Podcast Database Part 5

Nicholas Guzman | Kristijan Hornung | Victor Cardenas

nguzman9@csustan.edu | khornung@csustan.edu | vcardenas4@csustan.edu

Section 1:

Secti	on i.					
	CREATE TAI	BLE Per	rson (
	PersonID		HAR(11)	PRIM	ARY K	EY,
	Firstname		HAR(100)		NULL,	,
	Lastname		HAR(100)		NULL,	
	Nickname		HAR(50),		Ź	
	Title		HAR(30),			
	Age		GER(3)	NOT 1	NULL.	
	Bio		(30,000),		,	
	ProfilePic		(65,535),			
	Linkedin		HAR(100),			
	Twitter		HAR(100),			
	Facebook		HAR(100),			
	Wiki		HAR(100),			
	Youtube		HAR(100),			
	Instagram		HAR(100));			
	υ		(),,			
CREA	TE TABLE Po	dcaster	(
	PersonID		VARCHAR(1	1)	REFE	RENCES Person (PersonID),
	ExpertiseLeve	el	VARCHAR(3	o),		, , , , , , , , , , , , , , , , , , , ,
	NativeLangua		VARCHAR(3	, ,		
	Č	C	`	,,,		
Create	Table Guest (
	PersonID		VARCHAR(1	1)	REFE	RENCES Person (PersonID),
	Profession		VARCHAR(3	/		, , , , , , , , , , , , , , , , , , , ,
			`	,,,		
CREA	TE TABLE Po	dcast (
	PodID		VARC	HAR(1	1)	PRIMARY KEY,
	PodcastForma	ıt	VARCHAR(3		,	,
	Genre			HAR(4	(0),	
	Title			HAR(4		
	NumberOfEpi	isodes		GER(5)		NOT NULL,
	NumberOfSul			GER(15		,
	AgeRating			HAR(5	/ /	
	Description			(30,000	/ *	
				(,	//>	
CREA	TE TABLE Ep	isode (
	EpID		VARCHAR(1	1)	PRIM	ARY KEY,
	1		-(-	,		,

EpDate DATETIME,

EpLength TIME NOT NULL,

EpisodeNumber INTEGER(5), ViewerCount INTEGER(8),

DownloadSize INTEGER(7) NOT NULL,

Blurb TEXT(30,000));

CREATE TABLE CuratedCollection (

CurCollectionID VARCHAR(11) PRIMARY KEY,

Name VARCHAR(50), TrackSize INTEGER(5),

DateStarted DATE, DateFinished DATE,

MaturityRating VARCHAR(30), Description TEXT(30,000));

CREATE TABLE Artwork (

ArtworkID VARCHAR(11) PRIMARY KEY,

Artist VARCHAR(100), Theme VARCHAR(50), Title VARCHAR(100),

DateCreated DATE,

SubjectMatter VARCHAR(100), MeaningDescription TEXT(30,000));

CREATE TABLE Platform (

PlatformID VARCHAR(11) PRIMARY KEY,

Name VARCHAR(50), WebsiteURL VARCHAR(100), Device VARCHAR(50), DownloadLink VARCHAR(100), Cost FLOAT(3,2));

CREATE TABLE Rating (

EpID VARCHAR(11),
Review Comment VARCHAR(200),
NumberOfStars VARCHAR(1),
TimeAndDate DATETIME,
Username VARCHAR(50),
NumberOfLikes INTEGER(11),

CONSTRAINT Rating_TimeAndDate_Username_EpID_pk PRIMARY KEY (TimeAndDate, Username, EpID),

CONSTRAINT Rating_EpID_fk FOREIGN KEY (EpID) REFERENCES Episode (EpID) ON DELETE CASCADE);

CREATE TABLE RaterProfile (

Username VARCHAR(50) PRIMARY KEY,

RaterPic BLOB(65535));

CONSTRAINT RaterProfile Username pk PRIMARY KEY (Username),

CONSTRAINT RaterProfile_Username_fk FOREIGN KEY (Username) REFERENCES Rating (Username) ON DELETE CASCADE);

CREATE TABLE Hosts (

PersonID VARCHAR(11), PodID VARCHAR(11),

CONSTRAINT Hosts PersonID PodID PRIMARY KEY (PersonID, PodID),

CONSTRAINT Hosts_PersonID_fk FOREIGN KEY (PersonID) REFERENCES Podcaster (PersonID) ON DELETE CASCADE,

CONSTRAINT Hosts_PodID_fk FOREIGN KEY (PodID) REFERENCES Podcast (PodID) ON DELETE CASCADE);

CREATE TABLE AppearsIn (

PersonID VARCHAR(11),

EpID VARCHAR(11),

CONSTRAINT AppearsIn PersonID_EpID_pk PRIMARY KEY (PersonID, EpID),

CONSTRAINT AppearsIn_fk FOREIGN KEY (PersonID) REFERENCES Guest (PersonID) ON DELETE CASCADE,

CONSTRAINT AppearsIn_fk FOREIGN KEY (EpID) REFERENCES Episode (EpID) ON DELETE CASCADE);

CREATE TABLE RunOn (

PlatformID VARCHAR(11), PodID VARCHAR(11),

CONSTRAINT RunOn_PlatformID_PodID_pk PRIMARY KEY (PlatformID, PodID),

CONSTRAINT RunOn_PlatformID_fk FOREIGN KEY (PlatformID) REFERENCES Platform (PlatformID),

CONSTRAINT RunOn_PodID_fk FOREIGN KEY (PodID) REFERENCES Podcast (PodID));

CREATE TABLE Offers

PodID VARCHAR(11),

CurCollectionID VARCHAR(11),

CONSTRAINT Offers_PodID_CurCollectionID_pk PRIMARY KEY(PodID, CurCollectionID),

CONSTRAINT Offers_PodID_fk FOREIGN KEY (PodID) REFERENCES Podcast (PodID) ON DELETE CASCADE,

CONSTRAINT Offers_CurCollectionID_fk FOREIGN KEY (CurCollectionID) REFERENCES Curated Collection (CurCollectionID) ON DELETE CASCADE);

CREATE TABLE Debuts (

PodID VARCHAR(11), EpID VARCHAR(11),

CONSTRAINT Debuts EpID pk PRIMARY KEY (EpID),

CONSTRAINT Debuts_PodID_fk FOREIGN KEY (PodID) REFERENCES Podcast (PodID) ON DELETE CASCADE,

CONSTRAINT Debuts_EpID_fk FOREIGN KEY (EpID) REFERENCES Episode (EpID) ON DELETE CASCADE);

CREATE TABLE IsListedUnder (

CurCollectionID VARCHAR(11), EpID VARCHAR(11),

CONSTRAINT IsListedUnder_CurCollectionID_EpID_pk PRIMARY KEY (CurCollectionID, EpID),

CONSTRAINT IsListedUnder_CurCollectionID_fk FOREIGN KEY(CurCollectionID) REFERENCES Curated Collection (CurCollectionID) ON DELETE CASCADE,

CONSTRAINT IsListedUnder_EpID_fk FOREIGN KEY(EpID) REFERENCES Episode (EpID) ON DELETE CASCADE);

CREATE TABLE Presents (

EpID VARCHAR(11),

ArtworkID VARCHAR(11),

CONSTRAINT Presents ArtworkID pk PRIMARY KEY (ArtworkID),

CONSTRAINT Presents_ArtworkID_fk FOREIGN KEY (ArtworkID) REFERENCES Artwork (ArtworkID) ON DELETE CASCADE,

CONSTRAINT Presents_EpID_fk FOREIGN KEY (EpID) REFERENCES Episode (EpID) ON DELETE CASCADE);

Explanation of changes to the above Initial Relations from Part 4:

-We split the table "Rating" into two smaller tables: "Rating" and "RaterProfile" because the original Rating table did not satisfy the criteria for 3NF and BCNF normal forms. The new tables are structured as: Rating(<u>TimeAndDate</u>, <u>Username</u>, <u>EpID</u>, Review, NumberOfStars, NumberOfLikes), and RaterProfile(<u>Username</u>, RaterPic), with Rating having as primary key attributes TimeAndDate, Username, and EpID, whereas in RaterProfile the key is Username. EpID and Username are set to be unique in both tables, while RaterPic is set to be unique in RaterProfile.

-Changes in data types of attributes:

- 1. Person.age from integer(3) to "int" due to suitability of "int" type
- 2. Person. Bio from text(30000) to medium text due to suitability of "medium text" type
- 3. Person. Profile Pic from BLOB (65,535) to simply "blob", the size could not be specified
- 4. Podcast.NumberOfEpisodes and Podcast.NumberOfSubscribers from integer(5) and integer(15) respectively to simply "int", the size could not be specified
- 5. Podcast.Description from text(30000) to "mediumtext" due to suitability of "mediumtext" type
- 6. Episode.EpDate, Episode.EpLength, Episode.EpisodeNumber, Episode.ViewerCount, Episode.DownloadSize from "datetime", "time", "integer(5)", "integer(8)", and "integer(7)" respectively, all changed to varchar(20) to avoid any potential date and/or time format errors; and to provide 20 chars for large EpisodeNumber, ViewerCount, or DownloadSize. Also, to enable entering "MB" or "GB" with download size.
- 7. Episode.Blurb from text(30000) to "mediumtext" due to suitability of "mediumtext" type for the purpose
- 8. CuratedCollection's attributes data types for attributes CurCollectionID, Name, TrackSize, DateStarted, DateFinished, MaturityRating, Description, changed to: varchar(50), varchar(20), varchar(20), varchar(20), warchar(20), mediumtext respectively (from varchar(11), varchar(50), integer(5), date, date, varchar(30), text(30000), respectively) due to suitability of the varchar(50), varchar(20) and mediumtext types for the purpose, and to avoid date/time format-related errors, or track size format errors.
- 9. Artwork.SubjectMatter data type increased in size from varchar(100) to varchar(1000) to allow for longer artwork subject descriptions. Artwork.MeaningDescription data type changed from text(30000) to mediumtext due to better suitability of "mediumtext" type.
- 10. Platform.Cost data type changed from float(3,2) to varchar(20) to allow for currency signs (\$ for example).
- 11. Rating.NumberOfStars data type increased from varchar(1) to varchar(20) to allow for textual input ("one", "two",...), Rating.NumberOfLikes changed from integer(11) to int due to integer(11) not being available.
- WebsiteURL in relation "Platform" is no longer considered a candidate key, and is therefore not designated as a key in the database table, and not given a status of "unique" attribute, because we realized that two different URLs can be made to point to the same website.

Section 2:

List of SQL CREATE TABLE commands:

1) Person:

```
CREATE TABLE 'Person' (
'PersonID' varchar(11) NOT NULL,
`Firstname` varchar(100) NOT NULL,
'Lastname' varchar(100) NOT NULL,
'Nickname' varchar(50) DEFAULT NULL,
'Title' varchar(30) DEFAULT NULL,
'Age' int DEFAULT NULL,
'Bio' mediumtext.
'ProfilePic' blob,
`Linkedin` varchar(100) DEFAULT NULL,
'Twitter' varchar(100) DEFAULT NULL,
'Facebook' varchar(100) DEFAULT NULL,
'Wiki' varchar(100) DEFAULT NULL,
'Youtube' varchar(100) DEFAULT NULL,
'Instagram' varchar(100) DEFAULT NULL,
PRIMARY KEY ('PersonID'),
UNIQUE KEY 'PersonID' ('PersonID')
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci
```

2) Podcaster:

```
CREATE TABLE 'Podcaster' (
'PersonID' varchar(11) NOT NULL,
'ExpertiseLevel' varchar(30) DEFAULT NULL,
'NativeLanguage' varchar(30) DEFAULT NULL,
PRIMARY KEY ('PersonID'),
UNIQUE KEY 'PersonID' ('PersonID'),
CONSTRAINT 'podcaster_ibfk_1' FOREIGN KEY ('PersonID') REFERENCES 'Person' ('PersonID') ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

3) Guest:

```
CREATE TABLE `Guest` (
    `PersonID` varchar(11) NOT NULL,
    `Profession` varchar(30) DEFAULT NULL,
    PRIMARY KEY (`PersonID`),
    UNIQUE KEY `PersonID` (`PersonID`),
    CONSTRAINT `guest_ibfk_1` FOREIGN KEY (`PersonID`) REFERENCES `Person` (`PersonID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

4) Podcast:

```
CREATE TABLE 'Podcast' (
    'PodID' varchar(11) NOT NULL.
    'PodcastFormat' varchar(30) DEFAULT NULL,
    'Genre' varchar(40) DEFAULT NULL,
    'Title' varchar(40) DEFAULT NULL,
    'NumberOfEpisodes' int NOT NULL,
    'NumberOfSubscribers' int DEFAULT NULL,
    'AgeRating' varchar(5) DEFAULT NULL,
    'Description' mediumtext,
    PRIMARY KEY ('PodID'),
    UNIQUE KEY 'PodID' ('PodID')
   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
5) Episode:
   CREATE TABLE 'Episode' (
    'EpID' varchar(11) NOT NULL,
    `EpDate` varchar(20) DEFAULT NULL,
    `EpLength` varchar(20) NOT NULL,
    `EpisodeNumber` varchar(20) DEFAULT NULL,
    'ViewerCount' varchar(20) DEFAULT NULL,
    'DownloadSize' varchar(20) NOT NULL,
    'Blurb' mediumtext.
    PRIMARY KEY ('EpID'),
    UNIQUE KEY 'EpID' ('EpID')
   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
6) CuratedCollection:
   CREATE TABLE `CuratedCollection` (
    `CurCollectionID` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4 0900 ai ci NOT
   NULL.
    'Name' varchar(50) DEFAULT NULL,
    'TrackSize' varchar(20) DEFAULT NULL,
    'DateStarted' varchar(20) DEFAULT NULL,
    `DateFinished` varchar(20) DEFAULT NULL,
    'MaturityRating' varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4 0900 ai ci
   DEFAULT NULL,
    'Description' mediumtext,
    PRIMARY KEY ('CurCollectionID'),
    UNIQUE KEY 'CurCollectionID' ('CurCollectionID')
   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci
```

7) Artwork:

```
CREATE TABLE `Artwork` (
    `ArtworkID` varchar(11) NOT NULL,
    `Artist` varchar(100) DEFAULT NULL,
    `Theme` varchar(50) DEFAULT NULL,
    `Title` varchar(100) DEFAULT NULL,
    `DateCreated` date DEFAULT NULL,
    `SubjectMatter` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
DEFAULT NULL,
    `MeaningDescription` mediumtext,
    PRIMARY KEY (`ArtworkID`),
    UNIQUE KEY `ArtworkID` ('ArtworkID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

8) Platform:

```
CREATE TABLE 'Platform' (
    'PlatformID' varchar(11) NOT NULL,
    'Name' varchar(50) DEFAULT NULL,
    'WebsiteURL' varchar(100) DEFAULT NULL,
    'Device' varchar(50) DEFAULT NULL,
    'DownloadLink' varchar(100) DEFAULT NULL,
    'Cost' varchar(20) DEFAULT NULL,
    PRIMARY KEY ('PlatformID'),
    UNIQUE KEY 'PlatformID' ('PlatformID')
    ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

9) Rating:

```
CREATE TABLE `Rating` (
    `EpID` varchar(11) NOT NULL,
    `ReviewComment` varchar(200) DEFAULT NULL,
    `NumberOfStars` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
    DEFAULT NULL,
    `TimeAndDate` datetime NOT NULL,
    `Username` varchar(50) NOT NULL,
    `NumberOfLikes` int DEFAULT NULL,
    PRIMARY KEY (`TimeAndDate`,`Username`,`EpID`),
    UNIQUE KEY `EpID` (`EpID`),
    UNIQUE KEY `Username_2` (`Username`),
    CONSTRAINT `Rating_EpID_fk` FOREIGN KEY (`EpID`) REFERENCES `Episode` (`EpID`) ON
    DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci
```

10) Hosts:

11) AppearsIn:

12) RunOn:

13) Offers:

14) Debuts:

```
CREATE TABLE `Debuts` (
    `PodID` varchar(11) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
    `EpID` varchar(11) NOT NULL,
    PRIMARY KEY (`EpID`),
    UNIQUE KEY `EpID` (`EpID`),
    KEY `debuts_ibfk_1` (`PodID`),
    CONSTRAINT `Debuts_EpID_fk` FOREIGN KEY (`EpID`) REFERENCES `Episode` (`EpID`)
    ON DELETE CASCADE,
    CONSTRAINT `debuts_ibfk_1` FOREIGN KEY (`PodID`) REFERENCES `Podcast` (`PodID`)
    ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

15) IsListedUnder:

```
CREATE TABLE 'IsListedUnder' (
'CurCollectionID' varchar(11) NOT NULL,
'EpID' varchar(11) NOT NULL,
PRIMARY KEY ('CurCollectionID', 'EpID'),
KEY 'EpID' ('EpID'),
CONSTRAINT 'IsListedUnder_CurCollectionID_fk' FOREIGN KEY ('CurCollectionID')
REFERENCES 'CuratedCollection' ('CurCollectionID') ON DELETE CASCADE,
CONSTRAINT 'islistedunder_ibfk_1' FOREIGN KEY ('EpID') REFERENCES 'Episode' ('EpID')
ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

16) Presents:

```
CREATE TABLE `Presents` (
    `EpID` varchar(11) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
    `ArtworkID` varchar(11) NOT NULL,
    PRIMARY KEY (`ArtworkID`),
    UNIQUE KEY `ArtworkID` (`ArtworkID`),
    KEY `Presents_EpID_fk` (`EpID`),
    CONSTRAINT `Presents_ArtworkID_fk` FOREIGN KEY (`ArtworkID`) REFERENCES `Artwork`
(`ArtworkID`) ON DELETE CASCADE,
    CONSTRAINT `Presents_EpID_fk` FOREIGN KEY (`EpID`) REFERENCES `Episode` (`EpID`)
    ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

17) RaterProfile:

```
CREATE TABLE `RaterProfile` (
    `RaterPic` blob,
    `Username` varchar(50) NOT NULL,
    PRIMARY KEY (`Username`),
    UNIQUE KEY `Username_2` (`Username`),
    CONSTRAINT `raterprofile_ibfk_1` FOREIGN KEY (`Username`) REFERENCES `Rating`
    (`Username`) ON DELETE CASCADE ON UPDATE CASCADE
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4 0900 ai ci
```

Section 3:

AppearsIn EXPLAIN:

EXPLAIN AppearsIn

+ Options

Field	Туре	Null	Key	Default	Extra
PersonID	varchar(11)	NO	PRI	NULL	
EpID	varchar(11)	NO	PRI	NULL	

AppearsIn SELECT COUNT:

Your SQL query has been executed successfully.

SELECT COUNT(*) FROM AppearsIn

+ Options
COUNT(*)
45

Artwork EXPLAIN:

EXPLAIN Artwork

+ Options

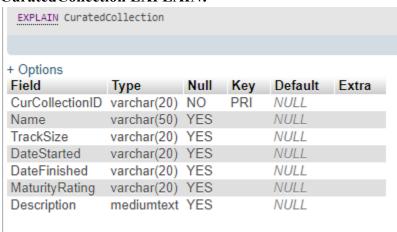
Field	Туре	Null	Key	Default	Extra
ArtworkID	varchar(11)	NO	PRI	NULL	
Artist	varchar(100)	YES		NULL	
Theme	varchar(50)	YES		NULL	
Title	varchar(100)	YES		NULL	
DateCreated	date	YES		NULL	
SubjectMatter	varchar(1000)	YES		NULL	
MeaningDescription	mediumtext	YES		NULL	

Artwork SELECT COUNT:

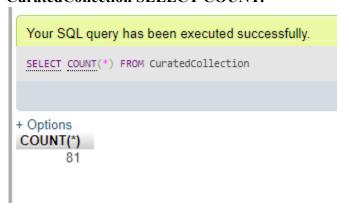
```
SELECT COUNT(*) FROM Artwork

+ Options
COUNT(*)
40
```

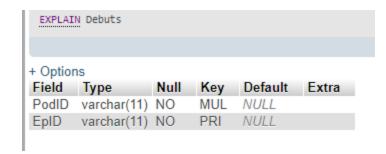
CuratedCollection EXPLAIN:



CuratedCollection SELECT COUNT:



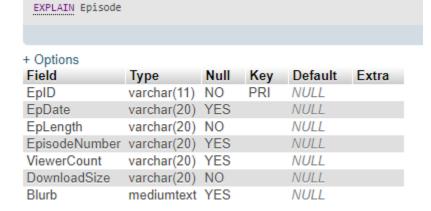
Debuts EXPLAIN:



Debuts SELECT COUNT:



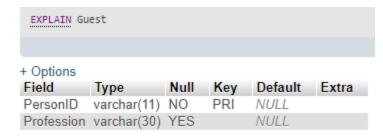
Episode EXPLAIN:



Episode SELECT COUNT:



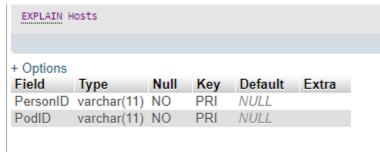
Guest EXPLAIN:



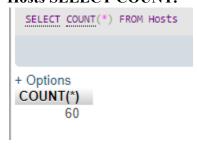
Guest SELECT COUNT



Hosts EXPLAIN:



Hosts SELECT COUNT:



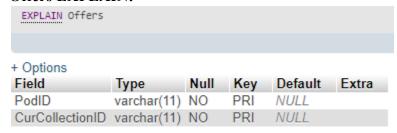
IsListedUnder EXPLAIN:

EXPLAIN IsListe	dUnder				
+ Options	-			D (1	
+ Options Field	Туре	Null	Key	Default	Extra
			Key PRI	Default NULL	Extra
Field		NO			Extra

IsListedUnder SELECT COUNT:



Offers EXPLAIN:



Offers SELECT COUNT:

SELECT COUNT(*) FROM Offers

+ Options
COUNT(*)
41

Person EXPLAIN:

EXPLAIN Person

+ Options

Field	Туре	Null	Key	Default	Extra
PersonID	varchar(11)	NO	PRI	NULL	
Firstname	varchar(100)	NO		NULL	
Lastname	varchar(100)	NO		NULL	
Nickname	varchar(50)	YES		NULL	
Title	varchar(30)	YES		NULL	
Age	int	YES		NULL	
Bio	mediumtext	YES		NULL	
ProfilePic	blob	YES		NULL	
Linkedin	varchar(100)	YES		NULL	
Twitter	varchar(100)	YES		NULL	
Facebook	varchar(100)	YES		NULL	
Wiki	varchar(100)	YES		NULL	
Youtube	varchar(100)	YES		NULL	
Instagram	varchar(100)	YES		NULL	

Person SELECT COUNT:

Your SQL query has been executed successfully.

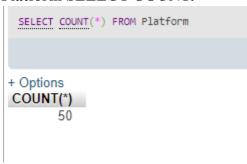
SELECT COUNT(*) FROM Person

+ Options
COUNT(*)
94

Platform EXPLAIN:

	LAIII.				
EXPLAIN Platfo	orm				
+ Options					
Field	Type	Null	Key	Default	Extra
PlatformID	varchar(11)	NO	PRI	NULL	
Name	varchar(50)	YES		NULL	
WebsiteURL	varchar(100)	YES		NULL	
Device	varchar(50)	YES		NULL	
DownloadLink	varchar(100)	YES		NULL	
Cost	varchar(20)	YES		NULL	

Platform SELECT COUNT:



Podcast EXPLAIN:

Type varchar(11)	Null NO	Key	Default	Extra
				Extra
				Extra
varchar(11)	NO	DDI	NILILI	
		FRI	NULL	
varchar(30)	YES		NULL	
varchar(40)	YES		NULL	
varchar(40)	YES		NULL	
int	NO		NULL	
int	YES		NULL	
varchar(5)	YES		NULL	
mediumtext	YES		NULL	
1	varchar(30) varchar(40) varchar(40) int int varchar(5)	varchar(30) YES varchar(40) YES varchar(40) YES int NO int YES	varchar(30) YES varchar(40) YES varchar(40) YES int NO int YES varchar(5) YES	varchar(30) YES NULL varchar(40) YES NULL varchar(40) YES NULL int NO NULL int YES NULL varchar(5) YES NULL

Podcast SELECT COUNT:



Podcaster EXPLAIN:

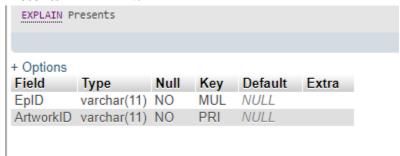


Podcaster SELECT COUNT:

```
+ Options
COUNT(*)

48
```

Presents EXPLAIN:



Presents SELECT COUNT:



RaterProfile EXPLAIN:



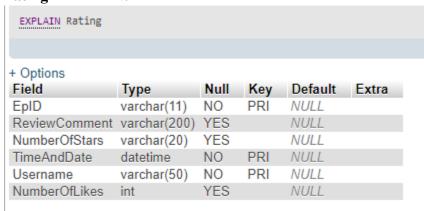
RaterProfile SELECT COUNT:

```
Your SQL query has been executed successfully.

SELECT COUNT(*) FROM RaterProfile

+ Options
COUNT(*)
84
```

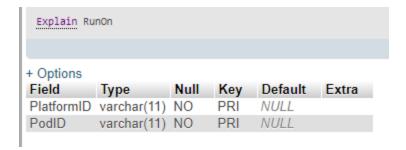
Rating EXPLAIN:



Rating SELECT COUNT:



RunOn EXPLAIN:



RunOn SELECT COUNT:

```
Your SQL query has been executed successfully.

SELECT COUNT (*) FROM RunOn

+ Options
COUNT(*)
49
```

Section 4:

INSERT data from the Podcast "Science Vs":

Insert Statement and data

```
INSERT INTO Podcast (PodID, PodcastFormat, Genre, Title,
NumberOfEpisodes, AgeRating, Description)
VALUES ('102', 'Informational', 'Educational', 'Science Vs', '151',
'M', '
There are a lot of fads, blogs and strong opinions, but then there's
SCIENCE. Science Vs is the show from Gimlet that finds out what's fact,
what's not, and what's somewhere in between. We do the hard work of
sifting through all the science so you don't have to and cover
everything from 5G and Pandemics, to Vaping and Fasting Diets.');
```

Successful Insertion



DELETE Person 98 Who's Nickname is:

Selection of PersonID #98



Successful Deletion



UPDATE data from Person "PersonID#14":

Update Statement

```
UPDATE Person SET Firstname = 'Joseph', Lastname = 'Rogan', Nickname = 'Joe', Title = 'Mr.', Age = '53', Bio = 'Joseph James Rogan (born August 11, 1967) is an American comedian, podcast host, and UFC color commentator. He has also worked as a television host and an actor. Rogan began his career in comedy in August 1988 in the Boston area. ... From 2001 to 2006, he was the host of Fear Factor.', LinkedIn = 'https://www.linkedin.com/in/joe-rogan-b86574111', Twitter = '@joerogan', Facebook = 'https://www.facebook.com/JOEROGAN', Wiki = 'https://en.wikipedia.org/wiki/Joe_Rogan', Youtube = 'https://www.youtube.com/user/PowerfulJRE', Instagram = '@joerogan' WHERE PersonId = 14;
```

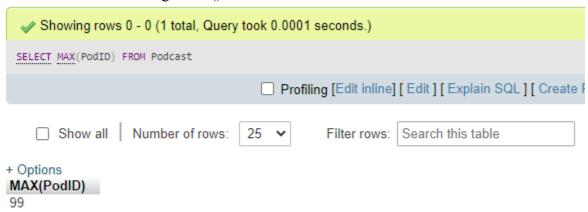
Successful Update



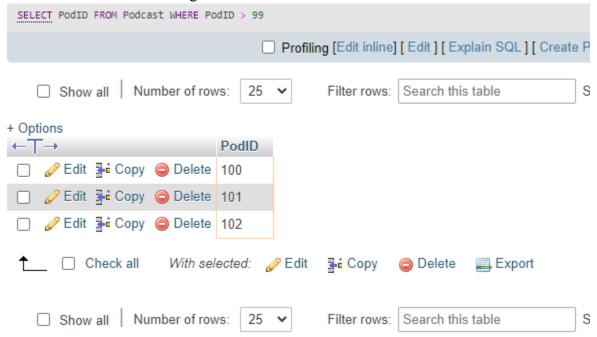
OBSERVATIONS:

Observation 1) At a certain point we wanted to see the greatest PodcastID value by using a SELECT and MAX() command. However the command didn't quite reach the max value so we used an alternative command.

Here is the selection using MAX()

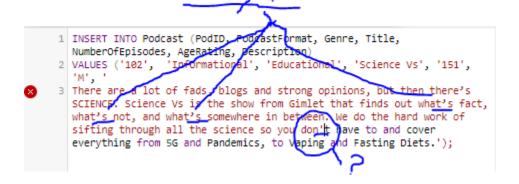


And here is the selection using '>' to double check if it was indeed the max.



I don't quite understand why the MAX() wouldn't return 102. Whatever the case there is more than one way to check the maximum value anyway.

Observation 2) While inserting tuples and specifically text data we had occasional error messages regarding the mediumtext data. It would sometimes trip on the use of (') in words such as "don't". This problem was rare and inconsistent but still created a problem.



1. INSERT, UPDATE, and DELETE on AppearsIn table

Observations: We noticed that we can edit (update) all attribute fields in one UPDATE statements, by separating them with commas

2. INSERT, UPDATE, and DELETE on Artwork table

```
1 row affected. (Query took 0.0020 seconds.)

DELETE FROM 'Artwork' WHERE ArtworkID ="1"
```

Observation: After executing the intended commands in the Artwork Table, there were no alarming alterations. However, changes can be seen outside of the table, such in the Presents Table, where the tuple containing ArtworkID ="1" can no longer be found.

3. INSERT, UPDATE, and DELETE on CurCollection table

Observation: We observed that spaces are not necessary next to "=" sign in UPDATE statement

4. INSERT, UPDATE, and DELETE on Debuts table

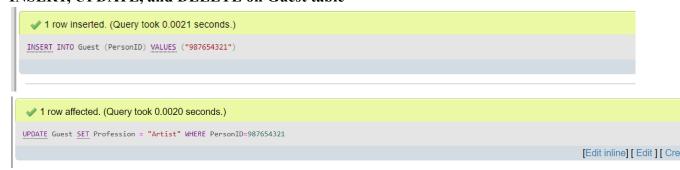
Observation: A noticeable observation that can be seen in the Debuts tables is that in order to update the table the EpID also needed to be stated. This was in order to prevent any duplicates from being accidently updated, For example, to prevent a Podcast with ID "1" from having 10 tuples of the same Episode, each EpID is required to be stated for uniqueness.

5. INSERT, UPDATE, and DELETE on Episode table



Observation: This table was relatively simple in terms of processing commands.

6. INSERT, UPDATE, and DELETE on Guest table



```
1 row affected. (Query took 0.0020 seconds.)

DELETE FROM Guest WHERE PersonID = 987654321

[Edit inline] [Edit] [Create PHP code]
```

Observation: We noticed that we do not need to specify all the attributes in parentheses in INSERT statement; we can just specify the PersonID

7. INSERT, UPDATE, and DELETE on Hosts table



Observations: It is enough to specify the value fort just one attribute for the DELETE statement to erase the whole tuple

8. INSERT, UPDATE, and DELETE on IsListedUnder table

```
# 1 row inserted. (Query took 0.0020 seconds.)

INSERT INTO 'IsListedUnder' ('CurCollectionID', 'EpID') VALUES ("2000", "5000")

### 1 row affected. (Query took 0.0021 seconds.)

UPDATE 'IsListedUnder' SET 'CurCollectionID'="2000", 'EpID'="350" WHERE EpID="5000"

#### 0 rows affected. (Query took 0.0002 seconds.)

DELETE FROM 'IsListedUnder' WHERE EpID ="100"
```

Observation: A few observations can be made in the IsListedUnder table, which connects the Curated Collection and Episode tables. When attempting to insert a new tuple into the table, the system considered the table to be a child table with constraints preventing the action from occurring. Updating and Deleting worked fine, where a change in the Epsiode table can be seen in the IsListedUnder table. The EpisodeID value was set at 2000, but changed in Episode which simultaneously changed the ID in the IsListedUnder Table.

9. INSERT, UPDATE, and DELETE on Offers table



Observation: No problems occurred while executing the given commands on the table.

10. INSERT, UPDATE, and DELETE on Person table

```
INSERT INTO Person (PersonID, Firstname, Lastname, Nickname, Title, Age, Bio, Linkedin, Twitter, Facebook, Wiki, Youtube, Instagram) VALUES ('98', 'Angela', 'Bennet', 'Angie', 'Ms.', '26', 'id consequat in consequat ut nulla sed accumsan felis ut at dolor quis odio', 'https://robohash.org/ettotamodit.png?size=50x50&set=set1', 'http://google.fr/convallis/eget/eleifend/luctus/ultricies/eu/nibh.js?pharetra=convallis&magn', 'http://homestead.com/dictumst/morbi.xml? faucibus=in&cursus=felis&urna', 'https://docs.microsoft.com/en-us/learn/certifications/browse/? jobrole=developer', 'https://login.libproxy.csustan.edu', 'https://blog.lime.link/4-ways-to-check-the-popularity-of-a-podcast/')

[Edit inline] [Edit] [Create PHP code]
```

Show query box

```
✓ 1 row affected. (Query took 0.0021 seconds.)

UPDATE Person SET PersonID = '99' WHERE PersonID = '98'
```

✓ 1 row affected. (Query took 0.0020 seconds.)
DELETE FROM Person WHERE PersonID = '99'

Observation: PersonID key effected multiple tables, so we had to be aware of the order of commands we executed. For instance we could not show INSERT, UPDATE and DELETE commands for a tuple in 'Hosts' if we already had done so for the tuple with the same PersonID in the Person table.

11. INSERT, UPDATE, and DELETE on Platform table

```
INSERT INTO 'Platform' ('PlatformID', 'Name', 'WebsiteURL', 'Device', 'DownloadLink', 'Cost') VALUES ("1000", "The Mdeia", "www.google.com", "Computer", "https://dropbox.com/google", "$100.00")

[E]

### 1 row affected. (Query took 0.0021 seconds.)

### UPDATE 'Platform' SET 'PlatformID'="1000", 'Name'="CS4250 Club", 'WebsiteURL'="https://www.cs.csustan.edu/~mthomas/cs4250/project5.html", 'Cost'="$2.50" WHERE PlatformID ="1000"

[Edit inline] [Edit ] [ Create Platform of the Computer of th
```

Observation: The execution of the commands on the Platform table saw no errors.

12. INSERT, UPDATE, and DELETE on Podcast table



Observations: We can use single quotation marks ('') to designate the attribute values in both UPDATE and DELETE statements

13. INSERT, UPDATE, and DELETE on Podcaster table



Observations: It is possible to write the whole statement in a single line, we do not need to separate DELETE from WHERE part and put them in different lines

14. INSERT, UPDATE, and DELETE on Presents table

Observation: When inserting data into the Presents table, the execution of the insert command failed to process. The reason behind this is due to the lack of data in the Epsidoe table. Originally there was no Episode with ID 5000, and this prevented the system from inserting data since no such Episode with ID 5000 existed.

15. INSERT, UPDATE, and DELETE on RaterProfile table

Observation: The RaterProfile table did not see any errors when going through the commands.

16. INSERT, UPDATE, and DELETE on Rating table

```
Instraction (Query took 0.0020 seconds.)

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES ("ikik", "1111", "2020-05-12 02:30:02")

INSERT INTO Rating (Username, EpID, TimeAndDate) VALUES (Username, E
```

Observation:

We noticed that TimeAndDate required a specific time/date format in INSERT, with date and then time of the day specified

17. INSERT, UPDATE, and DELETE on RunOn table

```
## 1 row inserted. (Query took 0.0020 seconds.)

INSERT INTO 'RunOn' ('PlatformID', 'PodID') VALUES ("1000", "5000")

## 1 row affected. (Query took 0.0021 seconds.)

UPDATE 'RunOn' SET 'PlatformID'="20", 'PodID'="5000" WHERE PodID ="5000"

## 1 row affected. (Query took 0.0020 seconds.)

DELETE FROM 'RunOn' WHERE PodID="5000"

## 1 row affected. (Query took 0.0020 seconds.)
```

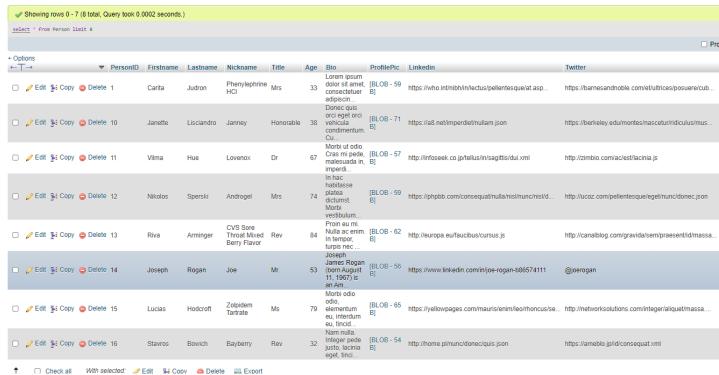
Observation: No alarming errors occurred when executing the commands for the RunOn table. However when updating the tuple with PodID "5000" the PlatformID was going to be set to 10, but since the PlatformID "10" was deleted in the above example, this gave an error saying no such Platform entry existed.

Section 5:

Most of the data found in our podcast database was generated from the Mockaroo website https://mockaroo.com/ that was provided as a useful tool in the resource section of the assignment page. The data was carefully examined to match attributes for our real world Podcast examples and constructed to be unique for each corresponding key, so that no duplicates were made in the process. The method of acquiring the data used in our database was to save the data files provided by Mockaroo as .CSV files and use the import feature found in phpmyadmin. This method allowed for a straightforward and relatively seamless import on all our relation tables. However on a few occasions while importing data, a certain issue would occur where myphpadmin viewed certain data that we were trying to import as incorrect datatypes. What resulted was the need to change the datatypes for the few attributes where the problem arose. These changes can be noticed primarily in our Episode relation, where the attributes were changed to varchar to avoid the data format issues.

Section 6: Samples of tuples for each relation

1) "Person" table:

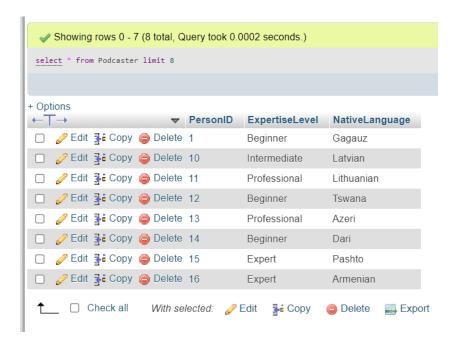


(Continues on next page)



Facebook	Wiki	Youtube	Instagram
https://abc.net.au/platea/dictumst.html	https://columbia.edu/enim/leo/rhoncus.js	https://twitter.com/sodales.jsp	http://mozilla.com/velit/id/pretium/iaculis/diam/e
https://quantcast.com/diam/erat/fermentum.jsp	http://wufoo.com/odio/elementum/eu/interdum/eu/tin	https://pcworld.com/amet/justo/morbi/ut/odio/cras/	http://ihg.com/integer/aliquet.jsp
http://sitemeter.com/nunc/viverra/dapibus/nulla/su	https://gnu.org/non/lectus/aliquam/sit/amet.html	http://squidoo.com/varius.png	http://wp.com/at/nulla/suspendisse/potenti/cras.js
http://zdnet.com/vitae/ipsum/aliquam/non/mauris/mo	http://wisc.edu/habitasse/platea/dictumst/morbi/ve	http://zdnet.com/magna/at.xml	https://hibu.com/augue/vestibulum/rutrum/rutrum/ne
https://themeforest.net/vestibulum.png	https://auda.org.au/massa/donec/dapibus.jpg	https://liveinternet.ru/sed/accumsan/felis/ut/at/d	https://nih.gov/fringilla/rhoncus/mauris/enim/leo
https://www.facebook.com/JOEROGAN/	https://en.wikipedia.org/wiki/Joe_Rogan	https://www.youtube.com/user/PowerfulJRE	@joerogan
http://msu.edu/tincidunt/lacus/at/velit/vivamus/ve	http://mozilla.com/justo/in/blandit.png	http://ocn.ne.jp/curae/donec/pharetra.jsp	https://ovh.net/consequat.xml
https://imdb.com/at/feugiat/non/pretium/quis.json	http://gizmodo.com/turpis/enim/blandit/mi/in.xml	https://nps.gov/mattis/pulvinar/nulla/pede/ullamco	http://flickr.com/pharetra/magna/ac/consequat/metu

2) "Podcaster" table:



3) "Guest" table:



4) "Podcast" table:



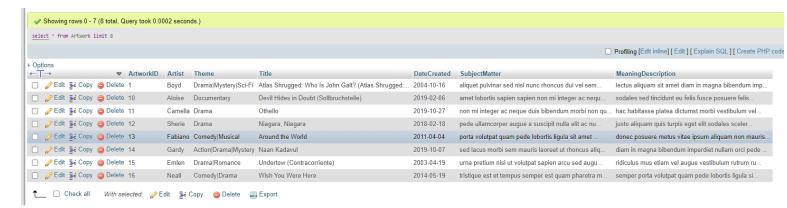
5) "Episode" table:



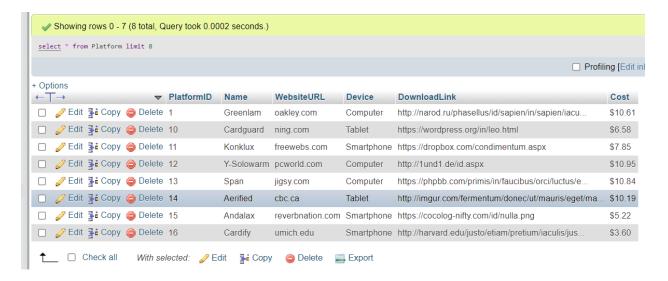
6) "CuratedCollection" table:



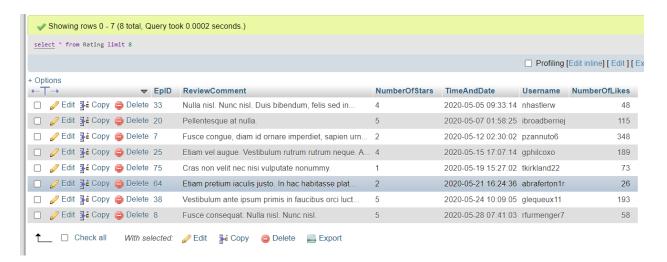
7) "Artwork" table:



8) "Platform" table:



9) "Rating" table:



10) "RaterProfile" table:



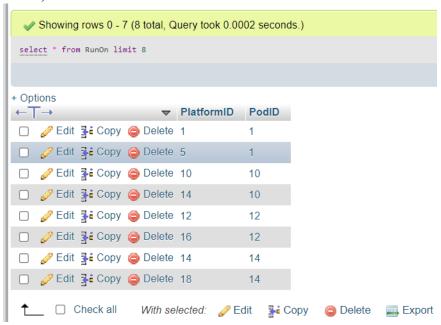
11) "Hosts" table:



12) "AppearsIn" table:



13) "RunOn" table:



14) "Offers" table:



15) "Debuts" table:



16) "IsListedUnder" table:



17) "Presents" table:



Section 7:

Project members contact information and contribution:

Victor Cardenas (email: <u>vcardenas4@csustan.edu</u>); Helped load the relations in the database in Section 2 as well as import data into the database. Also worked on Section 5 of the assignment. Helped with Section 4 (trying out the INSERT, DELETE and UPDATE commands), and provided screenshots.

Nicholas Guzman (email: nguzman9@csustan.edu); Helped write and load tables into database in Section 2. Screenshotted executions of SELECT COUNT and EXPLAIN commands in Section 3. Worked on Section 4 by sampling different commands, and made some observations. Helped with Section 4 (trying out the INSERT, DELETE and UPDATE commands), and provided screenshots.

Kristijan Hornung (email: khornung@csustan.edu); helped explain changes to the initial relations from Part 4 in Section 1, provided the SQL CREATE TABLE statements in Section 2, helped set the foreign key constraints in relationship tables to match the schema requirements, set certain attributes to "Unique" in all tables where that was applicable, helped with section 6 (samples of tuples for each relation). Helped with Section 4 (trying out the INSERT, DELETE and UPDATE commands).