By

Aishwarya Soundararajan

Personalized Subscription Management System For Online Entertainment

Index

Project Summary………………………………….2

Business Rules…………………………………….3

Entity Attribute Table……………………………..4

Entity Relationship Diagram……………………...9

Database Infrastructure…………………………...10

Creation of Tables………………………………...10

Insertion of Values………………………………..14

Major Data Question ………..................................21

Forms……………………………………………...23

Reports…………………………………………….27

Feedback Implementation…………………………31

Project Summary

This project focusses on designing a database that provides personalized subscriptions on movies and TV shows which are hosted on different streaming platforms such as Amazon prime, Netflix, Hulu, Google Play Store and YouTube. The current scenario is that personalized subscriptions are provided to the customers of the respective individual platforms. The proposed database would help the customers to take decisions about watching the movies and TV shows on respective platforms based on several attributes that are stored in the system such as platform, names of the movies and TV shows, description, genres, ratings, price, location, language and running time as per their interest.

The number of online entertainment streaming platforms are increasing widely which makes it difficult for the customers to decide upon which platform they could opt for watching movies and TV shows in their favorite genre with low price, highest ratings available in their location. It takes a lot of time and becomes a tedious task if the customer tries to consider these several deciding factors before he/she could watch their most preferred movie or TV show. This system makes this tiresome task as easy and simple as it could make for the customer. Using this system which has integrated several factors, it makes it easier for the customers to optimize their effort, time and money and make quick decisions. The database is designed in such a way that the customers can query the database as per his/her choices and need not go through the individual sites of different platforms but can have the details available on a single web page.

This report comprises of the system implementation for the proposed system, with the functionalities, dependencies and relationships between various entities and attributes. Business rules and several data questions are stated along with a fully attributed Entity Relationship Diagram (ERD) addressing the major issues in this system. The implementation report also consists of queries and information which may be required by various users of this system.

Business rules

A customer can look for movies and TV shows streamed on multiple platforms.

A movie or TV show can have different prices in different geographic locations.

A particular movie and TV show are available on multiple platforms for each geographic location.

Movies and TV shows are available in one or more than one language.

Movies and TV shows are available in one or more than one genre.

Movies and TV shows are available in one or more than one location.

A movie or a TV show can have only one description.

A movie or a TV show can have only one rating.

Customer can watch movies and TV shows in more than one preferred language.

Customers can watch movies and TV shows in more than one preferred genre.

The price for a movie or TV show includes the price of the movie / TV show and the subscription fee.

Subscription fee consists of the yearly subscription fee for the particular platform and not the monthly subscription fee.

The price of a movie or TV show is given different currency units as per the geographic location.

Entity and Attribute table

1. Customers– This entity captures information about the customers/users of the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Customers | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | CustID | VARCHAR(20) | NOT NULL | Unique Identifier for Customer |
| FOREIGN KEY | LocationID | VARCHAR(20) | NOT NULL | Foreign key identifying Location |
| OTHER ATTRIBUTES | CustFName | VARCHAR(30) | NOT NULL | First Name of the Customer |
|  | CustLName | VARCHAR(30) | NOT NULL | Last Name of the Customer |
|  | CustEmail | VARCHAR(30) | NOT NULL | Email ID of the Customer |

1. Location– This entity captures information about the location of customers and the location in which the media content is available

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Location | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | LocationID | VARCHAR(20) | NOT NULL | Unique Identifier of the location |
| OTHER ATTRIBUTES | LocationName | VARCHAR(40) | NOT NULL | Name of the location |
|  | LocationCurrencyType | VARCHAR(10) | NOT NULL | Currency type in the respective location |

1. Platforms– This entity captures information about the different platforms in which the media content is streamed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Platforms | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | PlatformID | VARCHAR(20) | NOT NULL | Primary Key for the Platforms |
| OTHER ATTRIBUTES | PlatformName | VARCHAR(30) | NOT NULL | Name of the Platforms |

1. Genre– This entity captures the information about the different genres.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Genre | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | GenreID | VARCHAR(20) | NOT NULL | Primary key of the genre |
| OTHER ATTRIBUTES | GenreName | VARCHAR(30) | NOT NULL | Name of the genre |

1. Price– This entity captures the information about the prices for each media content.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Price | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | PriceID | VARCHAR(20) | NOT NULL | Primary key for the available prices |
| OTHER ATTRIBUTES | Price | Decimal(6,2) | NOT NULL | Prices of the media content |

1. Language– This entity captures the information about the languages in which each media content is available and the location of the customers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  MediaLanguage | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | LanguageID | VARCHAR(20) | NOT NULL | Primary Key for the language |
| OTHER ATTRIBUTES | Language Name | VARCHAR(40) | NOT NULL | Name of the language |

1. EntertainmentDescription– This entity captures the information about the description of the media content.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  EntertainmentDescription | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | DescID | VARCHAR(20) | NOT NULL | Primary key for the description |
| OTHER ATTRIBUTES | DescCast | VARCHAR(100) | NULL | Cast of the media content |
|  | DescSynopsis | VARCHAR(1000) | NULL | Synopsis of the movie/TV Show |
|  | DescDirector | VARCHAR(50) | NULL | Director of the movie/ TV Show |
|  | DescMusicComposer | VARCHAR(50) | NULL | Music Composer of the movie/TV show |
|  | DescRunningTime | VARCHAR(10) | NULL | Running Time of the media content |

1. Rating– This entity captures the information about the rating and reviewed source of the media content.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Rating | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | RatingID | VARCHAR(20) | NOT NULL | Primary Key for the Rating |
| OTHER ATTRIBUTES | RatingScore | Decimal(2,1) | NOT NULL | Rating Score |
|  | ReviewedBy | VARCHAR(10) | NOT NULL | Reviewed Source - ImDb |

1. MediaContent– This entity captures the information about the rating and reviewed source of the media content

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  MediaContent | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY | MediaContentID | VARCHAR(20) | NOT NULL | Primary key for the Media Content |
| OTHER ATTRIBUTES | MediaContentName | VARCHAR(30) | NOT NULL | Name of the Movie/TV show |
|  | MediaContentType | VARCHAR(30) | NOT NULL | Type of the media content – Movie or TV Show |
| FOREIGN KEY | DescID | VARCHAR(20) | NOT NULL | Foreign key identifying the description of the mediacontent |
| FOREIGN KEY | RatingID | VARCHAR(20) | NOT NULL | Foreign key identifying the rating of the media |

1. SubscriptionFee– This entity captures the information about the subscription details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  SubscriptionFee | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | LocationID | VARCHAR(20) | NOT NULL | Composite Key identifying the location |
| PRIMARY KEY, FOREIGN KEY | PlatformID | VARCHAR(20) | NOT NULL | Composite Key identifying the platform |
| OTHER ATTRIBUTES | SubscriptionFee | Decimal(6,2) | NOT NULL | Describes the subscription fee on each platform |

1. MediaContentLocDetails - This composite table comprises of media content ID, location ID, price ID, PlatformID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  MediaContentLocDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | MediaContentID | VARCHAR(20) | NOT NULL | Composite key for media content |
| PRIMARY KEY, FOREIGN KEY | LocationID | VARCHAR(20) | NOT NULL | Composite key for location |
| PRIMARY KEY, FOREIGN KEY | PlatformID | VARCHAR(20) | NOT NULL | Composite key identifying platform |
| PRIMARY KEY, FOREIGN KEY | PriceID | DECIMAL(6,2) | NOT NULL | Composite key identifying Price |

1. ContentGenreDetails - This composite table comprises of media content ID, GenreID which describes the different genres of the media content

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  ContentGenreDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | GenreID | VARCHAR(20) | NOT NULL | Composite key identifying different genres |
| PRIMARY KEY, FOREIGN KEY | MediaContentID | VARCHAR(20) | NOT NULL | Composite Key identifying the media content names |

1. MediaContentLangDetails - This composite table comprises of media content ID, LanguageID which identifies the different languages in which the media content is available

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  MediaContentLangDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | LanguageID | VARCHAR(20) | NOT NULL | Composite key identifying different languages |
| PRIMARY KEY, FOREIGN KEY | MediaContentID | VARCHAR(20) | NOT NULL | Composite Key identifying the media content names |

1. PlatformUsageDetails- This composite table identifies the platform usage by different customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  PlatformUsageDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | PlatformID | VARCHAR(20) | NOT NULL | Composite key identifying different platforms |
| PRIMARY KEY, FOREIGN KEY | CustID | VARCHAR(20) | NOT NULL | Composite Key identifying different customers |

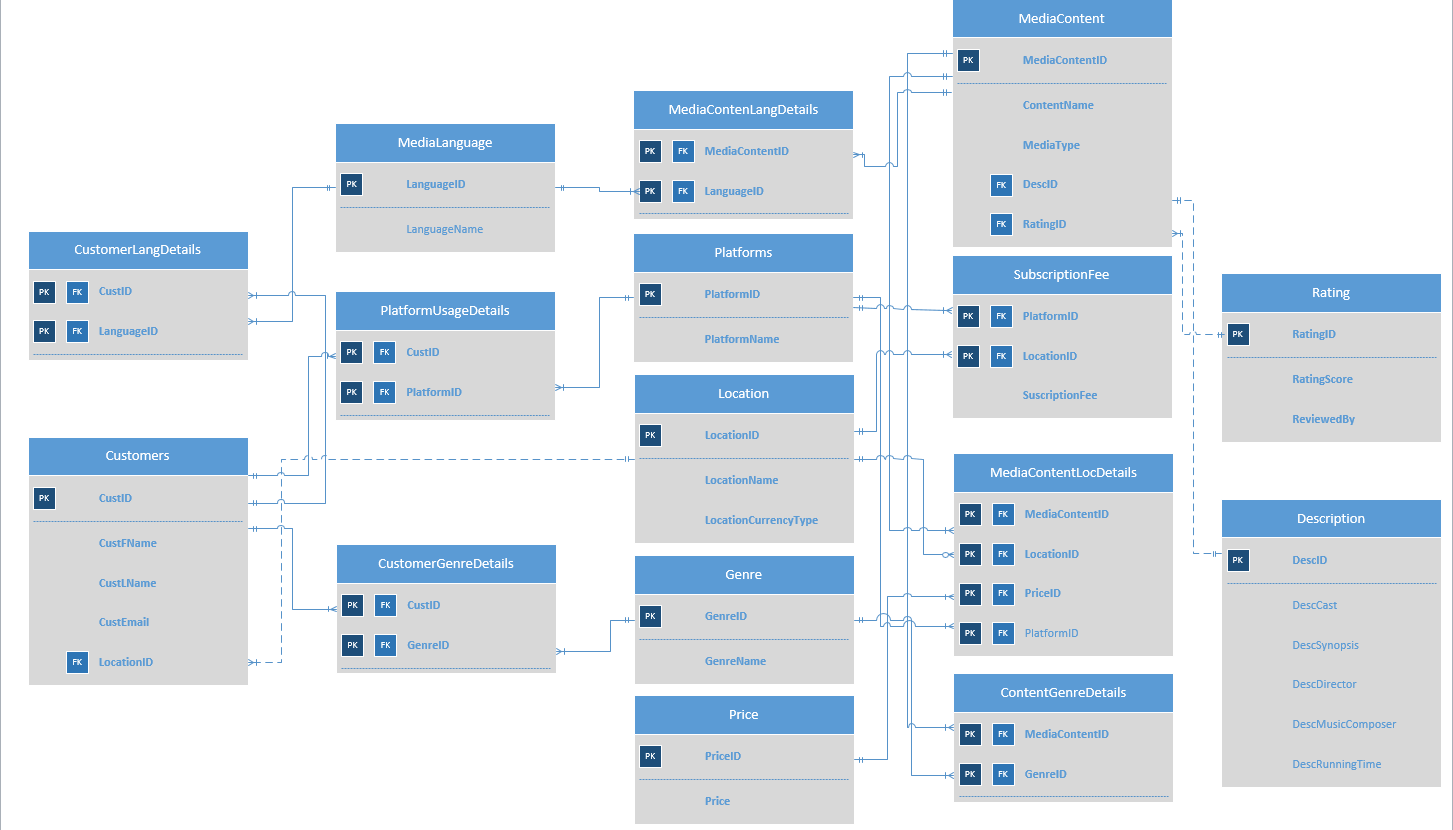
1. CustomerGenreDetails- This composite table identifies the genres preferred by different customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  CustomerGenreDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | GenreID | VARCHAR(20) | NOT NULL | Composite key identifying different genres |
| PRIMARY KEY, FOREIGN KEY | CustID | VARCHAR(20) | NOT NULL | Composite Key identifying different customers |

1. CustomerLangDetails- This composite table identifies the languages used by different customers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  CustomerLangDetails | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| PRIMARY KEY, FOREIGN KEY | LanguageID | VARCHAR(20) | NOT NULL | Composite key identifying different languages |
| PRIMARY KEY, FOREIGN KEY | CustID | VARCHAR(20) | NOT NULL | Composite Key identifying different customers |

ENTITY-RELATIONSHIP DIAGRAM:



DATABASE INFRASTRUCTURE

The database system infrastructure used is based on a client-server model. SQL Server is used as the database engine and Access is used as the interface design tool. Data is inserted, deleted, updated, and queried from the SQL Server database with the help of forms on Access. Useful data stored on the SQL Server database can also be viewed with the help of reports generated through Access

CREATION OF TABLES

CREATE TABLE EntertainmentDescription(

DescID VARCHAR(20) NOT NULL PRIMARY KEY,

DescCast VARCHAR(100),

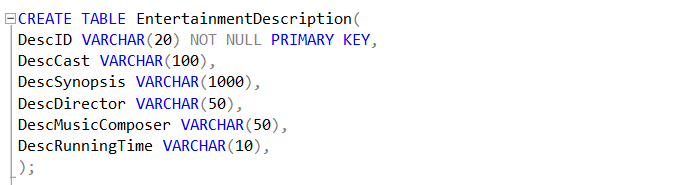
DescSynopsis VARCHAR(1000),

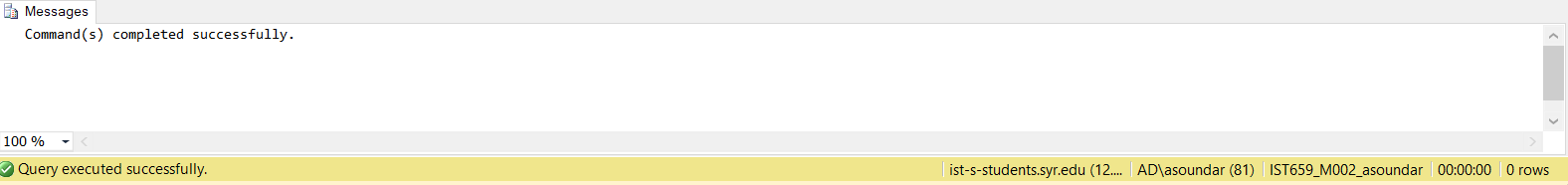
DescDirector VARCHAR(50),

DescMusicComposer VARCHAR(50),

DescRunningTime VARCHAR(10),

);





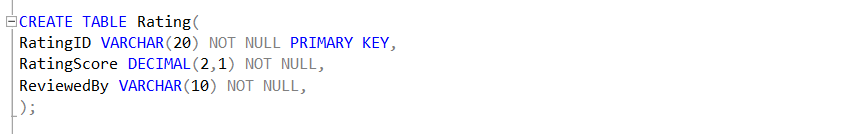
CREATE TABLE Rating(

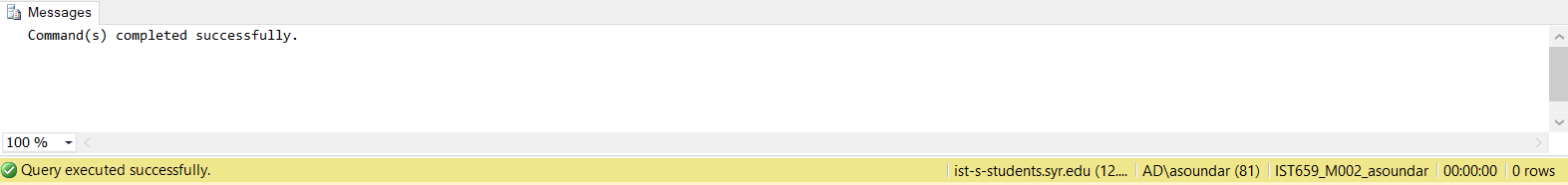
RatingID VARCHAR(20) NOT NULL PRIMARY KEY,

RatingScore DECIMAL(2,1) NOT NULL,

ReviewedBy VARCHAR(10) NOT NULL,

);





CREATE TABLE MediaContent(

MediaContentID VARCHAR(20) NOT NULL PRIMARY KEY,

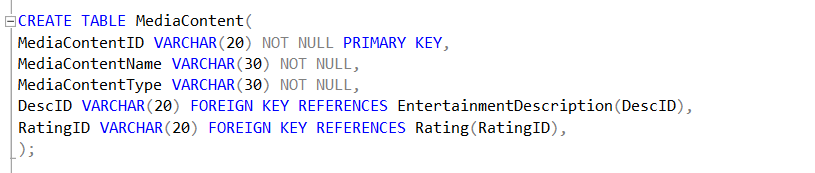
MediaContentName VARCHAR(30) NOT NULL,

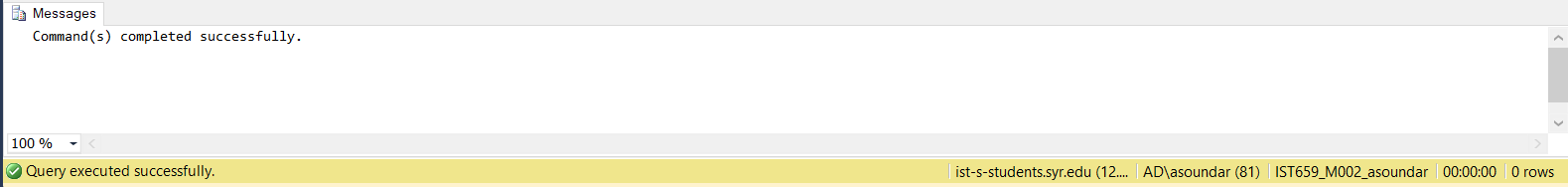
MediaContentType VARCHAR(30) NOT NULL,

DescID VARCHAR(20) FOREIGN KEY REFERENCES EntertainmentDescription(DescID),

RatingID VARCHAR(20) FOREIGN KEY REFERENCES Rating(RatingID),

);



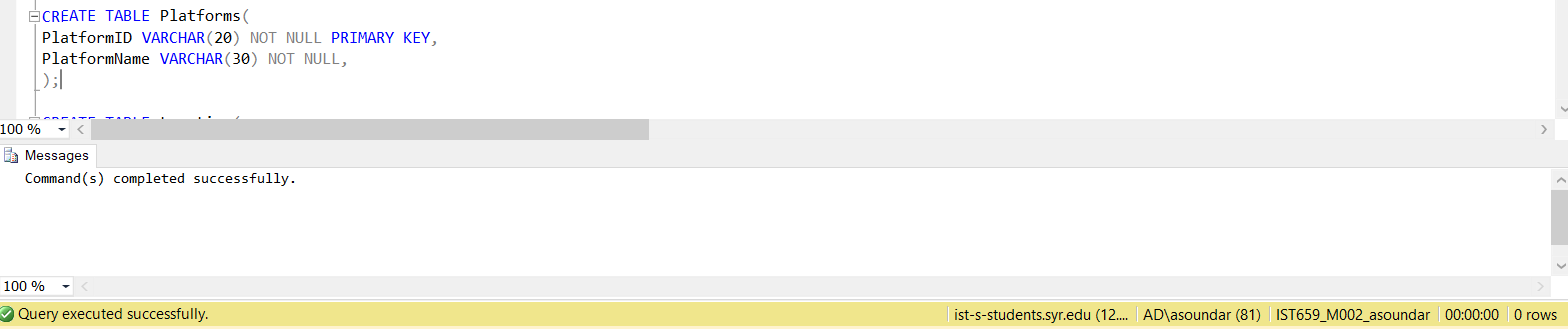


CREATE TABLE Platforms(

PlatformID VARCHAR(20) NOT NULL PRIMARY KEY,

PlatformName VARCHAR(30) NOT NULL,

);



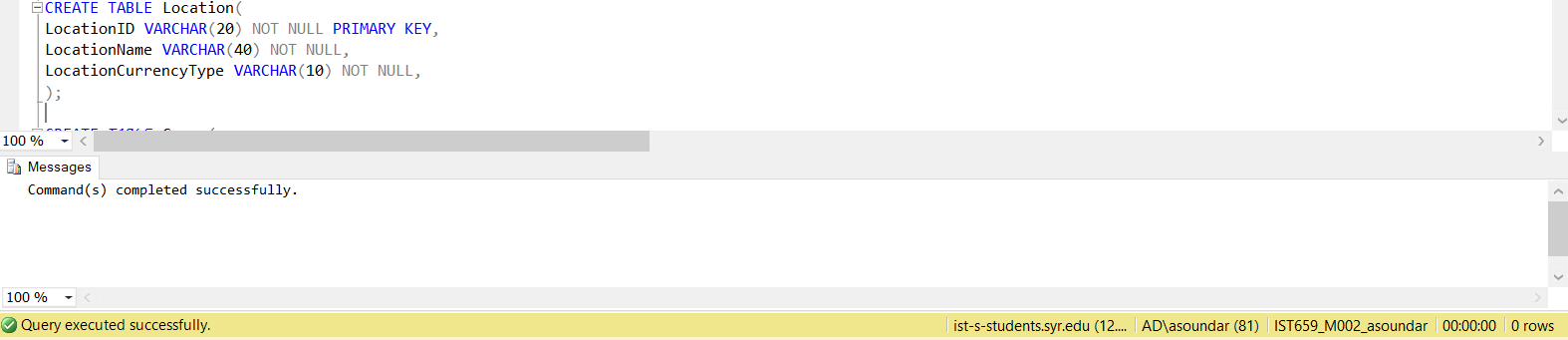
CREATE TABLE Location(

LocationID VARCHAR(20) NOT NULL PRIMARY KEY,

LocationName VARCHAR(40) NOT NULL,

LocationCurrencyType VARCHAR(10) NOT NULL,

);

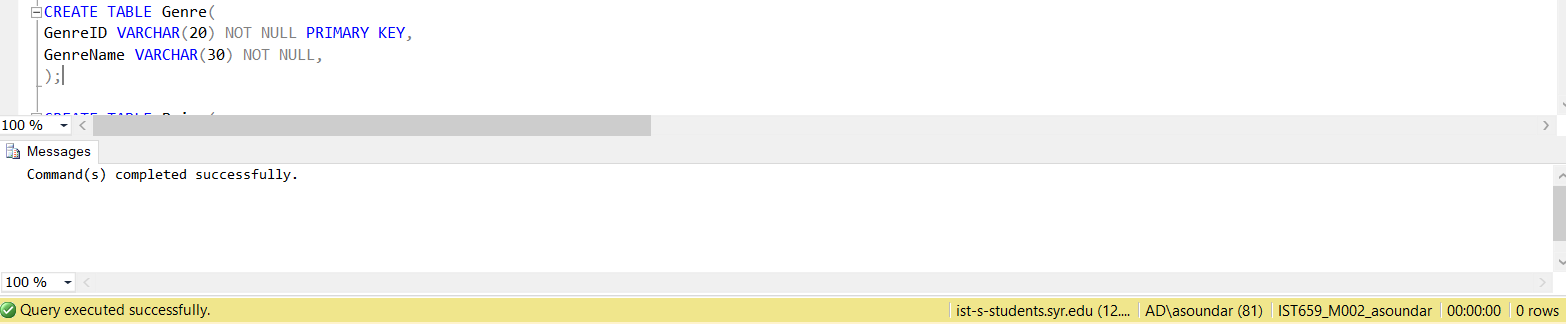


CREATE TABLE Genre(

GenreID VARCHAR(20) NOT NULL PRIMARY KEY,

GenreName VARCHAR(30) NOT NULL,

);

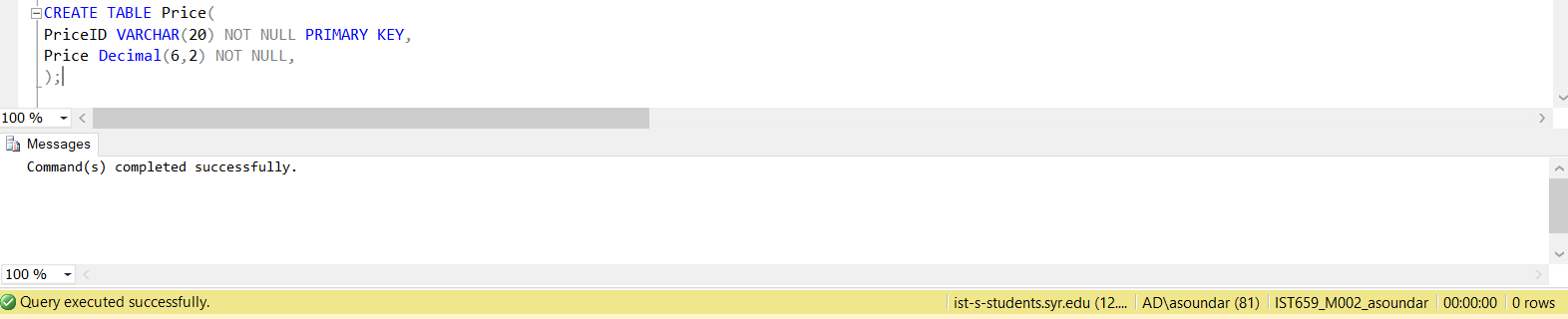


CREATE TABLE Price(

PriceID VARCHAR(20) NOT NULL PRIMARY KEY,

Price Decimal(6,2) NOT NULL,

);



CREATE TABLE MediaContentLocDetails(

MediaContentID VARCHAR(20) NOT NULL,

LocationID VARCHAR(20) NOT NULL,

PriceID VARCHAR(20) NOT NULL,

PlatformID VARCHAR(20) NOT NULL

constraint MediaContentLocDetails\_PK PRIMARY KEY (MediaContentID, LocationID, PriceID, PlatformID),

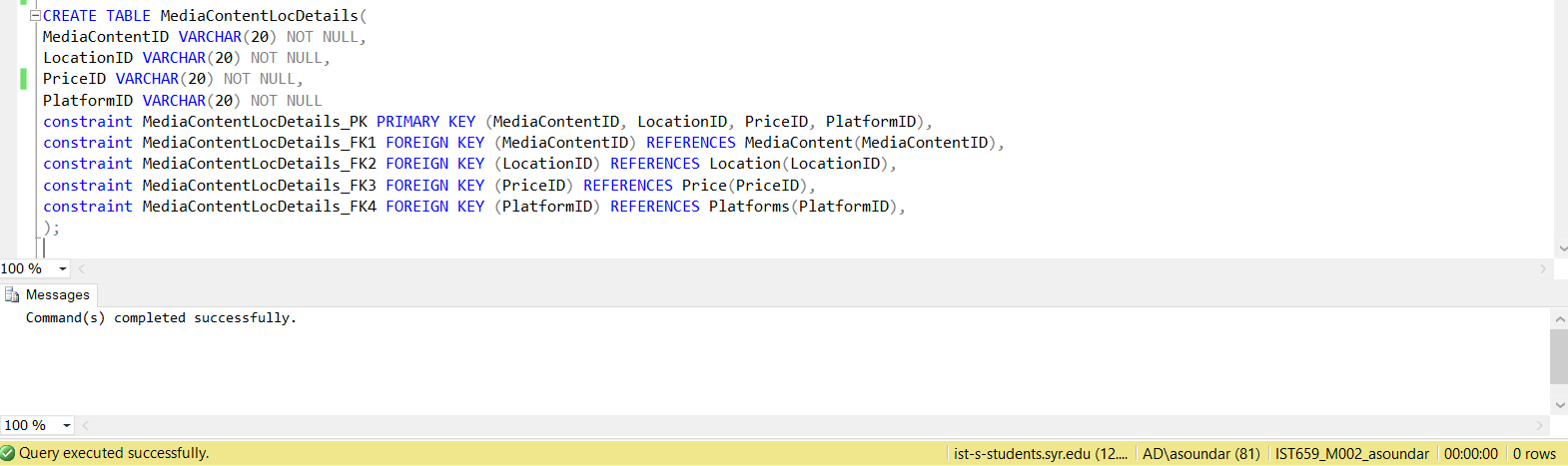
constraint MediaContentLocDetails\_FK1 FOREIGN KEY (MediaContentID) REFERENCES MediaContent(MediaContentID),

constraint MediaContentLocDetails\_FK2 FOREIGN KEY (LocationID) REFERENCES Location(LocationID),

constraint MediaContentLocDetails\_FK3 FOREIGN KEY (PriceID) REFERENCES Price(PriceID),

constraint MediaContentLocDetails\_FK4 FOREIGN KEY (PlatformID) REFERENCES Platforms(PlatformID),

);



CREATE TABLE SubscriptionFee(

LocationID VARCHAR(20) NOT NULL,

PlatformID VARCHAR(20) NOT NULL,

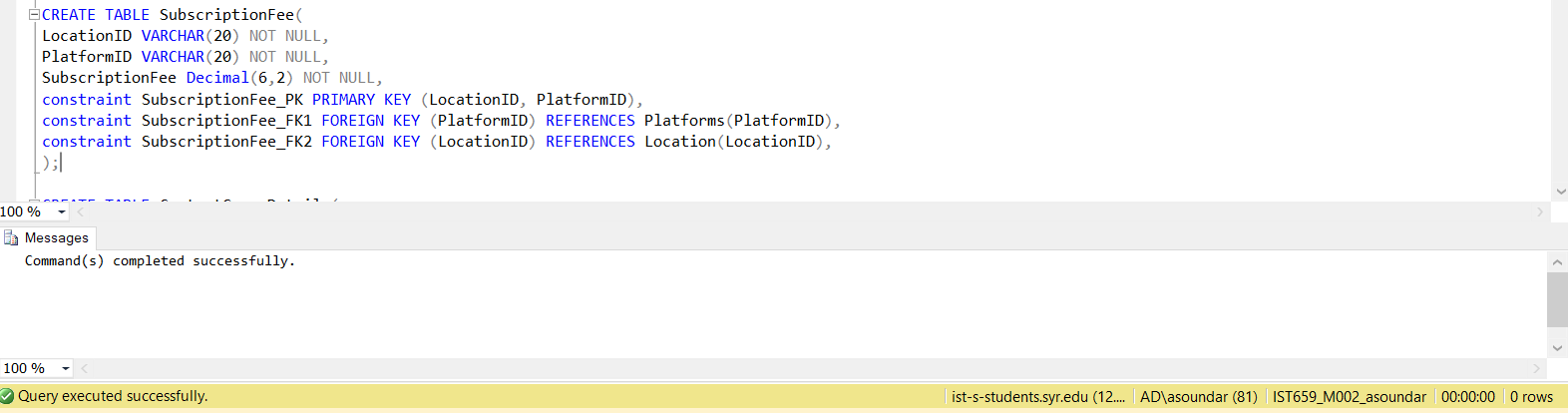
SubscriptionFee Decimal(6,2) NOT NULL,

constraint SubscriptionFee\_PK PRIMARY KEY (LocationID, PlatformID),

constraint SubscriptionFee\_FK1 FOREIGN KEY (PlatformID) REFERENCES Platforms(PlatformID),

constraint SubscriptionFee\_FK2 FOREIGN KEY (LocationID) REFERENCES Location(LocationID),

);



CREATE TABLE ContentGenreDetails(

MediaContentID VARCHAR(20) NOT NULL,

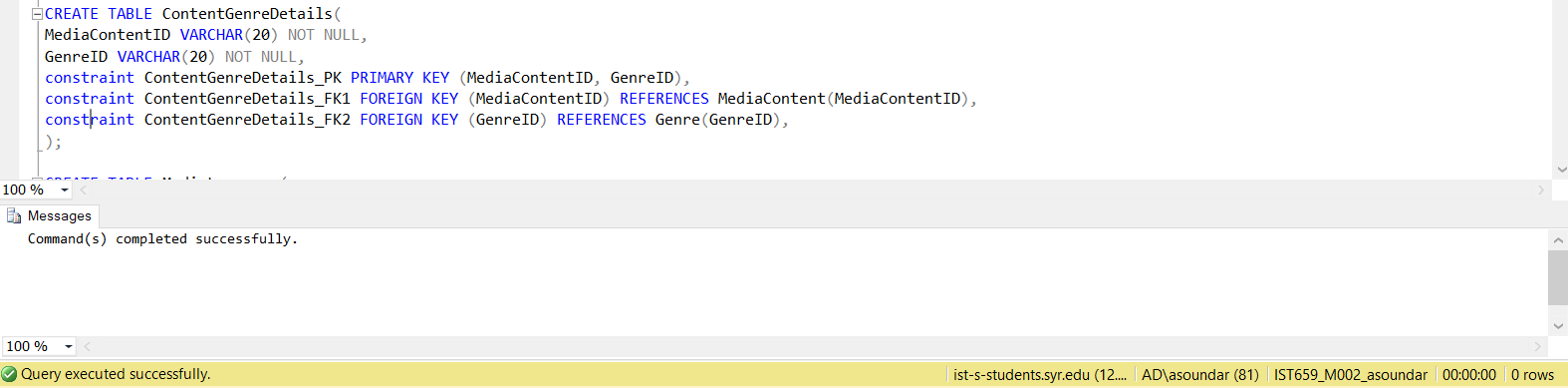
GenreID VARCHAR(20) NOT NULL,

constraint ContentGenreDetails\_PK PRIMARY KEY (MediaContentID, GenreID),

constraint ContentGenreDetails\_FK1 FOREIGN KEY (MediaContentID) REFERENCES MediaContent(MediaContentID),

constraint ContentGenreDetails\_FK2 FOREIGN KEY (GenreID) REFERENCES Genre(GenreID),

);

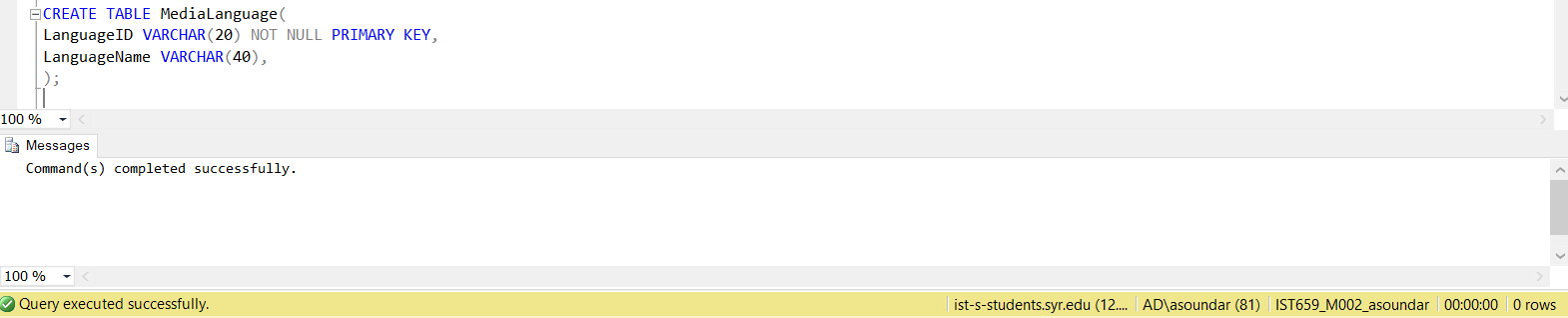


CREATE TABLE MediaLanguage(

LanguageID VARCHAR(20) NOT NULL PRIMARY KEY,

LanguageName VARCHAR(40),

);



CREATE TABLE Customers(

CustID VARCHAR(20) NOT NULL PRIMARY KEY,

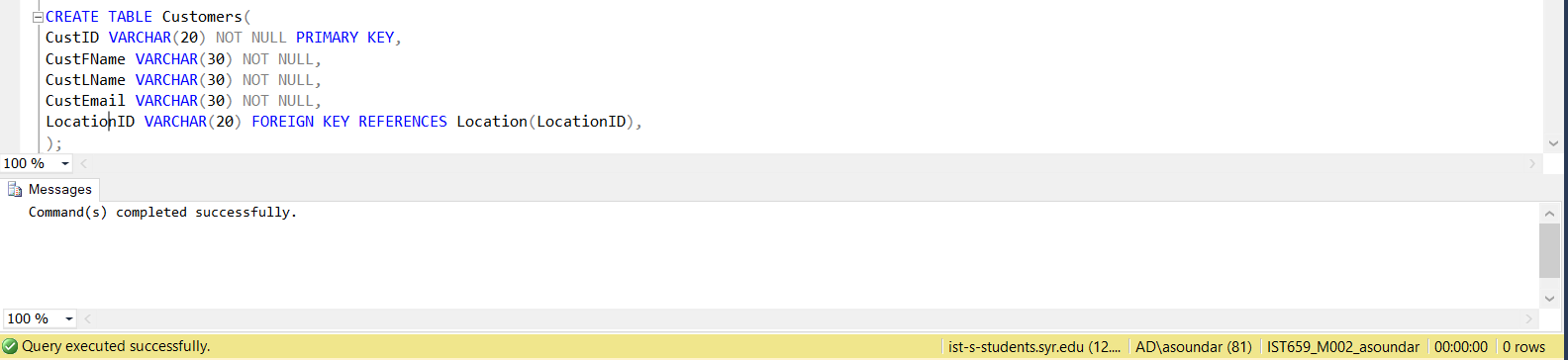
CustFName VARCHAR(30) NOT NULL,

CustLName VARCHAR(30) NOT NULL,

CustEmail VARCHAR(30) NOT NULL,

LocationID VARCHAR(20) FOREIGN KEY REFERENCES Location(LocationID),

);



CREATE TABLE CustomerLangDetails(

CustID VARCHAR(20) NOT NULL,

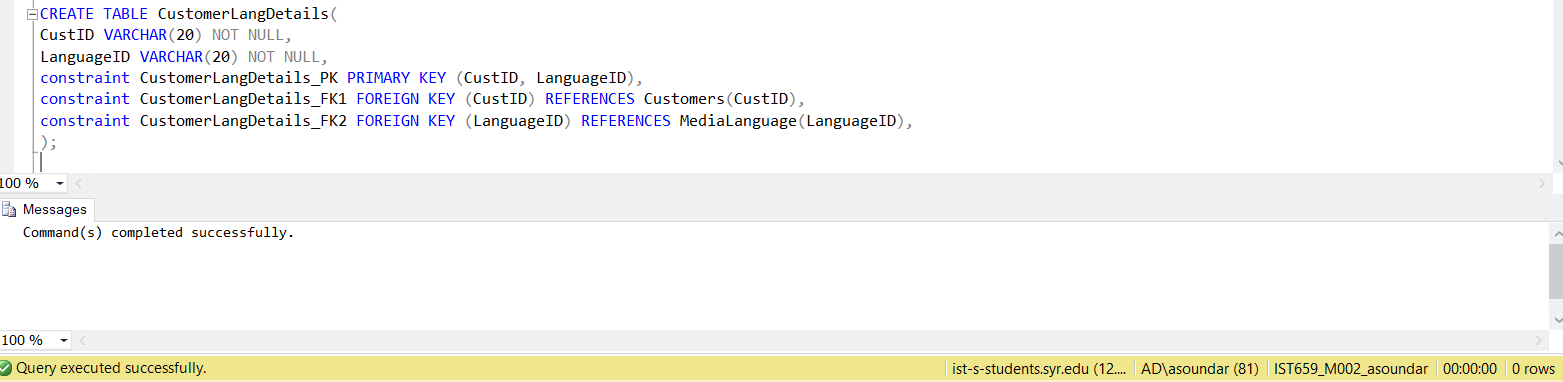
LanguageID VARCHAR(20) NOT NULL,

constraint CustomerLangDetails\_PK PRIMARY KEY (CustID, LanguageID),

constraint CustomerLangDetails\_FK1 FOREIGN KEY (CustID) REFERENCES Customers(CustID),

constraint CustomerLangDetails\_FK2 FOREIGN KEY (LanguageID) REFERENCES MediaLanguage(LanguageID),

);



CREATE TABLE PlatformUsageDetails(

CustID VARCHAR(20) NOT NULL,

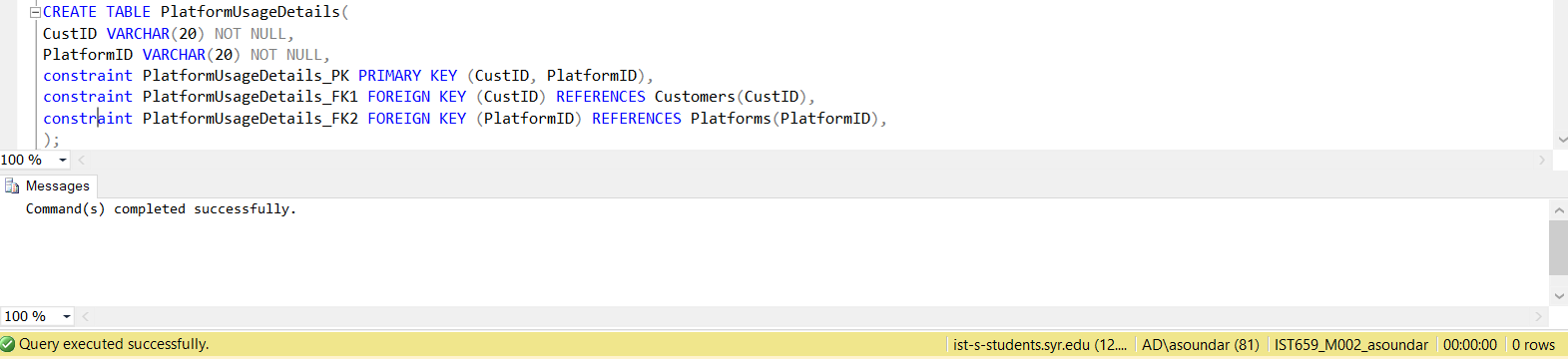
PlatformID VARCHAR(20) NOT NULL,

constraint PlatformUsageDetails\_PK PRIMARY KEY (CustID, PlatformID),

constraint PlatformUsageDetails\_FK1 FOREIGN KEY (CustID) REFERENCES Customers(CustID),

constraint PlatformUsageDetails\_FK2 FOREIGN KEY (PlatformID) REFERENCES Platforms(PlatformID),

);



CREATE TABLE CustomerGenreDetails(

CustID VARCHAR(20) NOT NULL,

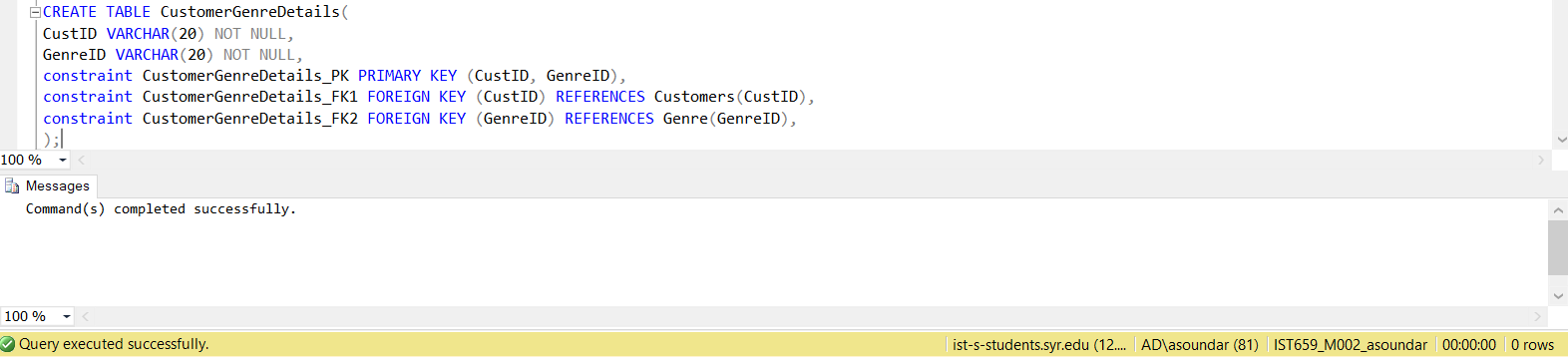
GenreID VARCHAR(20) NOT NULL,

constraint CustomerGenreDetails\_PK PRIMARY KEY (CustID, GenreID),

constraint CustomerGenreDetails\_FK1 FOREIGN KEY (CustID) REFERENCES Customers(CustID),

constraint CustomerGenreDetails\_FK2 FOREIGN KEY (GenreID) REFERENCES Genre(GenreID),

);



CREATE TABLE MediaContentLangDetails(

MediaContentID VARCHAR(20) NOT NULL,

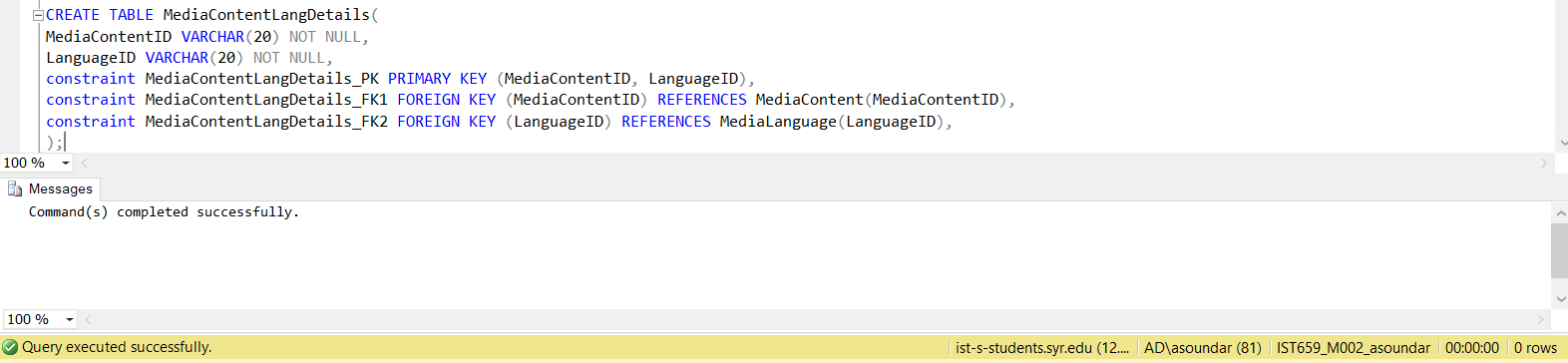
LanguageID VARCHAR(20) NOT NULL,

constraint MediaContentLangDetails\_PK PRIMARY KEY (MediaContentID, LanguageID),

constraint MediaContentLangDetails\_FK1 FOREIGN KEY (MediaContentID) REFERENCES MediaContent(MediaContentID),

constraint MediaContentLangDetails\_FK2 FOREIGN KEY (LanguageID) REFERENCES MediaLanguage(LanguageID),

);



Insertion of values

INSERT INTO EntertainmentDescription(DescID, DescCast, DescDirector, DescSynopsis, DescMusicComposer, DescRunningTime) VALUES (1111, 'Akshay Kumar, Mouni Roy, Kunal Kapoor', 'Reema Kagti', 'Set in 1948, the historical story of first Olympic medal posts the independence of India', 'Sachin-Jigar', 145)

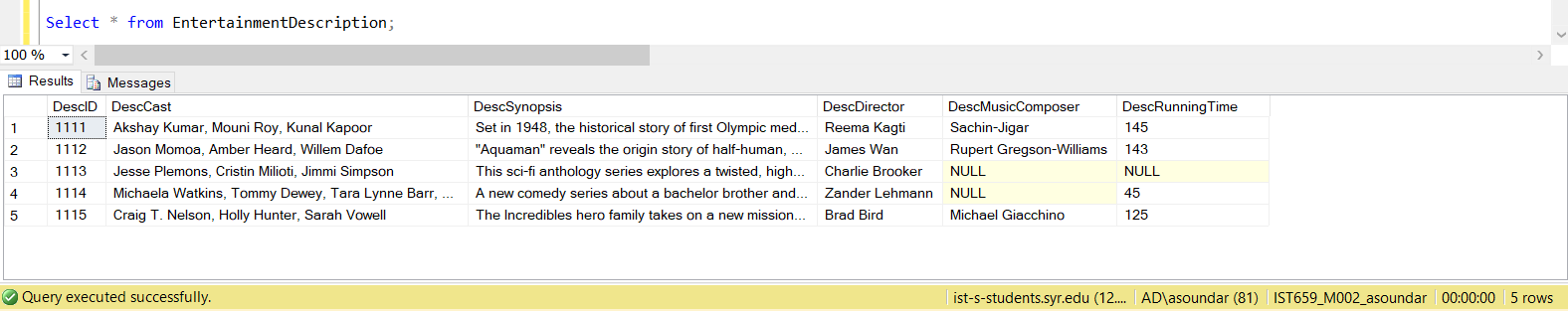
INSERT INTO EntertainmentDescription(DescID, DescCast, DescDirector, DescSynopsis, DescMusicComposer, DescRunningTime) VALUES (1112, 'Jason Momoa, Amber Heard, Willem Dafoe', 'James Wan', '"Aquaman" reveals the origin story of half-human, half-Atlantean Arthur Curry and takes him on the journey of his lifetime-to discover if he is worthy of who he was born to be a king.', 'Rupert Gregson-Williams', 143)

INSERT INTO EntertainmentDescription(DescID, DescCast, DescDirector, DescSynopsis, DescMusicComposer, DescRunningTime) VALUES (1113, 'Jesse Plemons, Cristin Milioti, Jimmi Simpson', 'Charlie Brooker', 'This sci-fi anthology series explores a twisted, high-tech near-future where greatest innovations and darkest instincts of humans collide',NULL,NULL)

INSERT INTO EntertainmentDescription(DescID, DescCast, DescDirector, DescSynopsis, DescMusicComposer, DescRunningTime) VALUES (1114, 'Michaela Watkins, Tommy Dewey, Tara Lynne Barr, Nyasha Hatendi, Frances Conroy, Julie Berman', 'Zander Lehmann', 'A new comedy series about a bachelor brother and his newly divorced sister living under one roof again.',NULL, 45)

INSERT INTO EntertainmentDescription(DescID, DescCast, DescDirector, DescSynopsis, DescMusicComposer, DescRunningTime) VALUES (1115, 'Craig T. Nelson, Holly Hunter, Sarah Vowell', 'Brad Bird', 'The Incredibles hero family takes on a new mission, which involves a change in family roles', 'Michael Giacchino', 125)

Select \* from EntertainmentDescription;



INSERT INTO Rating(RatingID, RatingScore, ReviewedBy) VALUES (1121, 7.5, 'IMDb')

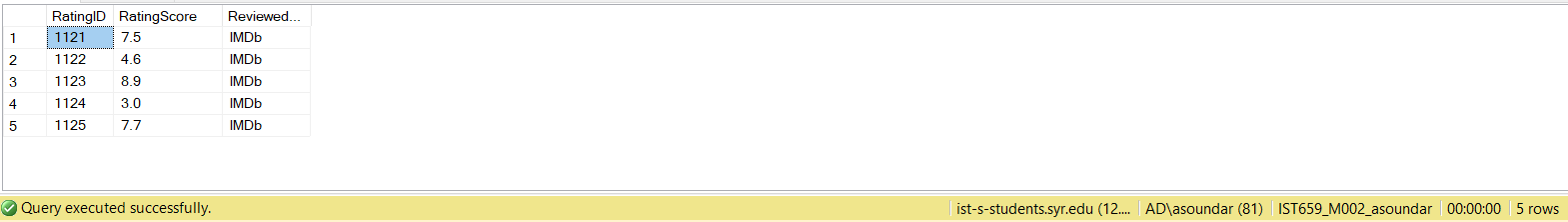
INSERT INTO Rating(RatingID, RatingScore, ReviewedBy) VALUES (1122, 4.6, 'IMDb')

INSERT INTO Rating(RatingID, RatingScore, ReviewedBy) VALUES (1123, 8.9, 'IMDb')

INSERT INTO Rating(RatingID, RatingScore, ReviewedBy) VALUES (1124, 3, 'IMDb')

INSERT INTO Rating(RatingID, RatingScore, ReviewedBy) VALUES (1125, 7.7, 'IMDb')

Select \* from Rating;



INSERT INTO MediaContent(MediaContentID, MediaContentName, MediaContentType, DescID, RatingID) VALUES ('M1131', 'Gold', 'Movie', 1111, 1121)

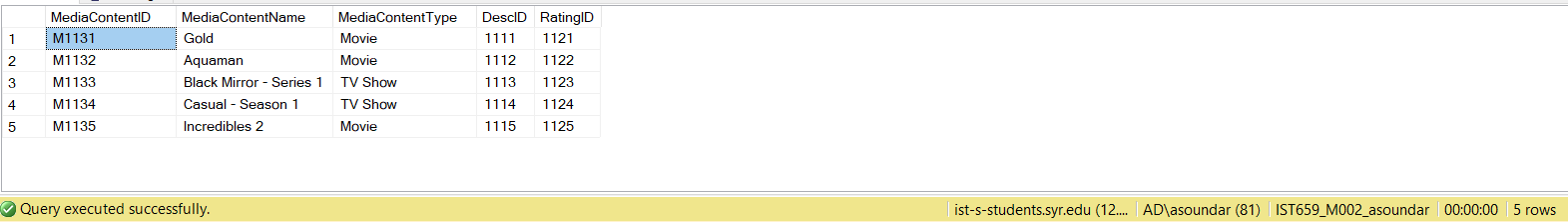
INSERT INTO MediaContent(MediaContentID, MediaContentName, MediaContentType, DescID, RatingID) VALUES ('M1132', 'Aquaman', 'Movie', 1112, 1122)

INSERT INTO MediaContent(MediaContentID, MediaContentName, MediaContentType, DescID, RatingID) VALUES ('M1133', 'Black Mirror - Series 1', 'TV Show', 1113, 1123)

INSERT INTO MediaContent(MediaContentID, MediaContentName, MediaContentType, DescID, RatingID) VALUES ('M1134', 'Casual - Season 1', 'TV Show', 1114, 1124)

INSERT INTO MediaContent(MediaContentID, MediaContentName, MediaContentType, DescID, RatingID) VALUES ('M1135', 'Incredibles 2', 'Movie', 1115, 1125)

Select \* from MediaContent;



INSERT INTO Platforms(PlatformID, PlatformName) VALUES ('A1141', 'Amazon Prime')

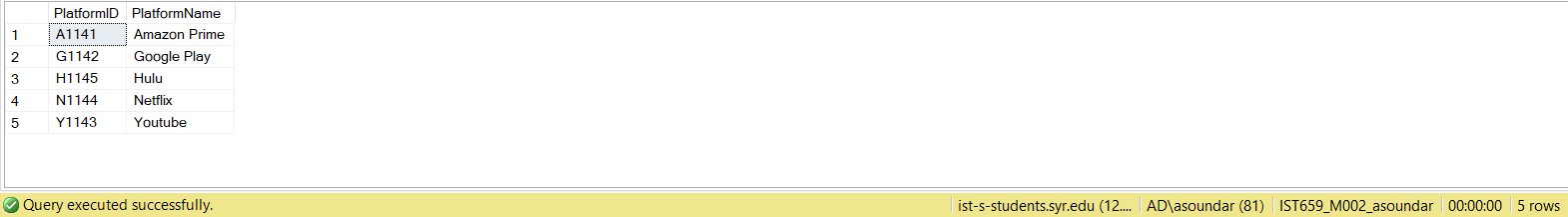
INSERT INTO Platforms(PlatformID, PlatformName) VALUES ('G1142', 'Google Play')

INSERT INTO Platforms(PlatformID, PlatformName) VALUES ('Y1143', 'Youtube')

INSERT INTO Platforms(PlatformID, PlatformName) VALUES ('N1144', 'Netflix')

INSERT INTO Platforms(PlatformID, PlatformName) VALUES ('H1145', 'Hulu')

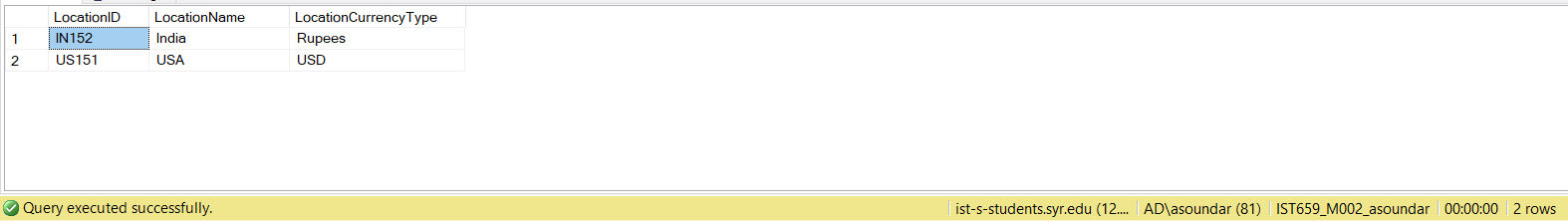
Select \* from Platforms;



INSERT INTO Location(LocationID, LocationName, LocationCurrencyType) VALUES ('US151', 'USA', 'USD')

INSERT INTO Location(LocationID, LocationName, LocationCurrencyType) VALUES ('IN152', 'India', 'Rupees')

Select \* from Location;



INSERT INTO Genre(GenreID, GenreName) VALUES ('S1165', 'Sci-Fi')

INSERT INTO Genre(GenreID, GenreName) VALUES ('A1165', 'Action')

INSERT INTO Genre(GenreID, GenreName) VALUES ('C1164', 'Comedy')

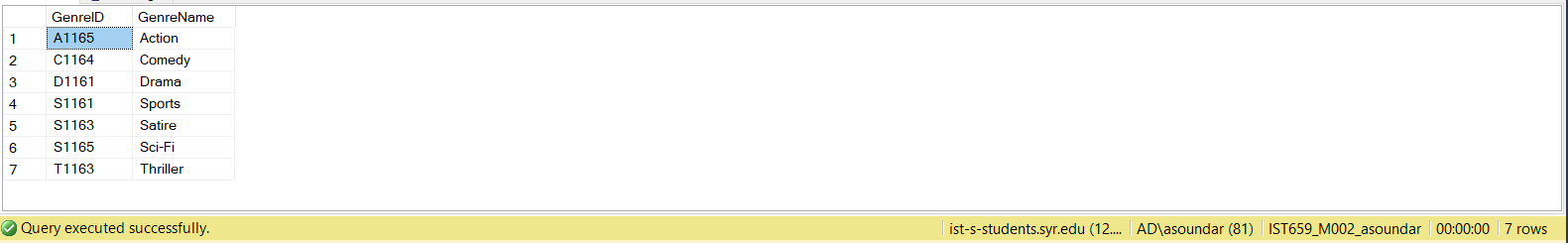
INSERT INTO Genre(GenreID, GenreName) VALUES ('S1161', 'Sports')

INSERT INTO Genre(GenreID, GenreName) VALUES ('D1161', 'Drama')

INSERT INTO Genre(GenreID, GenreName) VALUES ('S1163', 'Satire')

INSERT INTO Genre(GenreID, GenreName) VALUES ('T1163', 'Thriller')

Select \* from Genre;



INSERT INTO Price(PriceID, Price) VALUES ('USD1171', 0)

INSERT INTO Price(PriceID, Price) VALUES ('USD1172', 12.99)

INSERT INTO Price(PriceID, Price) VALUES ('RS1172', 2500)

INSERT INTO Price(PriceID, Price) VALUES ('USD1173', 27.25)

INSERT INTO Price(PriceID, Price) VALUES ('USD1174', 6.99)

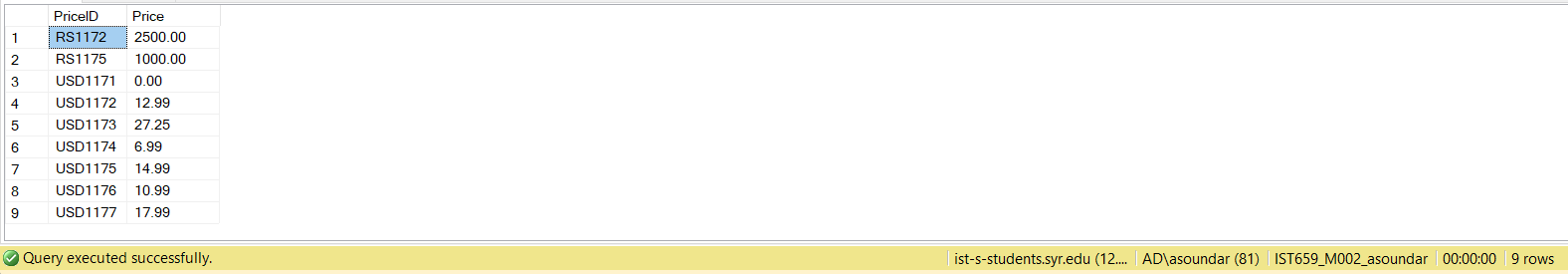
INSERT INTO Price(PriceID, Price) VALUES ('USD1175', 14.99)

INSERT INTO Price(PriceID, Price) VALUES ('RS1175', 1000)

INSERT INTO Price(PriceID, Price) VALUES ('USD1176', 10.99)

INSERT INTO Price(PriceID, Price) VALUES ('USD1177', 17.99)

Select \* from Price;



INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1131', 'US151', 'USD1171', 'A1141')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1132', 'US151', 'USD1172', 'G1142')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1132', 'IN152', 'RS1172', 'A1141')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1133', 'US151', 'USD1173', 'A1141')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1133', 'US151', 'USD1176', 'Y1143')

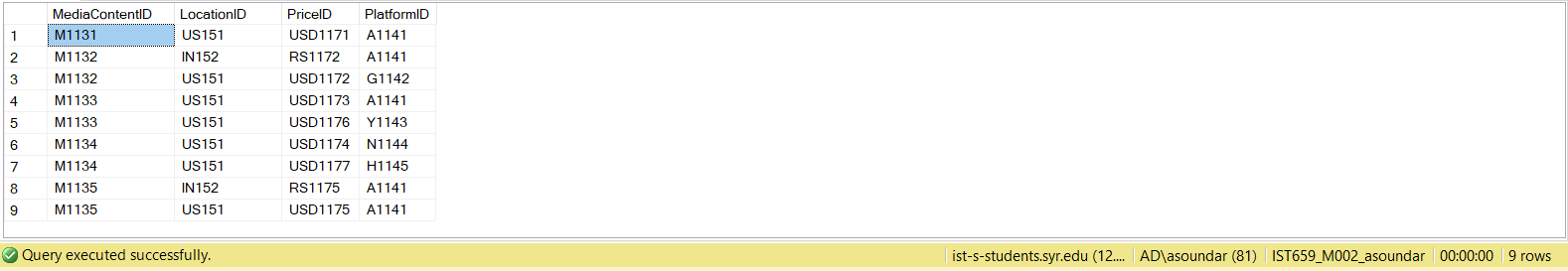
INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1134', 'US151', 'USD1174', 'N1144')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1134', 'US151', 'USD1177', 'H1145')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1135', 'US151', 'USD1175', 'A1141')

INSERT INTO MediaContentLocDetails(MediaContentID, LocationID, PriceID, PlatformID) VALUES ('M1135', 'IN152', 'RS1175', 'A1141')

Select \* from MediaContentLocDetails;



INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1131', 'S1161')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1131', 'D1161')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1132', 'A1165')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1133', 'S1163')

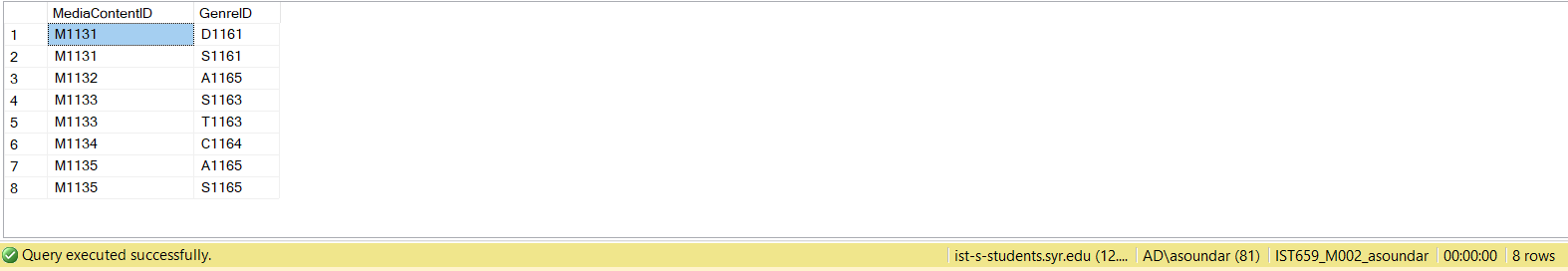
INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1133', 'T1163')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1134', 'C1164')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1135', 'S1165')

INSERT INTO ContentGenreDetails (MediaContentID, GenreID) VALUES ('M1135', 'A1165')

Select \* from ContentGenreDetails;

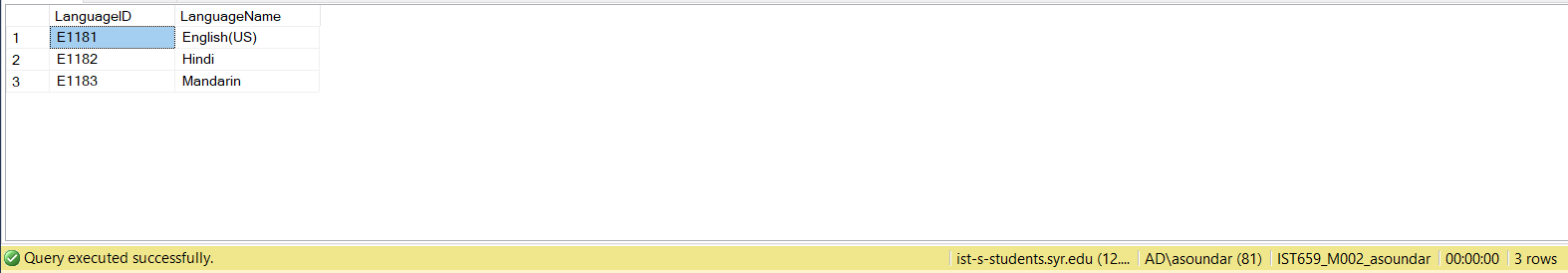


INSERT INTO MediaLanguage(LanguageID, LanguageName) VALUES ('E1181', 'English(US)')

INSERT INTO MediaLanguage(LanguageID, LanguageName) VALUES ('E1182', 'Hindi')

INSERT INTO MediaLanguage(LanguageID, LanguageName) VALUES ('E1183', 'Mandarin')

Select \* from MediaLanguage;



INSERT INTO Customers(CustID, CustFName, CustLName, CustEmail, LocationID) VALUES ('C1191', 'Alex', 'McClurg', 'alex@gmail.com', 'US151')

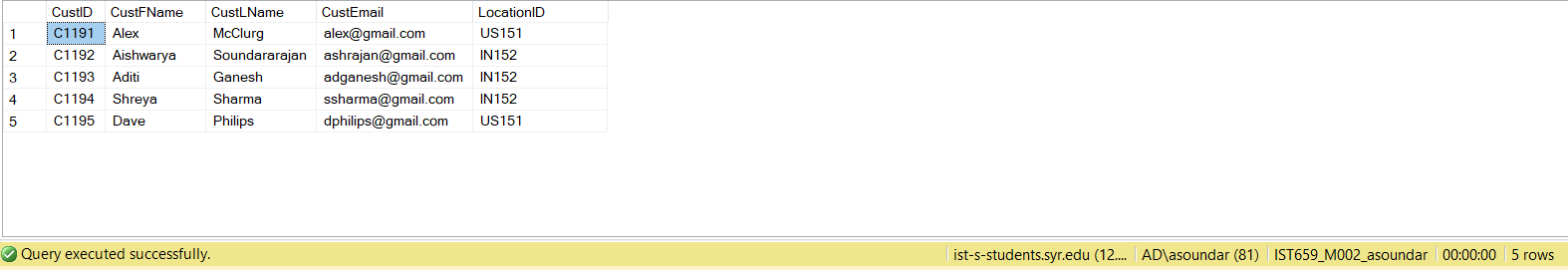
INSERT INTO Customers(CustID, CustFName, CustLName, CustEmail,LocationID) VALUES ('C1192', 'Aishwarya', 'Soundararajan', 'ashrajan@gmail.com', 'IN152')

INSERT INTO Customers(CustID, CustFName, CustLName, CustEmail, LocationID) VALUES ('C1193', 'Aditi', 'Ganesh', 'adganesh@gmail.com', 'IN152')

INSERT INTO Customers(CustID, CustFName, CustLName, CustEmail, LocationID) VALUES ('C1194', 'Shreya', 'Sharma', 'ssharma@gmail.com', 'IN152')

INSERT INTO Customers(CustID, CustFName, CustLName, CustEmail, LocationID) VALUES ('C1195', 'Dave', 'Philips', 'dphilips@gmail.com', 'US151')

Select \* from Customers;



INSERT INTO CustomerLangDetails(CustID,LanguageID) VALUES ('C1191', 'E1181')

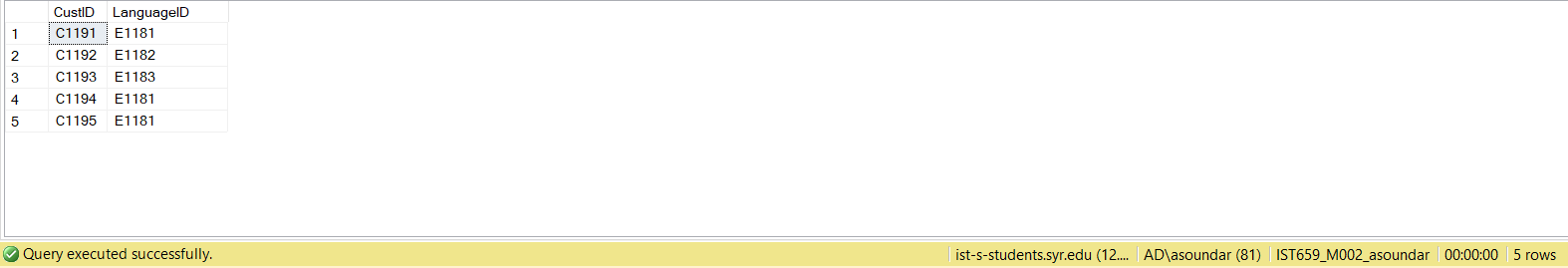
INSERT INTO CustomerLangDetails(CustID,LanguageID) VALUES ('C1192', 'E1182')

INSERT INTO CustomerLangDetails(CustID,LanguageID) VALUES ('C1193', 'E1183')

INSERT INTO CustomerLangDetails(CustID,LanguageID) VALUES ('C1194', 'E1181')

INSERT INTO CustomerLangDetails(CustID,LanguageID) VALUES ('C1195', 'E1181')

Select \* from CustomerLangDetails;



INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1191', 'A1141')

INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1191', 'G1142')

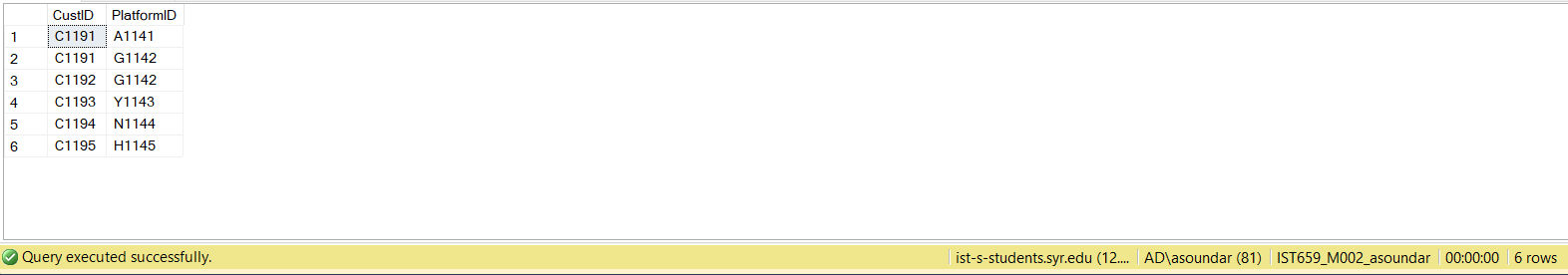
INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1192', 'G1142')

INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1193', 'Y1143')

INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1194', 'N1144')

INSERT INTO PlatformUsageDetails(CustID,PlatformID) VALUES ('C1195', 'H1145')

Select \* from PlatformUsageDetails;



INSERT INTO CustomerGenreDetails(CustID,GenreID) VALUES ('C1191', 'A1165')

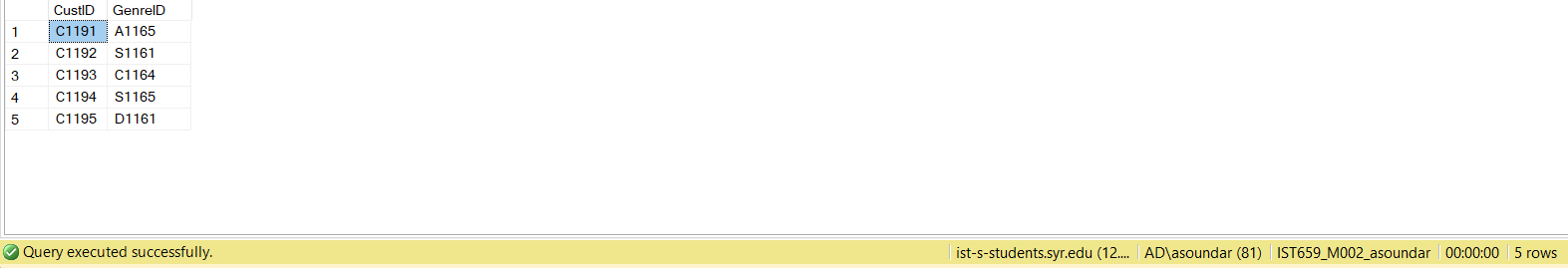
INSERT INTO CustomerGenreDetails(CustID,GenreID) VALUES ('C1192', 'S1161')

INSERT INTO CustomerGenreDetails(CustID,GenreID) VALUES ('C1193', 'C1164')

INSERT INTO CustomerGenreDetails(CustID,GenreID) VALUES ('C1194', 'S1165')

INSERT INTO CustomerGenreDetails(CustID,GenreID) VALUES ('C1195', 'D1161')

Select \* from CustomerGenreDetails;



INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1131', 'E1181')

INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1132', 'E1181')

INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1132', 'E1182')

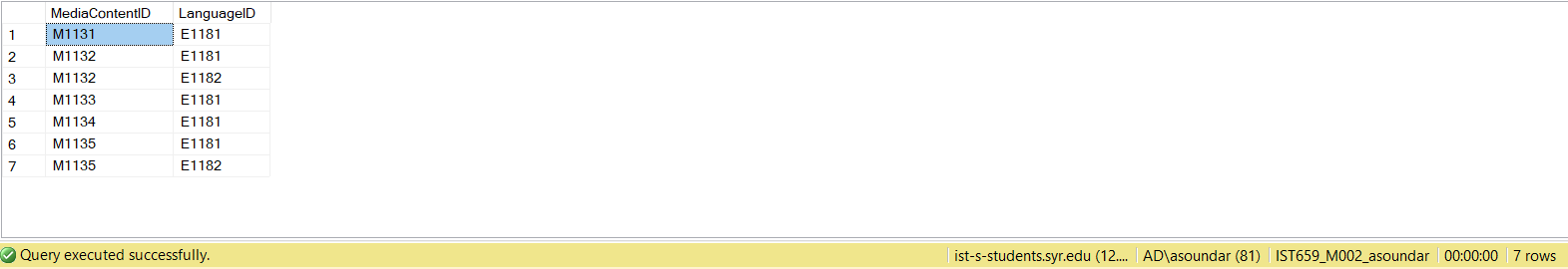
INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1133', 'E1181')

INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1134', 'E1181')

INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1135', 'E1181')

INSERT INTO MediaContentLangDetails(MediaContentID,LanguageID) VALUES ('M1135', 'E1182')

Select \* from MediaContentLangDetails;



INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('A1141', 'US151', 12.99)

INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('G1142', 'US151', 9.99)

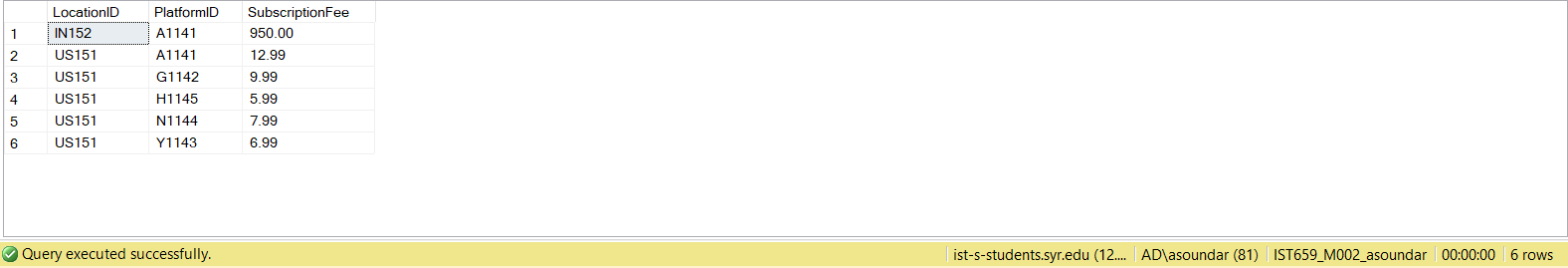
INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('A1141', 'IN152', 950)

INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('Y1143', 'US151', 6.99)

INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('N1144', 'US151', 7.99)

INSERT INTO SubscriptionFee(PlatformID,LocationID, SubscriptionFee) VALUES ('H1145', 'US151', 5.99)

Select \* from SubscriptionFee;



MAJOR DATA QUESTIONS:

-- 1) In which platform is ‘Aquaman’ movie available?

SELECT MediaContent.MediaContentName, Platforms.PlatformName FROM MediaContent

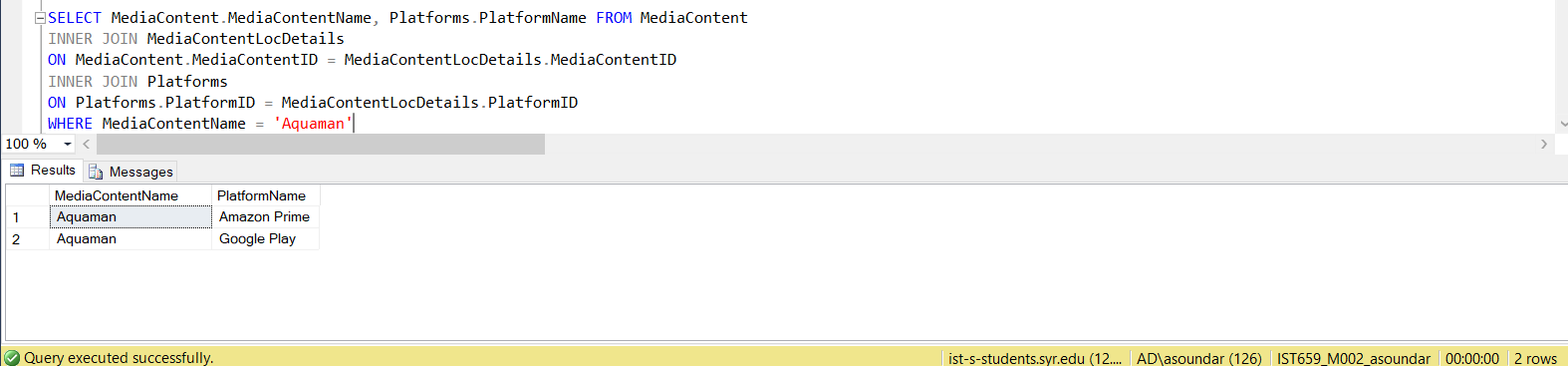
INNER JOIN MediaContentLocDetails

ON MediaContent.MediaContentID = MediaContentLocDetails.MediaContentID

INNER JOIN Platforms

ON Platforms.PlatformID = MediaContentLocDetails.PlatformID

WHERE MediaContentName = 'Aquaman'



-- 2) What are the available movie / TV shows in 'Action' genre?

SELECT MC.MediaContentName, G.GenreName FROM MediaContent as MC

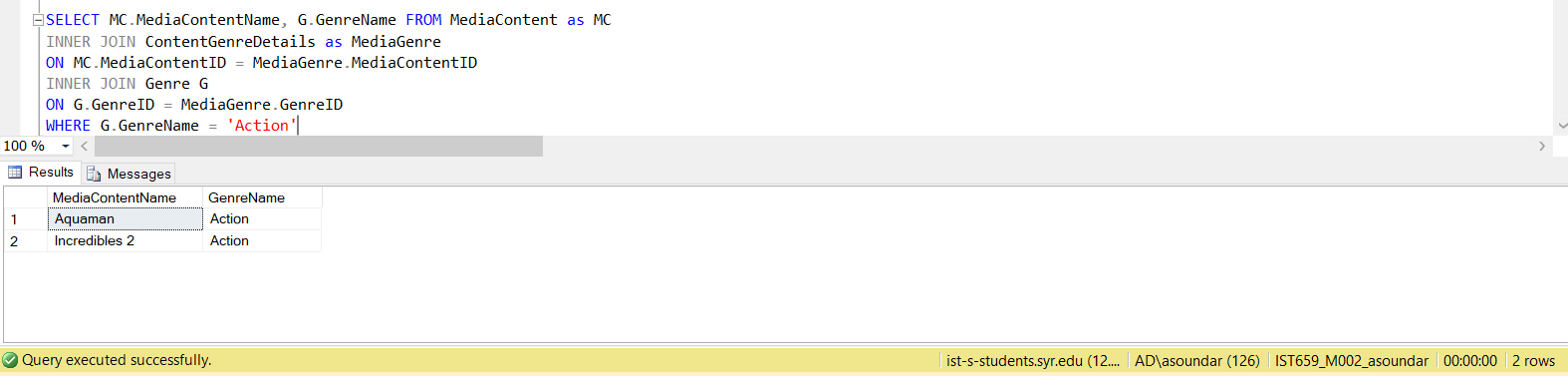
INNER JOIN ContentGenreDetails as MediaGenre

ON MC.MediaContentID = MediaGenre.MediaContentID

INNER JOIN Genre G

ON G.GenreID = MediaGenre.GenreID

WHERE G.GenreName = 'Action'



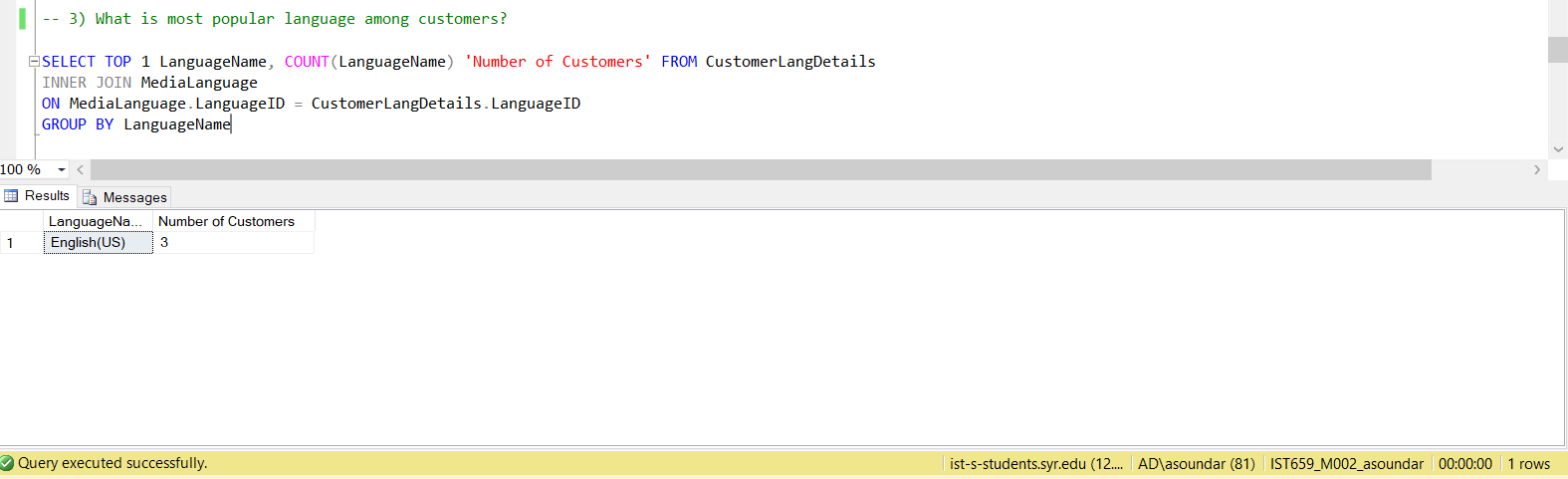
-- 3) What is most popular language among customers?

SELECT TOP 1 LanguageName, COUNT(LanguageName) 'Number of Customers' FROM CustomerLangDetails

INNER JOIN MediaLanguage

ON MediaLanguage.LanguageID = CustomerLangDetails.LanguageID

GROUP BY LanguageName



-- 4) What are the preferred movie genres for customerID 'C1195'?

SELECT G.GenreName, C.CustID, C.CustFName, C.CustLName FROM Genre as G

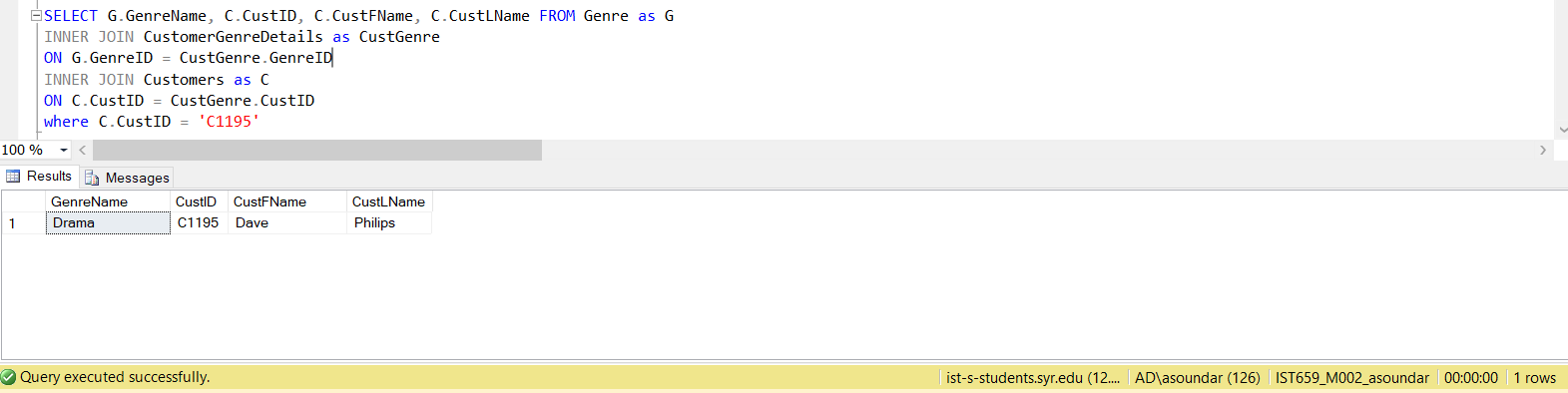
INNER JOIN CustomerGenreDetails as CustGenre

ON G.GenreID = CustGenre.GenreID

INNER JOIN Customers as C

ON C.CustID = CustGenre.CustID

where C.CustID = 'C1195'



--5) List of movies/TV shows available in different Platform along with their location and price details

SELECT MC.MediaContentType, MC.MediaContentName, PL.PlatformName, Location.LocationName,Location.LocationCurrencyType, P.Price,S.SubscriptionFee, (P.Price + S.SubscriptionFee) 'Total Price' FROM Price AS P

INNER JOIN MediaContentLocDetails AS MContent

ON MContent.PriceID = P.PriceID

INNER JOIN SubscriptionFee AS S

ON MContent.LocationID = S.LocationID AND MContent.PlatformID = S.PlatformID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

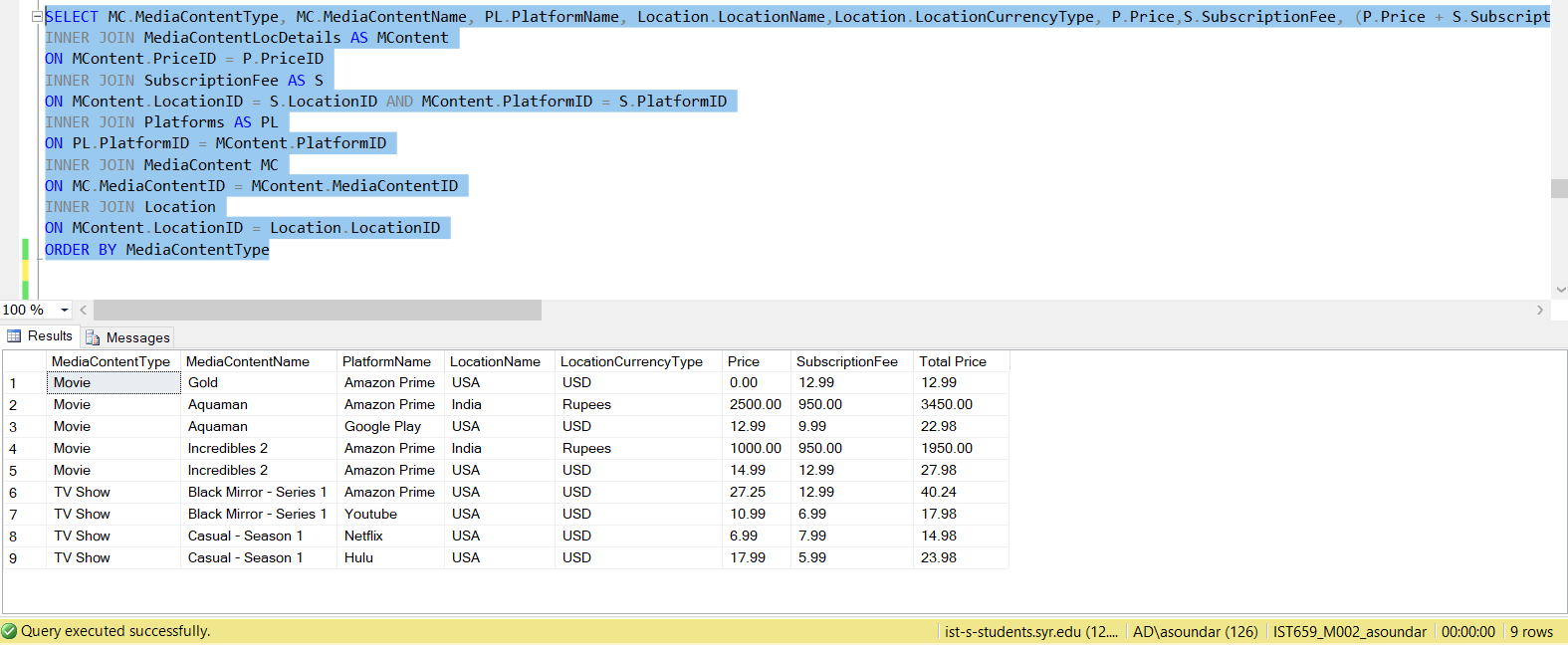
INNER JOIN MediaContent MC

ON MC.MediaContentID = MContent.MediaContentID

INNER JOIN Location

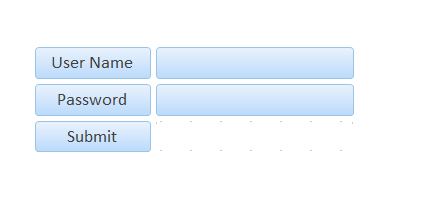
ON MContent.LocationID = Location.LocationID

ORDER BY MediaContentType

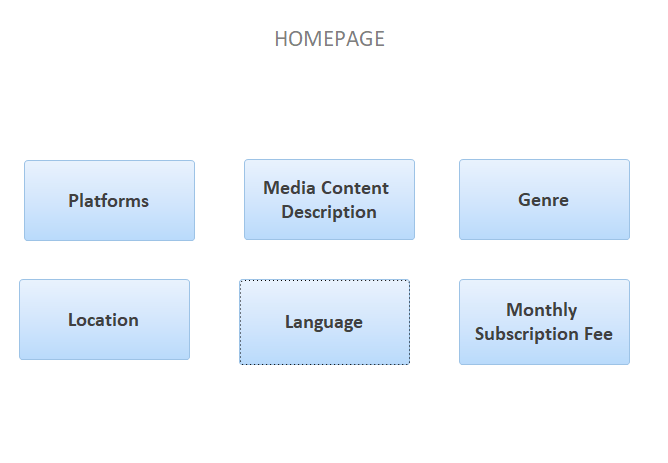


FORMS:

**Customer Login Page:**



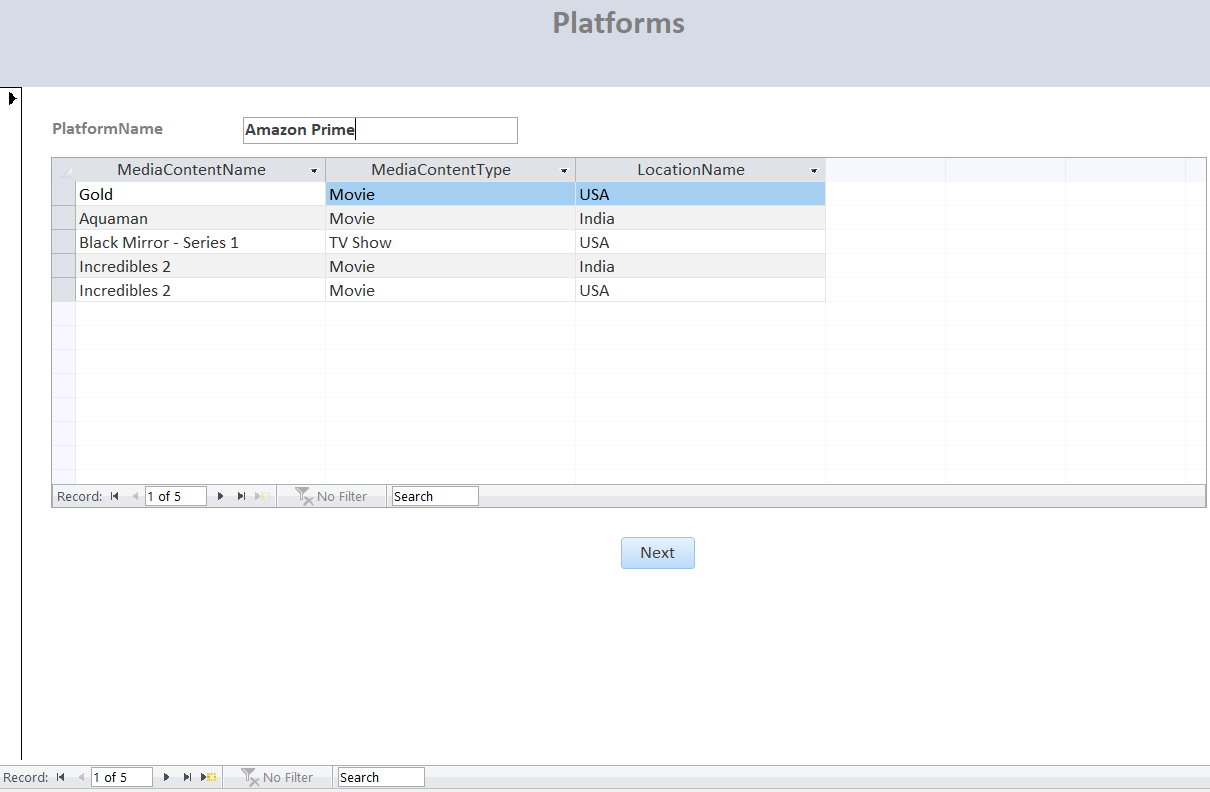
**Homepage:**



Given below are the UI pages displayed on clicking the buttons in the home page:

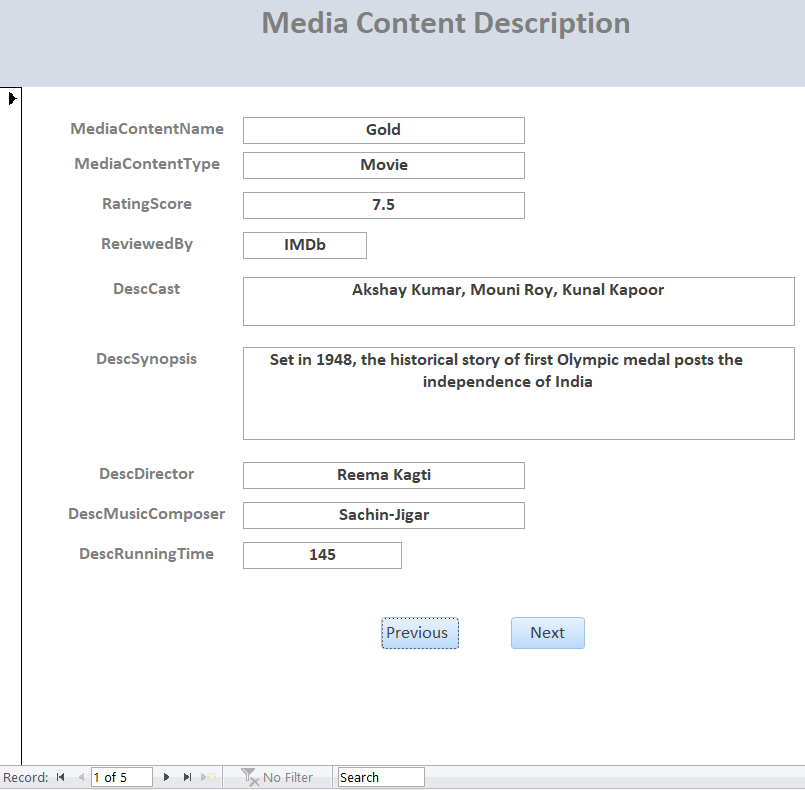
**Platform UI:**

Providing Media Content Name, Media Content Type and the Location based on Platforms



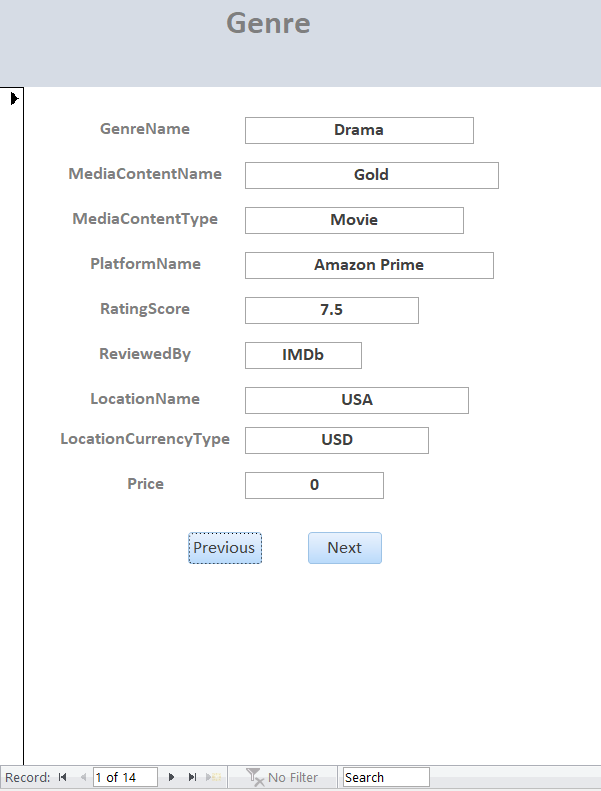
**Description UI:**

Providing Description details of the movies/TV Shows:



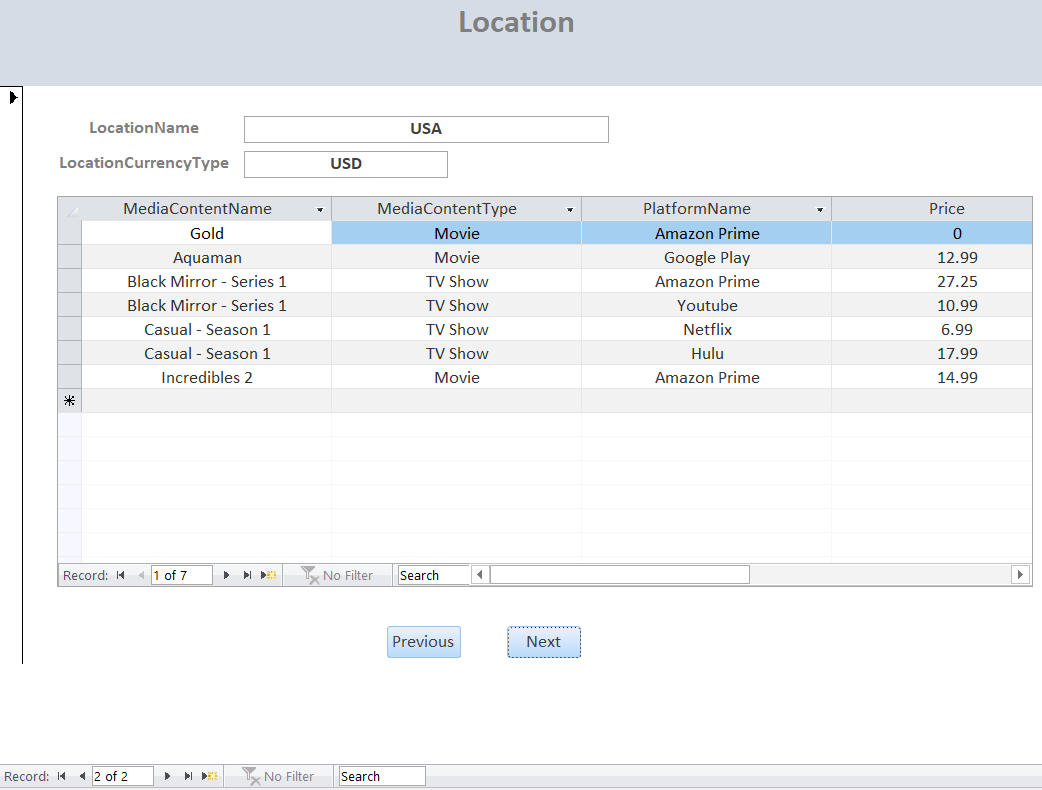
**Genre UI:**

Shows the genre of the movie along with media content type, rating, Platform Name, location in which it is available and its price.



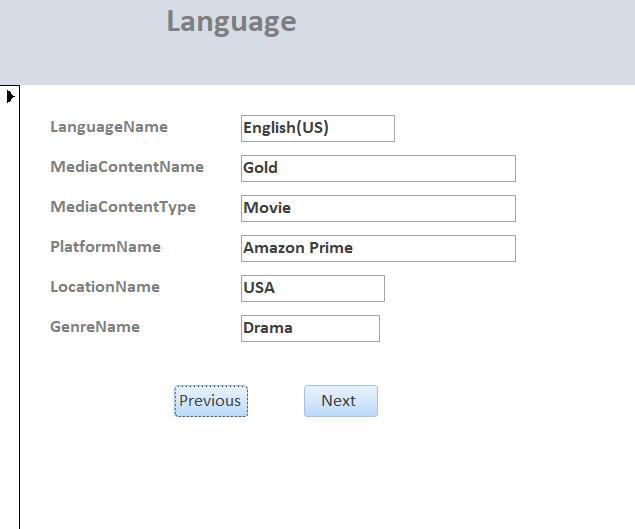
**Location UI:**

Finding out the Movie/ TV shows in different Platform based on the location:



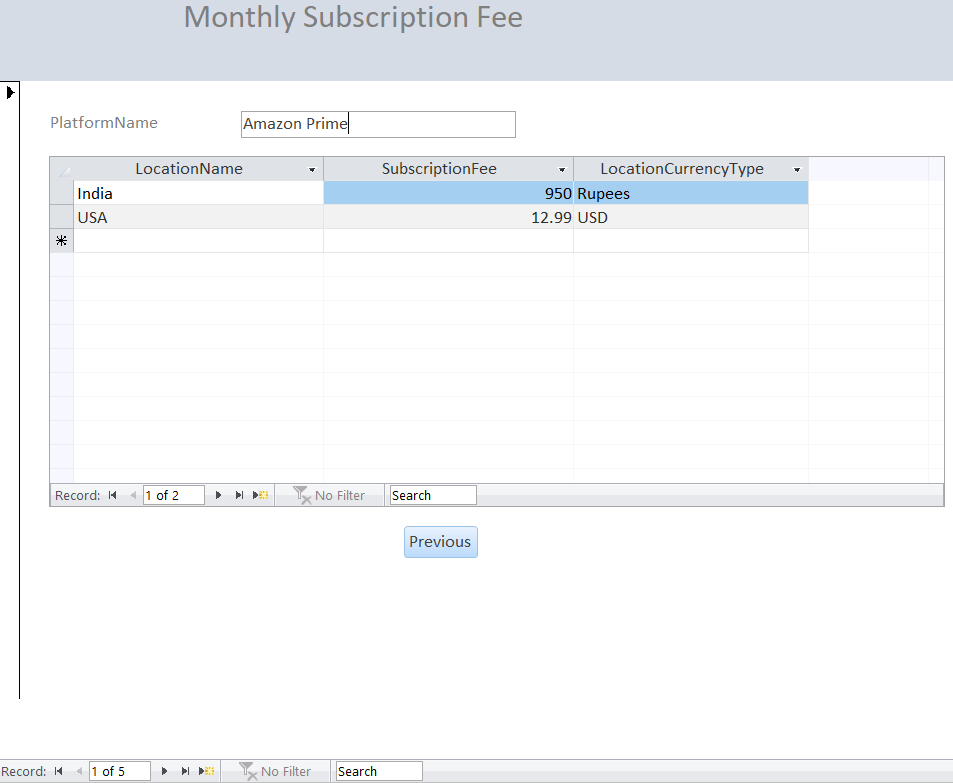
**Language UI:**

Language UI shows the availability of movies/TV shows in different languages, genres and platforms



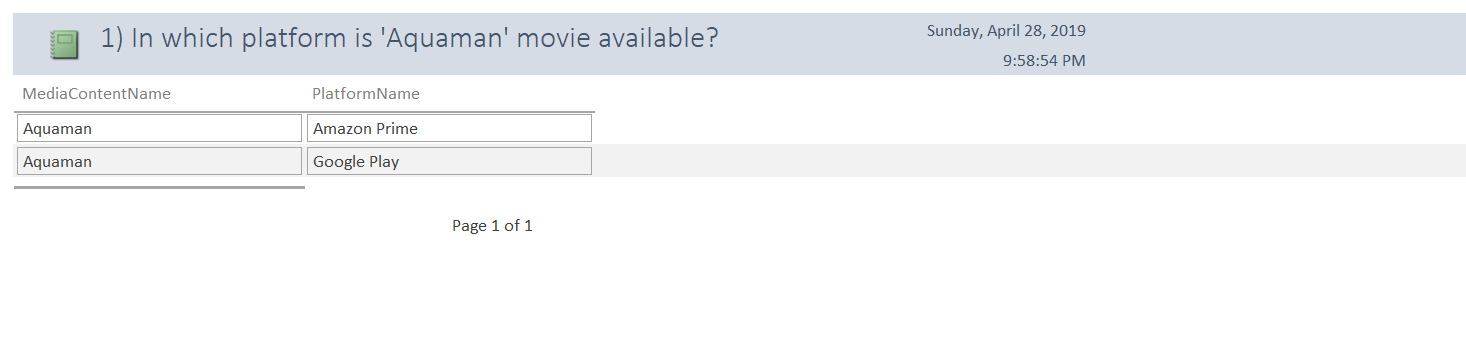
**Subscription UI:**

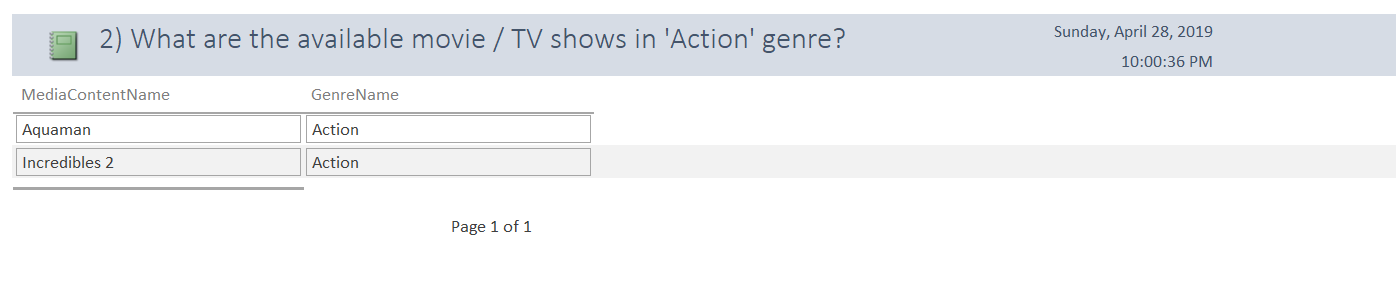
Subscription UI page of different Platforms in each locations along with the corresponding Subscription Fees



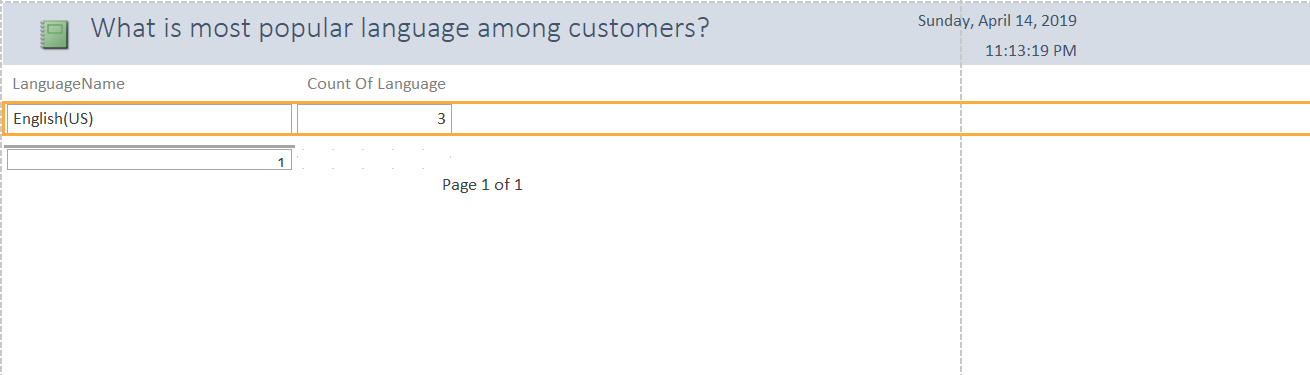
REPORTS:

1. In which platform is ‘Aquaman’ movie available?

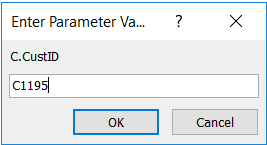


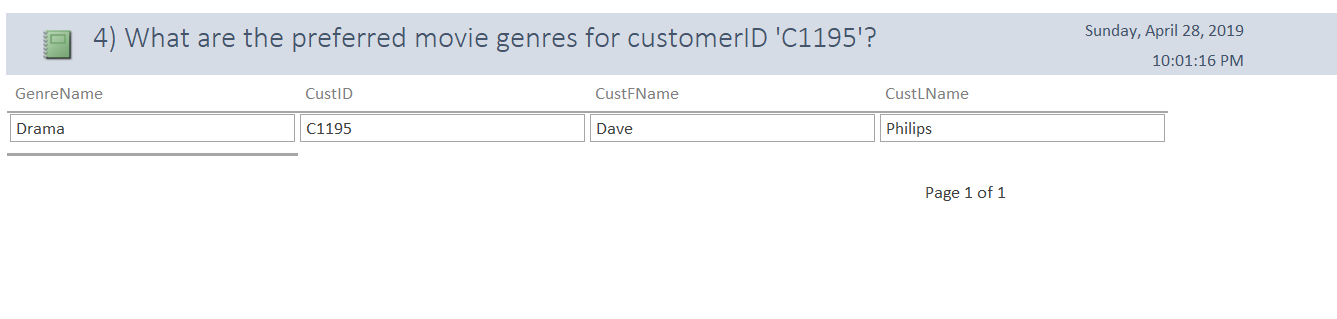
2) What are the available movie/TV shows in ‘Action’ genre?

3) What is the most popular language among customers?

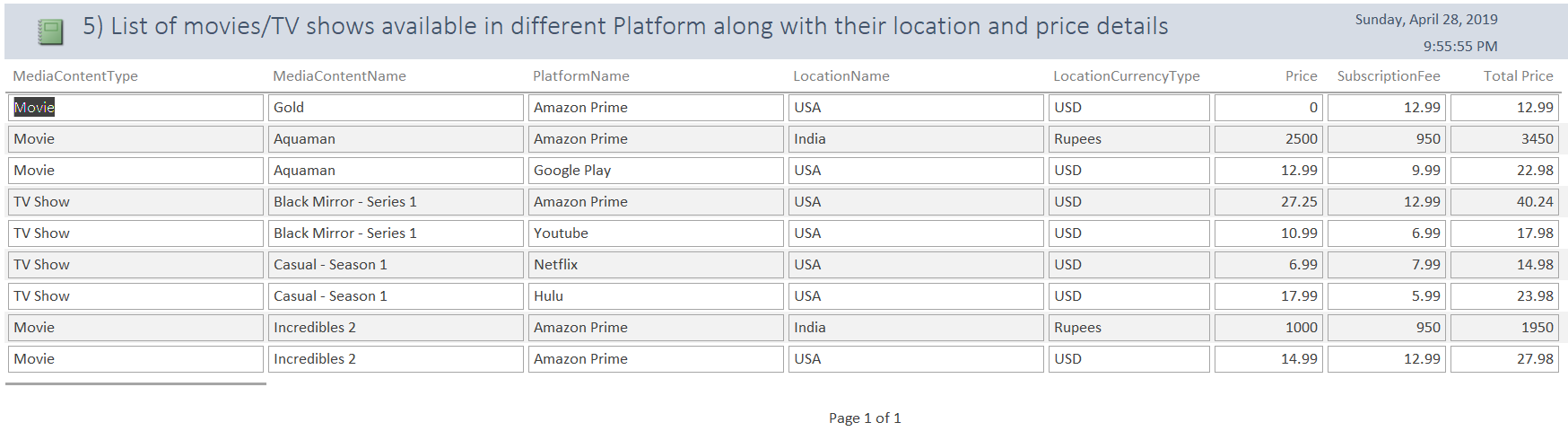


4) What are the preferred movie genres for customerID 'C1195'?





5) List of movies/TV shows available in different Platform along with their location and price details



FEEDBACK IMPLEMENTATION:

* Professor – Yang Wang’s Feedback:
* Given a movie, show the cheapest offering

-- Given a movie, show the cheapest offering

SELECT MediaContentType, MediaContentName, PlatformName, LocationName,LocationCurrencyType, Price,SubscriptionFee, (Price + SubscriptionFee) 'Total Price' FROM(SELECT MC.MediaContentType, MC.MediaContentName, PL.PlatformName, Location.LocationName,Location.LocationCurrencyType, P.Price,S.SubscriptionFee, (P.Price + S.SubscriptionFee) 'Total Price', rank() over(partition by MC.MediaContentName, Location.LocationName,Location.LocationCurrencyType order by (P.Price + S.SubscriptionFee))as rnk FROM Price AS P

INNER JOIN MediaContentLocDetails AS MContent

ON MContent.PriceID = P.PriceID

INNER JOIN SubscriptionFee AS S

ON MContent.LocationID = S.LocationID AND MContent.PlatformID = S.PlatformID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

INNER JOIN MediaContent MC

ON MC.MediaContentID = MContent.MediaContentID

INNER JOIN Location

ON MContent.LocationID = Location.LocationID)rnk

where MediaContentName = 'Black Mirror - Series 1' and LocationName = 'USA' and rnk = 1;



* Show the movie info

-- Show the movie info

SELECT MC.MediaContentType, MC.MediaContentName, D.DescSynopsis, D.DescCast, D.DescDirector, D.DescMusicComposer, D.DescRunningTime, R.RatingScore FROM MediaContent AS MC

INNER JOIN EntertainmentDescription AS D

ON D.DescID = MC.DescID

INNER JOIN Rating AS R

ON R.RatingID = MC.RatingID

ORDER BY MediaContentType



* Top rated movies of a particular genre and under certain price

-- top rated movies of a particular genre and under certain price

SELECT MC.MediaContentType, MC.MediaContentName, PL.PlatformName, G.GenreName, Max(R.RatingScore) AS Rating, P.Price, S.SubscriptionFee, (P.Price + S.SubscriptionFee) 'Total Price', Location.LocationName, Location.LocationCurrencyType FROM Price AS P

INNER JOIN MediaContentLocDetails AS MContent

ON MContent.PriceID = P.PriceID

INNER JOIN SubscriptionFee AS S

ON MContent.LocationID = S.LocationID AND MContent.PlatformID = S.PlatformID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

INNER JOIN MediaContent MC

ON MC.MediaContentID = MContent.MediaContentID

INNER JOIN ContentGenreDetails AS CG

ON CG.MediaContentID = MC.MediaContentID

INNER JOIN Genre AS G

ON G.GenreID = CG.GenreID

INNER JOIN Rating AS R

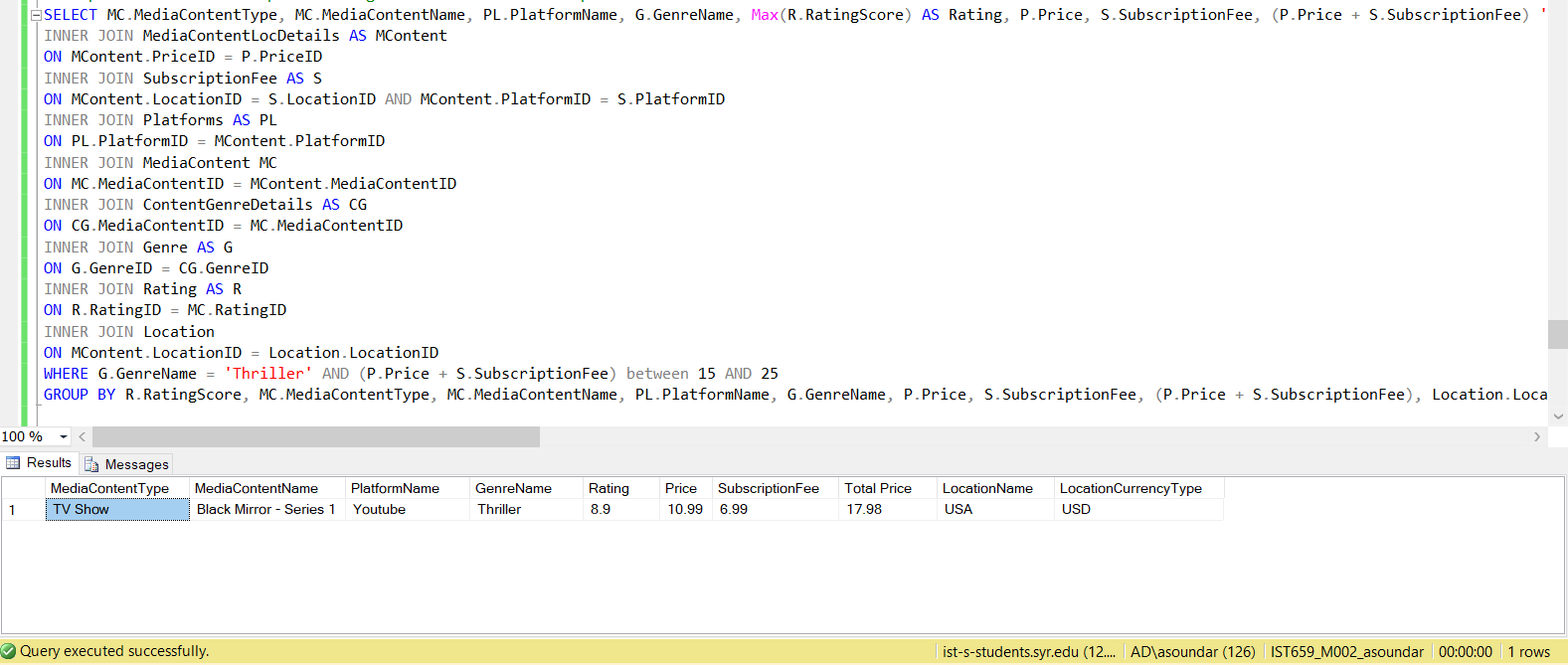
ON R.RatingID = MC.RatingID

INNER JOIN Location

ON MContent.LocationID = Location.LocationID

WHERE G.GenreName = 'Thriller' AND (P.Price + S.SubscriptionFee) between 15 AND 25

GROUP BY R.RatingScore, MC.MediaContentType, MC.MediaContentName, PL.PlatformName, G.GenreName, P.Price, S.SubscriptionFee, (P.Price + S.SubscriptionFee), Location.LocationName, Location.LocationCurrencyType



* Add a new report : list all movies/group by platform, genre

-- Add a new report : list all movies/group by platform, genre

SELECT PL.PlatformName, G.GenreName, MC.MediaContentType, MC.MediaContentName FROM MediaContent AS MC

INNER JOIN MediaContentLocDetails AS MContent

ON MC.MediaContentID = MContent.MediaContentID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

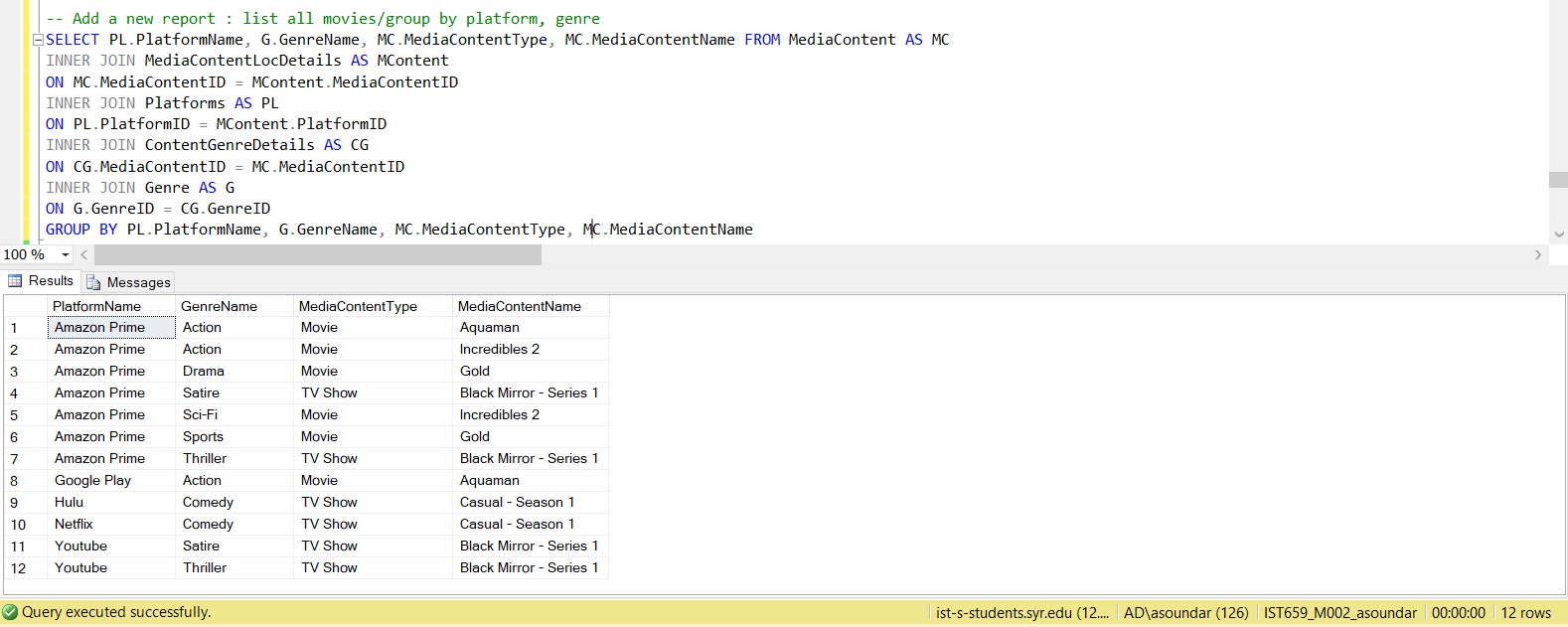
INNER JOIN ContentGenreDetails AS CG

ON CG.MediaContentID = MC.MediaContentID

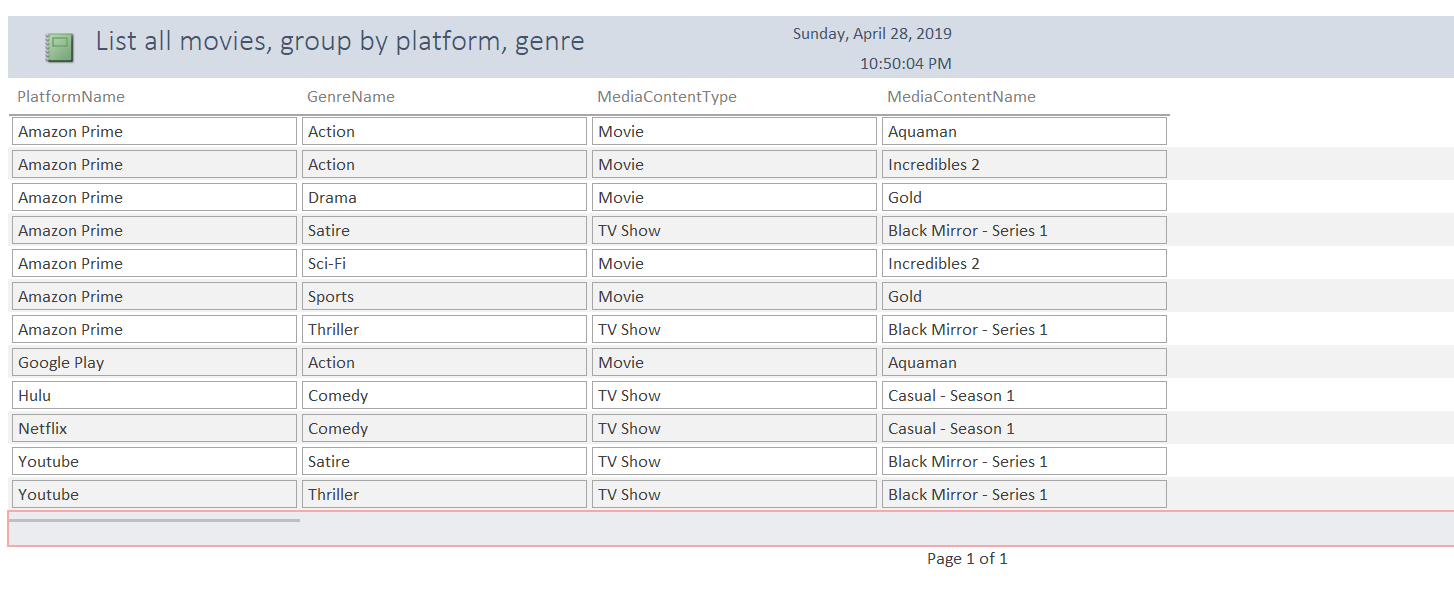
INNER JOIN Genre AS G

ON G.GenreID = CG.GenreID

GROUP BY PL.PlatformName, G.GenreName, MC.MediaContentType, MC.MediaContentName



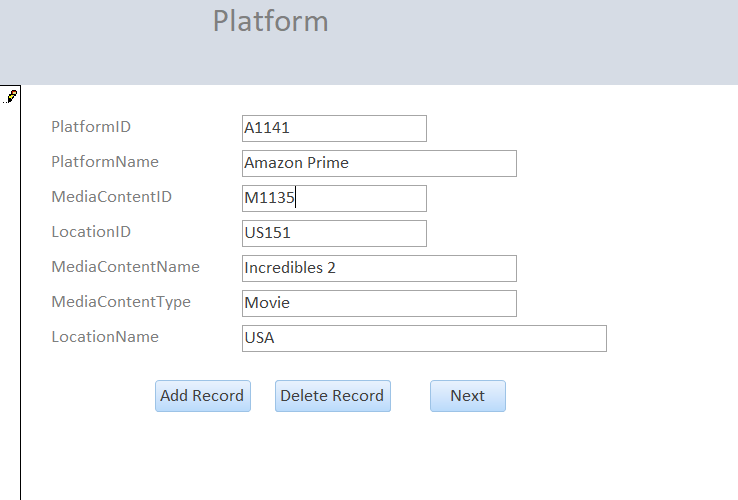
Report:



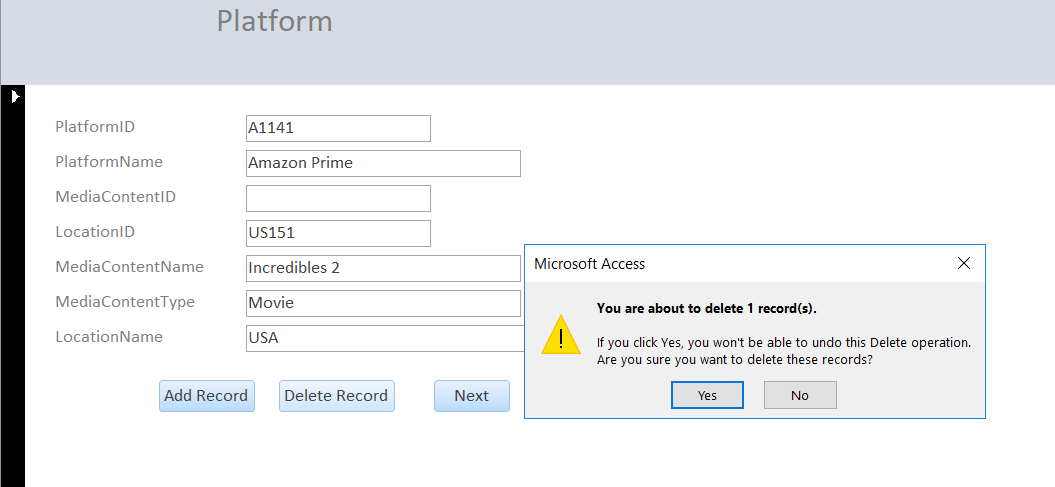
* Add forms where you can add/delete records, and traverse records

Admin users can add/ delete records and can also navigate through the pages. One such instance is given below:

Add record:



Delete record:



* Harsh Mehta:

“The overall project done by her is excellent. I think a few more data questions can be added.”

* What are the movies/ Tv shows available in 'English(US)' language?

SELECT MC.MediaContentName, ML.LanguageName FROM MediaContent AS MC

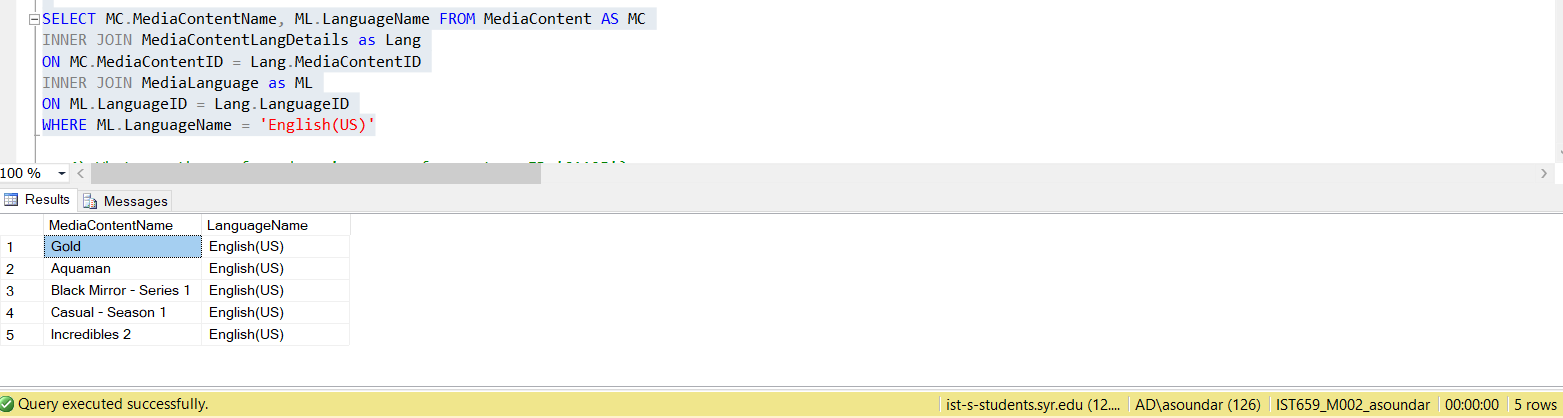
INNER JOIN MediaContentLangDetails as Lang

ON MC.MediaContentID = Lang.MediaContentID

INNER JOIN MediaLanguage as ML

ON ML.LanguageID = Lang.LanguageID

WHERE ML.LanguageName = 'English(US)'



* Jing Wang:

“Nice Idea and good job.”

* Addressing the questions related to currency type changes and inconsistencies raised by Mingkang Zhang, Yi Zhong:

I have added the field LocationCurrencyType in the Location table in ERD to specify different currency types used across different locations. When this information is joined with other tables such as MediaContent, Platform, SubscriptionFee and MediaContentLocDetails, it will help in identifying the price of each movie in different locations with their appropriate currency values.

For example,

The Movie Aquaman in India will have the Price as 2500 Rupees and the same movie in USA will be 12.99 USD.

I have also implemented and considered this while answering the data questions.

* Addressing the questions raised by Mingkang Zhang, Jessica Dischaive and Vidisha about adding a column named currency type:

I have addressed this in 5th data question that lists the movies in different platforms along with their location, platform and price, which is as follows:

SELECT MC.MediaContentType, MC.MediaContentName, PL.PlatformName, Location.LocationName,Location.LocationCurrencyType, P.Price,S.SubscriptionFee, (P.Price + S.SubscriptionFee) 'Total Price' FROM Price AS P

INNER JOIN MediaContentLocDetails AS MContent

ON MContent.PriceID = P.PriceID

INNER JOIN SubscriptionFee AS S

ON MContent.LocationID = S.LocationID AND MContent.PlatformID = S.PlatformID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

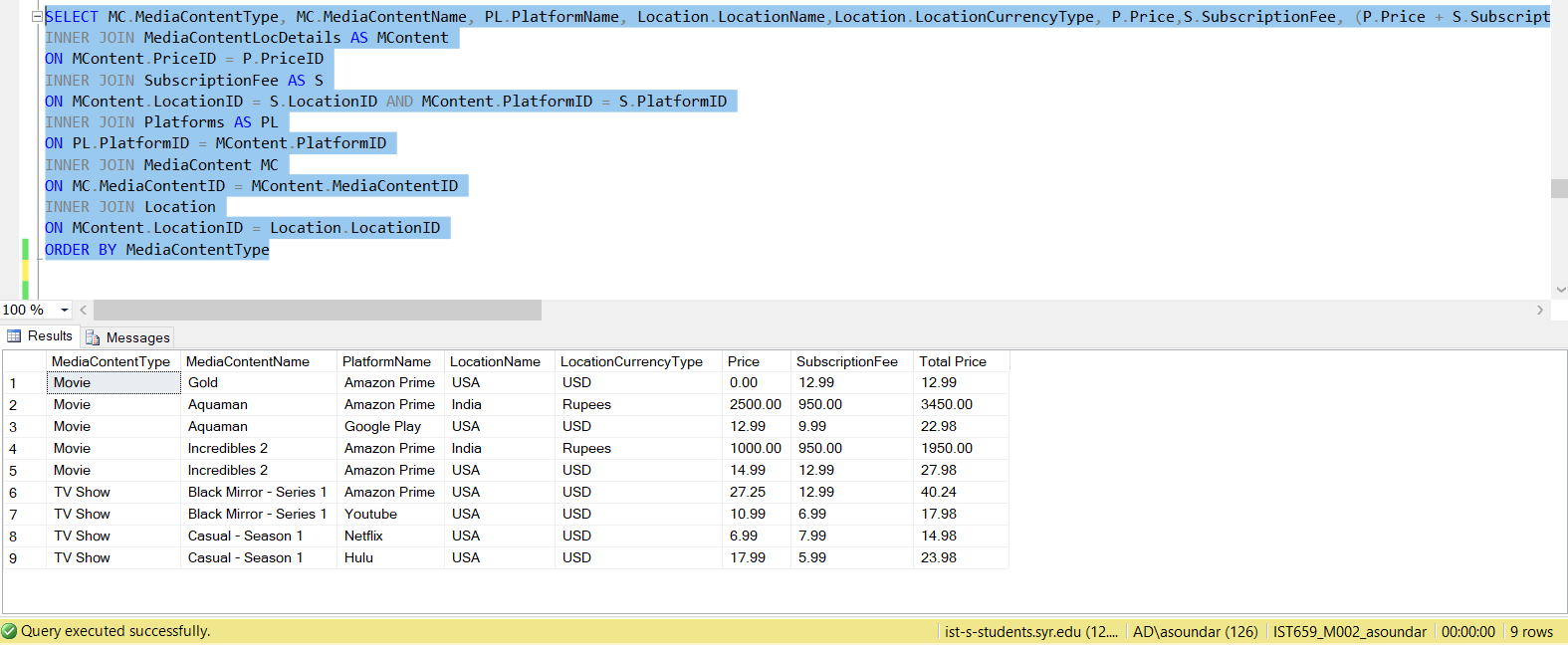
INNER JOIN MediaContent MC

ON MC.MediaContentID = MContent.MediaContentID

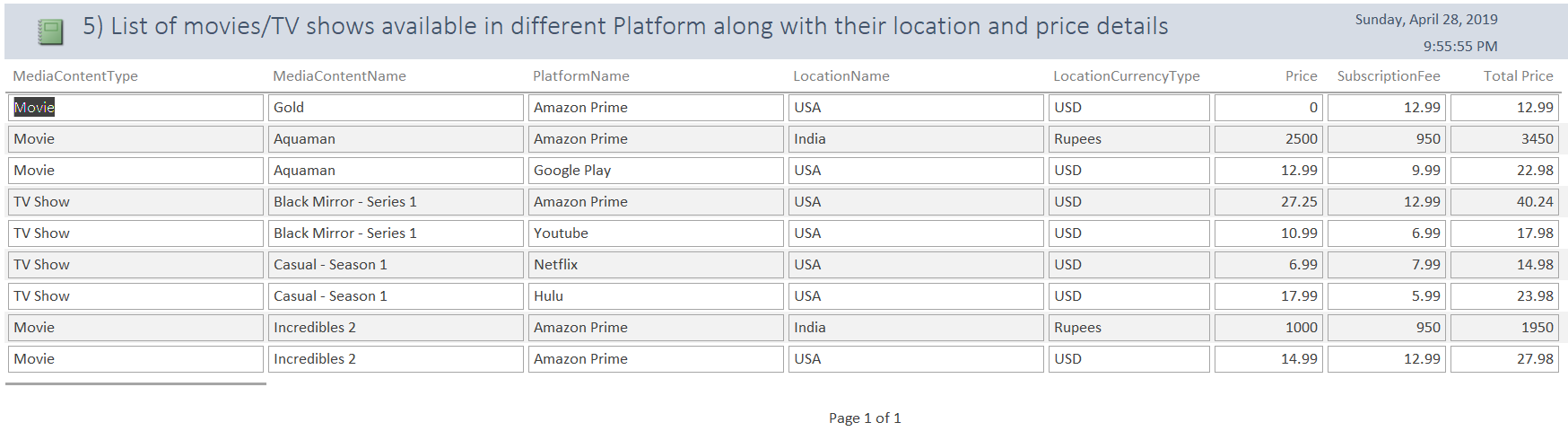
INNER JOIN Location

ON MContent.LocationID = Location.LocationID

ORDER BY MediaContentType



Report:



* Sanket:

“ A data question could be to check the cheapest movie on particular platform”

-- Cheapest movie on Amazon Prime

SELECT MediaContentType, MediaContentName, PlatformName, LocationName,LocationCurrencyType, Price,SubscriptionFee, (Price + SubscriptionFee) 'Total Price' FROM(SELECT MC.MediaContentType, MC.MediaContentName, PL.PlatformName, Location.LocationName,Location.LocationCurrencyType, P.Price,S.SubscriptionFee, (P.Price + S.SubscriptionFee) 'Total Price', rank() over(partition by PL.PlatformName order by (P.Price + S.SubscriptionFee))as rnk FROM Price AS P

INNER JOIN MediaContentLocDetails AS MContent

ON MContent.PriceID = P.PriceID

INNER JOIN SubscriptionFee AS S

ON MContent.LocationID = S.LocationID AND MContent.PlatformID = S.PlatformID

INNER JOIN Platforms AS PL

ON PL.PlatformID = MContent.PlatformID

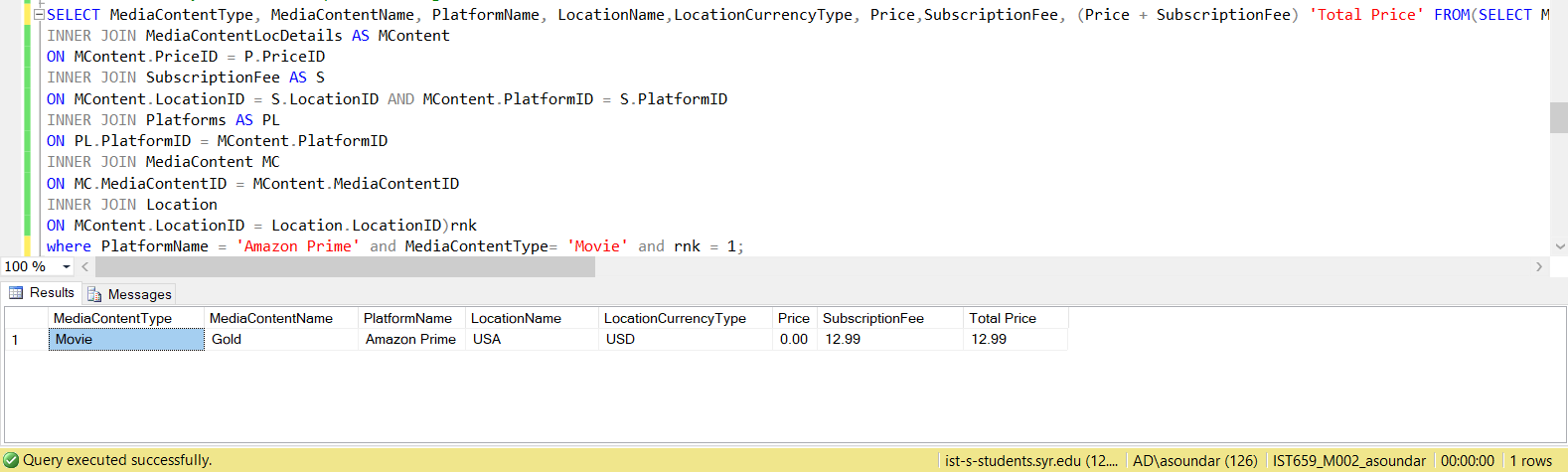
INNER JOIN MediaContent MC

ON MC.MediaContentID = MContent.MediaContentID

INNER JOIN Location

ON MContent.LocationID = Location.LocationID)rnk

where PlatformName = 'Amazon Prime' and MediaContentType= 'Movie' and rnk = 1;



* Rishie: “Complexity of ERD is resolved easily by the user interfaces which is commendable”
* Addressing the questions related to adding input section in forms by Jessica Dischiave and Shashank:

I have already addressed this feedback along with the Professor’s feedback which is as follows:

Add record:

