ONLINE MOVIE STORE

CPS 510 Project

Munevver Coskun 500923319
Nabil Haque 500772479
Zaiba Amla 500943055

Table of Contents

Application Description

Physical movie rentals are no longer the norm which is why the demand for online movie stores has increased. The movie store will allow customers to rent and buy movies online to watch from the comfort of their homes.

Platform purchases rights to sell movies from production companies and display them on their site. Customers create accounts and can view the selection of movies available. Customers will choose a movie to rent/buy and use credit/debit information.

The movie rental and purchasing website has various entities. First, the users must create an account, starting with the following attributes: first name, last name, email address, password, billing address, and card/debit information. To ensure customer uniqueness, each customer will be provided with a customer ID which will be the primary key attribute, not visible to the customer but saved in the database.

Rights to sell movies will be bought from production companies which will have similar attributes to users, including their own ID number as a primary key attribute. The following about the movies will be noted, production company name, password, movie title, genre, and direct deposit information.

The application will be able to provide information to the customers regarding:

- Movies that are rented
- Movies that are available in the storage of the application
- Movie purchase options
- Movie rental options
- How long can a movie be rented for
- Return conditions/policies

The application will be able to provide information to the seller regarding:

- Previously rented movies
- Movies that are available in the storage of the application
- Movies that are bought
- Customer account information
- Purchase/rental history
- Production company rights limitations and restrictions

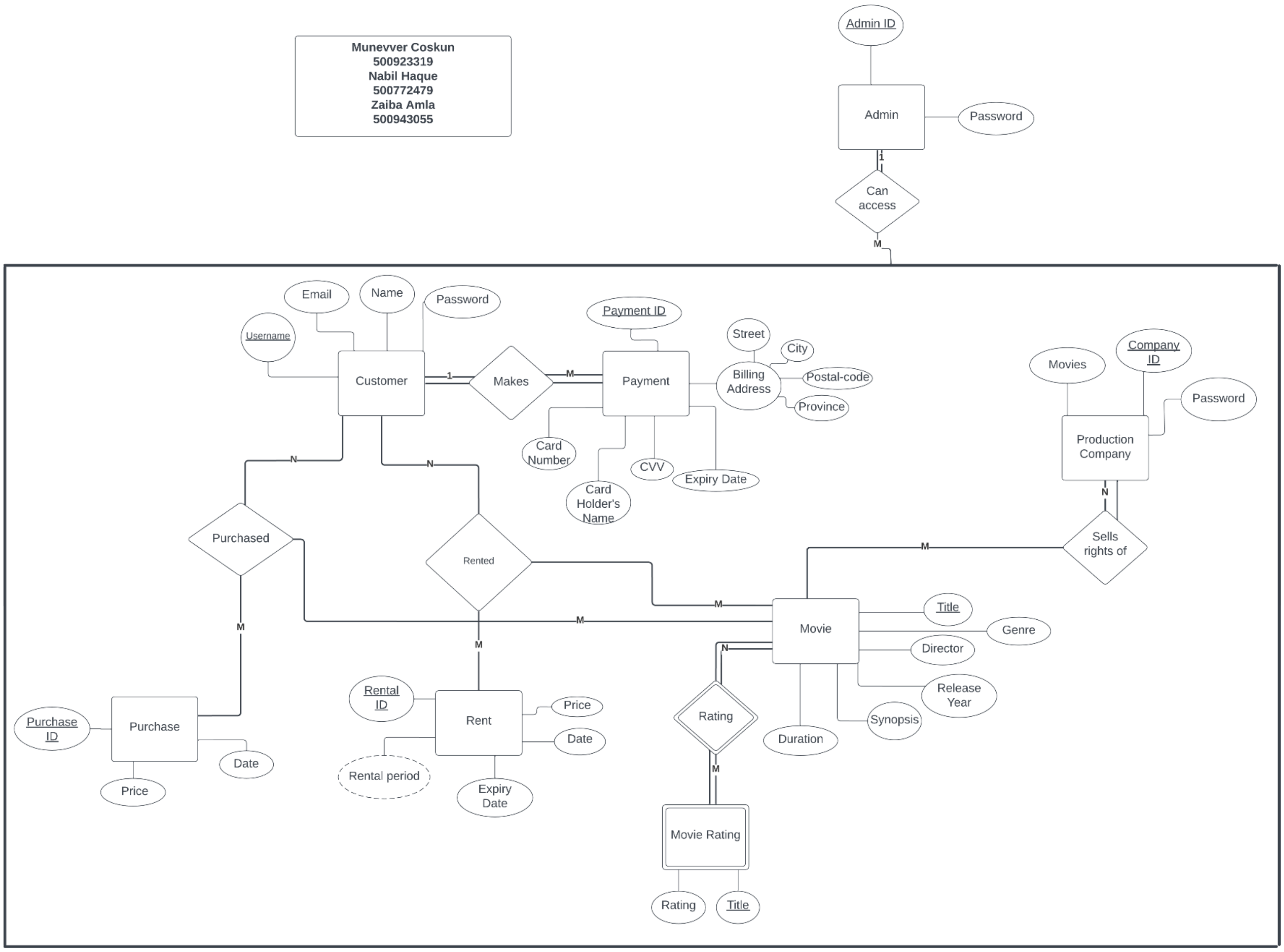
Role	Description
Remove Bought Movies	This function removed a movie from within our database when its purchased
Add New Movies	This function adds a new movie into our database when its rights are bought from the production company
Rent a Movie	This function allows customers to rent movies and adds the new information to our database
Expiry Date	This function sets a certain time period for customers that rented the movie to see the movie
Adding New Customer	This function creates a new information into our database about the new customer (Email, Name, Password, Username etc.)
Renting/Purchasing a Movie	This function(s) collects information about customer's payment method (Card number, Card Holder's Name, CVV, Expiry Date etc.) and stores it in our database.

Entities:

- Customer
- Movie
- Production Company
- Movie Rating (weak)
- Rent
- Purchase
- Admin

ER Model

Munevver Coskun
500923319
Nabil Haque
500772479
Zaiba Amla
500943055



Schema Design

```
CREATE TABLE Customer(
    Username VARCHAR2(100) PRIMARY KEY,
    Name VARCHAR2(100),
    Email VARCHAR2(100),
    Password VARCHAR2(100)
  );

CREATE TABLE Payment(
    Payment_ID int PRIMARY KEY,
    Card_Holder_Name VARCHAR2(100),
    Card_Number VARCHAR2(100),
    CVV VARCHAR2(100),
    Expiry_date VARCHAR2(100)
  );

CREATE TABLE Billing_Address(
    Street VARCHAR2(100),
    City VARCHAR2(100),
    Postal_Code VARCHAR2(100),
    Province VARCHAR2(100),
    Payment_ID int PRIMARY KEY,
    FOREIGN KEY (Payment_ID) REFERENCES Payment(Payment_ID)
    );
```

```
CREATE TABLE Production_Company(

   Company_ID int PRIMARY KEY,

   Password VARCHAR2(100),

   Movies VARCHAR2(100)

 );

CREATE TABLE Movie(

   Title VARCHAR2(100) PRIMARY KEY,

   Genre VARCHAR2(100),

   Director VARCHAR2(100),

   Release_Year int,

   Synopsis VARCHAR2(100),

   Duration int

 );
CREATE TABLE Rent(

   Rental_ID int PRIMARY KEY,

   Price VARCHAR2(100),

   Date_of_rental DATE,

   Expiry_Date DATE,

   Rental_Period VARCHAR2(100)

 );

CREATE TABLE Purchase(

   Purchase_ID int PRIMARY KEY,

   Price VARCHAR2(100),

   Buy_Date DATE

 );
```

```
CREATE TABLE Movie_Rating(

   Rating int PRIMARY KEY,

   Title VARCHAR2(100)

   );

  CREATE TABLE Admin(
   Admin_ID int PRIMARY KEY,
   password int

  );

  CREATE TABLE Can_Access(

   Admin_ID int,

   Payment_ID int,

   Username VARCHAR2(100),

   Purchase_ID int,

   Rental_ID int,

   Title VARCHAR2(100),

   Company_ID int,

   PRIMARY KEY (Admin_ID, Payment_ID, Username, Purchase_ID, Rental_ID, Title,

Company_ID),

   FOREIGN KEY (Admin_ID) REFERENCES Admin(Admin_ID),

   FOREIGN KEY (Payment_ID) REFERENCES Payment(Payment_ID),

   FOREIGN KEY (Username) REFERENCES Customer(Username),

   FOREIGN KEY (Purchase_ID) REFERENCES Purchase(Purchase_ID),

   FOREIGN KEY (Rental_ID) REFERENCES Rent(Rental_ID),

   FOREIGN KEY (Title) REFERENCES Movie(Title),

   FOREIGN KEY (Company_ID) REFERENCES Production_Company(Company_ID)

   );
```

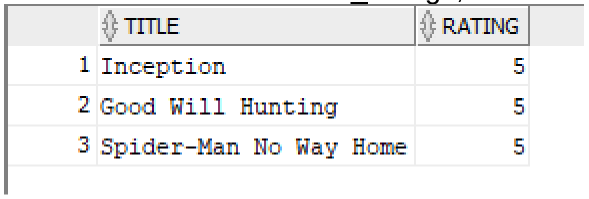
```
CREATE TABLE Makes(
  Payment_ID int,
  Username VARCHAR2(100),
  PRIMARY KEY (Payment_ID, Username),
  FOREIGN KEY (Payment_ID) REFERENCES Payment(Payment_ID),
  FOREIGN KEY (Username) REFERENCES Customer(Username)
  );

CREATE TABLE Rented( Username
  VARCHAR(100),
  Rental_ID int PRIMARY KEY,
  Title VARCHAR(100),
  FOREIGN KEY (Username) REFERENCES Customer(Username),
  FOREIGN KEY (Rental_ID) REFERENCES Rent(Rental_ID),
  FOREIGN KEY (Title) REFERENCES Movie(Title)
  );
CREATE TABLE Purchased(
  Username VARCHAR2(100) REFERENCES Customer(Username),
  Purchase_ID int REFERENCES Purchase(Purchase_ID),
  Title VARCHAR2(100) REFERENCES Movie(Title),
  FOREIGN KEY (Username) REFERENCES Customer(Username),
  FOREIGN KEY (Purchase_ID) REFERENCES Purchase(Purchase_ID),
  FOREIGN KEY (Title) REFERENCES Movie(Title)
);
CREATE TABLE Sells_Rights(
    Company_ID int PRIMARY KEY,
    Title VARCHAR2(100),
    FOREIGN KEY (Title) REFERENCES Movie(Title),
    FOREIGN KEY (Company_ID) REFERENCES Production_Company(Company_ID)
    );
CREATE TABLE Rating(
    Title VARCHAR2(100) PRIMARY KEY,
    Rating int,
    FOREIGN KEY (Title) REFERENCES Movie(Title),
    FOREIGN KEY (Rating) REFERENCES Movie_Rating(Rating)
    );
```

Simple Queries

```
SELECT DISTINCT Username, Card_Holder_Name
FROM Customer, Payment
ORDER BY Username DESC

SELECT Payment.Card_Holder_Name, payment.Card_Number
FROM Payment
INNER JOIN Billing_Address
ON Billing_Address.City = 'Toronto' AND Payment.Payment_ID =
Billing_Address.Payment_ID;

SELECT Company_ID
FROM Production_Company
WHERE Production_Company.Movies = 'Mean girls';

SELECT Rent.Price
FROM Rent
INNER JOIN Rented
ON Rented.Rental_ID = Rent.Rental_ID;
```

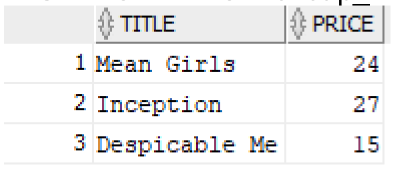
```
SELECT Price
FROM Purchase, Purchased
WHERE Purchased.Title = 'Inception' AND Purchased.Purchase_ID =
Purchase.Purchase_ID

SELECT Price
FROM Rent, Rented
WHERE Rented.Title = 'Mean Girls' AND Rented.Rental_ID =
Rent.Rental_ID

SELECT RENT.DATE_OF_RENTAL, rent.expiry_date
FROM rent
WHERE rental_id =1;

SELECT movie_rating.title, movie_rating.rating
FROM movie_rating
WHERE movie_Rating.rating = 5;

SELECT Can_Access.Username, Can_Access.Title
FROM Can_Access, Admin
WHERE Admin.Admin_ID = Can_Access.admin_ID;

SELECT Rating, COUNT(*)
FROM Movie_Rating
GROUP BY Rating;
```

```
 CREATE VIEW Movie_Ratings AS
   (SELECT movie_rating.title, movie_rating.rating
   FROM movie_rating
   WHERE movie_rating.rating = 5);

   SELECT * FROM Movie_ratings;
```



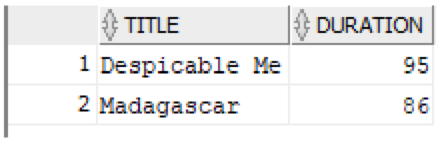
	TITLE	RATING
1	Inception	5
2	Good Will Hunting	5
3	Spider-Man No Way Home	5

```
   REPLACE VIEW Cheap_Movies_to_Buy AS
   (SELECT purchased.title, purchase.price
   FROM purchased, purchase
   WHERE purchase.purchase_ID = purchased.purchase_ID AND purchase.price
< 20);

   SELECT * FROM cheap_movies_to_buy;
```



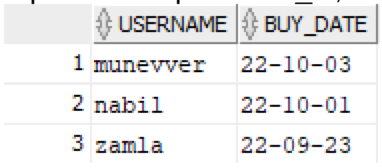
	TITLE	PRICE
1	Mean Girls	24
2	Inception	27
3	Despicable Me	15

```
   CREATE VIEW Under_2Hrs AS
   (SELECT movie.title, movie.duration
   FROM movie
   WHERE movie.duration < 120);

   SELECT * FROM Under_2Hrs;
```



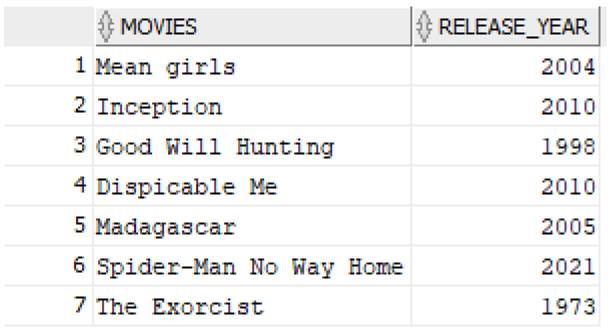
	TITLE	DURATION
1	Despicable Me	95
2	Madagascar	86

```
  SELECT Customer.username, purchase.Buy_date
  FROM Customer, Purchase, Purchased
  WHERE Customer.username = purchased.username AND purchase.purchase_id
= purchased.purchase_id;
```



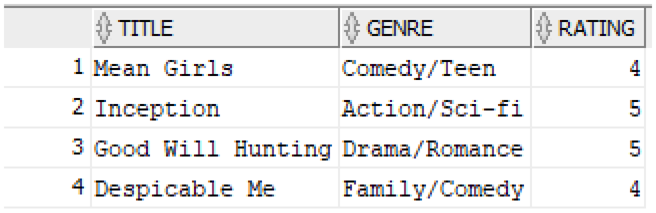
	USERNAME	BUY_DATE
1	munevver	22-10-03
2	nabil	22-10-01
3	zamla	22-09-23

```
  SELECT production_company.movies, movie.release_year
  FROM movie, sells_rights, production_company
  WHERE production_company.company_id = sells_rights.company_id AND
movie.title = sells_rights.title;
```



	MOVIES	RELEASE_YEAR
1	Mean girls	2004
2	Inception	2010
3	Good Will Hunting	1998
4	Dispicable Me	2010
5	Madagascar	2005
6	Spider-Man No Way Home	2021
7	The Exorcist	1973

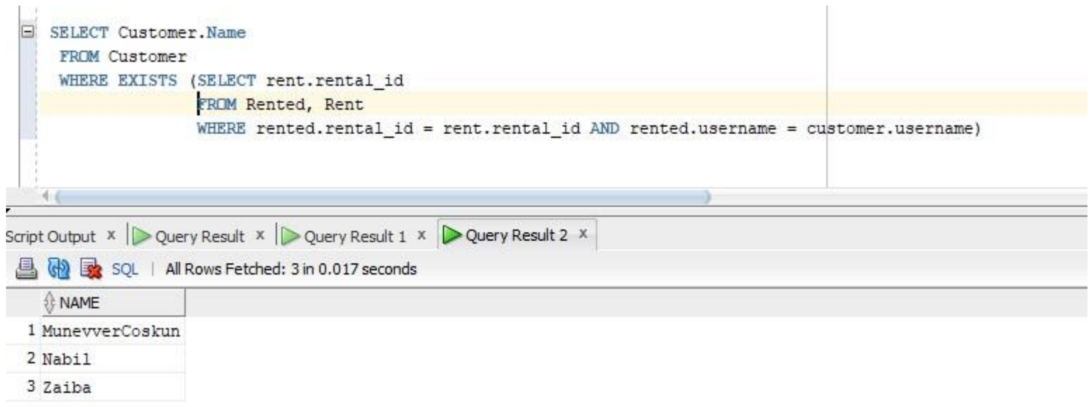
```
SELECT movie.title, movie.genre, movie_rating.rating
FROM movie, rating, movie_rating
WHERE movie.title = rating.title AND movie_rating.title = rating.title;
```



	TITLE	GENRE	RATING
1	Mean Girls	Comedy/Teen	4
2	Inception	Action/Sci-fi	5
3	Good Will Hunting	Drama/Romance	5
4	Despicable Me	Family/Comedy	4

Advanced Queries by UNIX Shell Implementation

1.



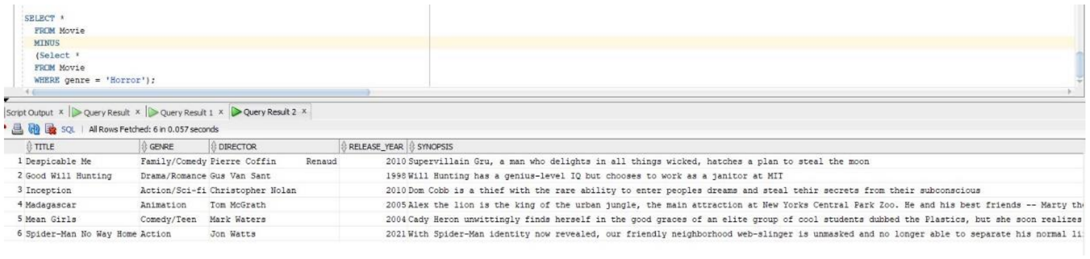
```
SELECT Customer.Name
 FROM Customer
 WHERE EXISTS (SELECT rent.rental_id
              FROM Rented, Rent
              WHERE rented.rental_id = rent.rental_id AND rented.username = customer.username)
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 3 in 0.017 seconds

	NAME
1	MunevverCoskun
2	Nabil
3	Zaiba

2.



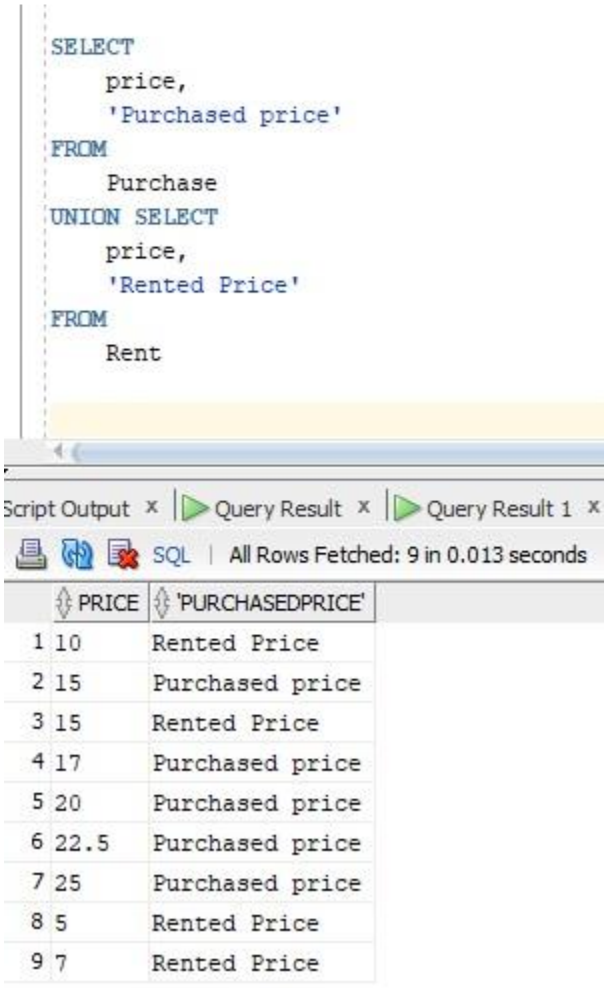
```
SELECT *
  FROM Movie
  MINUS
  (Select *
  FROM Movie
  WHERE genre = 'Horror');
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 6 in 0.057 seconds

	TITLE	GENRE	DIRECTOR	RELEASE_YEAR	SYNOPSIS
1	Despicable Me	Family/Comedy	Pierre Coffin Renaud	2010	Supervillain Gru, a man who delights in all things wicked, hatches a plan to steal the moon
2	Good Will Hunting	Drama/Romance	Gus Van Sant	1998	Will Hunting has a genius-level IQ but chooses to work as a janitor at MIT
3	Inception	Action/Sci-fi	Christopher Nolan	2010	Dom Cobb is a thief with the rare ability to enter peoples dreams and steal tehir secrets from their subconscious
4	Madagascar	Animation	Tom McGrath	2005	Alex the lion is the king of the urban jungle, the main attraction at New Yorks Central Park Zoo. He and his best friends -- Marty th
5	Mean Girls	Comedy/Teen	Mark Waters	2004	Cady Heron unwittingly finds herself in the good graces of an elite group of cool students dubbed the Plastics, but she soon realizes
6	Spider-Man No Way Home	Action	Jon Watts	2021	With Spider-Man identity now revealed, our friendly neighborhood web-slinger is unmasked and no longer able to separate his normal li

3.



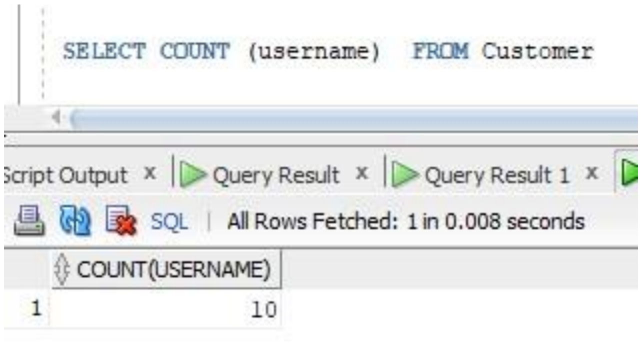
```
SELECT
    price,
    'Purchased price'
FROM
    Purchase
UNION SELECT
    price,
    'Rented Price'
FROM
    Rent
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 9 in 0.013 seconds

	PRICE	'PURCHASEDPRICE'
1	10	Rented Price
2	15	Purchased price
3	15	Rented Price
4	17	Purchased price
5	20	Purchased price
6	22.5	Purchased price
7	25	Purchased price
8	5	Rented Price
9	7	Rented Price

4.



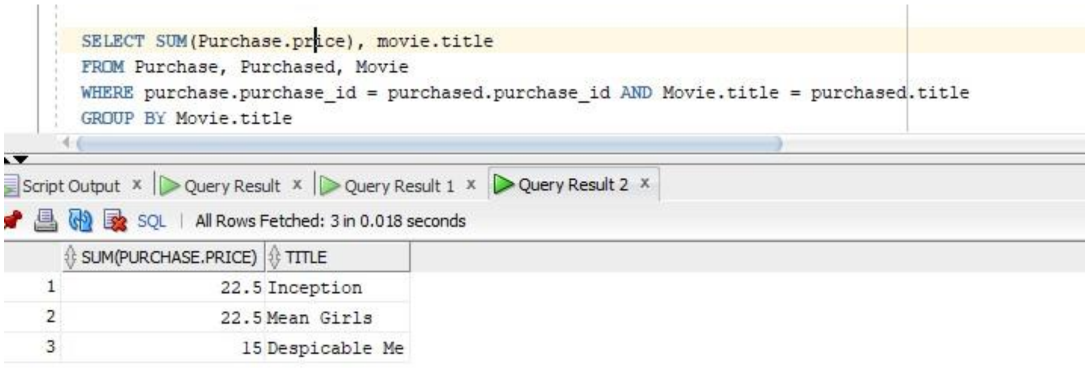
```
SELECT COUNT (username)  FROM Customer
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 1 in 0.008 seconds

	COUNT(USERNAME)
1	10

5.



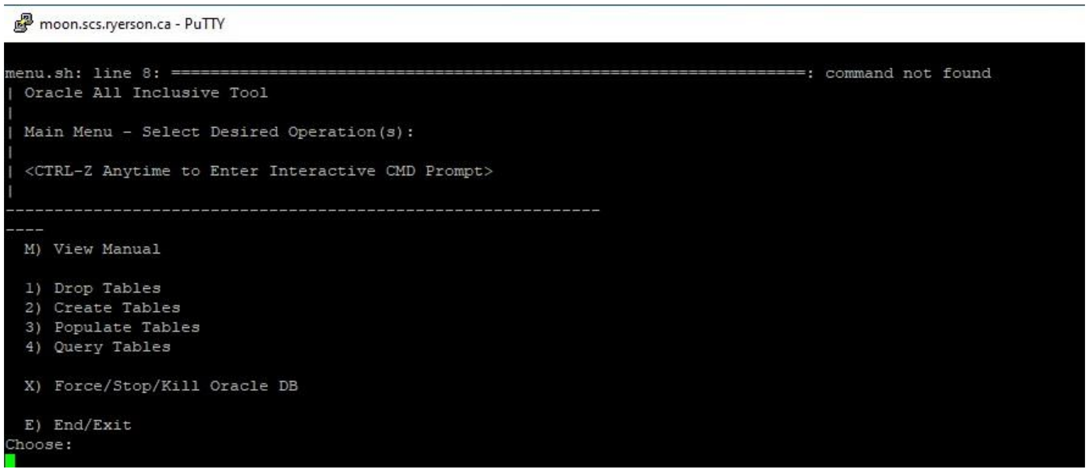
```
SELECT SUM(Purchase.price), movie.title
FROM Purchase, Purchased, Movie
WHERE purchase.purchase_id = purchased.purchase_id AND Movie.title = purchased.title
GROUP BY Movie.title
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 3 in 0.018 seconds

	SUM(PURCHASE.PRICE)	TITLE
1	22.5	Inception
2	22.5	Mean Girls
3	15	Despicable Me

Result of "bash menu.sh" command:

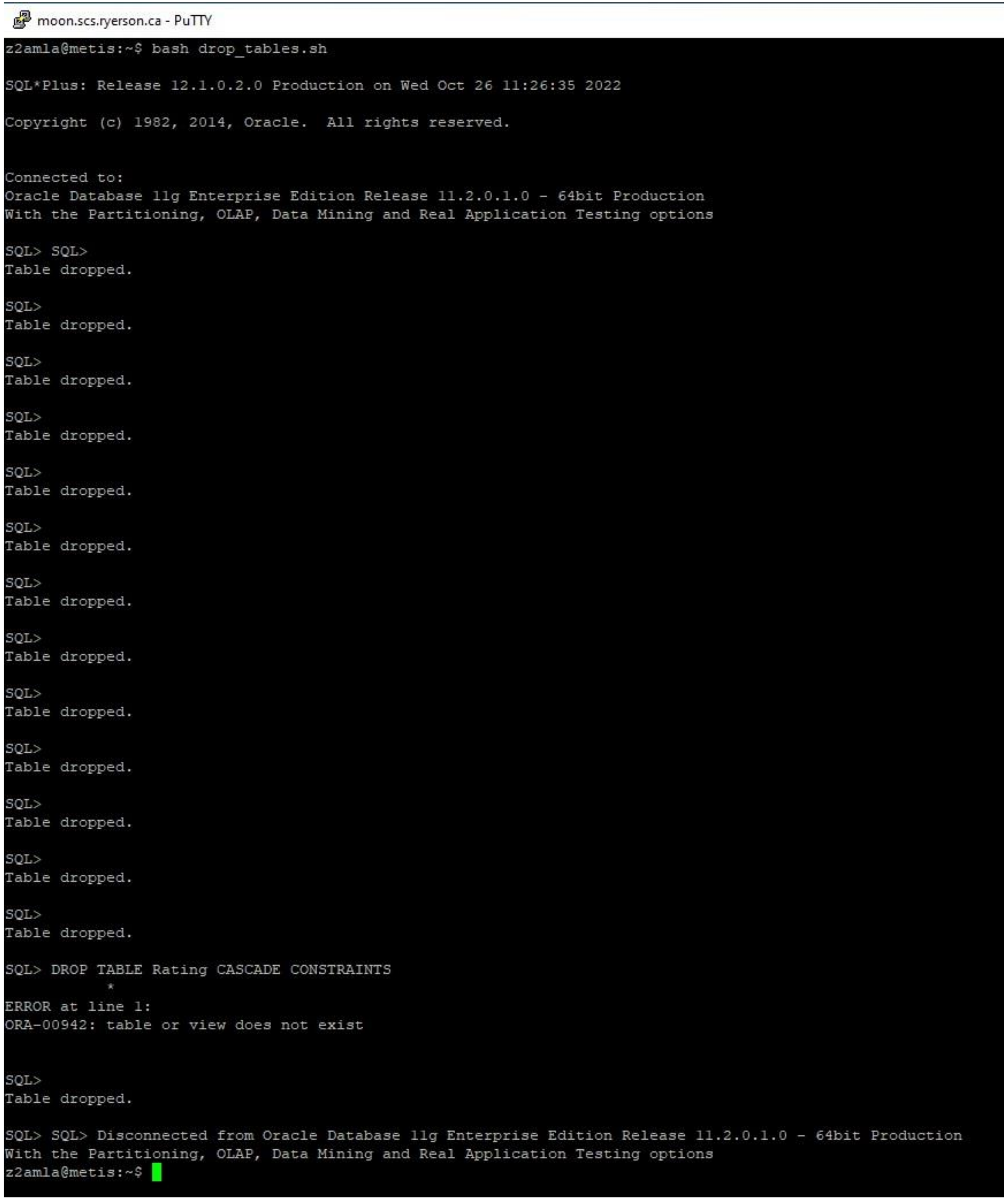


moon.scs.ryerson.ca - PuTTY

```
menu.sh: line 8: =================================================================: command not found
| Oracle All Inclusive Tool
|
| Main Menu - Select Desired Operation(s):
|
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
|
------------------------------------------------------------
----
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
```

Result of “bash drop_tables.sh” command:

Using the command “bash menu.sh >> 1” to drop tables also works successfully.

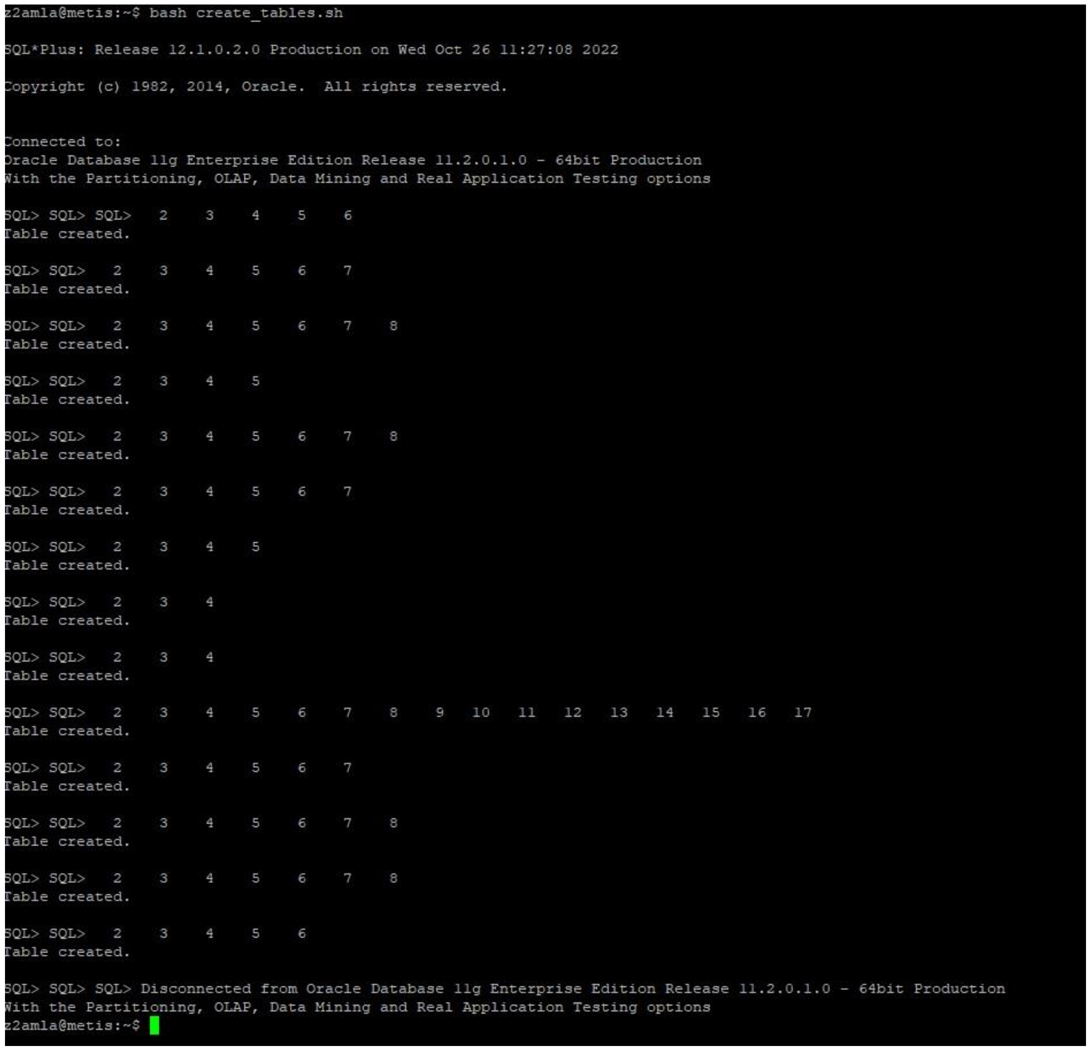


moon.scs.ryerson.ca - PuTTY

```
z2amla@metis:~$ bash drop_tables.sh

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 26 11:26:35 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL> DROP TABLE Rating CASCADE CONSTRAINTS
           *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL>
Table dropped.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
z2amla@metis:~$
```

Result of “bash create_tables.sh” command:

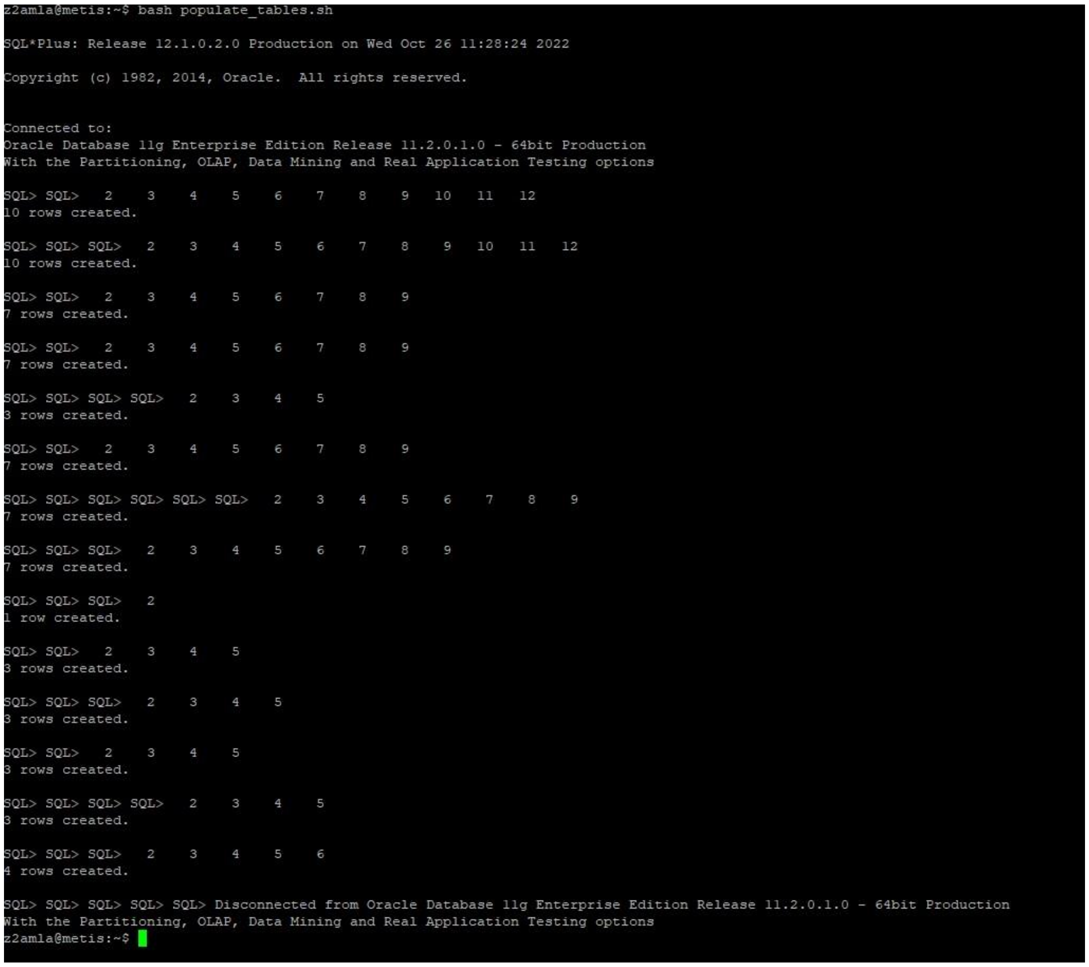
Using the command “bash menu.sh >> 2” to create tables also works successfully.



```
z2amla@metis:~$ bash create_tables.sh

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 26 11:27:08 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4
Table created.

SQL> SQL>   2    3    4
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
z2amla@metis:~$
```

Result of “bash populate_tables.sh” command:

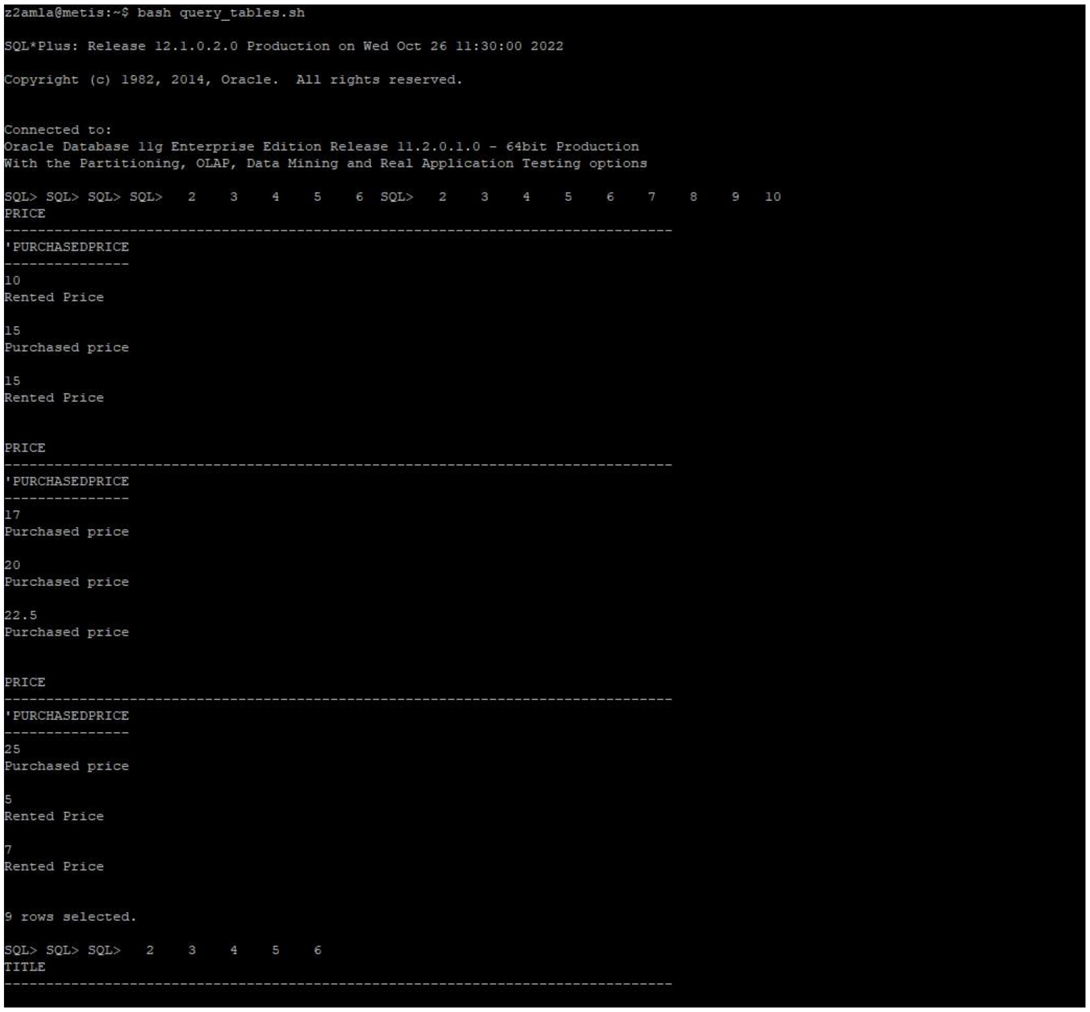
Using the command “bash menu.sh >> 3” to populate tables also works successfully.



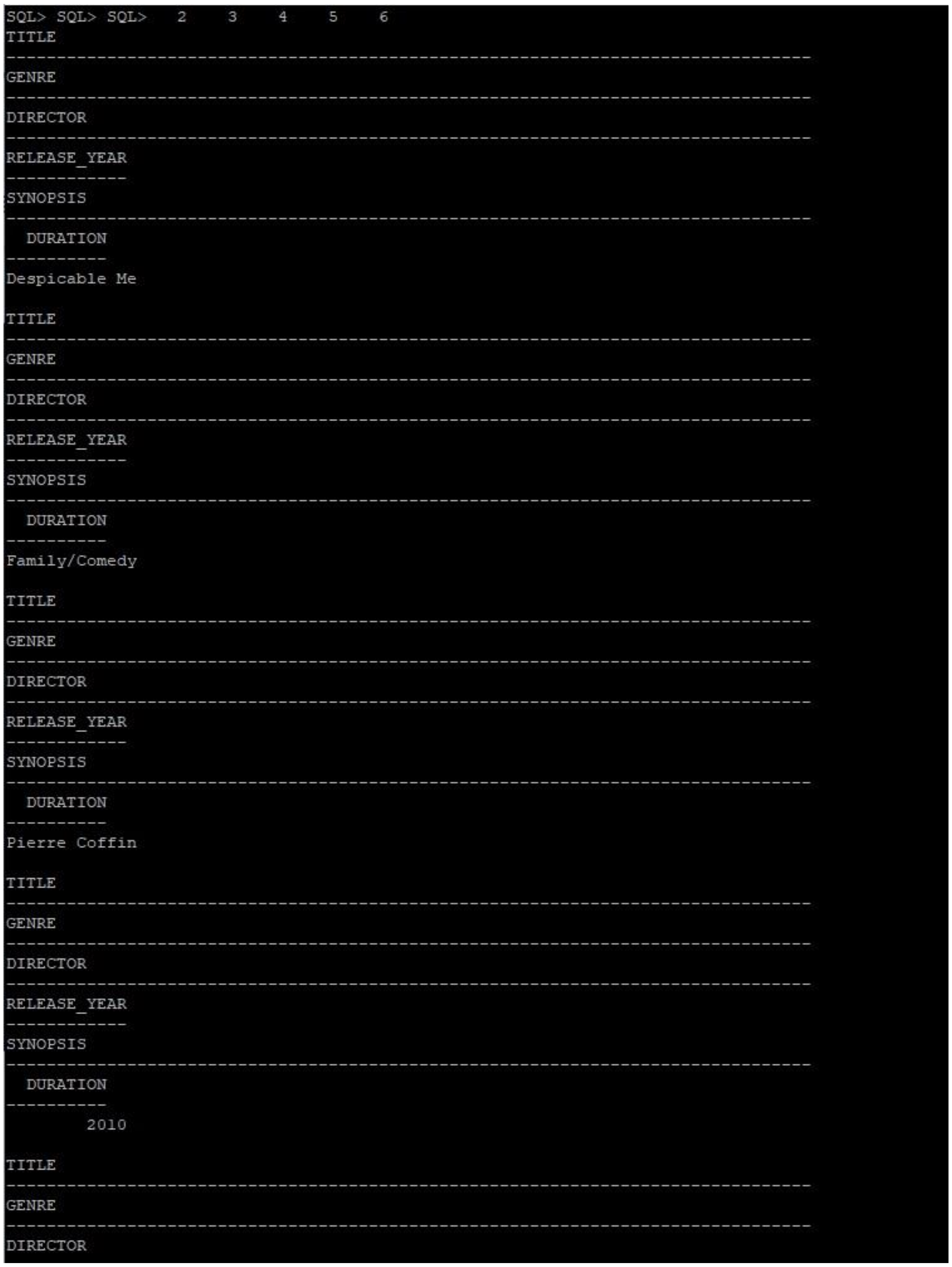
```
z2amla@metis:~$ bash populate_tables.sh

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 26 11:28:24 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12
10 rows created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12
10 rows created.

SQL> SQL>   2    3    4    5    6    7    8    9
7 rows created.

SQL> SQL>   2    3    4    5    6    7    8    9
7 rows created.

SQL> SQL> SQL> SQL>   2    3    4    5
3 rows created.

SQL> SQL>   2    3    4    5    6    7    8    9
7 rows created.

SQL> SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8    9
7 rows created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9
7 rows created.

SQL> SQL> SQL>   2
1 row created.

SQL> SQL>   2    3    4    5
3 rows created.

SQL> SQL> SQL>   2    3    4    5
3 rows created.

SQL> SQL>   2    3    4    5
3 rows created.

SQL> SQL> SQL> SQL>   2    3    4    5
3 rows created.

SQL> SQL> SQL>   2    3    4    5    6
4 rows created.

SQL> SQL> SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
z2amla@metis:~$
```

Result of “bash query_tables.sh” command:

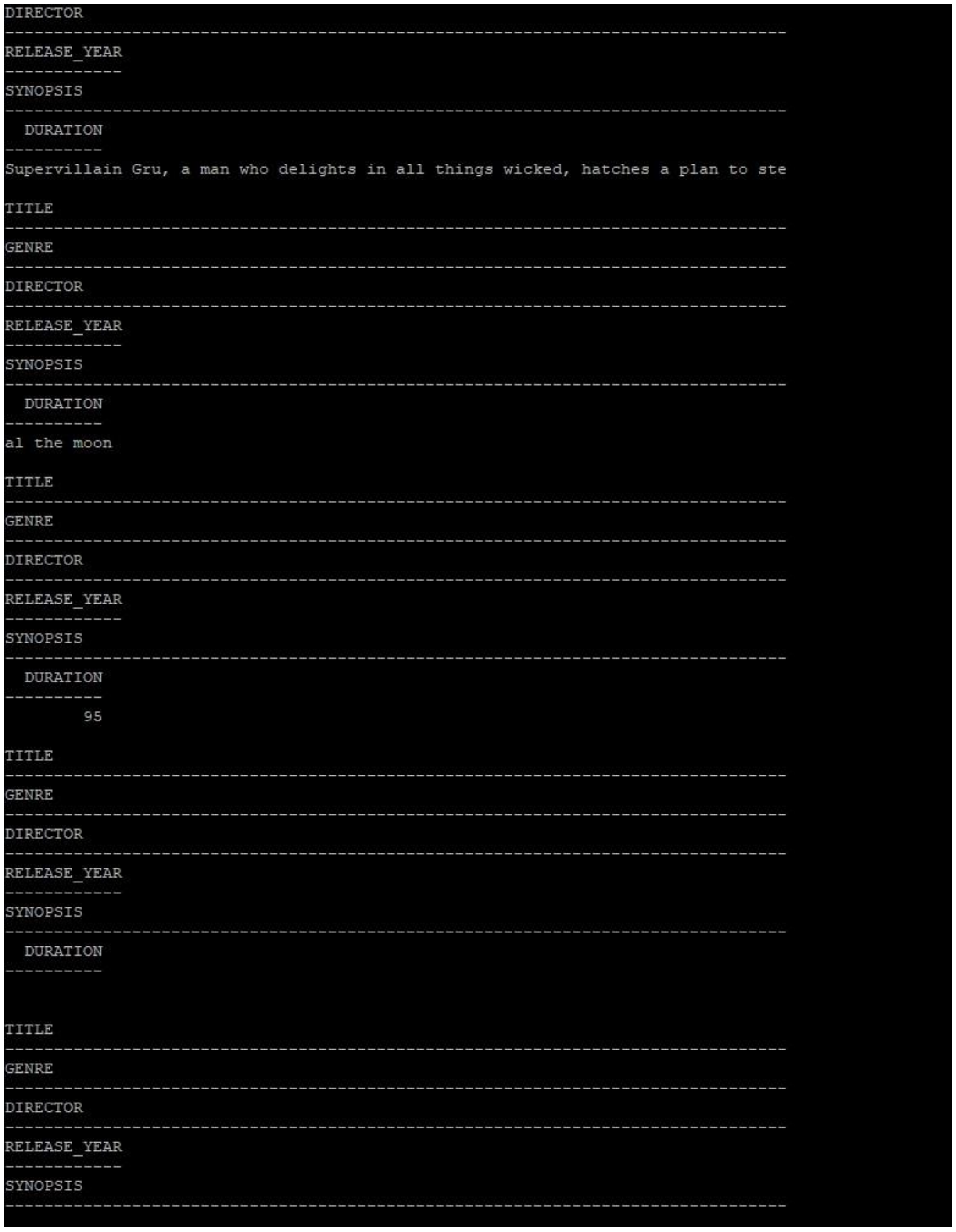
Using the command “bash menu.sh >> 4” to execute queries tables also works successfully.



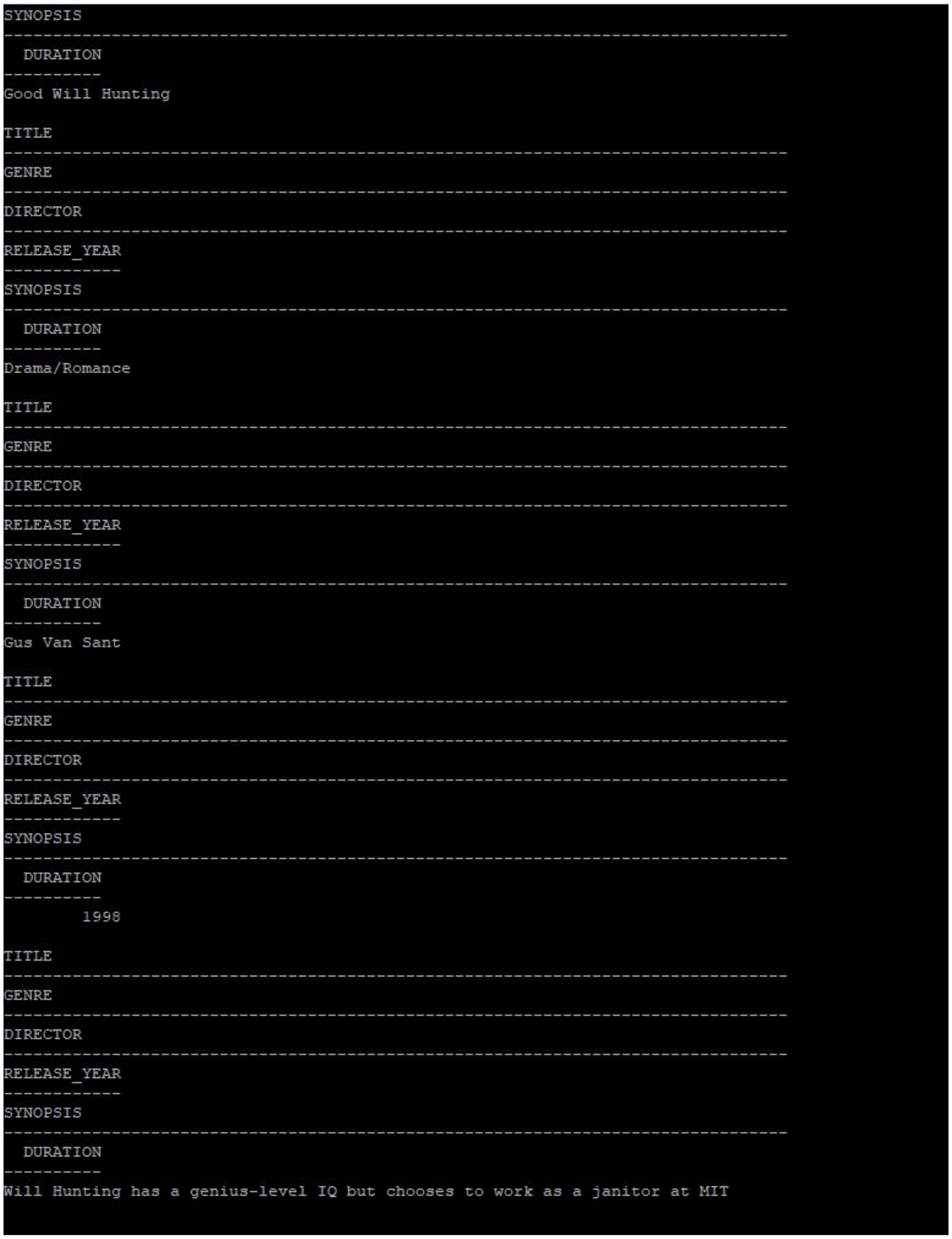
```
z2amla@metis:~$ bash query_tables.sh

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 26 11:30:00 2022

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>   2    3    4    5    6  SQL>   2    3    4    5    6    7    8    9   10
PRICE
--------------------------------------------------------------------------------
'PURCHASEDPRICE
---------------
10
Rented Price

15
Purchased price

15
Rented Price


PRICE
--------------------------------------------------------------------------------
'PURCHASEDPRICE
---------------
17
Purchased price

20
Purchased price

22.5
Purchased price


PRICE
--------------------------------------------------------------------------------
'PURCHASEDPRICE
---------------
25
Purchased price

5
Rented Price

7
Rented Price


9 rows selected.

SQL> SQL> SQL>   2    3    4    5    6
TITLE
--------------------------------------------------------------------------------
```



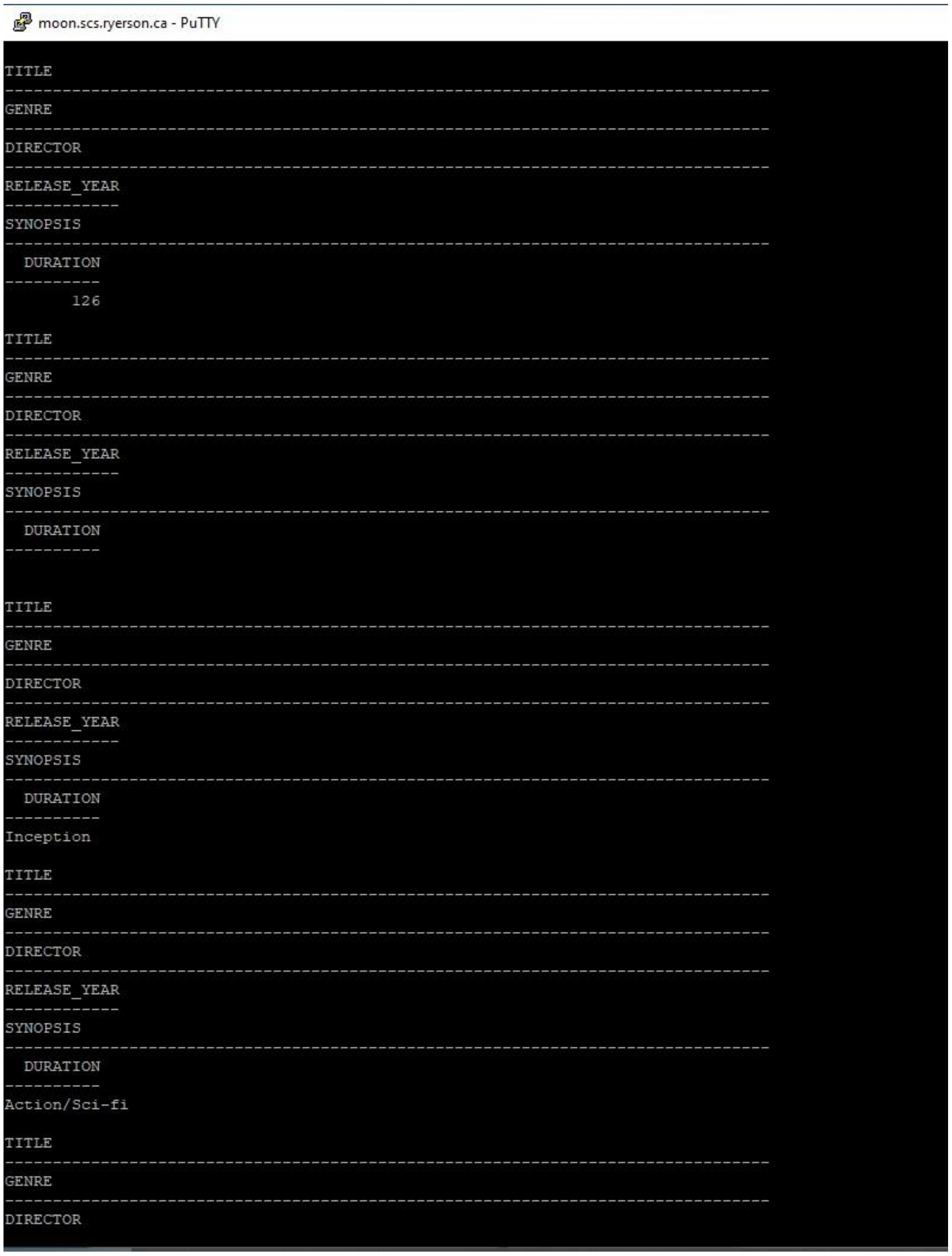
```
SQL> SQL> SQL>   2    3    4    5    6
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Despicable Me

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Family/Comedy

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Pierre Coffin

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        2010

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
```



```
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Supervillain Gru, a man who delights in all things wicked, hatches a plan to ste

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
al the moon

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        95

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
```

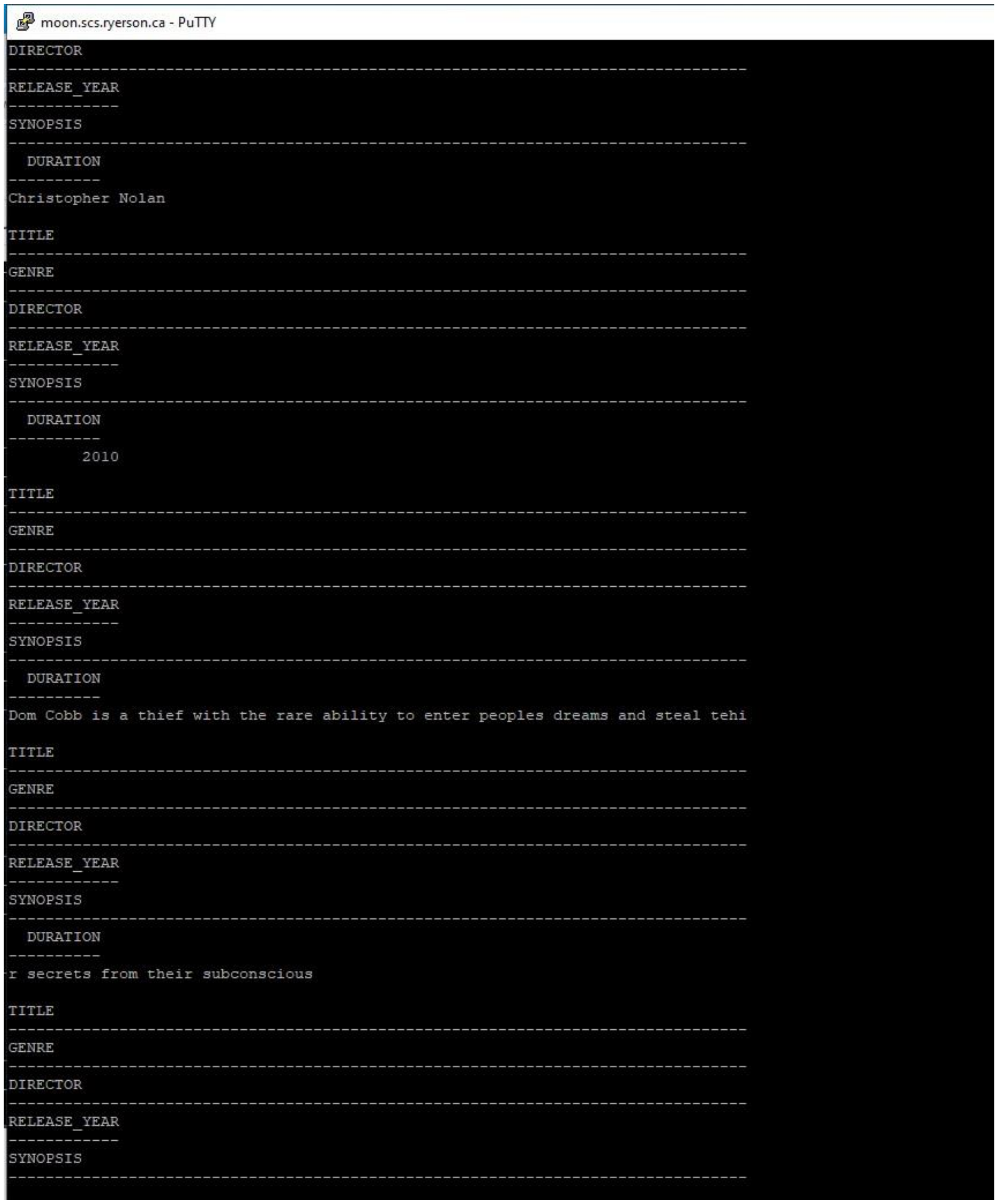


```
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Good Will Hunting

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Drama/Romance

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Gus Van Sant

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        1998

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Will Hunting has a genius-level IQ but chooses to work as a janitor at MIT
```

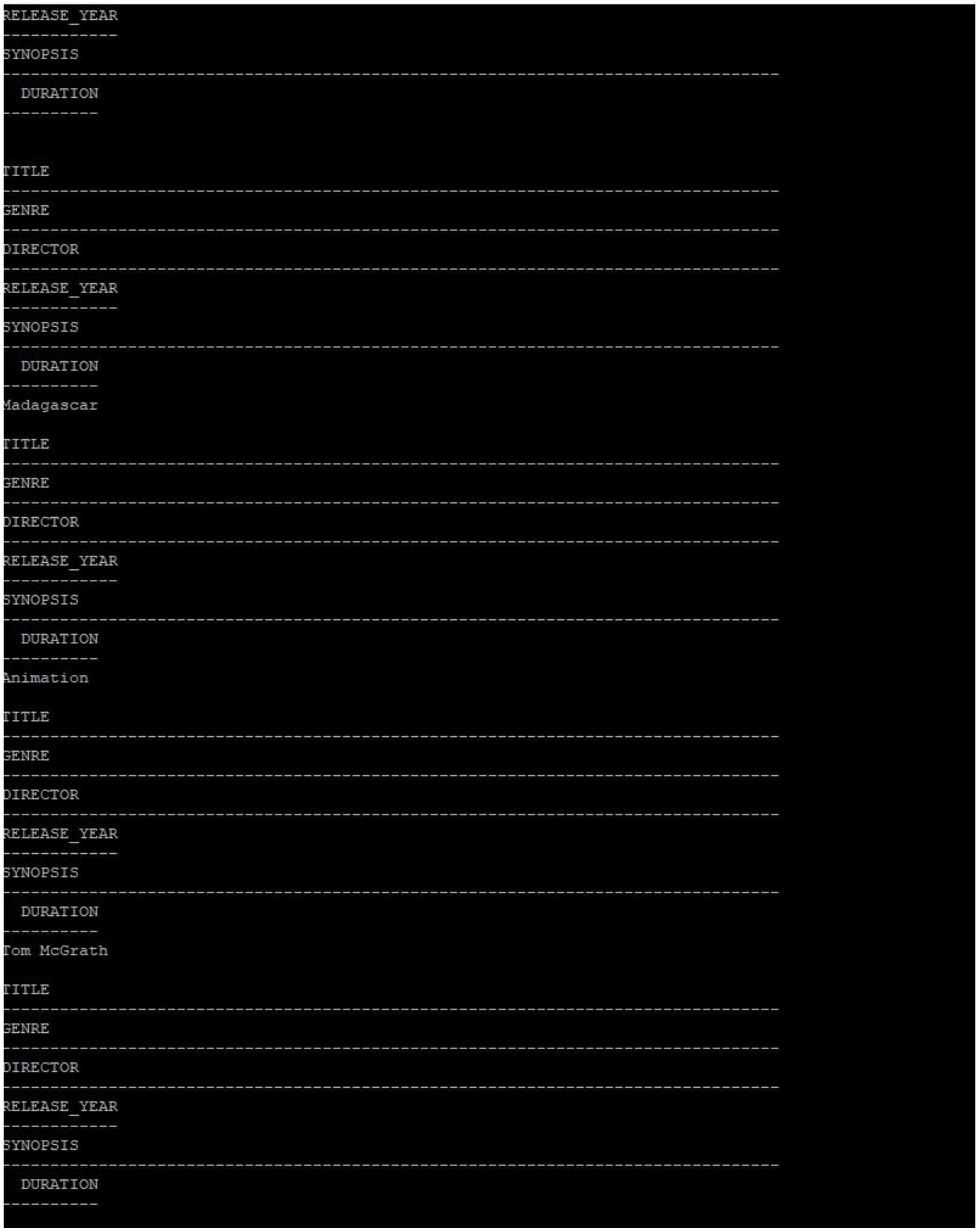


moon.scs.ryerson.ca - PuTTY

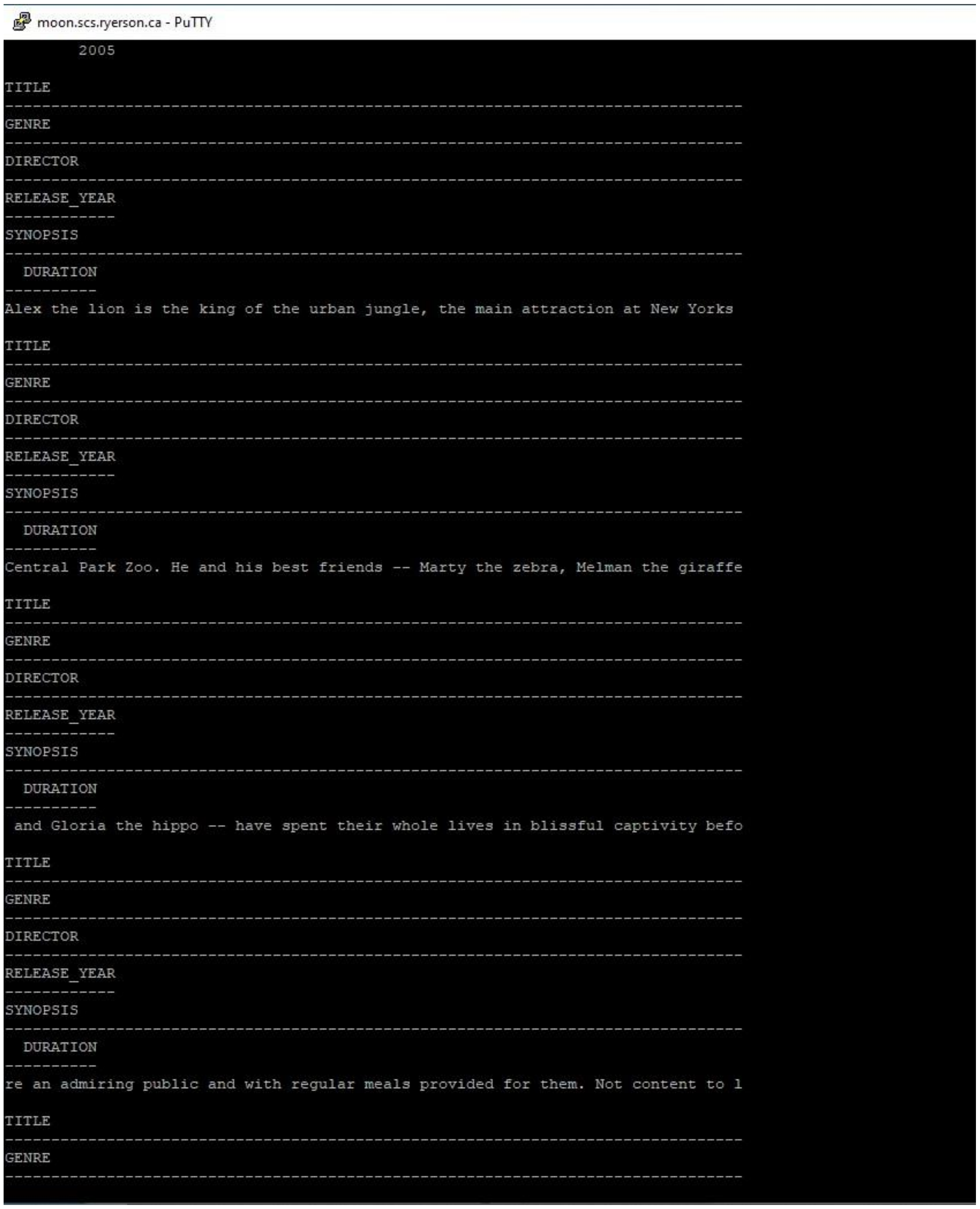
```
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
       126

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Inception

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Action/Sci-fi

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
```


moon.scs.ryerson.ca - PuTTY

```
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Christopher Nolan

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        2010

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Dom Cobb is a thief with the rare ability to enter peoples dreams and steal tehi

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
r secrets from their subconscious

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
```

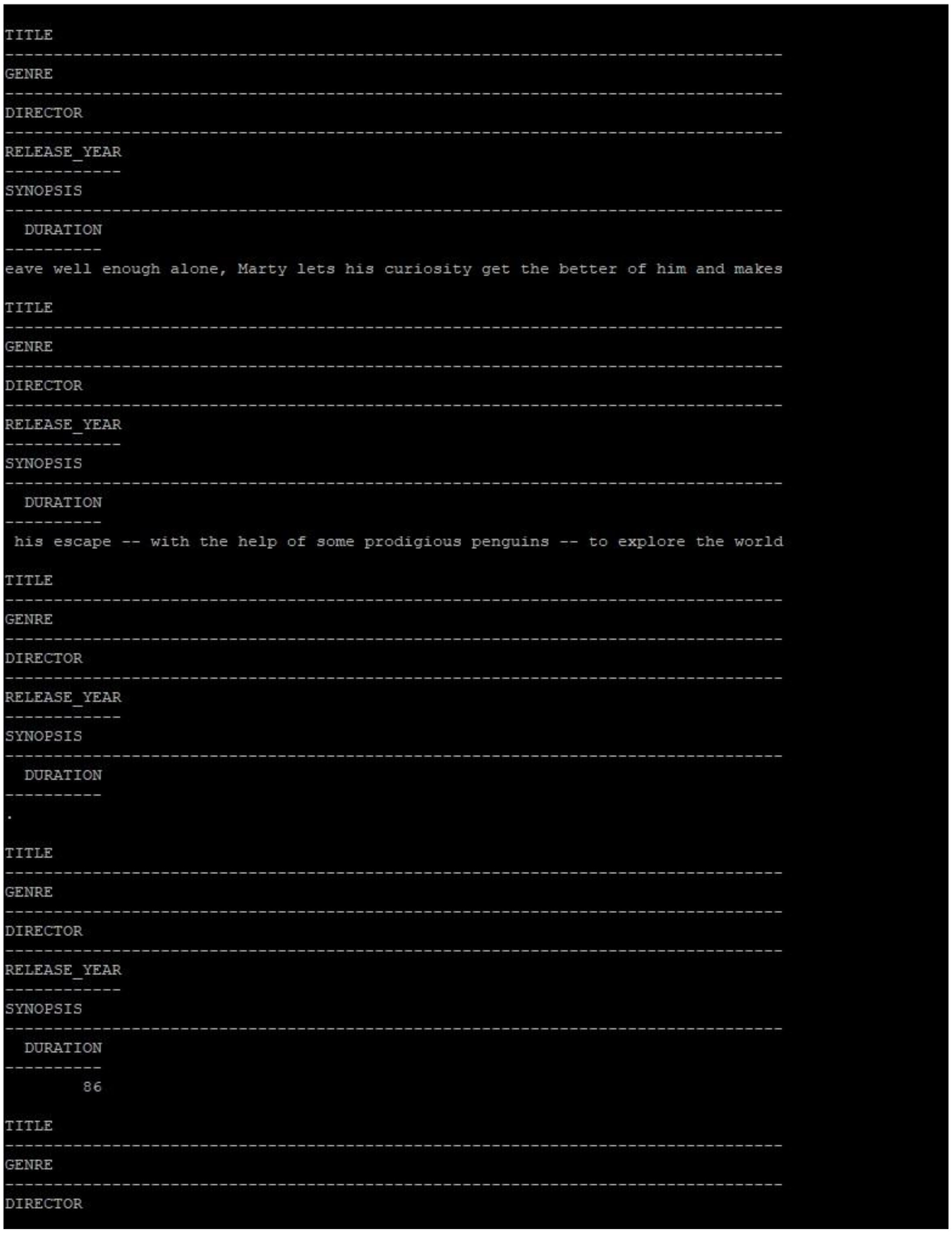


```
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Madagascar

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Animation

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Tom McGrath

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
```

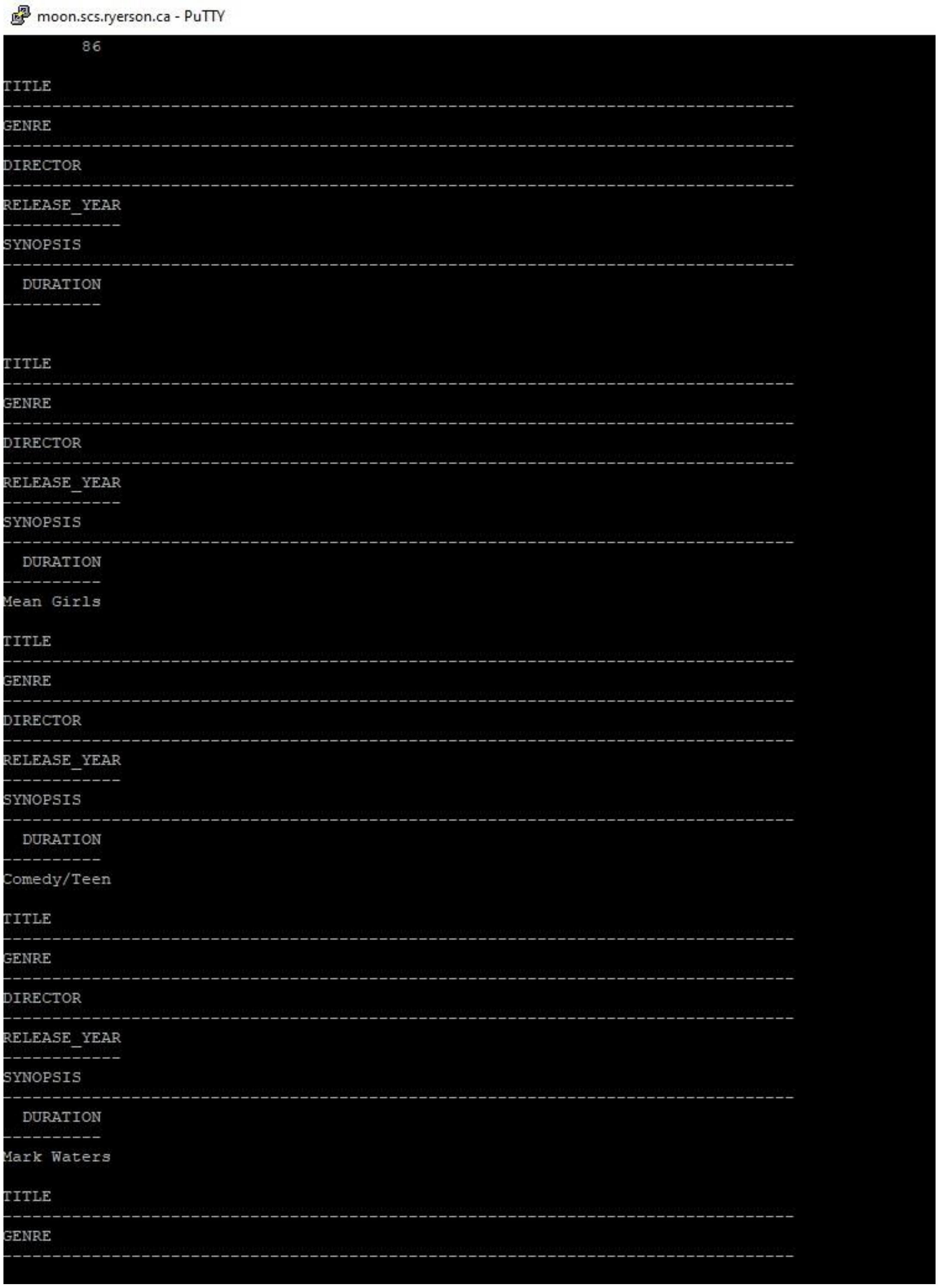


moon.scs.ryerson.ca - PuTTY

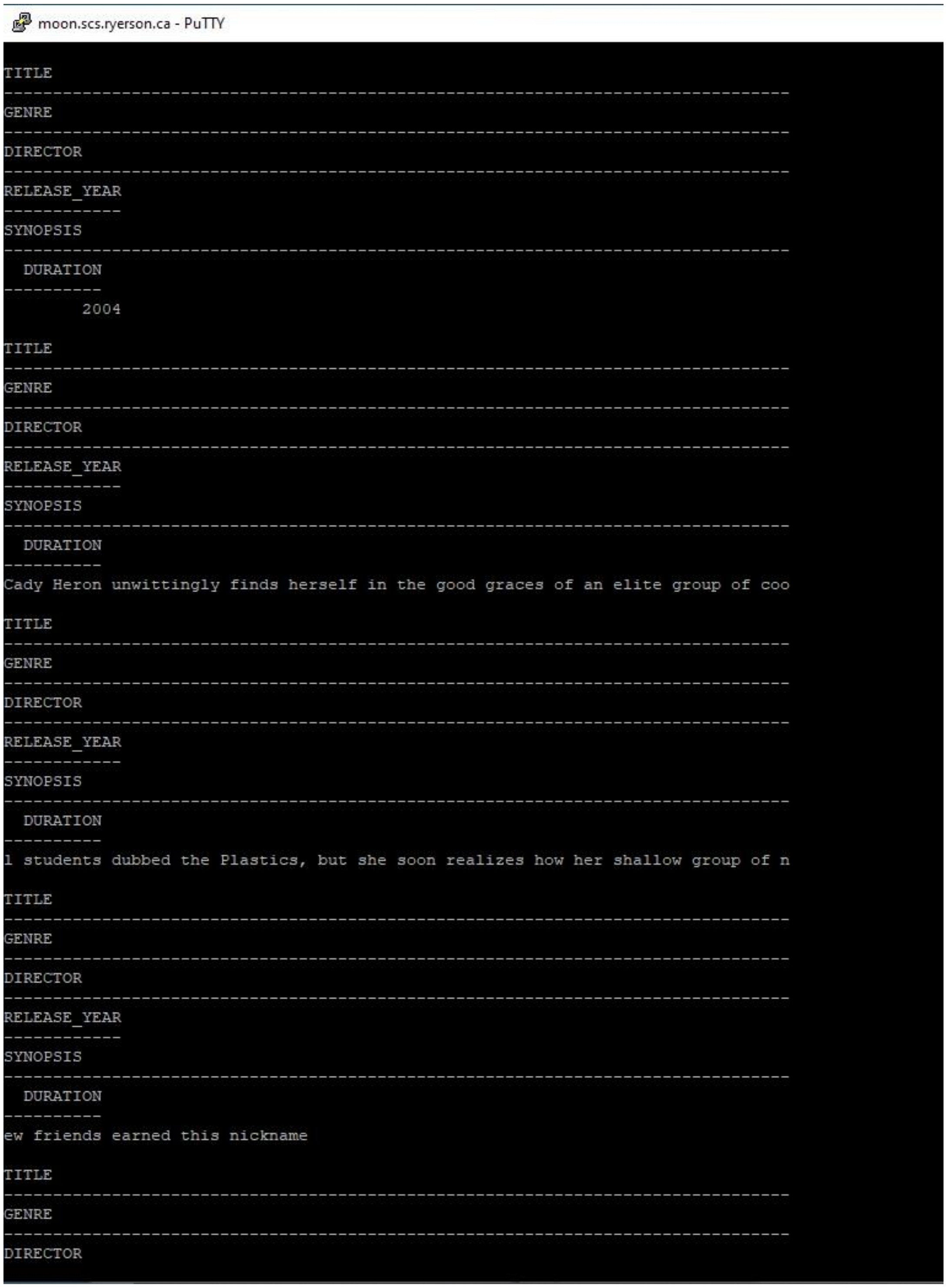
```
        2005

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Alex the lion is the king of the urban jungle, the main attraction at New Yorks

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Central Park Zoo. He and his best friends -- Marty the zebra, Melman the giraffe

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
 and Gloria the hippo -- have spent their whole lives in blissful captivity befo

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
re an admiring public and with regular meals provided for them. Not content to l

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
```



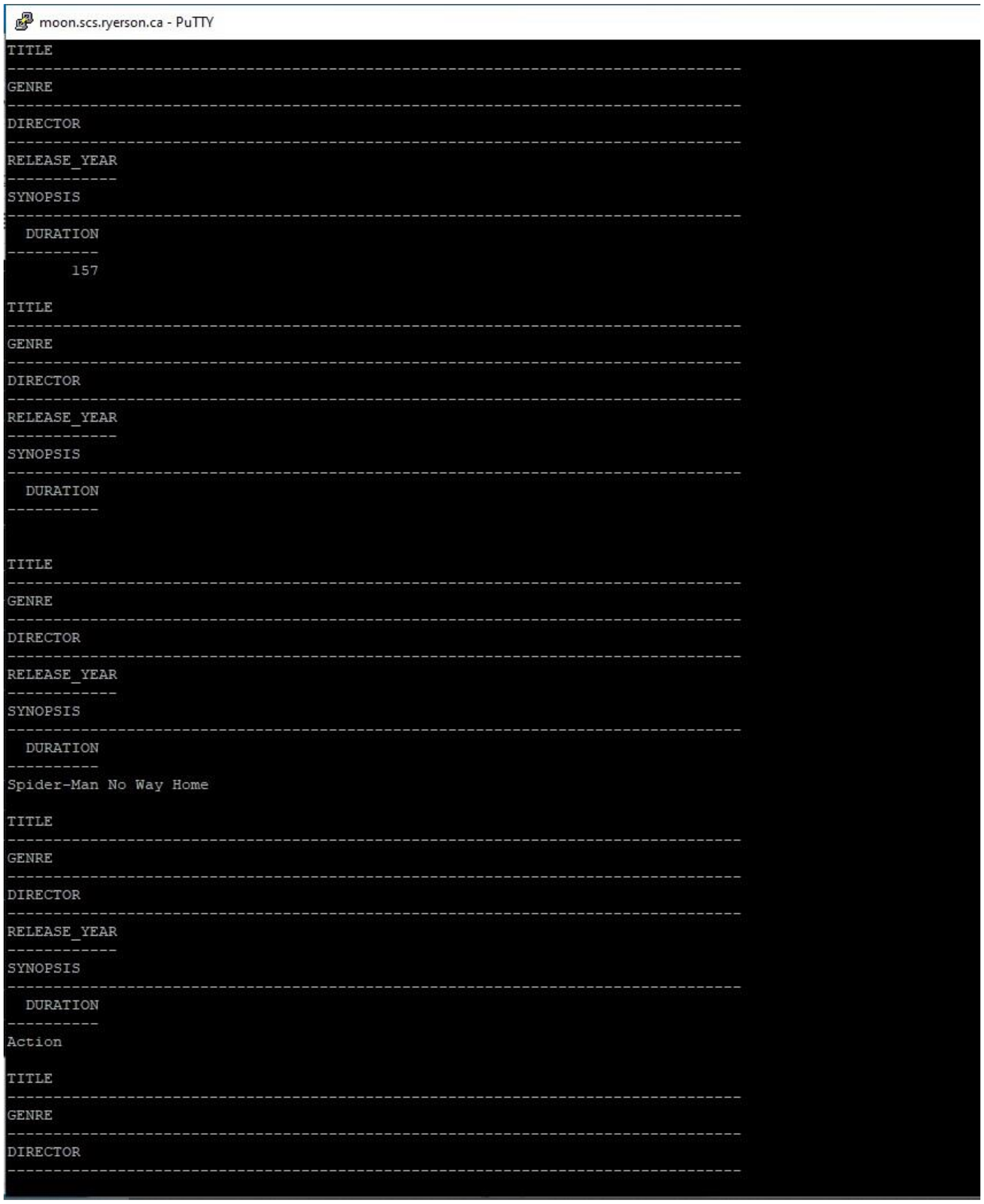
```
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
eave well enough alone, Marty lets his curiosity get the better of him and makes

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
 his escape -- with the help of some prodigious penguins -- to explore the world

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
.

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        86

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
```



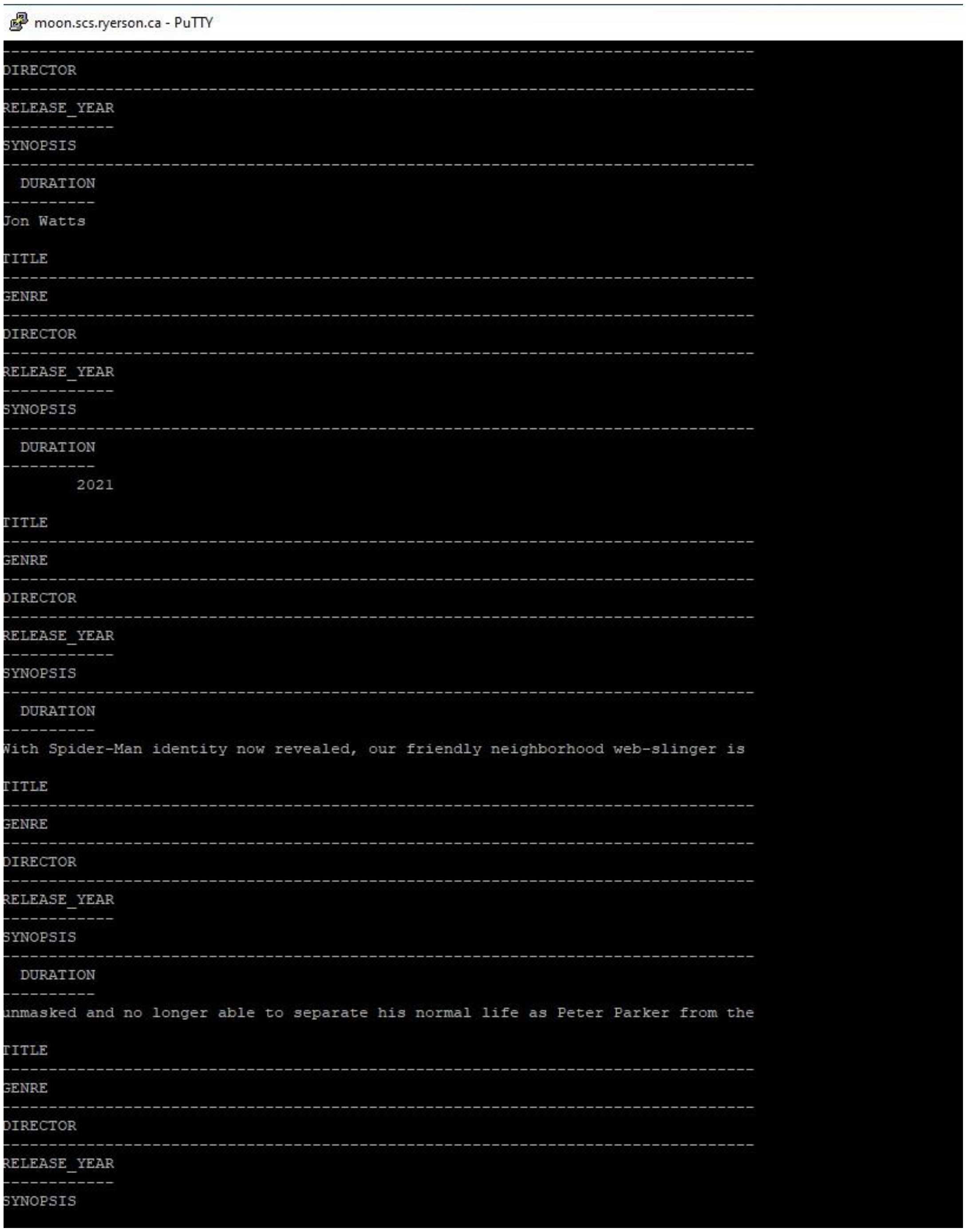
```
        86

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Mean Girls

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Comedy/Teen

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Mark Waters

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
```


moon.scs.ryerson.ca - PuTTY

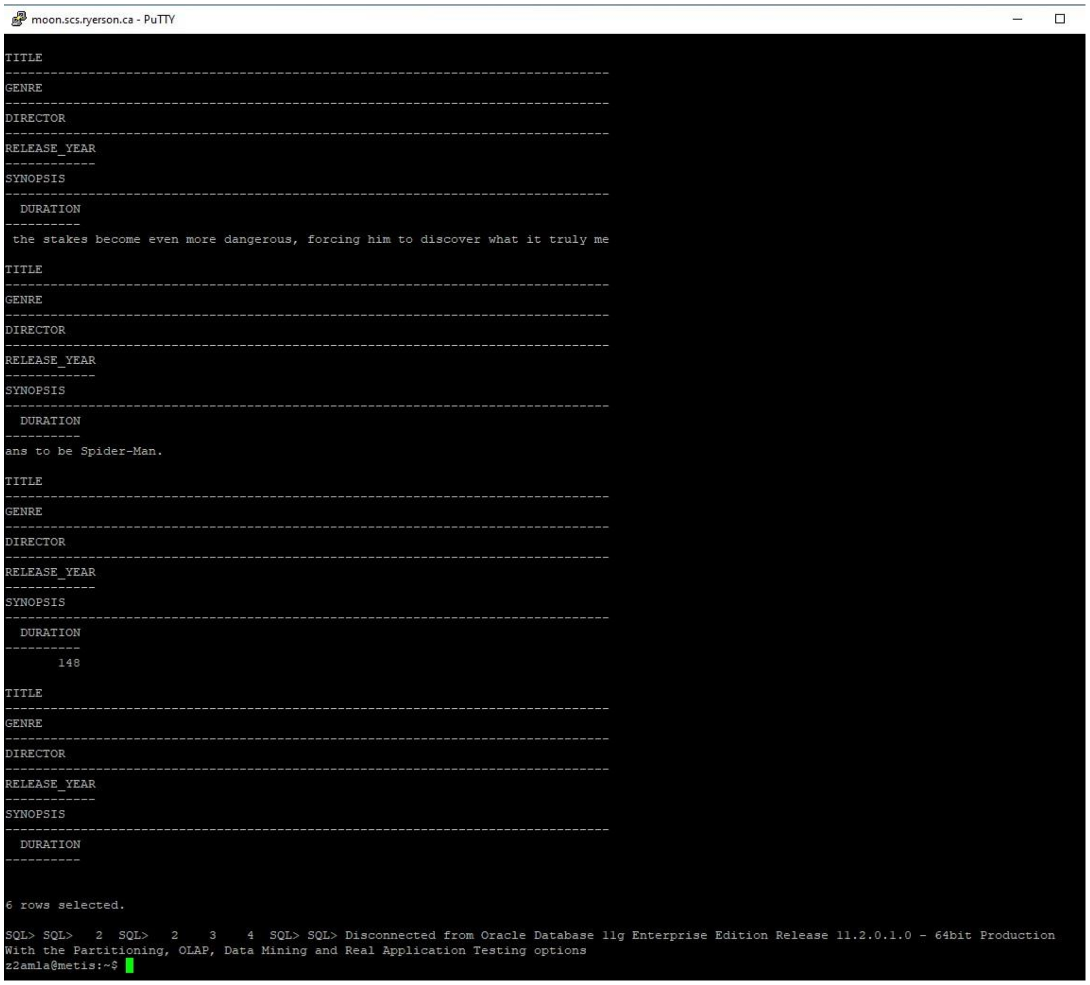
```
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        2004

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Cady Heron unwittingly finds herself in the good graces of an elite group of coo

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
l students dubbed the Plastics, but she soon realizes how her shallow group of n

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
ew friends earned this nickname

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
```



```
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
       157

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Spider-Man No Way Home

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Action

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
```


moon.scs.ryerson.ca - PuTTY

```
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
Jon Watts

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
        2021

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
With Spider-Man identity now revealed, our friendly neighborhood web-slinger is

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
unmasked and no longer able to separate his normal life as Peter Parker from the

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
```



```
TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
 the stakes become even more dangerous, forcing him to discover what it truly me

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
ans to be Spider-Man.

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------
       148

TITLE
--------------------------------------------------------------------------------
GENRE
--------------------------------------------------------------------------------
DIRECTOR
--------------------------------------------------------------------------------
RELEASE_YEAR
------------
SYNOPSIS
--------------------------------------------------------------------------------
  DURATION
----------


6 rows selected.

SQL> SQL>   2  SQL>   2    3    4  SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
z2amla@metis:~$
```

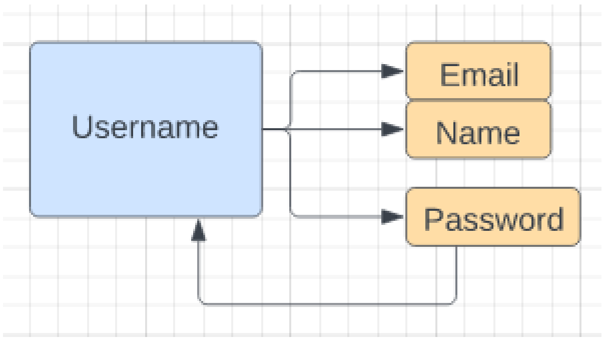
Normalization of the database/Functional Dependencies (1NF, 2NF, 3NF, BCNF)

Entity Tables

All Tables are 1NF because all values are atomic
All Tables are 2NF because they are all 1NF and all non-key attributes are fully functional and dependent on the primary key
All Tables are 3NF because they are all 1NF, 2NF, and all non-key attributes are non-transitively dependent on the primary key
All Tables are BCNF because every nontrivial, left irreducible functional dependency has a candidate key as its determinant

1. Customer

Customer(Username, Email, Name, Password)
Functional Dependency;
Username → Email, Name, Password
Password → Username



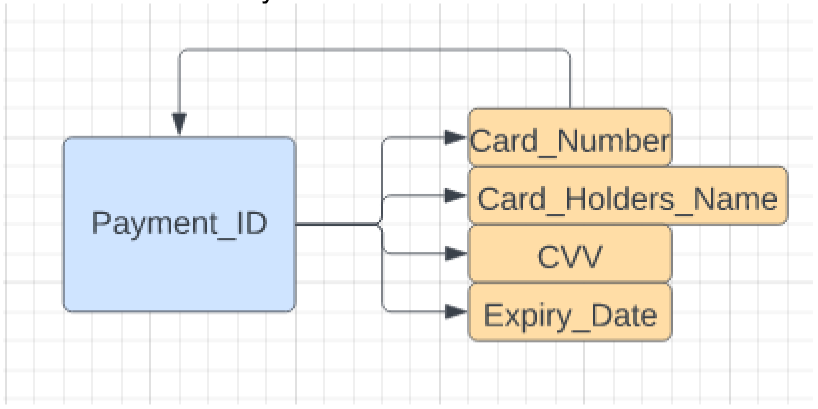
Username is dependent on password but since username is a non-candidate key the table is still 3NF.

Decomposition is present here since Username can relate to Email, Name, Password and password can relate back to username.

Password is a candidate key, so BCNF still holds

2. Payment

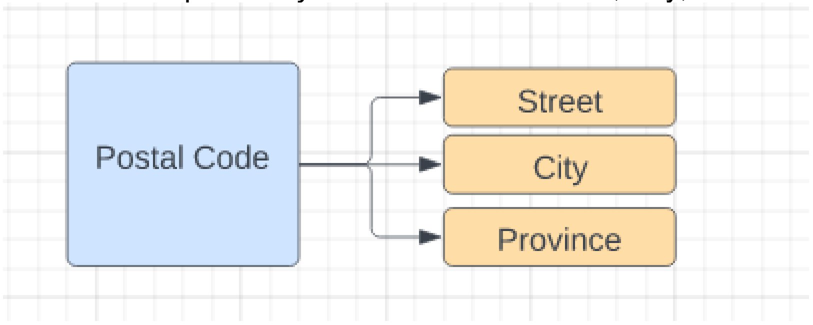
Payment (Payment ID, Card Number, Card Holders Name, CVV, Expiry Date)
Functional Dependency: Payment ID → Card Number, Card Holders Name, CVV, Expiry Date
Card Number → Payment ID



Payment_ID is dependent on card_number but since payment_ID is a non-candidate key the table is still 3NF
Decomposition is present here since Payment_ID can relate to Card Number, Card Holders Name, CVV, Expiry Date and Card Number can relate back to payment_ID.
Card_Number is a candidate key so BCNF still holds

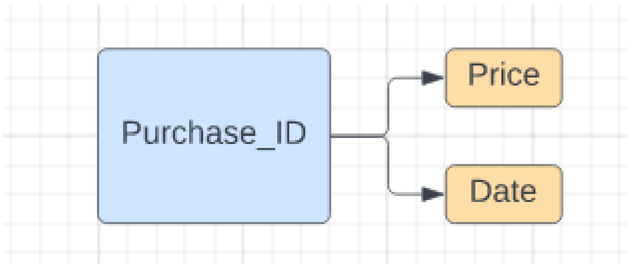
3. Billing Address

Billing_Address (Postal-Code, Street, City, Province)
Functional Dependency: Postal-Code → Street, City, Province



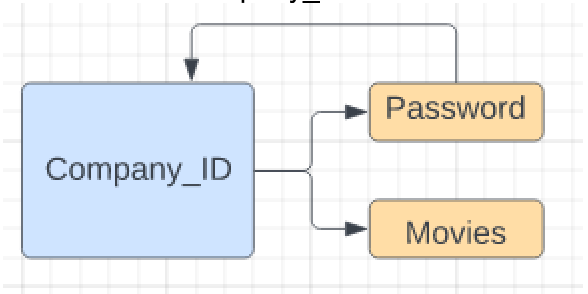
4. Purchase

Purchase(Purchase_ID, Price, Date)
Functional Dependency; Purchase_ID → Price, Date



5. Production_Company

Production_Company(Company_ID, Password, Movies)
Functional Dependency; Company_ID → Password, Movies
Password → Company_ID

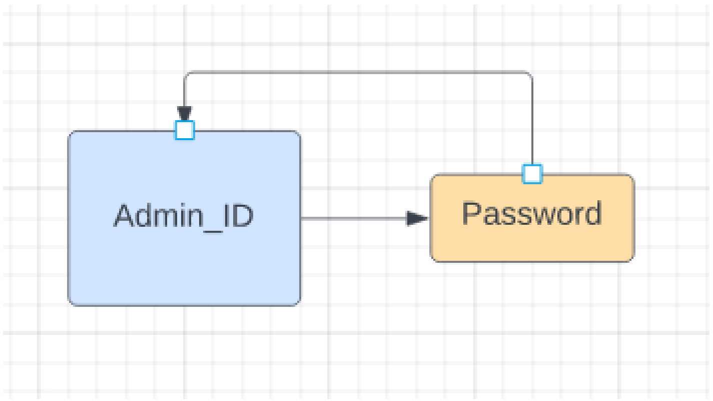


Company_ID depends on password but since Company_ID is a non-candidate key the table is still 3NF
Decomposition is present here since Company_ID can relate to Password and movies and Password can relate back to Company_ID.

Company_ID is a candidate key so BCNF still holds

6. Admin

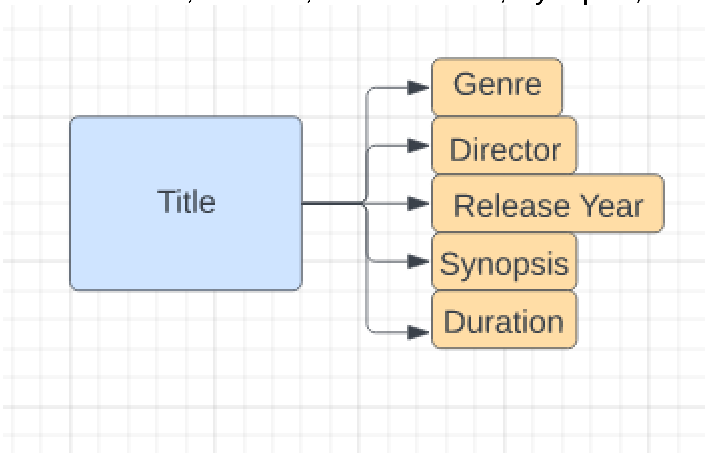
Admin(<u>Admin ID</u>, Password)
Functional Dependency: Admin ID → Password
Password → Admin_ID



Admin_ID depends on password but since Admin_ID is a non-candidate key the table is still 3NF
Decomposition is present here since Admin_ID can relate to Password and Password can relate back to Admin_ID.
Admin_ID is a candidate key so BCNF still holds

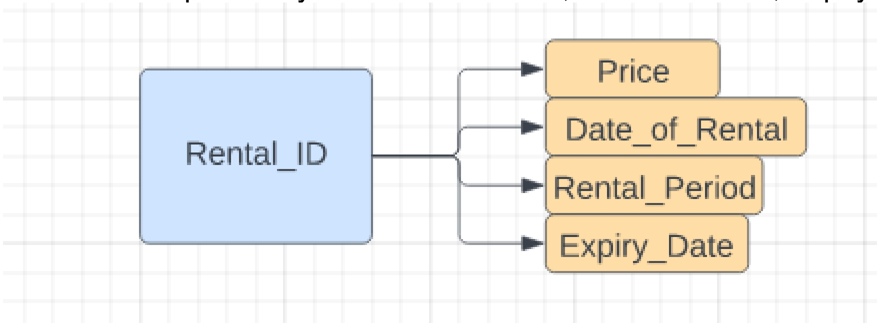
7. Movie

Movie(<u>Title,</u> Genre, Director, Release Year, Synopsis, Duration)
Functional Dependency
Title → Genre, Director, Release Year, Synopsis, Duration



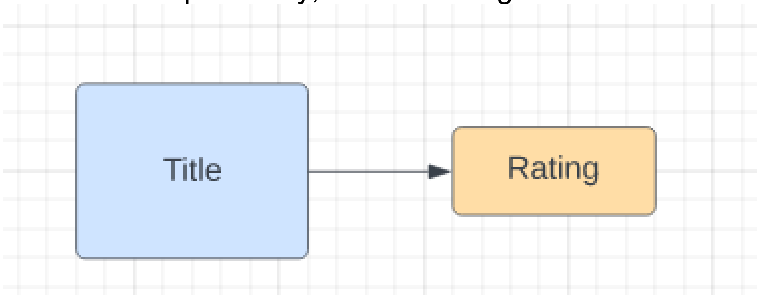
8. Rent

Rent(Rental_ID, Price, Date Of Rental, Expiry Date, Rental Period)
Functional Dependency: Rental ID → Price, Date Of Rental, Expiry Date, Rental Period



9. Movie_Rating

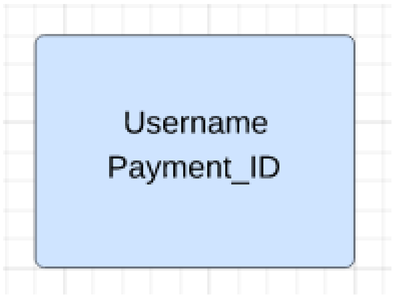
Movie_Rating(Title, Rating)
Functional Dependency; Title → Rating



Relationship Tables

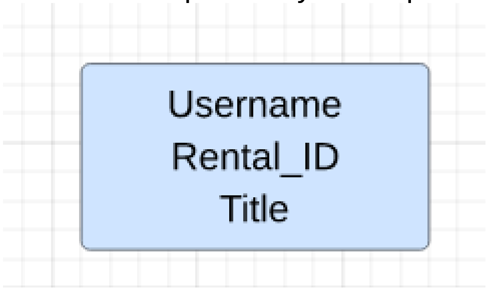
1. Makes

Makes(Username, Payment ID)
Functional Dependency: Username → Payment ID



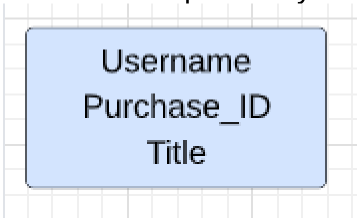
2. Rented

Rented(<u>Username</u>, <u>Rental_ID</u>, <u>Title</u>)
Functional Dependency: No dependency since many to many relation



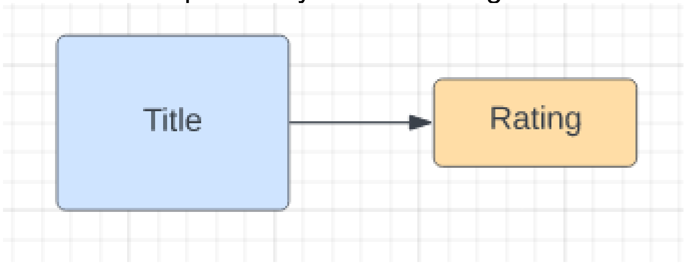
3. Purchased

Purchased(<u>Username</u>, <u>Purchase_ID</u>, <u>Title</u>)
Functional Dependency: No dependency since many to many relation



4. Rating

Rating(<u>Title</u>, rating)
Functional Dependency: Title → rating

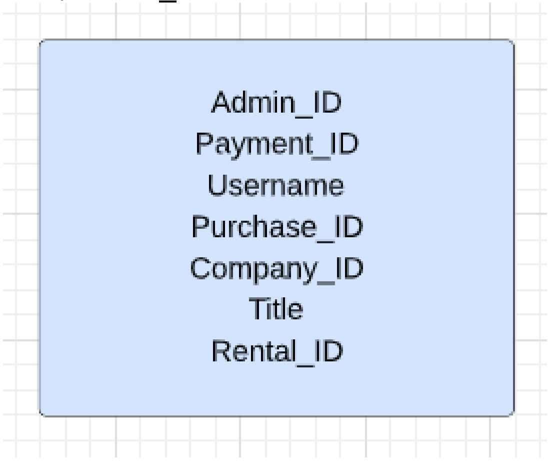


5. Sells_Rights_of

Sells_Rights_of(<u>Company ID</u>, <u>Title</u>)
Functional Dependency: No dependency since many to many relation

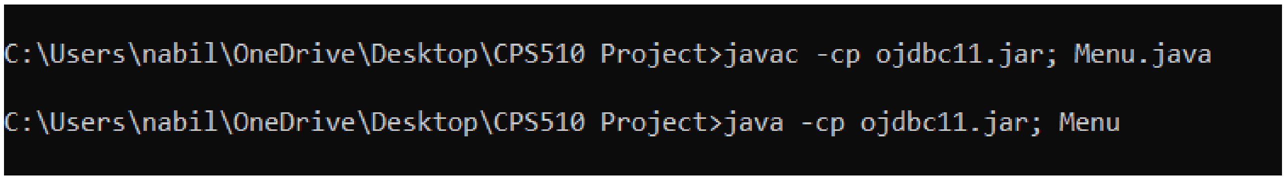
6. Can_Access

Can_Access(Admin_ID, Payment_ID, Username, Purchase_ID, Company_ID, Title, Rental_ID)
Functional Dependency; Admin_ID → Payment_ID, Username, Purchase_ID, Company_ID, Title, Rental_ID



Application using Graphical UIs and Java based GUI

In order to connect and run the java program, first we need to connect to our own local oracle database system. Then, we can use the code named "DatabaseConnection.java" below changing username and the password to our own. We would also need to download the ojdbc.jar file that is compatible with our version Oracle. We, then, go on our Unix-based operating system, compile the code with the command "javac -cp ojdc6.jar; Menu.java" and then run it with "java -cp ojdc6.jar; Menu"



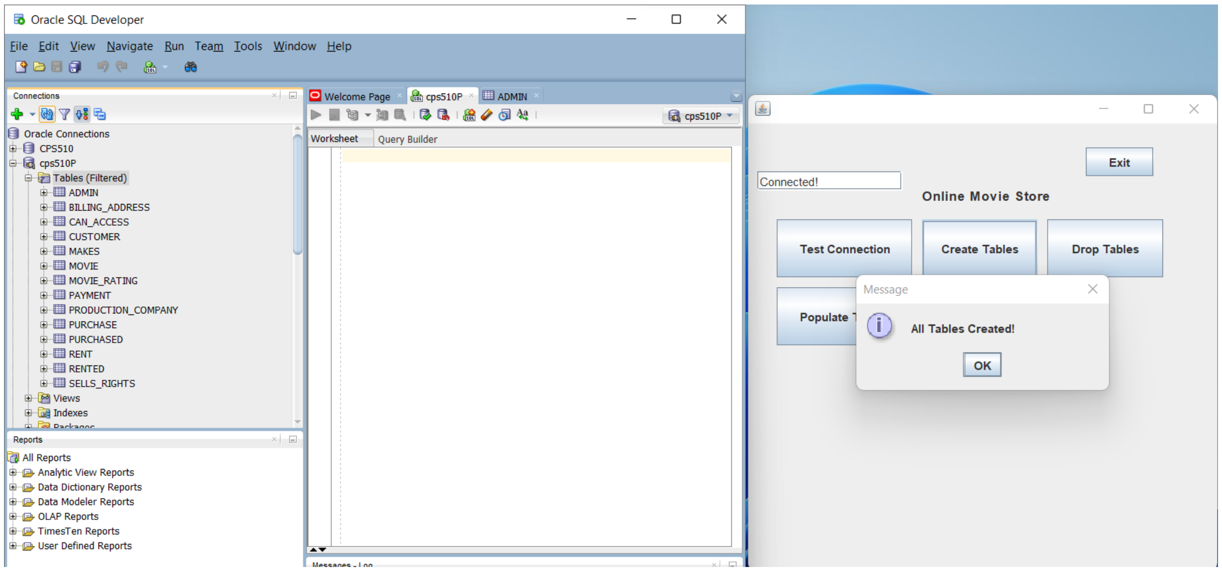
```
C:\Users\nabil\OneDrive\Desktop\CPS510 Project>javac -cp ojdbc11.jar; Menu.java

C:\Users\nabil\OneDrive\Desktop\CPS510 Project>java -cp ojdbc11.jar; Menu
```

DatabaseConnection.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class DatabaseConnection {
 public Connection connection = null;
 public boolean createConnection() {
 try {
   Class.forName("oracle.jdbc.driver.OracleDriver");
 } catch (ClassNotFoundException e) {
   return false;
 }
 try {
    connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521/XEPDB1",
"nabil", "Nabdude24");
    System.out.println("Connected!");
    return true;
 } catch (SQLException e) {
 e.printStackTrace();
   return false;
 }
 }
 public ResultSet executeQuery(String query) {
 try {
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery(query);
    return resultSet;
 }
 catch (SQLException e) {
    e.printStackTrace();
    return null;
 }
 }
}
```

CreateTables.java

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;


public class CreateTables{

    public static void createTable(JFrame f, DatabaseConnection databaseConnection, JButton
createBtn) {
        createBtn.addActionListener(actionEvent -> {
         databaseConnection.executeQuery("CREATE TABLE Customer(Username VARCHAR2(100)
PRIMARY KEY,Name VARCHAR2(100),Email VARCHAR2(100),Password VARCHAR2(100))");
         databaseConnection.executeQuery("CREATE TABLE Payment(Payment_ID int PRIMARY
KEY,Card_Holder_Name VARCHAR2(100),Card_Number VARCHAR2(100),CVV
VARCHAR2(100),Expiry_date VARCHAR2(100))");
         databaseConnection.executeQuery("CREATE TABLE Billing_Address(Street VARCHAR2(100),City
VARCHAR2(100),Postal_Code VARCHAR2(100),Province VARCHAR2(100),Payment_ID int PRIMARY
KEY,FOREIGN KEY(Payment_ID) REFERENCES Payment(Payment_ID))");
         databaseConnection.executeQuery("CREATE TABLE Production_Company(Company_ID int
PRIMARY KEY,Password VARCHAR2(100),Movies VARCHAR2(100))");
         databaseConnection.executeQuery("CREATE TABLE Movie(Title VARCHAR2(100) PRIMARY
KEY,Genre VARCHAR2(100),Director VARCHAR2(100),Release_Year int,Synopsis
VARCHAR2(1000),Duration int)");

         databaseConnection.executeQuery("CREATE TABLE Rent(Rental_ID int PRIMARY KEY,Price
int,Date_of_rental DATE,Expiry_Date DATE,Rental_Period VARCHAR2(100))");
         databaseConnection.executeQuery("CREATE TABLE Purchase(Purchase_ID int PRIMARY
KEY,Price int,Buy_Date DATE)");
         databaseConnection.executeQuery("CREATE TABLE Movie_Rating(Rating int,Title
VARCHAR2(100) PRIMARY KEY)");
       databaseConnection.executeQuery("CREATE TABLE Admin(Admin_ID int PRIMARY KEY,password
int)");
         databaseConnection.executeQuery("CREATE TABLE Can_Access(Admin_ID int,Payment_ID
int,Username VARCHAR2(100),Purchase_ID int,Rental_ID int,Title VARCHAR2(100),Company_ID
int,PRIMARY KEY (Admin_ID, Payment_ID, Username, Purchase_ID, Rental_ID, Title,
Company_ID),FOREIGN KEY (Admin_ID) REFERENCES Admin(Admin_ID),FOREIGN KEY (Payment_ID)
REFERENCES Payment(Payment_ID),FOREIGN KEY (Username) REFERENCES
Customer(Username),FOREIGN KEY (Purchase_ID) REFERENCES Purchase(Purchase_ID),FOREIGN KEY
(Rental_ID) REFERENCES Rent(Rental_ID),FOREIGN KEY (Title) REFERENCES Movie(Title),FOREIGN KEY
(Company_ID) REFERENCES Production_Company(Company_ID))");
        databaseConnection.executeQuery("CREATE TABLE Makes(Payment_ID int,Username
VARCHAR2(100),PRIMARY KEY (Payment_ID, Username),FOREIGN KEY (Payment_ID) REFERENCES
Payment(Payment_ID),FOREIGN KEY (Username) REFERENCES Customer(Username))");
        databaseConnection.executeQuery("CREATE TABLE Rented(Username VARCHAR(100),Rental_ID
int PRIMARY KEY,Title VARCHAR(100),FOREIGN KEY (Username) REFERENCES
Customer(Username),FOREIGN KEY (Rental_ID) REFERENCES Rent(Rental_ID),FOREIGN KEY (Title)
REFERENCES Movie(Title))");
        databaseConnection.executeQuery("CREATE TABLE Purchased(Username
VARCHAR2(100),Purchase_ID int,Title VARCHAR2(100),FOREIGN KEY (Username) REFERENCES
Customer(Username),FOREIGN KEY (Purchase_ID) REFERENCES Purchase(Purchase_ID),FOREIGN KEY
(Title) REFERENCES Movie(Title))");
        databaseConnection.executeQuery("CREATE TABLE Sells_Rights(Company_ID int PRIMARY
KEY,Title VARCHAR2(100),FOREIGN KEY (Title) REFERENCES Movie(Title),FOREIGN KEY (Company_ID)
REFERENCES Production_Company(Company_ID))");
        databaseConnection.executeQuery("commit");
        JOptionPane.showMessageDialog(f, "All Tables Created!");



    });

}
}
```



```
Oracle Connections
  CPS510
  cps510P
    Tables (Filtered)
      ADMIN
      BILLING_ADDRESS
      CAN_ACCESS
      CUSTOMER
      MAKES
      MOVIE
      MOVIE_RATING
      PAYMENT
      PRODUCTION_COMPANY
      PURCHASE
      PURCHASED
      RENT
      RENTED
      SELLS_RIGHTS
    Views
    Indexes
    Packages

All Reports
  Analytic View Reports
  Data Dictionary Reports
  Data Modeler Reports
  OLAP Reports
  TimesTen Reports
  User Defined Reports
```

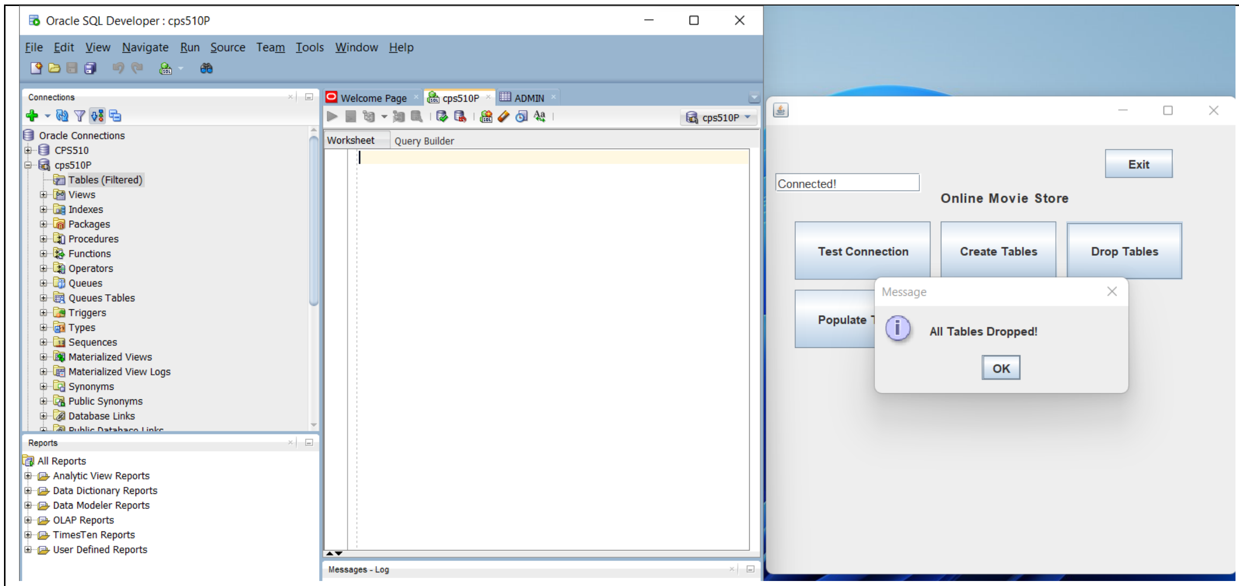
Connected!

Online Movie Store

All Tables Created!

DropTables.java

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;


public class DropTables{

    public static void dropTable(JFrame f, DatabaseConnection databaseConnection, JButton dropBtn) {
        dropBtn.addActionListener(actionEvent -> {
         databaseConnection.executeQuery("DROP TABLE Customer CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE PAYMENT CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE PURCHASE CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE RENT CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE MOVIE CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE PRODUCTION_COMPANY CASCADE
CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE MAKES CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE RENTED CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE PURCHASED CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE ADMIN CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE CAN_ACCESS CASCADE CONSTRAINTS
PURGE");
         databaseConnection.executeQuery("DROP TABLE BILLING_ADDRESS CASCADE CONSTRAINTS
PURGE");
         databaseConnection.executeQuery("DROP TABLE Sells_rights CASCADE CONSTRAINTS PURGE");
         databaseConnection.executeQuery("DROP TABLE Movie_Rating CASCADE CONSTRAINTS
PURGE");
         databaseConnection.executeQuery("commit");
         JOptionPane.showMessageDialog(f, "All Tables Dropped!");
        });
    }
}
```



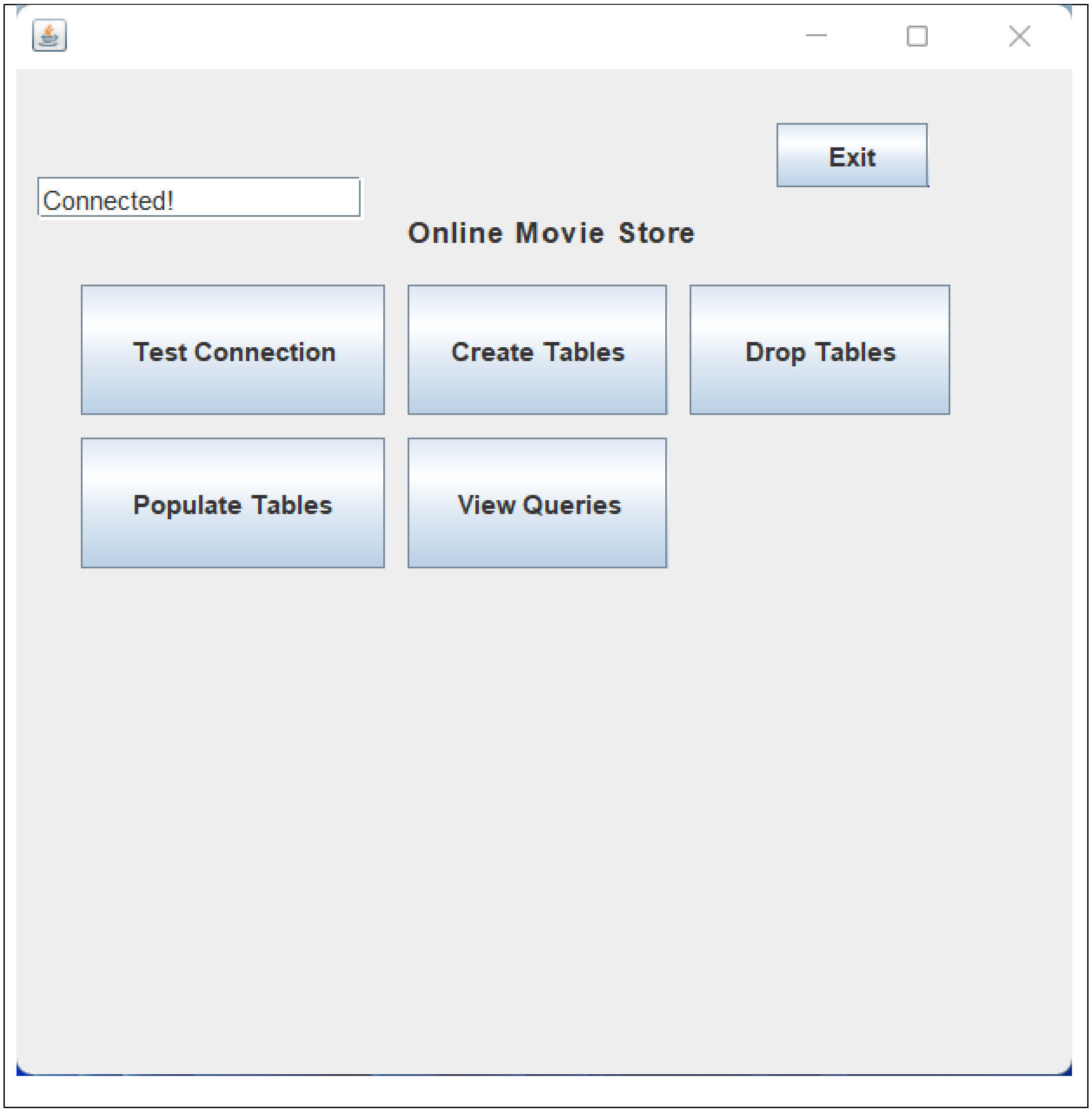
Connected!

All Tables Dropped!

Menu.java

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;

public class Menu {

    public static void main(String[] args) {

        Menu();

    }
   public static void Menu(){
      DatabaseConnection databaseConnection = new DatabaseConnection();
      JFrame f=new JFrame();//creating instance of JFrame

      JButton test=new JButton("Test Connection");
      test.setBounds(30,100,140,60);
      f.add(test);
      JButton createBtn=new JButton("Create Tables");//creating instance of JButton
      createBtn.setBounds(180,100,120, 60);//x axis, y axis, width, height
```

```
        f.add(createBtn);//adding button in JFrame

        JButton dropBtn = new JButton("Drop Tables");
        dropBtn.setBounds(310,100,120, 60);
        f.add(dropBtn);

        JLabel title = new JLabel("Online Movie Store");
        title.setFont(new Font("Arial", Font.BOLD, 14));
        title.setBounds(180, 50, 150, 50);
        f.add(title);

        JButton popBtn = new JButton("Populate Tables");
        popBtn.setBounds(30,170,140, 60);
        f.add(popBtn);

        JButton querybtn = new JButton("View Queries");
        querybtn.setBounds(180, 170, 120, 60);
        f.add(querybtn);
        final JTextField tf=new JTextField();
        tf.setBounds(10,50, 150,20);
        f.add(tf);

        JButton exit = new JButton("Exit");
        exit.setBounds(350,25,70,30);
        f.add(exit);


          test.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(databaseConnection.createConnection() == true){
                tf.setText("Connected!");
            }else{
                tf.setText("Not Connected");
            };
    }
 });
        CreateTables.createTable(f, databaseConnection, createBtn);
        DropTables.dropTable(f, databaseConnection, dropBtn);
        PopulateTables.popTable(f, databaseConnection, popBtn);



        querybtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

            ViewQueries.QueriesWindow(databaseConnection);

            }
        });

        exit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
           f.setVisible(false);
           f.dispose();



        }
  });

        f.setSize(500,500);//400 width and 500 height
        f.setLayout(null);//using no layout managers
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);



     }
 }
```



PopulateTables.java

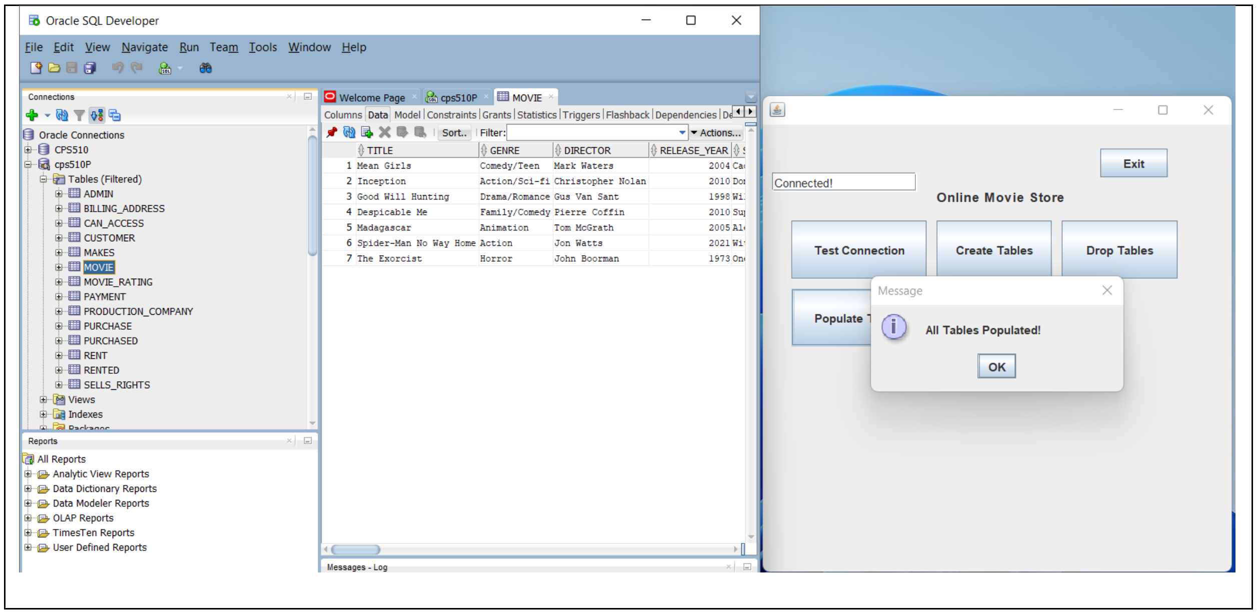
```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;


public class PopulateTables{

   public static void popTable(JFrame f, DatabaseConnection databaseConnection, JButton popBtn) {
      popBtn.addActionListener(actionEvent -> {

         databaseConnection.executeQuery("INSERT INTO Customer VALUES ('munevver',
'MunevverCoskun', 'munevver.coskun@ryerson.ca', 'mnvvrcskn')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('zamla', 'Zaiba', 'zaiba.amla@ryerson.ca', 'ilikePizza06')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('nabil', 'Nabil', 'nabil.@ryerson.ca', 'password')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('JohnH', 'John', 'john.Hanover@hotmail.com', 'BigPlate889')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('AshleyCake', 'Ashley', 'ashleyMakesCake@gmail.com',
'cupcakesWithSprinkles44')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('KinderFan', 'Neil', 'neilpatrick@yahoo.ca', 'FitzgararldTHE3')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('LillyPad', 'Lilly', 'lillyericson@gmail.com', 'OliveTheory73')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('Marshmellow', 'Marshell', 'Marshmellow_ericson@hotmail.ca',
'coolBeans62898')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('CampbellSoup', 'Nick', 'NickLoves_soup@gmail.com',
'ChiliIsthebettersoup_73')");
         databaseConnection.executeQuery("INSERT INTO Customer (Username, Name, Email,
Password) VALUES ('CraigsCookies', 'Craig', 'craigscookies@gmail.com', 'ChocolatechipCookie0098')");



         databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (1, 'MunevverCoskun', '785 660 932 272 5234',
'705',TO_DATE('08/12/24','YYYY-MM-DD'))");
         databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (2, 'Zaiba Amla', '670 839 607 635 2810', '445',TO_DATE
('02/05/23', 'YYYY-MM-DD'))");
         databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (3, 'Nabil Haque', '342 365 949 886 1437', '882',TO_DATE
('30/07/25', 'YYYY-MM-DD'))");
         databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (4, 'John Hanover', '516 537 272 969 2203', '313',
TO_DATE('2023-06-30', 'YYYY-MM-DD'))");
         databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (5, 'Ashley Oconnell', '404 417 309 051 4342', '668',
TO_DATE('2024-03-07', 'YYYY-MM-DD'))");
```

```
        databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (6, 'Neil Patrick', '773 409 364 529 4399',
'213',TO_DATE('2024-10-24', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (7, 'Lilly Ericson', '304 226 394 047 4286', '553',
TO_DATE('2025-01-13', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (8, 'Marshel Ericson', '874 052 748 879 0026', '831',
TO_DATE('2025-05-06', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (9, 'Nick Young', '258 907 173 426 1441', '055',
TO_DATE('2024-03-11', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Payment (Payment_ID,Card_Holder_Name,
Card_Number, CVV, Expiry_Date) VALUES (10, 'Craig Austin', '138 732 375 372 0891', '993',
TO_DATE('2024-06-11', 'YYYY-MM-DD'))");

        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (00001, 22.5, TO_DATE('2022-10-03', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (00002, 22.5, TO_DATE('2022-10-01', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (00003, 15, TO_DATE('2022-09-27', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (00004, 15, TO_DATE('2022-09-23', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (5, 20, TO_DATE('2022-10-04', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (6, 25, TO_DATE('2022-10-08', 'YYYY-MM-DD'))");
        databaseConnection.executeQuery("INSERT INTO Purchase(Purchase_ID, Price, Buy_Date)
VALUES (7, 17, TO_DATE('2022-10-12', 'YYYY-MM-DD'))");


         databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental,
Expiry_Date, Rental_Period) VALUES (0001, 15, TO_DATE('2022-03-10', 'YYYY-MM-DD'),
TO_DATE('2022-10-10','YYYY-MM-DD'), '1w')");
         databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental,
Expiry_Date, Rental_Period) VALUES (0002, 15, TO_DATE('2022-01-10', 'YYYY-MM-DD'),
TO_DATE('2022-08-10','YYYY-MM-DD'), '1w')");
         databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental,
Expiry_Date, Rental_Period) VALUES (0003, 10, TO_DATE('2022-09-29', 'YYYY-MM-DD'),
TO_DATE('2022-04-10','YYYY-MM-DD'), '1w')");
         databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental,
Expiry_Date, Rental_Period) VALUES (0004, 10, TO_DATE('2022-09-23', 'YYYY-MM-DD'),
TO_DATE('2022-09-30','YYYY-MM-DD'), '1w')");
         databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental,
Expiry_Date, Rental_Period) VALUES (5, 7, TO_DATE('2022-10-11', 'YYYY-MM-DD'), TO_DATE('2022-
10-18', 'YYYY-MM-DD'), '1w')");
```

```
            databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental, 
Expiry_Date, Rental_Period) VALUES (6, 10, TO_DATE('2022-10-07', 'YYYY-MM-DD'), TO_DATE('2022-
10-14', 'YYYY-MM-DD'), '1w')");
            databaseConnection.executeQuery("INSERT INTO Rent (Rental_ID, Price, Date_of_rental, 
Expiry_Date, Rental_Period) VALUES (7, 5, TO_DATE('2022-10-03', 'YYYY-MM-DD'), TO_DATE('2022-
10-09', 'YYYY-MM-DD'), '1w')");



            databaseConnection.executeQuery("INSERT INTO Billing_Address(Street, City, Postal_code, 
Province, Payment_ID) VALUES ('90 Bellows Ave', 'Mississuaga', 'L8R 9Y2', 'ON', 0001)");
            databaseConnection.executeQuery("INSERT INTO Billing_Address(Street, City, Postal_code, 
Province, Payment_ID) VALUES ('30 Dundas St', 'Toronto', 'M3K 1R5', 'ON', 0002)");
            databaseConnection.executeQuery("INSERT INTO Billing_Address(Street, City, Postal_code, 
Province, Payment_ID) VALUES ('44 Elmwood Ave', 'North York', 'M3V 4N9', 'ON', 0003)");



        databaseConnection.executeQuery("INSERT INTO Production_Company (Company_ID, 
Password, Movies) VALUES (001, 'BestMovieMakers123', 'Mean girls')");
        databaseConnection.executeQuery("INSERT INTO Production_Company (Company_ID, 
Password, Movies) VALUES (002, 'BeesAreCool', 'Inception')");
        databaseConnection.executeQuery("INSERT INTO Production_Company (Company_ID, 
Password, Movies) VALUES (003, 'Pinapple972', 'Good Will Hunting')");
        databaseConnection.executeQuery("INSERT INTO Production_Company (Company_ID, 
Password, Movies) VALUES (004, 'Kevin26', 'Dispicable Me')");
        databaseConnection.executeQuery("INSERT INTO Production_Company(Company_ID, 
Password, Movies) VALUES (5, 'YungShmoney$', 'Madagascar')");
        databaseConnection.executeQuery("INSERT INTO Production_Company(Company_ID, 
Password, Movies) VALUES (6, 'Password123', 'Spider-Man No Way Home')");
        databaseConnection.executeQuery("INSERT INTO Production_Company(Company_ID, 
Password, Movies) VALUES (7, 'Lebron212', 'The Exorcist')");

            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director, 
Release_Year, Synopsis, Duration) VALUES ('Mean Girls', 'Comedy/Teen', 'Mark Waters', 2004, 'Cady 
Heron unwittingly finds herself in the good graces of an elite group of cool students dubbed the 
Plastics, but she soon realizes how her shallow group of new friends earned this nickname', 157)");
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director, 
Release_Year, Synopsis, Duration) VALUES ('Inception', 'Action/Sci-fi', 'Christopher Nolan', 2010, 'Dom 
Cobb is a thief with the rare ability to enter peoples dreams and steal tehir secrets from their 
subconscious', 148)");
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director, 
Release_Year, Synopsis, Duration) VALUES ('Good Will Hunting', 'Drama/Romance', 'Gus Van Sant', 
1998, 'Will Hunting has a genius-level IQ but chooses to work as a janitor at MIT', 126)");
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director, 
Release_Year, Synopsis, Duration) VALUES ('Despicable Me', 'Family/Comedy', 'Pierre Coffin', 2010, 
'Supervillain Gru, a man who delights in all things wicked, hatches a plan to steal the moon', 95)");
```

```
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director,
Release_Year, Synopsis, Duration) VALUES ('Madagascar', 'Animation', 'Tom McGrath', 2005, 'Alex the
lion is the king of the urban jungle, the main attraction at New Yorks Central Park Zoo. He and his best
friends -- Marty the zebra, Melman the giraffe and Gloria the hippo -- have spent their whole lives in
blissful captivity before an admiring public and with regular meals provided for them. Not content to
leave well enough alone, Marty lets his curiosity get the better of him and makes his escape -- with
the help of some prodigious penguins -- to explore the world.', 86)");
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director,
Release_Year, Synopsis, Duration) VALUES ('Spider-Man No Way Home', 'Action', 'Jon Watts', 2021,
'With Spider-Man identity now revealed, our friendly neighborhood web-slinger is unmasked and no
longer able to separate his normal life as Peter Parker from the high stakes of being a superhero.
When Peter asks for help from Doctor Strange, the stakes become even more dangerous, forcing him
to discover what it truly means to be Spider-Man.', 148)");
            databaseConnection.executeQuery("INSERT INTO Movie (Title, Genre, Director,
Release_Year, Synopsis, Duration) VALUES ('The Exorcist', 'Horror', 'John Boorman', 1973, 'One of the
most profitable horror movies ever made, this tale of an exorcism is based loosely on actual events.
When young Regan (Linda Blair) starts acting odd -- levitating, speaking in tongues -- her worried
mother (Ellen Burstyn) seeks medical help, only to hit a dead end. A local priest (Jason Miller),
however, thinks the girl may be seized by the devil. The priest makes a request to perform an
exorcism, and the church sends in an expert (Max von Sydow) to help with the difficult job.', 132)");

            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (4,
'Mean Girls')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (5,
'Inception')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (5,
'Good Will Hunting')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (4,
'Despicable Me')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (4,
'Madagascar')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (5,
'Spider-Man No Way Home')");
            databaseConnection.executeQuery("INSERT INTO Movie_Rating (Rating, Title) VALUES (3.5,
'The Exorcist')");

            databaseConnection.executeQuery("INSERT INTO Admin VALUES (01, 5678)");

              databaseConnection.executeQuery("INSERT INTO Can_Access (Admin_ID, Payment_ID,
Username, Purchase_ID, Rental_ID, Title, Company_ID) VALUES (01, 0001, 'munevver', 00001, 0001,
'Mean Girls', 001)");
              databaseConnection.executeQuery("INSERT INTO Can_Access (Admin_ID, Payment_ID,
Username, Purchase_ID, Rental_ID, Title, Company_ID) VALUES (01, 0002, 'zamla', 00002, 0002,
'Inception', 002)");
              databaseConnection.executeQuery("INSERT INTO Can_Access (Admin_ID, Payment_ID,
Username, Purchase_ID, Rental_ID, Title, Company_ID) VALUES (01, 0003, 'nabil', 00003, 0003, 'Good
Will Hunting', 003)");
```

```
                databaseConnection.executeQuery("INSERT INTO Makes (Payment_ID, Username) VALUES
(0001, 'munevver')");
                databaseConnection.executeQuery("INSERT INTO Makes (Payment_ID, Username) VALUES
(0002, 'zamla')");
                databaseConnection.executeQuery("INSERT INTO Makes (Payment_ID, Username) VALUES
(0003, 'nabil')");

            databaseConnection.executeQuery("INSERT INTO Rented (Username, Rental_ID, Title)
VALUES ('zamla', 0001, 'Mean Girls')");
            databaseConnection.executeQuery("INSERT INTO Rented (Username, Rental_ID, Title)
VALUES ('munevver', 0004, 'Despicable Me')");
            databaseConnection.executeQuery("INSERT INTO Rented (Username, Rental_ID, Title)
VALUES ('nabil', 0002, 'Inception')");

            databaseConnection.executeQuery("INSERT INTO Purchased (Username, Purchase_ID, Title)
VALUES ('munevver', 00001, 'Mean Girls')");
            databaseConnection.executeQuery("INSERT INTO Purchased (Username, Purchase_ID, Title)
VALUES ('zamla', 00004, 'Despicable Me')");
            databaseConnection.executeQuery("INSERT INTO Purchased (Username, Purchase_ID, Title)
VALUES ('nabil', 00002, 'Inception')");

            databaseConnection.executeQuery("INSERT INTO Sells_Rights (Company_ID, Title) VALUES
(001, 'Mean Girls')");
            databaseConnection.executeQuery("INSERT INTO Sells_Rights (Company_ID, Title) VALUES
(002, 'Inception')");
            databaseConnection.executeQuery("INSERT INTO Sells_Rights (Company_ID, Title) VALUES
(003, 'Good Will Hunting')");
            databaseConnection.executeQuery("INSERT INTO Sells_Rights (Company_ID, Title) VALUES
(004, 'Despicable Me')");



          databaseConnection.executeQuery("commit");
          JOptionPane.showMessageDialog(f, "All Tables Populated!");

        });
    }
}
```



	TITLE	GENRE	DIRECTOR	RELEASE_YEAR
1	Mean Girls	Comedy/Teen	Mark Waters	2004
2	Inception	Action/Sci-fi	Christopher Nolan	2010
3	Good Will Hunting	Drama/Romance	Gus Van Sant	1998
4	Despicable Me	Family/Comedy	Pierre Coffin	2010
5	Madagascar	Animation	Tom McGrath	2005
6	Spider-Man No Way Home	Action	Jon Watts	2021
7	The Exorcist	Horror	John Boorman	1973

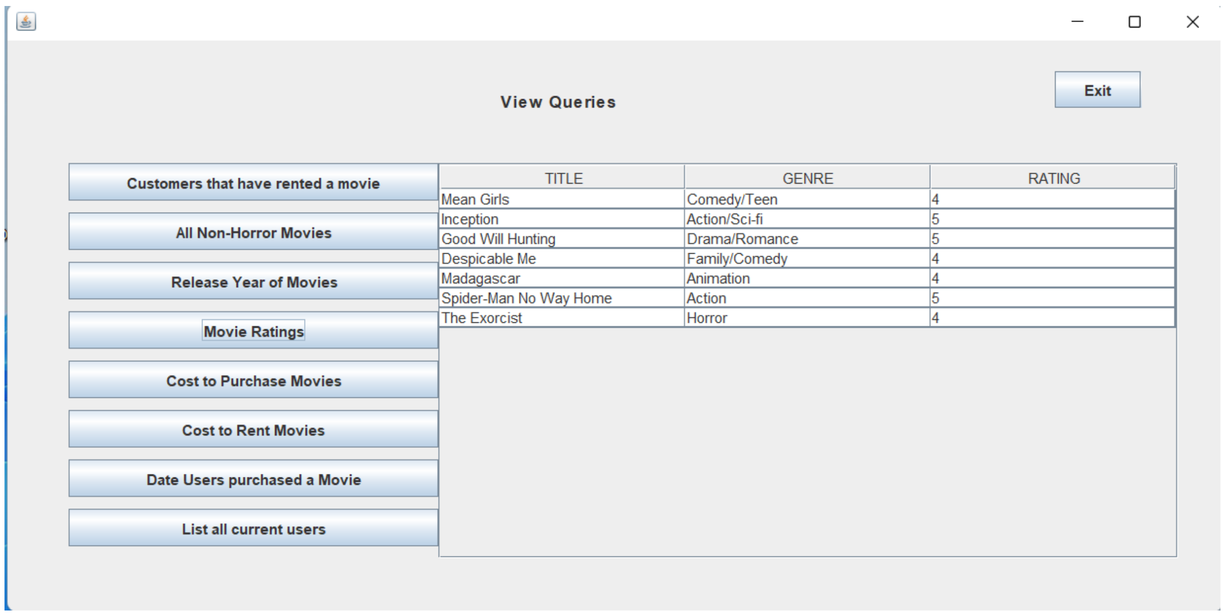
ViewQueries.java

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;

public class ViewQueries{




public static void QueriesWindow(DatabaseConnection databaseConnection){

    JFrame queryWindow=new JFrame();
    JLabel title = new JLabel("View Queries");
    title.setFont(new Font("Arial", Font.BOLD, 14));
    title.setBounds(400, 25, 100, 50);
    queryWindow.add(title);
    JButton query1 = new JButton("Customers that have rented a movie");
    query1.setBounds(50, 100, 300, 30);
    queryWindow.add(query1);
    JButton query2 = new JButton("All Non-Horror Movies");
    query2.setBounds(50,140,300,30);
```

```
 queryWindow.add(query2);

 JButton query3 = new JButton("Release Year of Movies");
 query3.setBounds(50,180,300,30);
 queryWindow.add(query3);

 JButton query4 = new JButton("Movie Ratings");
 query4.setBounds(50,220,300,30);
 queryWindow.add(query4);

 JButton query5 = new JButton("Cost to Purchase Movies");
 query5.setBounds(50,260,300,30);
 queryWindow.add(query5);

 JButton query6 = new JButton("Cost to Rent Movies");
 query6.setBounds(50,300,300,30);
 queryWindow.add(query6);

 JButton query7 = new JButton("Date Users purchased a Movie");
 query7.setBounds(50,340,300,30);
 queryWindow.add(query7);

 JButton query8 = new JButton("List all current users");
 query8.setBounds(50,380,300,30);
 queryWindow.add(query8);

 JButton exit = new JButton("Exit");
 exit.setBounds(850,25,70,30);
 queryWindow.add(exit);

 JTable queryResult = new JTable();


 JScrollPane sp = new JScrollPane(queryResult);

 sp.setBounds(350,100, 600, 320);
 //model.addColumn("customerName");

queryWindow.add(sp);

DefaultTableModel model =(DefaultTableModel)queryResult.getModel();
queryWindow.setSize(1000,500);//400 width and 500 height
queryWindow.setLayout(null);//using no layout managers
queryWindow.setVisible(true);



 query1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
           try{
              model.setRowCount(0);
              ResultSet res = databaseConnection.executeQuery("SELECT Customer.Name FROM
Customer WHERE EXISTS(SELECT rent.rental_id FROM Rent, Rented WHERE rented.rental_id =
rent.rental_id AND rented.username = customer.username)");
              ResultSetMetaData metdata = res.getMetaData();
              int colCount = metdata.getColumnCount();
              String[] colName = new String[colCount];

              for(int i=0; i<colCount; i++){
                 colName[i] = metdata.getColumnName(i+1);
              }
              model.setColumnIdentifiers(colName);
              String name;
              while(res.next()){
                 name=res.getString(1);
                 String[] row = {name};
                 model.addRow(row);
              }
           }
           catch (SQLException a) {
              a.printStackTrace();

            }

        }
 });

    query2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
           try{
              model.setRowCount(0);
              ResultSet res = databaseConnection.executeQuery("SELECT * FROM Movie MINUS(Select *
FROM Movie WHERE genre = 'Horror')");
              ResultSetMetaData metdata = res.getMetaData();
              int colCount = metdata.getColumnCount();
              String[] colName = new String[colCount];
              //DefaultTableModel model =(DefaultTableModel)queryResult.getModel();
              for(int i=0; i<colCount; i++){
                 colName[i] = metdata.getColumnName(i+1);
              }
              model.setColumnIdentifiers(colName);
              String title, genre, director, releaseYear, synopsis, duration;
              while(res.next()){
                 title=res.getString(1);
                 genre=res.getString(2);
                 director=res.getString(3);
```

```
                releaseYear=res.getString(4);
                synopsis=res.getString(5);
                duration=res.getString(6);
                String[] row = {title,genre,director,releaseYear,synopsis,duration};
                model.addRow(row);
            }

          }
          catch (SQLException a) {
             a.printStackTrace();

           }

       }
 });

   query3.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          try{
             model.setRowCount(0);
             ResultSet res = databaseConnection.executeQuery("SELECT production_company.movies,
movie.release_year FROM movie, sells_rights, production_company WHERE
production_company.company_id = sells_rights.company_id AND movie.title = sells_rights.title");
             ResultSetMetaData metdata = res.getMetaData();
             int colCount = metdata.getColumnCount();
             String[] colName = new String[colCount];

             for(int i=0; i<colCount; i++){
                colName[i] = metdata.getColumnName(i+1);
             }
             model.setColumnIdentifiers(colName);
             String movie, year;
             while(res.next()){
                movie=res.getString(1);
                year=res.getString(2);
                String[] row = {movie,year};
                model.addRow(row);
             }
          }
          catch (SQLException a) {
             a.printStackTrace();

           }

       }
 });

   query4.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          try{
             model.setRowCount(0);
             ResultSet res = databaseConnection.executeQuery("SELECT movie.title, movie.genre,
movie_rating.rating FROM movie, movie_rating WHERE movie.title = movie_rating.title");
             ResultSetMetaData metdata = res.getMetaData();
             int colCount = metdata.getColumnCount();
             String[] colName = new String[colCount];

             for(int i=0; i<colCount; i++){
                colName[i] = metdata.getColumnName(i+1);
             }
             model.setColumnIdentifiers(colName);
             String movie,genre, rating;
             while(res.next()){
                movie=res.getString(1);
                genre=res.getString(2);
                rating=res.getString(3);
                String[] row = {movie,genre,rating};
                model.addRow(row);
             }
          }
          catch (SQLException a) {
             a.printStackTrace();

           }

       }
 });

   query5.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
          try{
             model.setRowCount(0);
             ResultSet res = databaseConnection.executeQuery("SELECT movie.title, purchase.price
FROM movie, purchase, purchased WHERE purchase.purchase_id = purchased.purchase_id AND
movie.title = purchased.title");
             ResultSetMetaData metdata = res.getMetaData();
             int colCount = metdata.getColumnCount();
             String[] colName = new String[colCount];

             for(int i=0; i<colCount; i++){
                colName[i] = metdata.getColumnName(i+1);
             }
             model.setColumnIdentifiers(colName);
             String movie,price;
             while(res.next()){
                movie=res.getString(1);
            price=res.getString(2);

            String[] row = {movie,price};
            model.addRow(row);
          }
        }
        catch (SQLException a) {
          a.printStackTrace();

         }

      }
 });

   query6.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        try{
          model.setRowCount(0);
          ResultSet res = databaseConnection.executeQuery("SELECT movie.title, rent.price FROM
movie, rent, rented WHERE rent.rental_id = rented.rental_id AND movie.title = rented.title");
          ResultSetMetaData metdata = res.getMetaData();
          int colCount = metdata.getColumnCount();
          String[] colName = new String[colCount];

          for(int i=0; i<colCount; i++){
            colName[i] = metdata.getColumnName(i+1);
          }
          model.setColumnIdentifiers(colName);
          String movie,price;
          while(res.next()){
            movie=res.getString(1);
            price=res.getString(2);

            String[] row = {movie,price};
            model.addRow(row);
          }
        }
        catch (SQLException a) {
          a.printStackTrace();

         }

      }
 });

   query7.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        try{
          model.setRowCount(0);
          ResultSet res = databaseConnection.executeQuery("SELECT Customer.username, movie.title,
purchase.Buy_date FROM Customer, Movie, Purchase, Purchased WHERE Customer.username =
purchased.username AND purchase.purchase_id = purchased.purchase_id AND movie.title =
purchased.title");
          ResultSetMetaData metdata = res.getMetaData();
          int colCount = metdata.getColumnCount();
          String[] colName = new String[colCount];

          for(int i=0; i<colCount; i++){
            colName[i] = metdata.getColumnName(i+1);
          }
          model.setColumnIdentifiers(colName);
          String name,movie,price;
          while(res.next()){
            name=res.getString(1);
            movie=res.getString(2);
            price=res.getString(3);

            String[] row = {name,movie,price};
            model.addRow(row);
          }
        }
        catch (SQLException a) {
          a.printStackTrace();

         }

      }
 });

   query8.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        try{
          model.setRowCount(0);
          ResultSet res = databaseConnection.executeQuery("SELECT Username FROM Customer");
          ResultSetMetaData metdata = res.getMetaData();
          int colCount = metdata.getColumnCount();
          String[] colName = new String[colCount];

          for(int i=0; i<colCount; i++){
            colName[i] = metdata.getColumnName(i+1);
          }
          model.setColumnIdentifiers(colName);
          String name;
          while(res.next()){
            name=res.getString(1);
            String[] row = {name};
```

```
                model.addRow(row);
            }
        }
        catch (SQLException a) {
            a.printStackTrace();

         }

     }
 });

   exit.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
         queryWindow.setVisible(false);
         queryWindow.dispose();


     }
 });

   queryWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
 }
```



TITLE	GENRE	RATING
Mean Girls	Comedy/Teen	4
Inception	Action/Sci-fi	5
Good Will Hunting	Drama/Romance	5
Despicable Me	Family/Comedy	4
Madagascar	Animation	4
Spider-Man No Way Home	Action	5
The Exorcist	Horror	4

Relational Algebra (RA) Notation

1. List the username(in the order of first joining the system) and cardholder names for all users from the tables customer and payment respectfully.

SELECT DISTINCT Username, Card_Holder_Name
FROM Customer, Payment
ORDER BY Username DESC

τ Username desc **π** Username, Card_Holder_Name (Customer □□ payment)

2. This lists all the customers and their card numbers who have ‘Toronto’ in their billing address.

SELECT Payment.Card_Holder_Name, payment.Card_Number
FROM Payment
INNER JOIN Billing_Address
ON Billing_Address.City = 'Toronto' AND Payment.Payment_ID =
Billing_Address.Payment_ID;

π Card_Holder_Name, Card_Number **(**Payment $\bowtie_{\text{Billing\_Address.City = 'Toronto'}}$ Billing_Address)

3. List the company ID who's sells the rights of ‘Mean Girls’ to the service.

SELECT Company_ID
FROM Production_Company
WHERE Production_Company.Movies = 'Mean girls';

π Company_ID (***σ*** Production_Company. Movies = ‘Mean girls’ (Production_Company))

4. Join the tables rent and rented and display matching values in both tables.

SELECT Rent.Price
FROM Rent
INNER JOIN Rented
ON Rented.Rental_ID = Rent.Rental_ID;

π Rent.Price **(**Rent $\bowtie_{\text{Rented.Rental\_ID = Rent.Rental\_ID}}$ Rented)

5. List the purchase price of the movie ‘Inception’.

SELECT Price
FROM Purchase, Purchased
WHERE Purchased.Title = 'Inception' AND Purchased.Purchase_ID =
Purchase.Purchase_ID

π Price (***σ*** Title = ‘Inception’ (Purchased □□ Purchase))

6. List the rental price of the movie 'Mean Girls '

```
SELECT Price
FROM Rent, Rented
WHERE Rented.Title = 'Mean Girls' AND Rented.Rental_ID = Rent.Rental_ID
```

π Price (***σ*** Title = ‘Mean Girls’ (Rented □□ Rent))

7. List the date of rental and the expiry date where the rental ID is 1

```
SELECT RENT.DATE_OF_RENTAL, rent.expiry_date
FROM rent
WHERE rental_id =1;
```

π Date_of_rental, expiry_date (***σ*** rental_id =1(rent))

8. List the title and rating of a movie where the movie rating is equal to 5

```
SELECT movie_rating.title, movie_rating.rating
FROM movie_rating
WHERE movie_Rating.rating = 5;
```

π title, rating (***σ*** movie_Rating.rating = 5 (movie_rating))

9. List all the usernames and the associated movie title

```
SELECT Can_Access.Username, Can_Access.Title
FROM Can_Access, Admin
WHERE Admin.Admin_ID = Can_Access.admin_ID;
```

π Username, title (***σ*** Admin.Admin_ID = Can_Access.admin_ID (Can_Access □□ Admin))

10. list all the unique ratings and the number of movies that have such ratings

```
SELECT Rating, COUNT(*)
FROM Movie_Rating
GROUP BY Rating;
```

π Rating F_{COUNT} Rating (Movie_Rating)