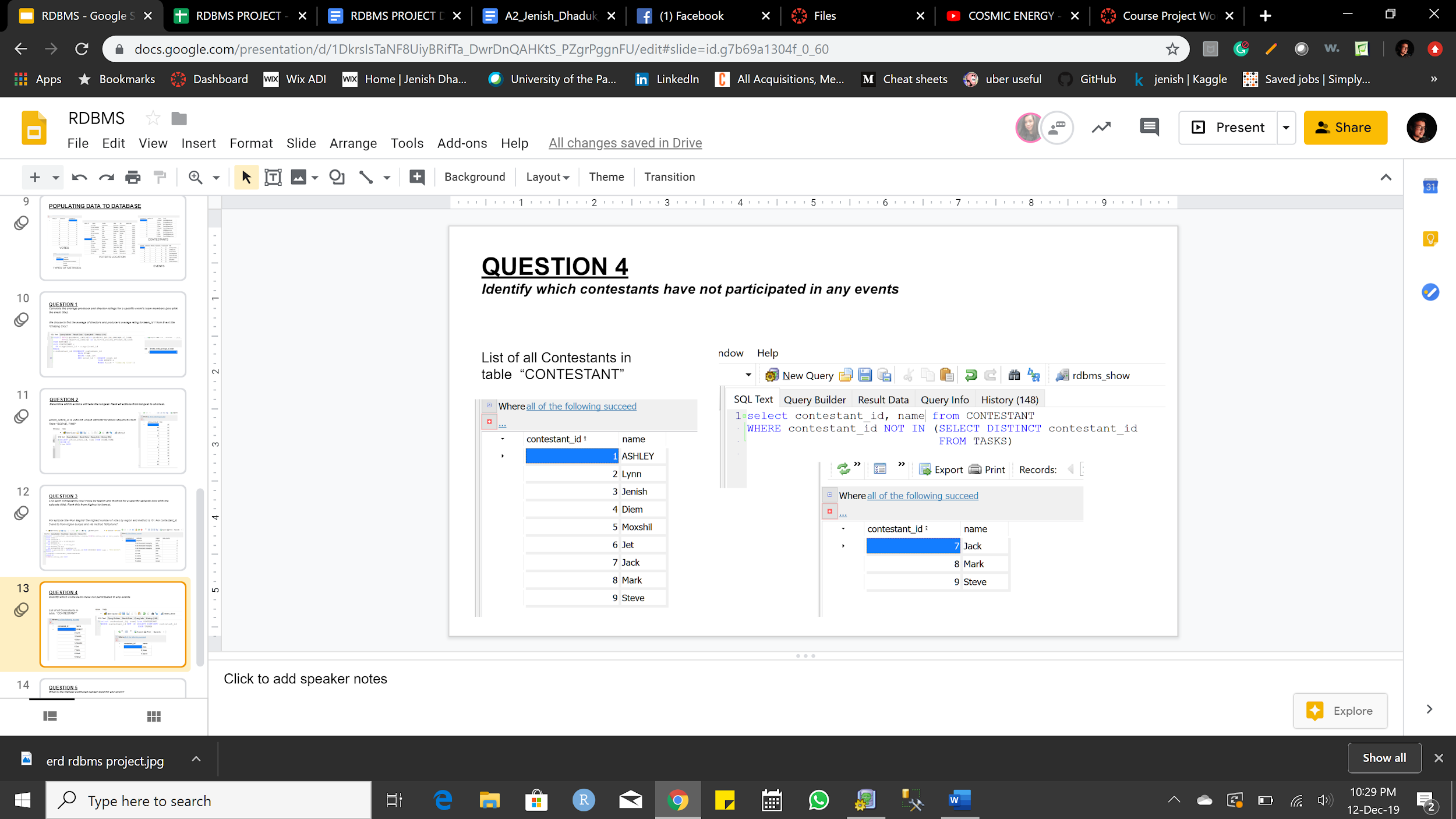
***QUERY***

select contestant\_id, name from CONTESTANT

WHERE contestant\_id NOT IN (SELECT DISTINCT contestant\_id

FROM TASKS)

Answer: Jack, Mark, and Steve are so far the 3 contestants that have not participated in any events.

******

**5. What is the highest estimated danger level for any event?**

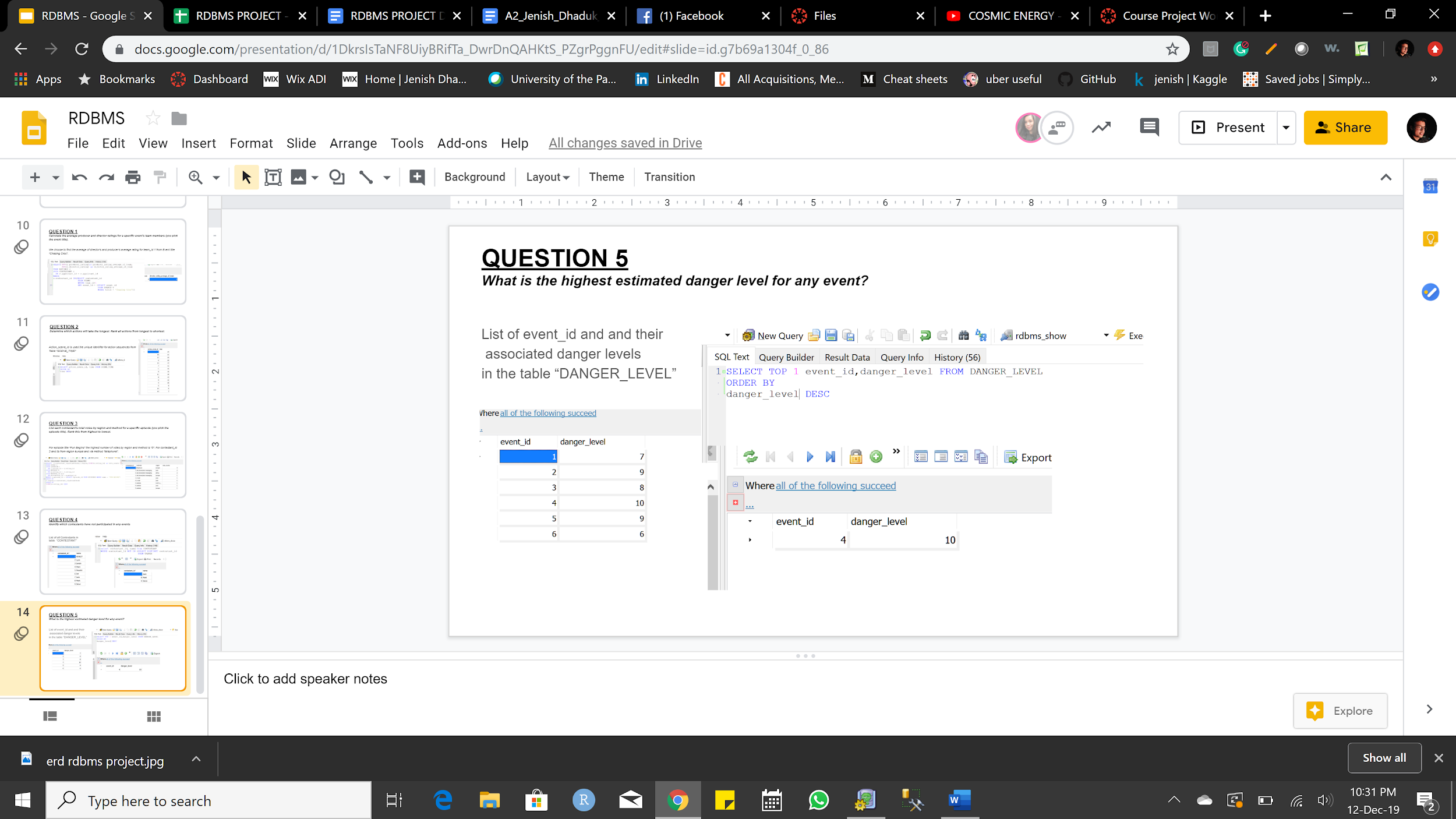
***QUERY***

SELECT TOP 1 event\_id,danger\_level FROM DANGER\_LEVEL

ORDER BY

danger\_level DESC

Answer: the event with the Id of 4 has the highest danger level- 10.

****

**6. Show a relational expression for any one query**

RELATIONAL EXPRESSION FOR QUERY4

Π contestant\_id,name(σ contestant\_id(CONTESTANT) - σ contestant\_id(TASKS))