CPS510 Project

Online Movie Store

Arshpreet Singh (501030338)
James Tan (501042067)
Sania Syed (501017695)

Table of Contents

Assignment 1 - Project Description: Online Movie Store

General Description

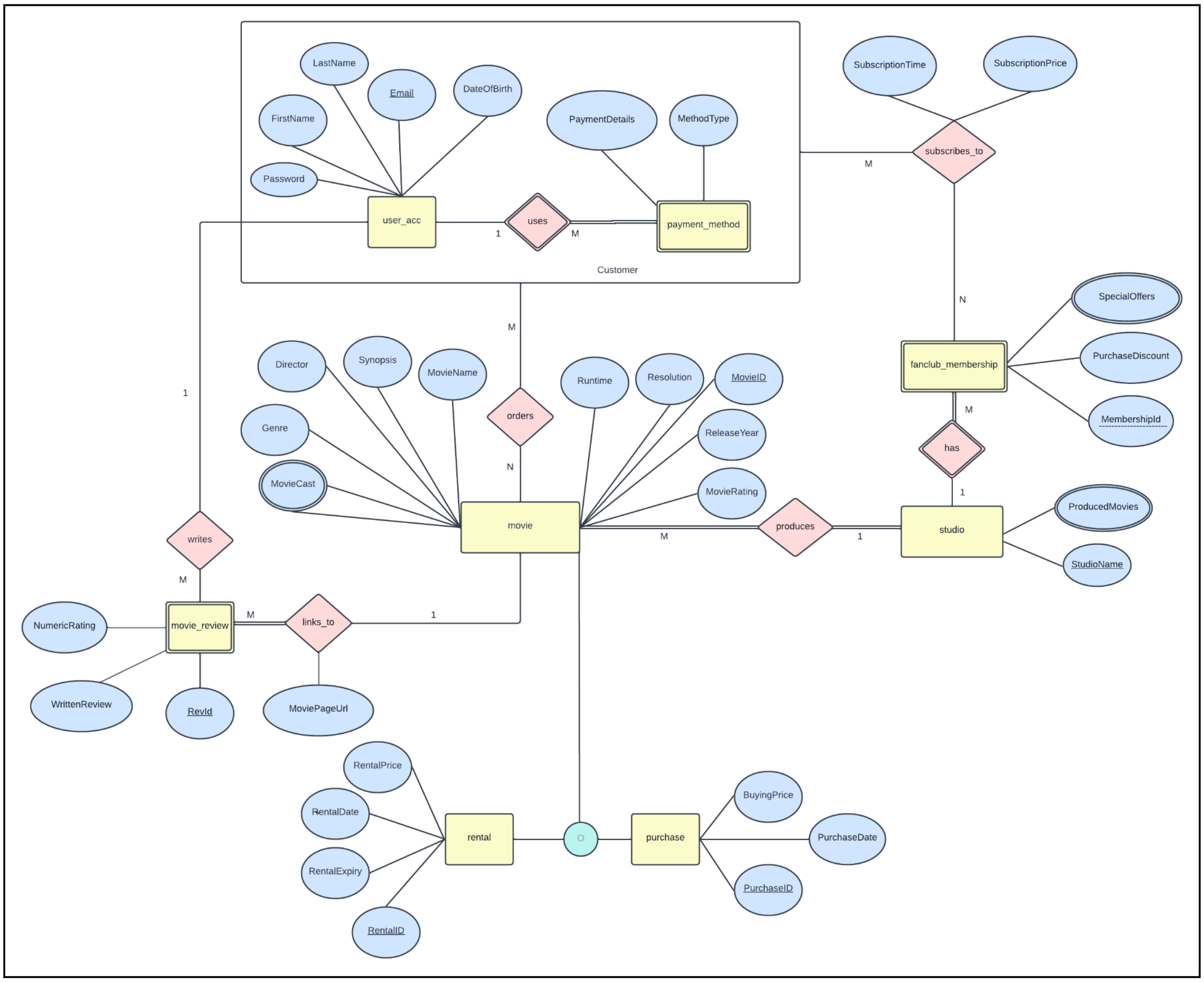
The purpose of our system is to manage information which enables users to purchase and/or rent movies. It will store information relevant to the movies, which will allow for queries to be made in the Online Movie Store which in turn would allow users to select and add movies of their interest to the shopping cart. The system will store relevant information regarding the user to enable the process of payment. Our system will also manage the information of movie studios to keep track of the rights we have to be hosting their movies.

How the System will Work

A user logs in to the Online Movie Store using their credentials like username and password. The studio of a movie gives the admins of the site rights to their movie/s to be put up on the website for rental/purchase. The user sees a list of moves and is able to query the database for certain movies based on criterias such as name, price and genre. The user can choose the option of renting or buying the movie. Upon selection of a movie on the website, it is added to the user's shopping cart. The user can checkout their current shopping cart and their account is charged the attributed costs. If the movie was rented, they can keep the movie for 2 weeks, and the amount of time that they have left to keep the movie will be shown on their account.

user_acc(Email, FirstName, LastName, DateOfBirth, Password)
payment_method(Email, PaymentDetails, MethodType)
movie(MovieID, MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast)
orders(Email, MovieID, PaymentDetails)
studio(StudioName, ProducedMovies)
produces(MovieID, StudioName)
rental(MovieID, RentalPrice, RentalDate, RentalExpiry)
purchase(MovieID, BuyingPrice, PurchaseDate)
movie_review(RevID, WrittenRev, NumericRating, Email, FirstName, LastName)
fanclub_membership(MembershipID, RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate)
has(MembershipID, StudioName)

Assignment 2 - ER Diagram



Assignment 3 - Schema Design

Create Tables - SQL Code

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';

CREATE TABLE user_acc (
        Email VARCHAR2(254) PRIMARY KEY,
        FirstName VARCHAR2(100) NOT NULL,
        LastName VARCHAR2(100) NOT NULL,
        DateOfBirth DATE NOT NULL,
        Password VARCHAR2(100) NOT NULL);

CREATE TABLE payment_method (
        PaymentDetails NUMBER PRIMARY KEY,
        MethodType VARCHAR2(16),
        Email VARCHAR2(254) REFERENCES user_acc(Email));

CREATE TABLE uses (
        Email VARCHAR2(254) REFERENCES user_acc(Email),
        PaymentDetails NUMBER REFERENCES payment_method(PaymentDetails),
        PRIMARY KEY (Email, PaymentDetails));

CREATE TABLE movie (
        MovieID NUMBER PRIMARY KEY,
        MovieName VARCHAR2(100) NOT NULL,
        Genre VARCHAR2(200) NOT NULL,
        Director VARCHAR2(100) NOT NULL,
        MovieRating VARCHAR2(10) NOT NULL,
        ReleaseYear NUMBER NOT NULL,
        Resolution VARCHAR2(10) NOT NULL,
        Runtime VARCHAR2(20) NOT NULL,
        Synopsis VARCHAR2(2000),
        MovieCast VARCHAR2(1000));

CREATE TABLE orders (
        Email VARCHAR2(254),
        PaymentDetails NUMBER,
        MovieID NUMBER REFERENCES movie(MovieID),
        PRIMARY KEY (Email, PaymentDetails, MovieID),
        FOREIGN KEY (Email, PaymentDetails) REFERENCES uses(Email, PaymentDetails));
```

```
CREATE TABLE studio (
        StudioName VARCHAR2(100) PRIMARY KEY,
        ProducedMovies VARCHAR2(1000));

CREATE TABLE produces (
        MovieID NUMBER REFERENCES movie(MovieID),
        StudioName VARCHAR2(100) REFERENCES studio(StudioName),
        PRIMARY KEY (StudioName, MovieID));

CREATE TABLE rental (
        MovieID NUMBER REFERENCES movie(MovieID),
        RentalPrice NUMBER NOT NULL,
        RentalDate DATE,
        RentalExpiry DATE,
        PRIMARY KEY (MovieID));

CREATE TABLE purchase (
        MovieID NUMBER REFERENCES movie(MovieID),
        BuyingPrice Float(2) NOT NULL,
        PurchaseDate DATE);

CREATE TABLE movie_review (
        RevID NUMBER PRIMARY KEY,
        WrittenRev VARCHAR2(400),
        NumericRating NUMBER,
        Email VARCHAR2(254) REFERENCES user_acc(Email));

CREATE TABLE links_to (
        RevID NUMBER REFERENCES movie_review(RevID),
        MovieID NUMBER REFERENCES movie(MovieID),
        MoviePageUrl VARCHAR2(2000),
        PRIMARY KEY(RevID, MovieID));

CREATE TABLE writes (
        RevID NUMBER REFERENCES movie_review(RevID),
        Email VARCHAR2(254) REFERENCES user_acc(Email),
        PRIMARY KEY(Email, RevID));

CREATE TABLE fanclub_membership (
        MembershipID NUMBER PRIMARY KEY,
        RentalDiscount FLOAT(2),
        PurchaseDiscount FLOAT(2),
        RentalSpecialOffers VARCHAR(200),
        PurchaseSpecialOffers VARCHAR(200),
        OffersRentalStartDate DATE,
        OffersRentalEndDate DATE,
        OffersPurchaseStartDate DATE,
        OffersPurchaseEndDate DATE);

CREATE TABLE subscribes_to (
        MembershipID NUMBER REFERENCES fanclub_membership(MembershipID),
        Email VARCHAR2(254),
        PaymentDetails NUMBER,
        SubscriptionTime NUMBER,
        SubscriptionPrice Float(2),
        SubscriptionDate DATE,
        SubscriptionExpiry DATE,
        FOREIGN KEY (Email,PaymentDetails) REFERENCES uses(Email,PaymentDetails),
        PRIMARY KEY (MembershipID, Email, PaymentDetails));

CREATE TABLE has (
        MembershipID NUMBER REFERENCES Fanclub_membership(MembershipID),
        StudioName VARCHAR2(100) REFERENCES studio(StudioName),
        PRIMARY KEY(MembershipID, StudioName));
```

Populate Tables - SQL Code

```
--Walt Disney Pictures
INSERT INTO movie VALUES
(1, 'The Lion King', 'Action-Adventure, Drama, Family, Musical', 'Jon Favreau', 'PG', 2019, 'HD',
'118 Minutes',
'Disney''s film journeys to the African savanna where a future king is born. Simba idolizes his
father, King Mufasa,
and takes to heart his own royal destiny. But not everyone in the kingdom celebrates the new
cub''s arrival. Scar,
Mufasa''s brother—and former heir to the throne—has plans of his own. The battle for Pride
Rock is ravaged with betrayal,
tragedy and drama, ultimately resulting in Simba''s exile. With help from a curious pair of
newfound friends, Simba will
have to figure out how to grow up and take back what is rightfully his. The all-star cast includes
Donald Glover as Simba,
Beyoncé Knowles-Carter as Nala, James Earl Jones as Mufasa, Chiwetel Ejiofor as Scar, Seth
Rogen as Pumbaa and Billy Eichner
as Timon, and utilizes pioneering filmmaking techniques to bring treasured characters to life in a
whole new way.',
```

```sql
'Donald Glover, Seth Rogen, Chiwetel Ejiofor, Alfre Woodard, Billy Eichner, John Kani, John Oliver, Florence Kasumba, Eric Andre, Keegan-Michael Key,
JD McCrary, Shahadi Joseph, Beyoncé Knowles-Carter, James Jones, Penny Johnson Jerald, Amy Sedaris, Chance Bennett, Josh McCrary, Phil LaMarr, J. Lee');

--Walt Disney Pictures
INSERT INTO movie VALUES
(2, 'Frozen', 'Animation, Family, Fantasy, Musical', 'Chris Buck, Jennifer Lee', 'PG', 2013, 'HD', '102 Minutes',
'Walt Disney Animation Studios, the studio behind "Tangled" and "Wreck-It Ralph," presents "Frozen," a stunning
big-screen comedy adventure. Fearless optimist Anna (voice of Kristen Bell) sets off on an epic journey—teaming
up with rugged mountain man Kristoff (voice of Jonathan Groff) and his loyal reindeer Sven—to find her sister
Elsa (voice of Idina Menzel), whose icy powers have trapped the kingdom of Arendelle in eternal winter. Encountering
Everest-like conditions, mystical trolls and a hilarious snowman named Olaf, Anna and Kristoff battle the elements
in a race to save the kingdom.',
'Kristen Bell, Idina Menzel, Jonathan Groff, Josh Gad, Santino Fontana, Alan Tudyk, Ciarán Hinds, Chris Williams, Stephen John Anderson, Maia Wilson,
Edie McClurg, Robert Pine');

--Walt Disney Pictures
INSERT INTO movie VALUES
(3, 'Tangled', 'Action-Adventure, Animation, Family, Fantasy, Musical', 'Nathan Grengo, Byron Howard', 'PG', 2010, 'HD', '100 Minutes',
'When the kingdom''s most wanted, and most charming, bandit Flynn Rider hides in a mysterious tower, the last thing he expects to
find is Rapunzel, a spirited teen with an unlikely superpower: 70 feet of magical golden hair! Together, the unlikely duo sets off
on a fantastic journey filled with surprising heroes, laughter and suspense.',
'Ron Perlman, Zachary Levi, Richard Kiel, M.C. Gainey, Jeffrey Tambor, Mandy Moore, Brad Garrett, Paul F. Tompkins, Donna Murphy');

--Walt Disney Pictures
INSERT INTO movie VALUES
(4, 'Big Hero 6', 'Action-Adventure, Animation, Family, Sci-Fi', 'Chris Williams, Don Hall', 'PG', 2014, 'HD', '102 Minutes',
'With all the heart and humor audiences expect from Walt Disney Animation Studios, Big Hero 6 is an action-packed comedy adventure
that introduces Baymax, a lovable, personal companion robot, who forms a special bond with robotics prodigy Hiro Hamada. When a devastating
```

```
turn of events catapults them into the midst of a dangerous plot unfolding in the streets of San
Fransokyo, Hiro turns to Baymax and his diverse
group of friends— adrenaline junkie Go Go Tomago, neatnik Wasabi, chemistry whiz Honey
Lemon and fanboy Fred — who transform into a band
of unlikely heroes. Bring home Disney''s Big Hero 6, featuring comic-book-style action and
hilarious, unforgettable characters — it''s
fun for the whole family!',
'Jamie Chung, Genesis Rodriguez, James Cromwell, Ryan Potter, Daniel Henney, Alan Tudyk,
Scott Adsit, T.J. Miller (canon), Damon Wayans Jr., Maya Rudolph');

--Warner Bros
INSERT INTO movie VALUES
(5, 'The Matrix', 'Action, Sci-Fi, Technology, Thriller', 'Andy Wachowski, Larry Wachowski', 'R',
1999, '4K HDR', '136 Minutes',
'Neo is a young software engineer and part-time hacker who is singled out by some mysterious
figures who want to introduce him
into the secret of ''the matrix''. The cops also seem to be after him, and he takes a chance on
discovering what he has always suspected:
that the world is not quite what it seems to be and a sinister conspiracy is at work.',
'Carrie-Anne Moss, Hugo Weaving, Joe Pantoliano, Keanu Reeves, Laurence Fishburne');

--Warner Bros
INSERT INTO movie VALUES
(6, 'Harry Potter and the Philosopher''s Stone', 'Fantasy, Adventure, Action', 'Chris Columbus',
'PG', 2001, 'HD', '152 Minutes',
'Harry Potter has learned to live with his bullying Uncle Vernon, his callous Aunt Petunia and the
constant whining of his greedy,
spoiled cousin Dudley. He''s even learned to live with sleeping in the cupboard under the stairs.
Harry''s relatives have just as
reluctantly learned to live with the unwelcome presence of their orphaned relation, a constant
reminder of Petunia''s "wayward" sister
and brother-in-law and their mysterious and untimely demise. Even the impending arrival of his
11th birthday offers no excitement for
Harry--as usual, there''s little chance of cards, presents or any kind of birthday treat. This year,
however, is different. On his 11th birthday,
Harry learns that he is the orphaned son of two powerful wizards and possesses unique magical
powers of his own. Invited to attend Hogwarts
School of Witchcraft and Wizardry, Harry embarks on the adventure of a lifetime. At Hogwarts,
he finds the home and family he has never had.',
'Daniel Radcliffe, Rupert Grint, Emma Watson, John Cleese, Robbie Coltrane, Warwick Davis,
Richard Griffiths, Richard Harris (I), Ian Hart, John Hurt,
Alan Rickman, Fiona Shaw');

--Paramount Pictures
```

```
INSERT INTO movie VALUES
(7, 'Mission: Impossible', 'Action, Adventure, Romance, Technology', 'Brian De Palma', 'PG-13',
1996, 'HD', '110 Minutes',
'A former Russian spy selling international intelligence on the black market... a list containing
names of the top undercover agents in the
world... a corrupt agent doubling for an unknown organization... a mysterious arms dealer... a
spy agency ready to disavow the actions or
existence of any of its members captured or killed... and one man on a mission that seems
impossible.',
'Tom Cruise, Jon Voight, Emmanuelle Bart, Henry Czerny, Jean Reno, Ving Rhames, Randall
Paul, John McLaughlin, Karel Dobry, Melissa Knatchbull');

--Paramount Pictures
INSERT INTO movie VALUES
(8, 'Interstellar', 'Action, Drama, Sci-Fi', 'Christopher Nolan', 'PG-13', 2014, 'HD', '169 Minutes',
'With our time on Earth coming to an end, a team of explorers undertakes the most important
mission in human history. Traveling beyond this galaxy
to discover whether mankind has a future among the stars.',
'Matthew McConaughey, Anne Hathaway, Jessica Chastain, Ellen Burstyn');

--DreamWorks Animation
INSERT INTO movie VALUES
(9, 'Madagascar', 'Animated, Kids, Family', 'Eric Darnell, Tom McGrath', 'PG', 2005, 'HD', '86
Minutes',
'Alex the Lion is the king of the urban jungle as the main attraction at New York''s Central Park
Zoo. He and his best friends Marty the Zebra,
Melman the Giraffe and Gloria the Hippo have lived their entire lives in blissful captivity,
complete with lavish meals and their own park views.
Not content to leave well enough alone, Marty allows his curiosity to get the better of him and,
with the help of some prodigious penguins, makes
his escape to explore the world he''s been missing, intending to return before morning. In the
middle of the night, Alex, Melman and Gloria discover
their friend missing and decide the only course of action is to break out of the zoo and get Marty
back home before anyone notices they''re gone.
Even in New York, a lion, giraffe and hippopotamus wandering the streets and riding the subway
are bound to attract some attention.',
'Ben Stiller, Chris Rock, David Schwimmer');

--Blue Sky Studios
INSERT INTO movie VALUES
(10, 'Ice Age', 'Animated, Family', 'Chris Wedge, Carlos Saldanha', 'PG', 2002, 'HD', '81
Minutes',
'It''s a story about a Mammoth, a Sloth, a Tiger and a Squirrel. They face the onslaught of the
new Ice Age.','Denis Leary, John Leguizamo, Ray Romano');
```

```
--DreamWorks Animation
INSERT INTO movie VALUES
(11, 'Kung Fu Panda', 'Animated, Family', 'Mark Osborne, John Stevenson', 'PG', 2008, 'HD', '92
Minutes',
'Enthusiastic, big and a little clumsy, Po is the biggest fan of Kung Fu around--which doesn''t
exactly come in handy while working every day in his family''s
noodle shop. Unexpectedly chosen to fulfill an ancient prophecy, Po''s dreams become reality
when he joins the world of Kung Fu and studies alongside his idols,
the legendary Furious Five--Tigress, Crane, Mantis, Viper and Monkey--under the leadership of
their guru, Master Shifu. But before they know it, the vengeful and
treacherous snow leopard Tai Lung is headed their way, and it''s up to Po to defend everyone
from the oncoming threat. Can he turn his dreams of becoming a Kung
Fu master into reality? Po puts his heart--and his girth--into the task, and the unlikely hero
ultimately finds that his greatest weaknesses turn out to be his
greatest strengths.',
'Jack Black, Ian McShane, Angelina Jolie');

--DreamWorks Animation
INSERT INTO movie Values
(12, 'Shrek', 'Animated, Family', 'Andrew Adamson, Vicky Jenson', 'PG', 2001, '4K HDR', '90
Minutes',
'Once upon a time, in a far away swamp, there lived an ornery ogre named Shrek whose
precious solitude is suddenly shattered by an invasion of annoying fairy tale
characters. There are blind mice in his food, a big, bad wolf in his bed, three little homeless pigs
and more, all banished from their kingdom by the evil Lord
Farquaad. Determined to save their home--not to mention his own--Shrek cuts a deal with
Farquaad and sets out to rescue the beautiful Princess Fiona to be
Farquaad''s bride. Accompanying him on his mission is wisecracking Donkey, who will do
anything for Shrek... except shut up. Rescuing the Princess from a
fire-breathing dragon may prove the least of their problems when the deep, dark secret she has
been keeping is revealed.',
'Eddie Murphy, Cameron Diaz, John Lithgow, Mike Myers, Vincent Cassel, Linda Hunt');

--Paramount Pictures
INSERT INTO movie VALUES
(13, 'Top Gun: Maverick', 'Action, Adventure, Drama', 'Joseph Kosinski', 'PG', 2002, 'HD', '130
Minutes',
'After more than thirty years of service as one of the Navy''s top aviators, Pete Mitchell is where
he belongs, pushing the envelope as a courageous test pilot
and dodging the advancement in rank that would ground him.',
'Tom Cruise, Miles Teller, Jennifer Connelly, Jon Hamm, Glen Powell');

--20th Century Studios
```

```
INSERT INTO movie VALUES
(14, 'Avatar', 'Sci-Fi, Action, Aventure', 'James Cameron', 'PG-13', 2009, 'HD', '162 Minutes',
'The story''s hero is Jake Sully, an ex-Marine confined to a wheelchair. Bitter and disillusioned,
he''s still a warrior at heart. All Jake ever wanted was
something worth fighting for and he ultimately finds it in the place he least expected- on a
distant world. Jake has been recruited to join an expedition to
the moon Pandora, which corporate interests are strip-mining for a mineral worth $20 million per
kilogram on Earth. To facilitate their work, the humans use
a link system that projects a person''s consciousness into a hybrid of humans and Pandora''s
indigenous humanoid race, the Na''vi. This human-Na''vi hybrid - a
fully living, breathing body that resembles the Na''vi but possesses the individual human''s
thoughts, feelings and personality - is known as an "avatar." In
his new avatar form, Jake can once again walk. His mission is to interact with and infiltrate the
Na''vi with the hope of enlisting their help - or at least
their acquiescence -- in mining the ore.',
'Michelle Rodriguez, Sam Worthington, Sigourney Weaver, Stephen Lang, Zoe Saldana');

--20th Century Studios
INSERT INTO movie VALUES
(15, 'Night at the Museum', 'Fantasy, Comedy, Family, Adventure', 'Shawn Levy', 'PG', 2006,
'HD', '109 Minutes',
'Larry Daley, a down-and-out dreamer whose imaginative ideas have never paid off, finds
himself in desperate need of a job. Larry has always believed he was
destined for big things. But he has no idea just how literally gargantuan and hairy a challenge he
will face when he grudgingly accepts the supposedly menial
graveyard shift as a security guard at the Natural History Museum. On his very first night on the
job, Larry is handed an over-sized flashlight and a dog-eared
instruction manual, then left all alone in the eerily quiet, cavernous museum. At least, he thinks
he''s alone. To his utter astonishment and disbelief, Larry
watches in shock and awe as, one by one, the primeval beasts and storied icons that surround
him stir magically to life--and total havoc ensues. Now, as
Tyrannosaurus Rex and Atilla The Hun carve a swath of destruction through the marble
corridors, and lions and monkeys prowl the fragile exhibits, Larry is at a
loss as to how to get the museum back under control.',
'Ben Stiller, Carla Gugino, Robin Williams, Dick Van Dyke, Mickey Rooney, Bill Cobbs, Jake
Cherry, Ricky Gervais, Kim Raver, Patrick Gallagher, Rami Malek,
Mizuo Peck');

--20th Century Studios
INSERT INTO movie VALUES
(16, 'Independance Day', 'Adventure, Sci-Fi, Action', 'Roland Emmerich', 'PG-13', 1996, 'HD',
'145 Minutes',
'It is an ordinary summer day. But then, without warning, something very extraordinary happens.
Enormous shadows fall across the land. Strange atmospheric
phenomena, ominous and mesmerizing, surface around the globe. All eyes turn upward. The
question of whether we''re alone in the universe has finally been
answered. And, in a matter of minutes, the lives of every person across the globe are forever
changed. With the fate of our planet at stake, the Fourth of July
is about to take on an entirely new meaning. No longer will it be an American holiday. It will be
known as the day the entire world fought back. The day we did
not go gentle into the good night... The day all of us on planet Earth celebrated our
independence day.',
'Will Smith, Bill Pullman, Jeff Goldblum, Mary McDonnell, Judd Hirsch, Margaret Colin, Randy
Quaid, Robert Loggia, James Rebhorn, Harvey Fierstein, Adam Baldwin,
Brent Spiner');

--Lionsgate Films
INSERT INTO movie VALUES
(17, 'The Hunger Games', 'Action, Adaptation, Adventure, Fantasy', 'Gary Ross', 'PG-13', 2012,
'HD', '143 Minutes',
'Every year in the ruins of what was once North America, the nation of Panem forces each of its
twelve districts to send a teenage boy and girl to compete in
the Hunger Games. Part twisted entertainment, part government intimidation tactic, the Hunger
Games are a nationally televised event in which "Tributes" must
fight with one another until one survivor remains. Pitted against highly-trained Tributes who have
prepared for these Games their entire lives, Katniss is forced
to rely upon her sharp instincts as well as the mentorship of drunken former victor Haymitch
Abernathy. If she''s ever to return home to District 12, Katniss must
make impossible choices in the arena that weigh survival against humanity and life against
love.',
'Lenny Kravitz, Toby Jones, Woody Harrelson, Wes Bentley, Stanley Tucci, Jennifer Lawrence,
Elizabeth Banks, Liam Hemsworth, Donald Sutherland, Isabelle Fuhrman,
Alexander Ludwig, Amandla Stenberg');

--Lionsgate Films
INSERT INTO movie VALUES
(18, 'Knives Out', 'Thriller, Mystery', 'Rian Johnson', 'PG-13', 2019, 'HD', '130 Minutes',
'When renowned crime novelist Harlan Thrombey is found dead at his estate just after his 85th
birthday, the inquisitive and debonair Detective Benoit Blanc is
mysteriously enlisted to investigate. From Harlan''s dysfunctional family to his devoted staff,
Blanc sifts through a web of red herrings and self-serving lies
to uncover the truth behind Harlan''s untimely death.',
'Daniel Craig, Chris Evans, Jamie Lee Curtis, Ana de Armas, Michael Shannon, Toni Collette,
LaKeith Stanfield, Don Johnson, Katherine Langford,
Christopher Plummer, Jaeden Martell');

--Universal Pictures
INSERT INTO movie VALUES
```

```
(19, 'Despicable Me', 'Animated, Comedy, Family', 'Chris Renaud, Pierre Coffin', 'PG', 2010, '4K
HDR', '95 Minutes',
'In a happy suburban neighborhood surrounded by white picket fences with flowering rose
bushes, sits a black house with a dead lawn. Unbeknownst to the neighbors,
hidden beneath this home is a vast secret hideout. Surrounded by a small army of minions, we
discover Gru planning the biggest heist in the history of the world.
He is going to steal the moon, yes, the moon. Gru delights in all things wicked. Armed with his
arsenal of shrink rays, freeze rays, and battle-ready vehicles for
land and air, he vanquishes all who stand in his way. Until the day he encounters the immense
will of three little orphaned girls who look at him and see
something that no one else has ever seen: a potential Dad. The world''s greatest villain has just
met his greatest challenge: three little girls named Margo,
Edith and Agnes.',
'Steve Carell, Jason Segel, Russell Brand, Julie Andrews, Will Arnett, Kristen Wiig, Danny
McBride, Miranda Cosgrove, Dana Gaier, Elsie Fisher, Pierre Coffin,
Chris Renaud');

--Columbia Pictures
INSERT INTO movie Values
(20, 'Men in Black', 'Sci-Fi, Comedy, Action', 'Barry Sonnenfeld', 'PG-13', 1997, 'HD', '98
Minutes',
'J and K, two federal agents aka "The Men in Black," are assigned to investigate all alien related
phenomena. The agents uncover an intergalactic plot to
assassinate two ambassadors from opposing galaxies who happen to reside in New York City.
Jay and Kay''s mission is to foil the plot by tracking down the
terrorist, thereby preventing the Earth from being destroyed.',
'Tommy Lee Jones, Will Smith, Linda Fiorentino, Vincent D''Onofrio, Rip Torn');

COMMIT;

ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';

INSERT INTO user_acc VALUES
('john.bob@gmail.com', 'John', 'Bob', DATE '2000-01-16', 'ApplePie');

INSERT INTO user_acc VALUES
('tommyross@gmail.com', 'Tommy', 'Ross', DATE '1986-04-17', 'TorontoMet');

Insert INTO user_acc VALUES
('franklynmoon@yahoo.com', 'Franklyn', 'Moon', DATE '1975-12-30', 'AppleCiderVinegar');

INSERT INTO user_acc VALUES
('amylucas@gmail.com', 'Amy', 'Lucas', DATE '2005-03-18', 'MapleSyrup');
```

```
INSERT INTO user_acc VALUES
('rosalindaflemming@gmail.com', 'Rosalinda', 'Flemming', DATE '2008-04-04',
'KoolaidJammers');

INSERT INTO user_acc VALUES
('marioodonnell@gmail.com', 'Mario', 'Odonnell', DATE '1985-10-08', 'Iphone14ProMax');

INSERT INTO user_acc VALUES
('courtneypage@gmail.com', 'Courney', 'Page', DATE '1993-02-28', 'StudentLearningCentre');

INSERT INTO user_acc VALUES
('beverlymeadows@gmail.com', 'Bevery', 'Meadows', DATE '1969-09-18', 'Ghostbusters1234');

INSERT INTO user_acc VALUES
('lesleywhite@yahoo.com', 'Lesley', 'White', DATE '2001-11-13', 'SamsungGalaxy');

INSERT INTO user_acc VALUES
('gregrobinson@yahoo.com', 'Greg', 'Robinson', DATE '1972-05-15', 'StarWars');

-----------------------------------------

INSERT INTO studio VALUES
('Walt Disney Pictures', 'The Lion King, Frozen, Tangled, Big Hero 6');

INSERT INTO studio VALUES
('Warner Bros', 'The Matrix, Harry Potter and the Philosopher''s Stone');

INSERT INTO studio VALUES
('Paramount Pictures', 'Mission: Impossible, Interstellar, Top Gun: Maverick');

INSERT INTO studio VALUES
('Columbia Pictures', 'Men In Black');

INSERT INTO studio VALUES
('DreamWorks Animation', 'Madagascar, Kung Fu Panda, Shrek');

INSERT INTO studio VALUES
('Blue Sky Studios', 'Ice Age');

INSERT INTO studio VALUES
('20th Century Studios', 'Night At the Museum, Avatar, Independence Day');

INSERT INTO studio VALUES
('Lionsgate Films', 'The Hunger Games, Knives Out');
```

```
INSERT INTO studio VALUES
('Universal Pictures', 'Despicable Me');
--------------------------------------------------
INSERT INTO produces VALUES
(1, 'Walt Disney Pictures');

INSERT INTO produces VALUES
(2, 'Walt Disney Pictures');

INSERT INTO produces VALUES
(3, 'Walt Disney Pictures');

INSERT INTO produces VALUES
(4, 'Walt Disney Pictures');

INSERT INTO produces VALUES
(5, 'Warner Bros');

INSERT INTO produces VALUES
(6, 'Warner Bros');

INSERT INTO produces VALUES
(7, 'Paramount Pictures');

INSERT INTO produces VALUES
(8, 'Paramount Pictures');

INSERT INTO produces VALUES
(9, 'DreamWorks Animation');

INSERT INTO produces VALUES
(10, 'Blue Sky Studios');

INSERT INTO produces VALUES
(11, 'DreamWorks Animation');

INSERT INTO produces VALUES
(12, 'DreamWorks Animation');

INSERT INTO produces VALUES
(13, 'Paramount Pictures');

INSERT INTO produces VALUES
```

```
(14, '20th Century Studios');

INSERT INTO produces VALUES
(15, '20th Century Studios');

INSERT INTO produces VALUES
(16, '20th Century Studios');

INSERT INTO produces VALUES
(17, 'Lionsgate Films');

INSERT INTO produces VALUES
(18, 'Lionsgate Films');

INSERT INTO produces VALUES
(19, 'Universal Pictures');

INSERT INTO produces VALUES
(20, 'Columbia Pictures');
--------------------------------------------
--Frozen, Walt Disney Studios
INSERT INTO purchase VALUES
(1, 24.99, DATE '2022-09-24');

-- The Matrix, Warner Bros
INSERT INTO purchase VALUES
(2, 15.99, DATE '2022-09-25');

--Frozen, Walt Disney Pictures
INSERT INTO purchase VALUES
(3, 24.99, DATE '2022-09-25');

--Interstellar, Paramount Pictures
INSERT INTO purchase VALUES
(4, 16.99, DATE '2022-09-26');

--Big Hero 6, Walt Disney Pictures
INSERT INTO purchase VALUES
(5, 13.99, DATE '2022-09-27');

--Frozen, Walt Disney Pictures
INSERT INTO purchase VALUES
(6, 13.99, DATE '2022-09-27');
```

```
--The Matrix, Warner Bros
INSERT INTO purchase VALUES
(7, 15.99, DATE '2022-09-28');

--Ice Age, Blue Sky Studios
INSERT INTO purchase VALUES
(8, 14.99, DATE '2022-09-29');

--Men in Black, Columbia Pictures
INSERT INTO purchase VALUES
(9, 6.99, DATE '2022-09-30');

--Madagascar, DreamWorks Animation
INSERT INTO purchase VALUES
(10, 12.99, DATE '2022-10-01');

--The Lion King, Walt Disney Pictures
INSERT INTO purchase VALUES
(11, 24.99, DATE '2022-10-01');

--Avatar, 20th Century Studios
INSERT INTO purchase VALUES
(12, 9.99, DATE '2022-10-02');

--Tangled, Walt Disney Pictures
INSERT INTO purchase VALUES
(13, 24.99, DATE '2022-10-03');

--Kung Fu Panda, DreamWorks Animation
INSERT INTO purchase VALUES
(14, 12.99, DATE '2022-10-04');

--Ice Age, Blue Sky Studios
INSERT INTO purchase VALUES
(15, 14.99, DATE '2022-10-04');

--
--------------------------------------------------------------------------------------------------------------------------------
----------------

--The Lion King, Walt Disney Pictures
INSERT INTO rental VALUES
(1, 4.99, TIMESTAMP '2022-09-24 23:14:16', TIMESTAMP '2022-10-08 23:14:16');
```

```
--Night at the Museum, 20th Century Studios
INSERT INTO rental VALUES
(2, 4.99, TIMESTAMP '2022-09-25 13:05:19', TIMESTAMP '2022-10-09 13:05:19');

--Knives Out, Lionsgate Films
INSERT INTO rental VALUES
(3, 3.99, TIMESTAMP '2022-09-25 08:34:34', TIMESTAMP '2022-10-09 08:34:34');

--Despicable Me, Universal Pictures
INSERT INTO rental VALUES
(4, 2.99, TIMESTAMP '2022-09-28 12:43:12', TIMESTAMP '2022-10-12 12:43:12');

--Knives Out, Lionsgate Films
INSERT INTO rental VALUES
(5, 2.99, TIMESTAMP '2022-09-29 18:35:12', TIMESTAMP '2022-10-13 18:35:12');

--Night at the Museum, 20th Century Studios
INSERT INTO rental VALUES
(6, 4.99, TIMESTAMP '2022-09-29 15:56:14', TIMESTAMP '2022-10-13 15:56:14');

--Mission: Impossible, Paramount Pictures
INSERT INTO rental VALUES
(7, 3.99, TIMESTAMP '2022-09-30 22:16:39', TIMESTAMP '2022-10-14 22:16:39');

--Shrek, Dreamworks Animation
INSERT INTO rental VALUES
(8, 4.99, TIMESTAMP '2022-09-30 16:53:23', TIMESTAMP '2022-10-14 16:53:23');

--Shrek, Dreamworks Animation
INSERT INTO rental VALUES
(9, 4.99, TIMESTAMP '2022-10-01 17:12:54', TIMESTAMP '2022-10-15 17:12:54');

--The Hunger Games, Lionsgate Films
INSERT INTO rental VALUES
(10, 3.99, TIMESTAMP '2022-10-02 19:13:43', TIMESTAMP '2022-10-16 19:13:43');

--
---------------------------------------------------------------------------------------------------------------------------------
----------------

INSERT INTO payment_method VALUES
(4894842049204922, 'Credit', 'john.bob@gmail.com');

INSERT INTO payment_method VALUES
```

```
(5384048443249893, 'Debit', 'john.bob@gmail.com');

INSERT INTO payment_method VALUES
(5455838554939584, 'Debit', 'tommyross@gmail.com');

INSERT INTO payment_method VALUES
(8748535454355090, 'Credit', 'franklynmoon@yahoo.com');

INSERT INTO payment_method VALUES
(4398554454355454, 'Credit', 'amylucas@gmail.com');

INSERT INTO payment_method VALUES
(9989545499005495, 'Debit', 'amylucas@gmail.com');

INSERT INTO payment_method VALUES
(5734535454350990, 'Debit', 'rosalindaflemming@gmail.com');

INSERT INTO payment_method VALUES
(9099434554355435, 'Credit', 'marioodonnell@gmail.com');

INSERT INTO payment_method VALUES
(0394040454353545, 'Debit', 'marioodonnell@gmail.com');

INSERT INTO payment_method VALUES
(5835593459409899, 'Debit', 'courtneypage@gmail.com');

INSERT INTO payment_method VALUES
(9458535409335549, 'Debit', 'beverlymeadows@gmail.com');

INSERT INTO payment_method VALUES
(7878989898984355, 'Credit', 'beverlymeadows@gmail.com');

INSERT INTO payment_method VALUES
(8494530385744204, 'Credit', 'lesleywhite@yahoo.com');

INSERT INTO payment_method VALUES
(7593932453959204, 'Debit', 'lesleywhite@yahoo.com');

INSERT INTO payment_method VALUES
(0495039465740343, 'Credit', 'gregrobinson@yahoo.com');

INSERT INTO payment_method VALUES
(9392893840405445, 'Debit', 'gregrobinson@yahoo.com');
```

```
---------------------------------------------------------------------------------------------------------------------
--------------------------------

INSERT INTO uses VALUES
('john.bob@gmail.com', 4894842049204922);

INSERT INTO uses VALUES
('john.bob@gmail.com', 5384048443249893);

INSERT INTO uses VALUES
('tommyross@gmail.com', 5455838554939584);

INSERT INTO uses VALUES
('franklynmoon@yahoo.com', 8748535454355090);

INSERT INTO uses VALUES
('amylucas@gmail.com', 4398554454355454);

INSERT INTO uses VALUES
('amylucas@gmail.com', 9989545499005495);

INSERT INTO uses VALUES
('rosalindaflemming@gmail.com', 5734535454350990);

INSERT INTO uses VALUES
('marioodonnell@gmail.com', 9099434554355435);

INSERT INTO uses VALUES
('marioodonnell@gmail.com', 0394040454353545);

INSERT INTO uses VALUES
('courtneypage@gmail.com', 5835593459409899);

INSERT INTO uses VALUES
('beverlymeadows@gmail.com',9458535409335549);

INSERT INTO uses VALUES
('beverlymeadows@gmail.com', 7878989898984355);

INSERT INTO uses VALUES
('lesleywhite@yahoo.com', 8494530385744204);

INSERT INTO uses VALUES
('lesleywhite@yahoo.com', 7593932453959204);
```

```
INSERT INTO uses VALUES
('gregrobinson@yahoo.com', 0495039465740343);

INSERT INTO uses VALUES
('gregrobinson@yahoo.com', 9392893840405445);

-----Orders
INSERT INTO orders VALUES
('john.bob@gmail.com', 4894842049204922, 2);

INSERT INTO orders VALUES
('john.bob@gmail.com', 5384048443249893, 5);

INSERT INTO orders VALUES
('john.bob@gmail.com', 4894842049204922, 1);

INSERT INTO orders VALUES
('john.bob@gmail.com', 5384048443249893, 15);

INSERT INTO orders VALUES
('tommyross@gmail.com', 5455838554939584, 2);

INSERT INTO orders VALUES
('franklynmoon@yahoo.com', 8748535454355090, 8);

INSERT INTO orders VALUES
('franklynmoon@yahoo.com', 8748535454355090, 4);

INSERT INTO orders VALUES
('franklynmoon@yahoo.com', 8748535454355090, 18);

INSERT INTO orders VALUES
('amylucas@gmail.com', 9989545499005495, 2);

INSERT INTO orders VALUES
('amylucas@gmail.com', 9989545499005495, 5);

INSERT INTO orders VALUES
('amylucas@gmail.com', 4398554454355454, 19);

INSERT INTO orders VALUES
('rosalindaflemming@gmail.com', 5734535454350990, 10);
```

```
INSERT INTO orders VALUES
('rosalindaflemming@gmail.com', 5734535454350990, 18);

INSERT INTO orders VALUES
('marioodonnell@gmail.com', 9099434554355435, 20);

INSERT INTO orders VALUES
('marioodonnell@gmail.com', 9099434554355435, 9);

INSERT INTO orders VALUES
('marioodonnell@gmail.com', 9099434554355435, 15);

INSERT INTO orders VALUES
('courtneypage@gmail.com', 5835593459409899, 1);

INSERT INTO orders VALUES
('courtneypage@gmail.com', 5835593459409899, 14);

INSERT INTO orders VALUES
('courtneypage@gmail.com', 5835593459409899, 7);

INSERT INTO orders VALUES
('beverlymeadows@gmail.com', 7878989898984355, 3);

INSERT INTO orders VALUES
('beverlymeadows@gmail.com', 7878989898984355, 12);

INSERT INTO orders VALUES
('lesleywhite@yahoo.com', 7593932453959204, 11);

INSERT INTO orders VALUES
('lesleywhite@yahoo.com', 7593932453959204, 12);

INSERT INTO orders VALUES
('gregrobinson@yahoo.com', 0495039465740343, 10);

INSERT INTO orders VALUES
('gregrobinson@yahoo.com', 0495039465740343, 17);


-------Movie Review
INSERT INTO movie_review VALUES
(1, 'It was not for me', 2, 'gregrobinson@yahoo.com');
```

```
INSERT INTO movie_review VALUES
(7, 'I fell asleep watching this, nice', 5, 'gregrobinson@yahoo.com');

INSERT INTO movie_review VALUES
(11, 'Wow, amazing, wow, amazing', 10, 'beverlymeadows@gmail.com');

INSERT INTO movie_review VALUES
(34, 'They''ve outdone themselves yet again! A great watch for the family', 10,
'marioodonnell@gmail.com');


-----writes
INSERT INTO writes VALUES
(1, 'gregrobinson@yahoo.com');

INSERT INTO writes VALUES
(7, 'gregrobinson@yahoo.com');

INSERT INTO writes VALUES
(11, 'beverlymeadows@gmail.com');

INSERT INTO writes VALUES
(34, 'marioodonnell@gmail.com');

----links to
INSERT INTO links_to VALUES
(1, 10, 'https://onlinemoviestore.ca/ice+age');

INSERT INTO links_to VALUES
(7, 17, 'https://onlinemoviestore.ca/the+hunger+games');

INSERT INTO links_to VALUES
(11, 12, 'https://onlinemoviestore.ca/shrek');

INSERT INTO links_to VALUES
(34, 9, 'https://onlinemoviestore.ca/madagascar');

------fanclub membership
INSERT INTO fanclub_membership VALUES
(1, 2.00, 10.00, NULL, NULL, NULL, NULL, NULL, NULL);

INSERT INTO fanclub_membership VALUES
(2, 2.00, 10.00, NULL, NULL, NULL, NULL, NULL, NULL);
```

```
INSERT INTO fanclub_membership VALUES
(3, 1.49, 8.99, '$2 Off Despicable Me', NULL, DATE '2022-09-28', DATE '2022-10-03', NULL,
NULL);

INSERT INTO fanclub_membership VALUES
(4, 1.49, 7.99, NULL, NULL, NULL, NULL, NULL, NULL);

INSERT INTO fanclub_membership VALUES
(5, 2.00, 5.99, NULL, '$8 Off Men in Black', NULL, NULL, DATE '2022-09-28', DATE
'2022-10-02');

INSERT INTO fanclub_membership VALUES
(6, 0.99, 6.99, NULL, 'BUY 1 GET 1 FREE', NULL, NULL, DATE '2022-09-29', DATE
'2022-10-05');

-----SUBSCRIBES_TO
INSERT INTO subscribes_to VALUES
(1, 'franklynmoon@yahoo.com', 8748535454355090, 3, 9.99, DATE '2022-08-17', DATE
'2022-11-17');

INSERT INTO subscribes_to VALUES
(2, 'amylucas@gmail.com', 9989545499005495, 3, 9.99, DATE '2022-08-19', DATE
'2022-08-19');

INSERT INTO subscribes_to VALUES
(3, 'amylucas@gmail.com', 9989545499005495, 6, 14.99, DATE '2022-05-22', DATE
'2022-11-22');

INSERT INTO subscribes_to VALUES
(4, 'rosalindaflemming@gmail.com', 5734535454350990, 1, 3.99, DATE '2022-10-20', DATE
'2022-11-20');

INSERT INTO subscribes_to VALUES
(5, 'marioodonnell@gmail.com', 9099434554355435, 12, 19.99, DATE '2022-01-15', DATE
'2023-01-15');

INSERT INTO subscribes_to VALUES
(6, 'courtneypage@gmail.com', 5835593459409899, 6, 14.99, DATE '2022-06-12', DATE
'2022-12-12');

-----HAS
INSERT INTO has VALUES
(1, 'Walt Disney Pictures');
```

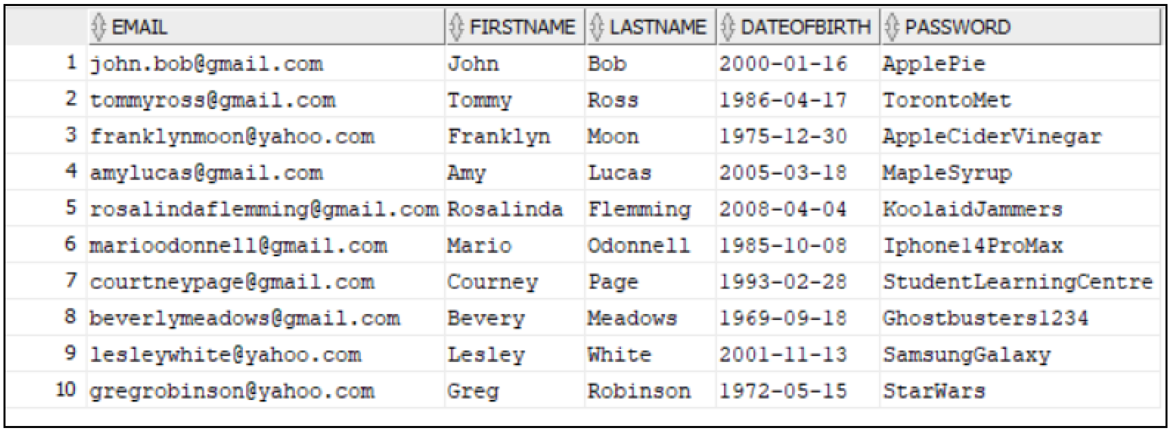
```
INSERT INTO has VALUES
(2, 'Walt Disney Pictures');

INSERT INTO has VALUES
(3, 'Universal Pictures');

INSERT INTO has VALUES
(4, 'Lionsgate Films');

INSERT INTO has VALUES
(5, 'Columbia Pictures');

INSERT INTO has VALUES
(6, '20th Century Studios');

COMMIT;
```

Drop Tables - SQL Code

```
DROP TABLE orders;
DROP TABLE uses CASCADE CONSTRAINTS;
DROP TABLE writes;
DROP TABLE links_to;
DROP TABLE produces;
DROP TABLE subscribes_to;
DROP TABLE has;
DROP TABLE fanclub_membership;
DROP TABLE payment_method CASCADE CONSTRAINTS;
DROP TABLE studio;
DROP TABLE movie CASCADE CONSTRAINTS;
DROP TABLE rental;
DROP TABLE purchase;
DROP TABLE movie_review;
DROP TABLE user_acc CASCADE CONSTRAINTS;
```

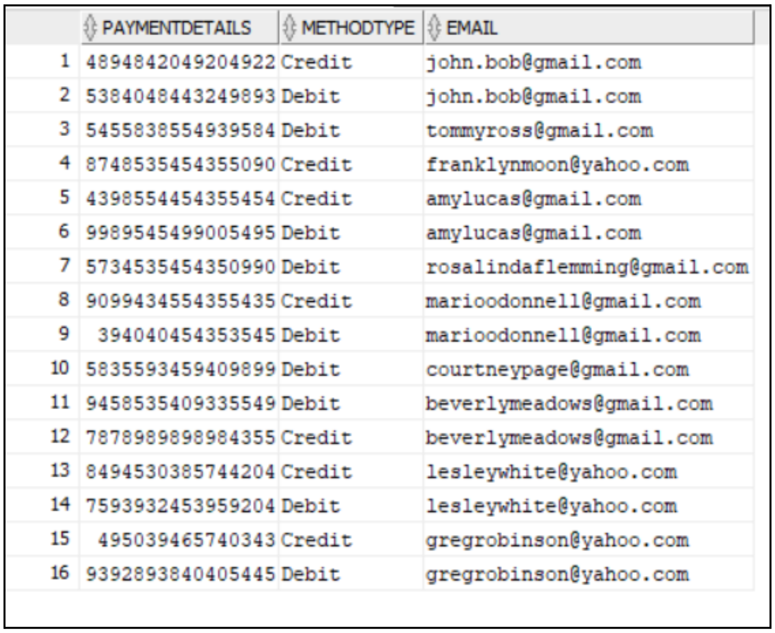
Screenshots of Tables

User_acc



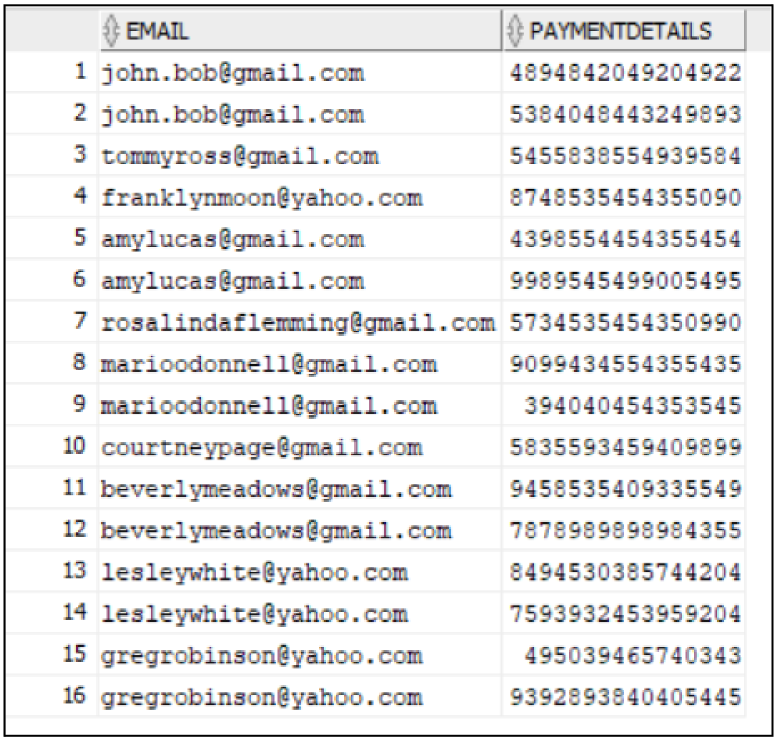
	EMAIL	FIRSTNAME	LASTNAME	DATEOFBIRTH	PASSWORD
1	john.bob@gmail.com	John	Bob	2000-01-16	ApplePie
2	tommyross@gmail.com	Tommy	Ross	1986-04-17	TorontoMet
3	franklynmoon@yahoo.com	Franklyn	Moon	1975-12-30	AppleCiderVinegar
4	amylucas@gmail.com	Amy	Lucas	2005-03-18	MapleSyrup
5	rosalindaflemming@gmail.com	Rosalinda	Flemming	2008-04-04	KoolaidJammers
6	marioodonnell@gmail.com	Mario	Odonnell	1985-10-08	Iphone14ProMax
7	courtneypage@gmail.com	Courney	Page	1993-02-28	StudentLearningCentre
8	beverlymeadows@gmail.com	Bevery	Meadows	1969-09-18	Ghostbusters1234
9	lesleywhite@yahoo.com	Lesley	White	2001-11-13	SamsungGalaxy
10	gregrobinson@yahoo.com	Greg	Robinson	1972-05-15	StarWars

Payment_method



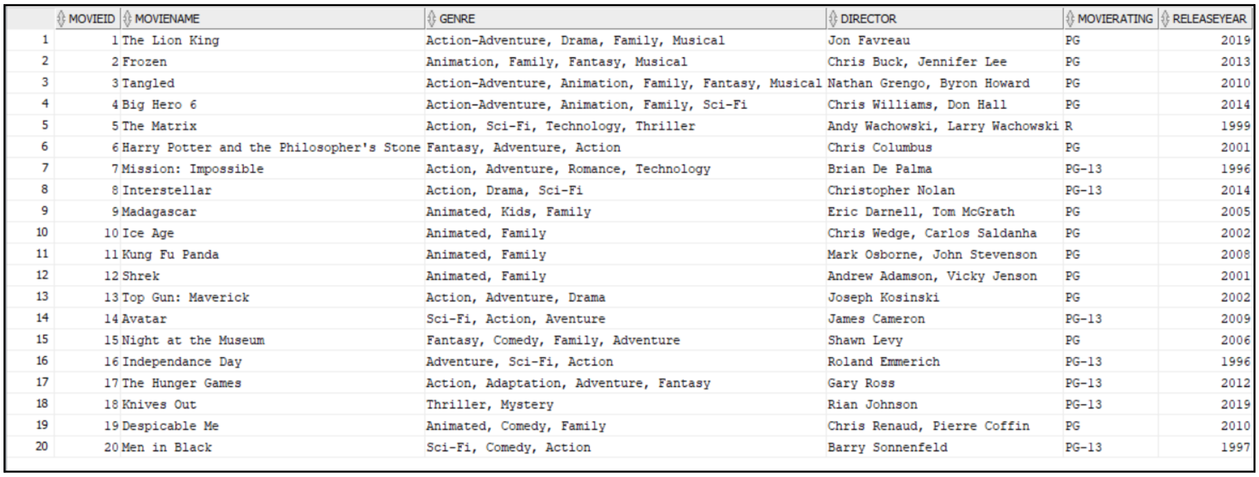
	PAYMENTDETAILS	METHODTYPE	EMAIL
1	4894842049204922	Credit	john.bob@gmail.com
2	5384048443249893	Debit	john.bob@gmail.com
3	5455838554939584	Debit	tommyross@gmail.com
4	8748535454355090	Credit	franklynmoon@yahoo.com
5	4398554454355454	Credit	amylucas@gmail.com
6	9989545499005495	Debit	amylucas@gmail.com
7	5734535454350990	Debit	rosalindaflemming@gmail.com
8	9099434554355435	Credit	marioodonnell@gmail.com
9	394040454353545	Debit	marioodonnell@gmail.com
10	5835593459409899	Debit	courtneypage@gmail.com
11	9458535409335549	Debit	beverlymeadows@gmail.com
12	7878989898984355	Credit	beverlymeadows@gmail.com
13	8494530385744204	Credit	lesleywhite@yahoo.com
14	7593932453959204	Debit	lesleywhite@yahoo.com
15	495039465740343	Credit	gregrobinson@yahoo.com
16	9392893840405445	Debit	gregrobinson@yahoo.com

Uses

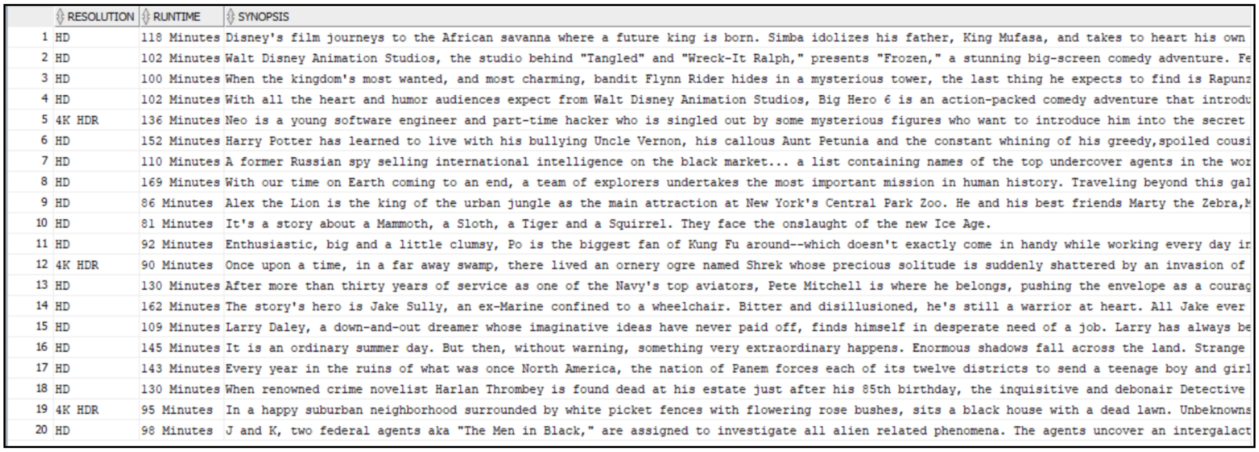


	EMAIL	PAYMENTDETAILS
1	john.bob@gmail.com	4894842049204922
2	john.bob@gmail.com	5384048443249893
3	tommyross@gmail.com	5455838554939584
4	franklynmoon@yahoo.com	8748535454355090
5	amylucas@gmail.com	4398554454355454
6	amylucas@gmail.com	9989545499005495
7	rosalindaflemming@gmail.com	5734535454350990
8	marioodonnell@gmail.com	9099434554355435
9	marioodonnell@gmail.com	394040454353545
10	courtneypage@gmail.com	5835593459409899
11	beverlymeadows@gmail.com	9458535409335549
12	beverlymeadows@gmail.com	7878989898984355
13	lesleywhite@yahoo.com	8494530385744204
14	lesleywhite@yahoo.com	7593932453959204
15	gregrobinson@yahoo.com	495039465740343
16	gregrobinson@yahoo.com	9392893840405445

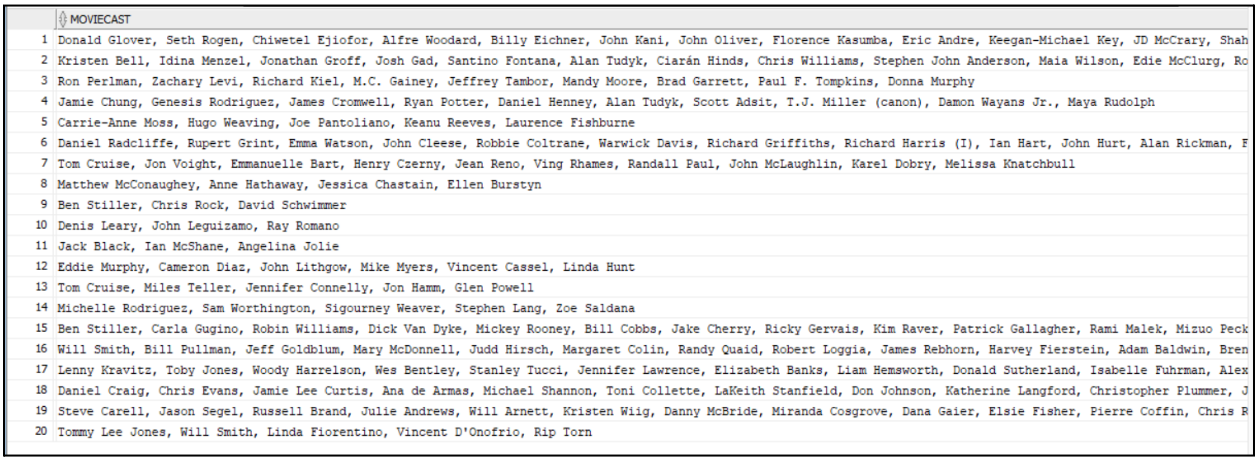
Movie



	MOVIEID	MOVIENAME	GENRE	DIRECTOR	MOVIERATING	RELEASEYEAR
1	1	The Lion King	Action-Adventure, Drama, Family, Musical	Jon Favreau	PG	2019
2	2	Frozen	Animation, Family, Fantasy, Musical	Chris Buck, Jennifer Lee	PG	2013
3	3	Tangled	Action-Adventure, Animation, Family, Fantasy, Musical	Nathan Grengo, Byron Howard	PG	2010
4	4	Big Hero 6	Action-Adventure, Animation, Family, Sci-Fi	Chris Williams, Don Hall	PG	2014
5	5	The Matrix	Action, Sci-Fi, Technology, Thriller	Andy Wachowski, Larry Wachowski	R	1999
6	6	Harry Potter and the Philosopher's Stone	Fantasy, Adventure, Action	Chris Columbus	PG	2001
7	7	Mission: Impossible	Action, Adventure, Romance, Technology	Brian De Palma	PG-13	1996
8	8	Interstellar	Action, Drama, Sci-Fi	Christopher Nolan	PG-13	2014
9	9	Madagascar	Animated, Kids, Family	Eric Darnell, Tom McGrath	PG	2005
10	10	Ice Age	Animated, Family	Chris Wedge, Carlos Saldanha	PG	2002
11	11	Kung Fu Panda	Animated, Family	Mark Osborne, John Stevenson	PG	2008
12	12	Shrek	Animated, Family	Andrew Adamson, Vicky Jenson	PG	2001
13	13	Top Gun: Maverick	Action, Adventure, Drama	Joseph Kosinski	PG	2002
14	14	Avatar	Sci-Fi, Action, Aventure	James Cameron	PG-13	2009
15	15	Night at the Museum	Fantasy, Comedy, Family, Adventure	Shawn Levy	PG	2006
16	16	Independance Day	Adventure, Sci-Fi, Action	Roland Emmerich	PG-13	1996
17	17	The Hunger Games	Action, Adaptation, Adventure, Fantasy	Gary Ross	PG-13	2012
18	18	Knives Out	Thriller, Mystery	Rian Johnson	PG-13	2019
19	19	Despicable Me	Animated, Comedy, Family	Chris Renaud, Pierre Coffin	PG	2010
20	20	Men in Black	Sci-Fi, Comedy, Action	Barry Sonnenfeld	PG-13	1997

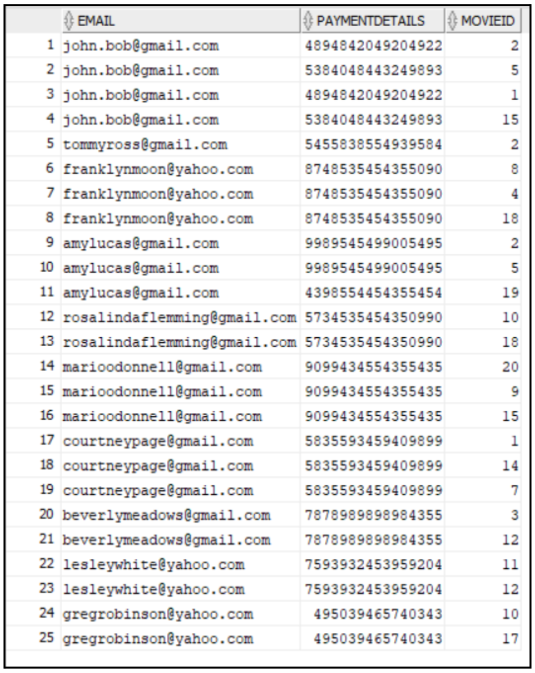


	RESOLUTION	RUNTIME	SYNOPSIS
1	HD	118 Minutes	Disney's film journeys to the African savanna where a future king is born. Simba idolizes his father, King Mufasa, and takes to heart his own
2	HD	102 Minutes	Walt Disney Animation Studios, the studio behind "Tangled" and "Wreck-It Ralph," presents "Frozen," a stunning big-screen comedy adventure. Fe
3	HD	100 Minutes	When the kingdom's most wanted, and most charming, bandit Flynn Rider hides in a mysterious tower, the last thing he expects to find is Rapunz
4	HD	102 Minutes	With all the heart and humor audiences expect from Walt Disney Animation Studios, Big Hero 6 is an action-packed comedy adventure that introdu
5	4K HDR	136 Minutes	Neo is a young software engineer and part-time hacker who is singled out by some mysterious figures who want to introduce him into the secret
6	HD	152 Minutes	Harry Potter has learned to live with his bullying Uncle Vernon, his callous Aunt Petunia and the constant whining of his greedy,spoiled cousi
7	HD	110 Minutes	A former Russian spy selling international intelligence on the black market... a list containing names of the top undercover agents in the wor
8	HD	169 Minutes	With our time on Earth coming to an end, a team of explorers undertakes the most important mission in human history. Traveling beyond this gal
9	HD	86 Minutes	Alex the Lion is the king of the urban jungle as the main attraction at New York's Central Park Zoo. He and his best friends Marty the Zebra,M
10	HD	81 Minutes	It's a story about a Mammoth, a Sloth, a Tiger and a Squirrel. They face the onslaught of the new Ice Age.
11	HD	92 Minutes	Enthusiastic, big and a little clumsy, Po is the biggest fan of Kung Fu around--which doesn't exactly come in handy while working every day in
12	4K HDR	90 Minutes	Once upon a time, in a far away swamp, there lived an ornery ogre named Shrek whose precious solitude is suddenly shattered by an invasion of
13	HD	130 Minutes	After more than thirty years of service as one of the Navy's top aviators, Pete Mitchell is where he belongs, pushing the envelope as a courag
14	HD	162 Minutes	The story's hero is Jake Sully, an ex-Marine confined to a wheelchair. Bitter and disillusioned, he's still a warrior at heart. All Jake ever
15	HD	109 Minutes	Larry Daley, a down-and-out dreamer whose imaginative ideas have never paid off, finds himself in desperate need of a job. Larry has always be
16	HD	145 Minutes	It is an ordinary summer day. But then, without warning, something very extraordinary happens. Enormous shadows fall across the land. Strange
17	HD	143 Minutes	Every year in the ruins of what was once North America, the nation of Panem forces each of its twelve districts to send a teenage boy and girl
18	HD	130 Minutes	When renowned crime novelist Harlan Thrombey is found dead at his estate just after his 85th birthday, the inquisitive and debonair Detective
19	4K HDR	95 Minutes	In a happy suburban neighborhood surrounded by white picket fences with flowering rose bushes, sits a black house with a dead lawn. Unbeknowns
20	HD	98 Minutes	J and K, two federal agents aka "The Men in Black," are assigned to investigate all alien related phenomena. The agents uncover an intergalact



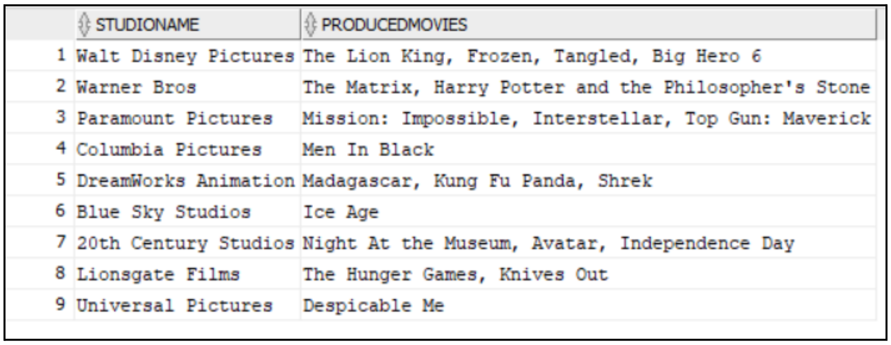
	MOVIECAST
1	Donald Glover, Seth Rogen, Chiwetel Ejiofor, Alfre Woodard, Billy Eichner, John Kani, John Oliver, Florence Kasumba, Eric Andre, Keegan-Michael Key, JD McCrary, Shah
2	Kristen Bell, Idina Menzel, Jonathan Groff, Josh Gad, Santino Fontana, Alan Tudyk, Ciarán Hinds, Chris Williams, Stephen John Anderson, Maia Wilson, Edie McClurg, Ro
3	Ron Perlman, Zachary Levi, Richard Kiel, M.C. Gainey, Jeffrey Tambor, Mandy Moore, Brad Garrett, Paul F. Tompkins, Donna Murphy
4	Jamie Chung, Genesis Rodriguez, James Cromwell, Ryan Potter, Daniel Henney, Alan Tudyk, Scott Adsit, T.J. Miller (canon), Damon Wayans Jr., Maya Rudolph
5	Carrie-Anne Moss, Hugo Weaving, Joe Pantoliano, Keanu Reeves, Laurence Fishburne
6	Daniel Radcliffe, Rupert Grint, Emma Watson, John Cleese, Robbie Coltrane, Warwick Davis, Richard Griffiths, Richard Harris (I), Ian Hart, John Hurt, Alan Rickman, F
7	Tom Cruise, Jon Voight, Emmanuelle Bart, Henry Czerny, Jean Reno, Ving Rhames, Randall Paul, John McLaughlin, Karel Dobry, Melissa Knatchbull
8	Matthew McConaughey, Anne Hathaway, Jessica Chastain, Ellen Burstyn
9	Ben Stiller, Chris Rock, David Schwimmer
10	Denis Leary, John Leguizamo, Ray Romano
11	Jack Black, Ian McShane, Angelina Jolie
12	Eddie Murphy, Cameron Diaz, John Lithgow, Mike Myers, Vincent Cassel, Linda Hunt
13	Tom Cruise, Miles Teller, Jennifer Connelly, Jon Hamm, Glen Powell
14	Michelle Rodriguez, Sam Worthington, Sigourney Weaver, Stephen Lang, Zoe Saldana
15	Ben Stiller, Carla Gugino, Robin Williams, Dick Van Dyke, Mickey Rooney, Bill Cobbs, Jake Cherry, Ricky Gervais, Kim Raver, Patrick Gallagher, Rami Malek, Mizuo Peck
16	Will Smith, Bill Pullman, Jeff Goldblum, Mary McDonnell, Judd Hirsch, Margaret Colin, Randy Quaid, Robert Loggia, James Rebhorn, Harvey Fierstein, Adam Baldwin, Bren
17	Lenny Kravitz, Toby Jones, Woody Harrelson, Wes Bentley, Stanley Tucci, Jennifer Lawrence, Elizabeth Banks, Liam Hemsworth, Donald Sutherland, Isabelle Fuhrman, Alex
18	Daniel Craig, Chris Evans, Jamie Lee Curtis, Ana de Armas, Michael Shannon, Toni Collette, LaKeith Stanfield, Don Johnson, Katherine Langford, Christopher Plummer, J
19	Steve Carell, Jason Segel, Russell Brand, Julie Andrews, Will Arnett, Kristen Wiig, Danny McBride, Miranda Cosgrove, Dana Gaier, Elsie Fisher, Pierre Coffin, Chris R
20	Tommy Lee Jones, Will Smith, Linda Fiorentino, Vincent D'Onofrio, Rip Torn

Orders



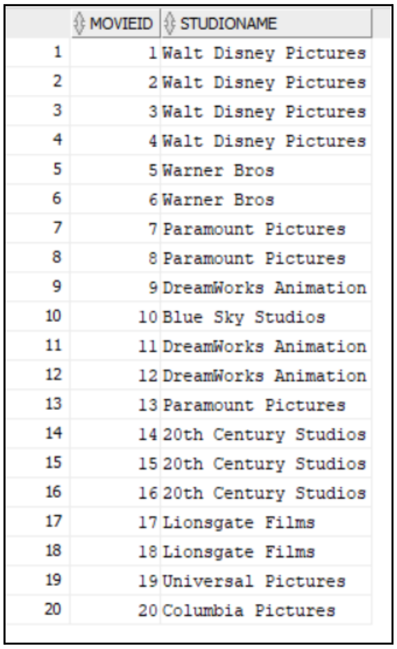
	EMAIL	PAYMENTDETAILS	MOVIEID
1	john.bob@gmail.com	4894842049204922	2
2	john.bob@gmail.com	5384048443249893	5
3	john.bob@gmail.com	4894842049204922	1
4	john.bob@gmail.com	5384048443249893	15
5	tommyross@gmail.com	5455838554939584	2
6	franklynmoon@yahoo.com	8748535454355090	8
7	franklynmoon@yahoo.com	8748535454355090	4
8	franklynmoon@yahoo.com	8748535454355090	18
9	amylucas@gmail.com	9989545499005495	2
10	amylucas@gmail.com	9989545499005495	5
11	amylucas@gmail.com	4398554454355454	19
12	rosalindaflemming@gmail.com	5734535454350990	10
13	rosalindaflemming@gmail.com	5734535454350990	18
14	marioodonnell@gmail.com	9099434554355435	20
15	marioodonnell@gmail.com	9099434554355435	9
16	marioodonnell@gmail.com	9099434554355435	15
17	courtneypage@gmail.com	5835593459409899	1
18	courtneypage@gmail.com	5835593459409899	14
19	courtneypage@gmail.com	5835593459409899	7
20	beverlymeadows@gmail.com	7878989898984355	3
21	beverlymeadows@gmail.com	7878989898984355	12
22	lesleywhite@yahoo.com	7593932453959204	11
23	lesleywhite@yahoo.com	7593932453959204	12
24	gregrobinson@yahoo.com	495039465740343	10
25	gregrobinson@yahoo.com	495039465740343	17

Studio



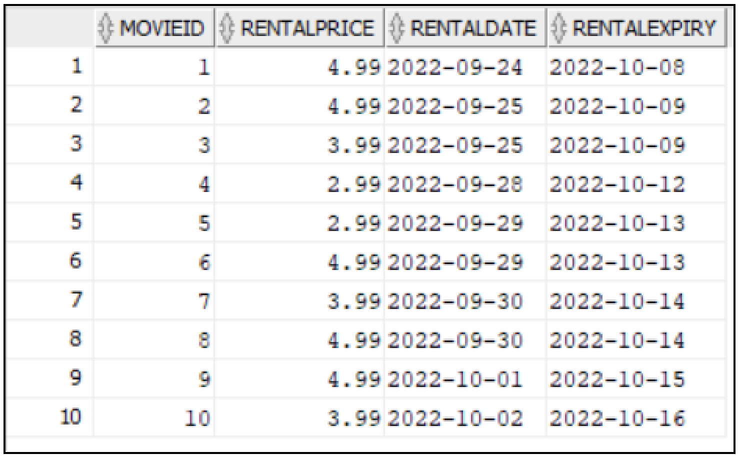
	STUDIONAME	PRODUCEDMOVIES
1	Walt Disney Pictures	The Lion King, Frozen, Tangled, Big Hero 6
2	Warner Bros	The Matrix, Harry Potter and the Philosopher's Stone
3	Paramount Pictures	Mission: Impossible, Interstellar, Top Gun: Maverick
4	Columbia Pictures	Men In Black
5	DreamWorks Animation	Madagascar, Kung Fu Panda, Shrek
6	Blue Sky Studios	Ice Age
7	20th Century Studios	Night At the Museum, Avatar, Independence Day
8	Lionsgate Films	The Hunger Games, Knives Out
9	Universal Pictures	Despicable Me

Produces



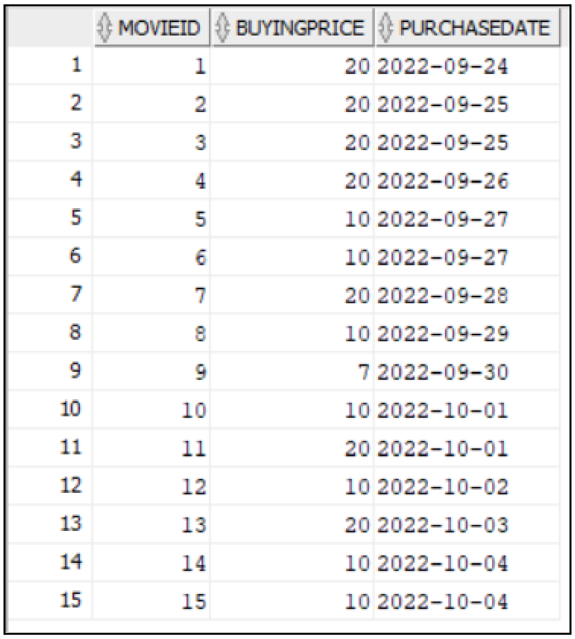
	MOVIEID	STUDIONAME
1	1	Walt Disney Pictures
2	2	Walt Disney Pictures
3	3	Walt Disney Pictures
4	4	Walt Disney Pictures
5	5	Warner Bros
6	6	Warner Bros
7	7	Paramount Pictures
8	8	Paramount Pictures
9	9	DreamWorks Animation
10	10	Blue Sky Studios
11	11	DreamWorks Animation
12	12	DreamWorks Animation
13	13	Paramount Pictures
14	14	20th Century Studios
15	15	20th Century Studios
16	16	20th Century Studios
17	17	Lionsgate Films
18	18	Lionsgate Films
19	19	Universal Pictures
20	20	Columbia Pictures

Rental



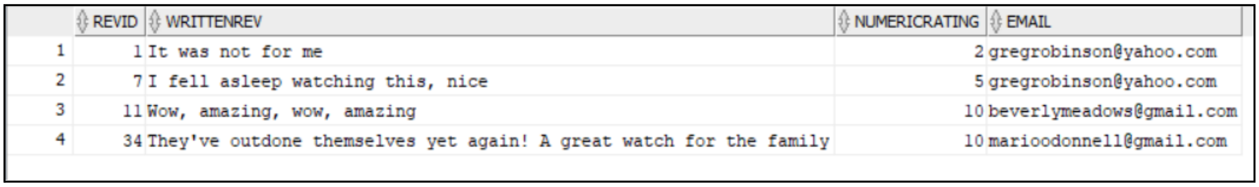
	MOVIEID	RENTALPRICE	RENTALDATE	RENTALEXPIRY
1	1	4.99	2022-09-24	2022-10-08
2	2	4.99	2022-09-25	2022-10-09
3	3	3.99	2022-09-25	2022-10-09
4	4	2.99	2022-09-28	2022-10-12
5	5	2.99	2022-09-29	2022-10-13
6	6	4.99	2022-09-29	2022-10-13
7	7	3.99	2022-09-30	2022-10-14
8	8	4.99	2022-09-30	2022-10-14
9	9	4.99	2022-10-01	2022-10-15
10	10	3.99	2022-10-02	2022-10-16

Purchase



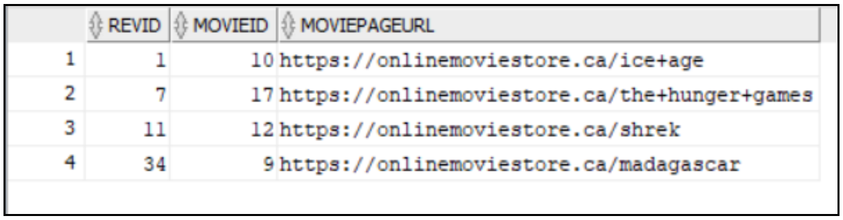
	MOVIEID	BUYINGPRICE	PURCHASEDATE
1	1	20	2022-09-24
2	2	20	2022-09-25
3	3	20	2022-09-25
4	4	20	2022-09-26
5	5	10	2022-09-27
6	6	10	2022-09-27
7	7	20	2022-09-28
8	8	10	2022-09-29
9	9	7	2022-09-30
10	10	10	2022-10-01
11	11	20	2022-10-01
12	12	10	2022-10-02
13	13	20	2022-10-03
14	14	10	2022-10-04
15	15	10	2022-10-04

Movie_review



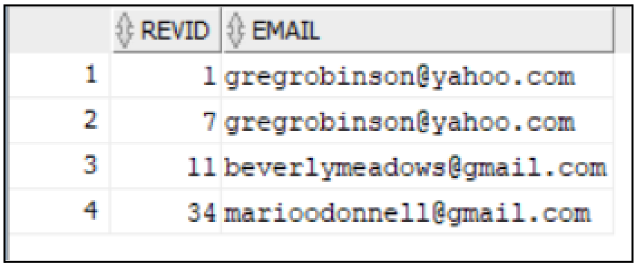
	REVID	WRITTENREV	NUMERICRATING	EMAIL
1	1	It was not for me	2	gregrobinson@yahoo.com
2	7	I fell asleep watching this, nice	5	gregrobinson@yahoo.com
3	11	Wow, amazing, wow, amazing	10	beverlymeadows@gmail.com
4	34	They've outdone themselves yet again! A great watch for the family	10	marioodonnell@gmail.com

Links_to



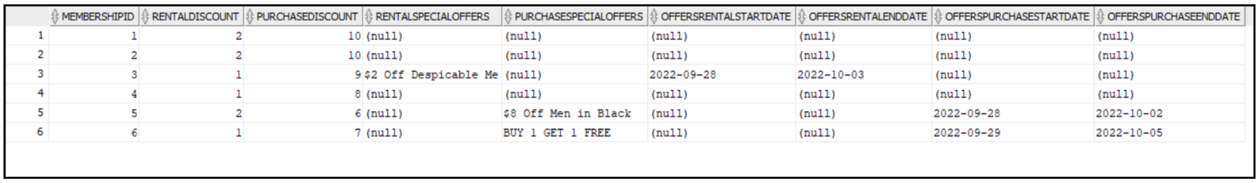
	REVID	MOVIEID	MOVIEPAGEURL
1	1	10	https://onlinemoviestore.ca/ice+age
2	7	17	https://onlinemoviestore.ca/the+hunger+games
3	11	12	https://onlinemoviestore.ca/shrek
4	34	9	https://onlinemoviestore.ca/madagascar

Writes



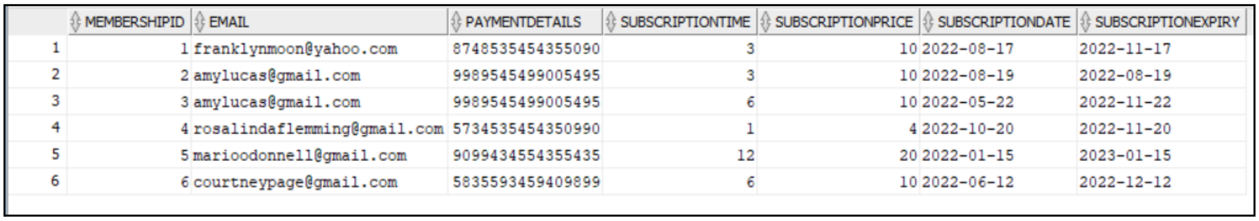
	REVID	EMAIL
1	1	gregrobinson@yahoo.com
2	7	gregrobinson@yahoo.com
3	11	beverlymeadows@gmail.com
4	34	marioodonnell@gmail.com

Fanclub_membership



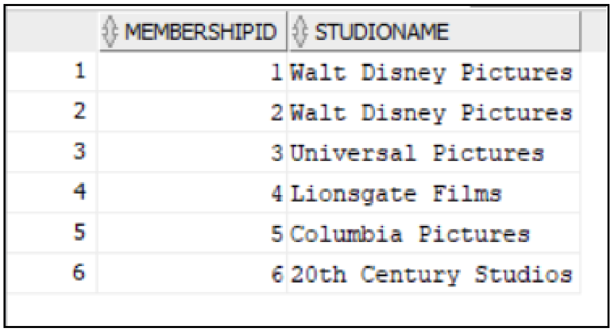
	MEMBERSHIPID	RENTALDISCOUNT	PURCHASEDISCOUNT	RENTALSPECIALOFFERS	PURCHASESPECIALOFFERS	OFFERSRENTALSTARTDATE	OFFERSRENTALENDDATE	OFFERSPURCHASESTARTDATE	OFFERSPURCHASEENDDATE
1	1	2	10	(null)	(null)	(null)	(null)	(null)	(null)
2	2	2	10	(null)	(null)	(null)	(null)	(null)	(null)
3	3	1	9	$2 Off Despicable Me	(null)	2022-09-28	2022-10-03	(null)	(null)
4	4	1	8	(null)	(null)	(null)	(null)	(null)	(null)
5	5	2	6	(null)	$8 Off Men in Black	(null)	(null)	2022-09-28	2022-10-02
6	6	1	7	(null)	BUY 1 GET 1 FREE	(null)	(null)	2022-09-29	2022-10-05

Subscribes_to



	MEMBERSHIPID	EMAIL	PAYMENTDETAILS	SUBSCRIPTIONTIME	SUBSCRIPTIONPRICE	SUBSCRIPTIONDATE	SUBSCRIPTIONEXPIRY
1	1	franklynmoon@yahoo.com	8748535454355090	3	10	2022-08-17	2022-11-17
2	2	amylucas@gmail.com	9989545499005495	3	10	2022-08-19	2022-08-19
3	3	amylucas@gmail.com	9989545499005495	6	10	2022-05-22	2022-11-22
4	4	rosalindaflemming@gmail.com	5734535454350990	1	4	2022-10-20	2022-11-20
5	5	marioodonnell@gmail.com	9099434554355435	12	20	2022-01-15	2023-01-15
6	6	courtneypage@gmail.com	5835593459409899	6	10	2022-06-12	2022-12-12

Has



	MEMBERSHIPID	STUDIONAME
1	1	Walt Disney Pictures
2	2	Walt Disney Pictures
3	3	Universal Pictures
4	4	Lionsgate Films
5	5	Columbia Pictures
6	6	20th Century Studios

Assignment 4 - Designing Views and Simple Queries

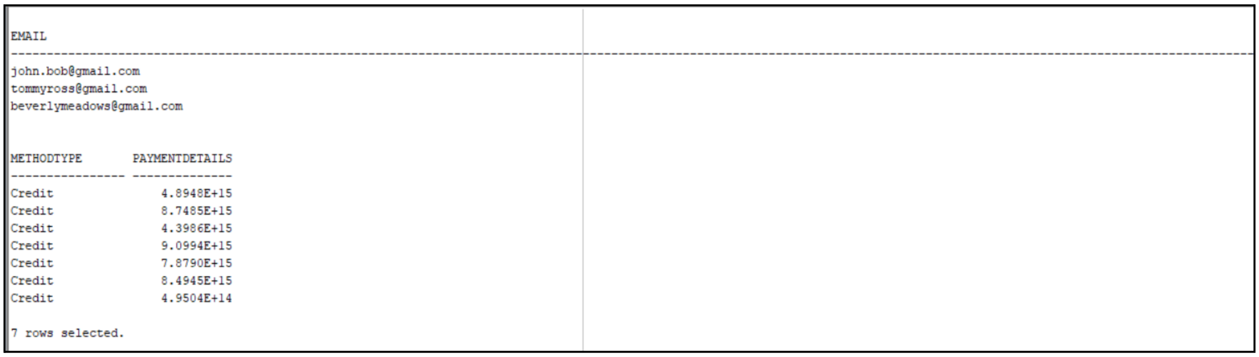
Part 1

Simple Queries - SQL Code

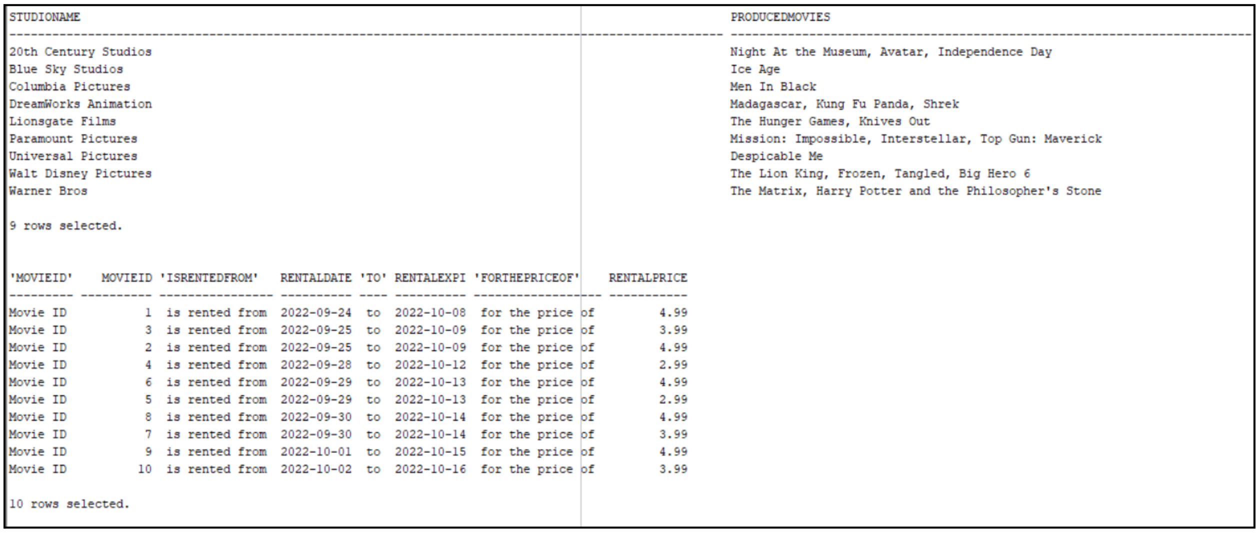
```
------- query for user_acc
SELECT Email,FirstName, LastName, DateOfBirth
FROM user_acc
WHERE DateOfBirth LIKE '%6%';

------- query for payment_method
SELECT MethodType, PaymentDetails
FROM payment_method
WHERE MethodType = 'Credit';

-------- query for studio
SELECT DISTINCT *
FROM studio
ORDER BY StudioName;

--------- query for rental
SELECT 'Movie ID ', MovieID, ' is rented from ', RentalDate, ' to ', RentalExpiry, ' for the price of
', RentalPrice
FROM rental
ORDER BY RentalDate ASC;

-------- query for purchase
SELECT *
FROM purchase
ORDER BY BuyingPrice DESC, PurchaseDate ASC;

-------- queries for movie
SELECT MovieName, ReleaseYear, Runtime
FROM movie
ORDER BY ReleaseYear DESC;

SELECT MovieName, Genre, Resolution, Runtime, MovieRating, ReleaseYear
FROM movie
WHERE Genre = 'Animated, Family'
ORDER BY ReleaseYear DESC;
```

```
----queries for payment details
SELECT PaymentDetails
FROM uses
WHERE Email LIKE '%gmail.com';

----queries for orders
SELECT Email, PaymentDetails, MovieID
FROM orders
WHERE Email = 'gregrobinson@yahoo.com';

---queries for produces
SELECT MovieID
FROM produces
WHERE StudioName = 'Walt Disney Pictures';

---queries fpr  movie_review
SELECT WrittenRev, NumericRating
FROM movie_review
WHERE Email = 'gregrobinson@yahoo.com';

---queries for links_to
SELECT MoviePageUrl
FROM links_to;

--queries for writes
SELECT Email
FROM writes
WHERE RevID = 34;

---queries for fanclub membership
SELECT *
FROM Fanclub_membership;

----queries for subscribes_to
SELECT *
FROM subscribes_to;

----queries for has
SELECT *
FROM has;
```

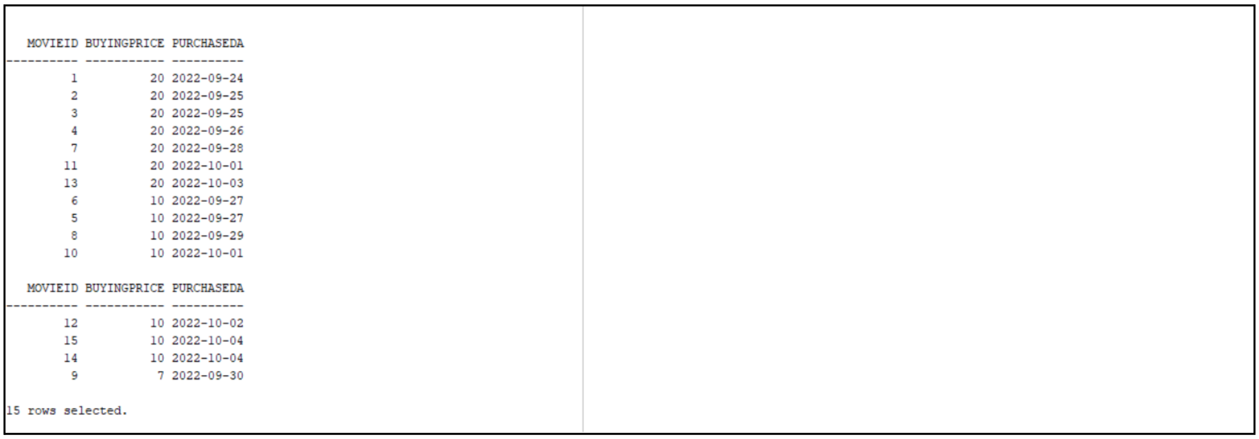
Output for Simple SQL Queries



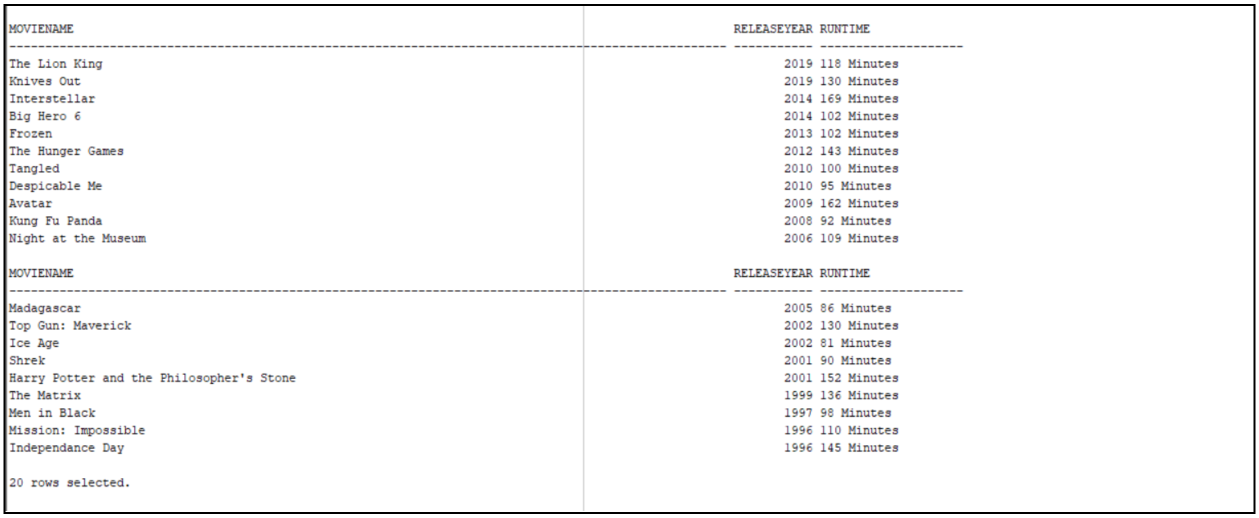
```
EMAIL
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------
john.bob@gmail.com
tommyross@gmail.com
beverlymeadows@gmail.com


METHODTYPE       PAYMENTDETAILS
---------------- --------------
Credit               4.8948E+15
Credit               8.7485E+15
Credit               4.3986E+15
Credit               9.0994E+15
Credit               7.8790E+15
Credit               8.4945E+15
Credit               4.9504E+14

7 rows selected.
```



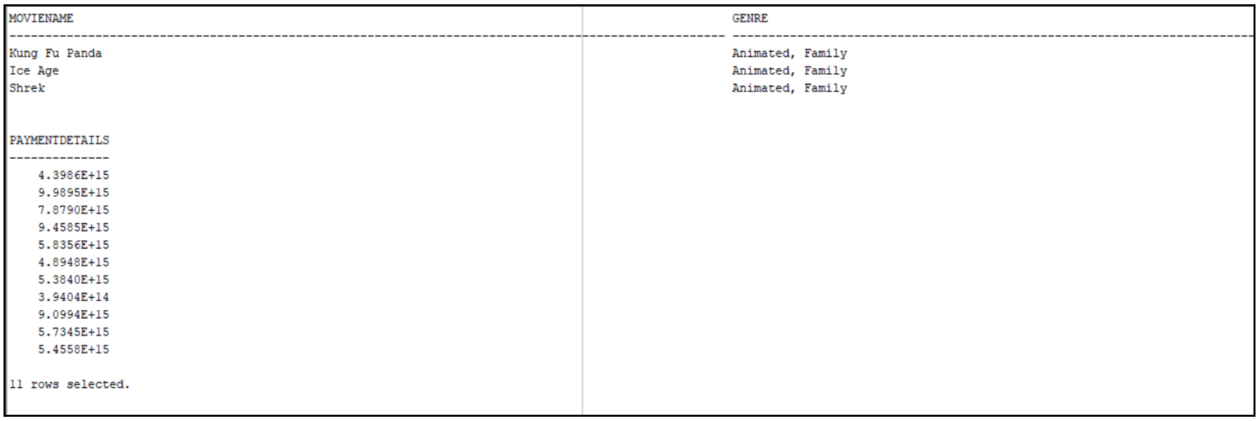
```
STUDIONAME                                                                                           PRODUCEDMOVIES
---------------------------------------------------------------------------------------------------- ---------------------------------------------------------------------------
20th Century Studios                                                                                 Night At the Museum, Avatar, Independence Day
Blue Sky Studios                                                                                     Ice Age
Columbia Pictures                                                                                    Men In Black
DreamWorks Animation                                                                                 Madagascar, Kung Fu Panda, Shrek
Lionsgate Films                                                                                      The Hunger Games, Knives Out
Paramount Pictures                                                                                   Mission: Impossible, Interstellar, Top Gun: Maverick
Universal Pictures                                                                                   Despicable Me
Walt Disney Pictures                                                                                 The Lion King, Frozen, Tangled, Big Hero 6
Warner Bros                                                                                          The Matrix, Harry Potter and the Philosopher's Stone

9 rows selected.


'MOVIEID'    MOVIEID 'ISRENTEDFROM'   RENTALDATE 'TO' RENTALEXPI 'FORTHEPRICEOF'    RENTALPRICE
--------- ---------- ---------------- ---------- ---- ---------- ------------------ -----------
Movie ID           1  is rented from  2022-09-24  to  2022-10-08  for the price of        4.99
Movie ID           3  is rented from  2022-09-25  to  2022-10-09  for the price of        3.99
Movie ID           2  is rented from  2022-09-25  to  2022-10-09  for the price of        4.99
Movie ID           4  is rented from  2022-09-28  to  2022-10-12  for the price of        2.99
Movie ID           6  is rented from  2022-09-29  to  2022-10-13  for the price of        4.99
Movie ID           5  is rented from  2022-09-29  to  2022-10-13  for the price of        2.99
Movie ID           8  is rented from  2022-09-30  to  2022-10-14  for the price of        4.99
Movie ID           7  is rented from  2022-09-30  to  2022-10-14  for the price of        3.99
Movie ID           9  is rented from  2022-10-01  to  2022-10-15  for the price of        4.99
Movie ID          10  is rented from  2022-10-02  to  2022-10-16  for the price of        3.99

10 rows selected.
```



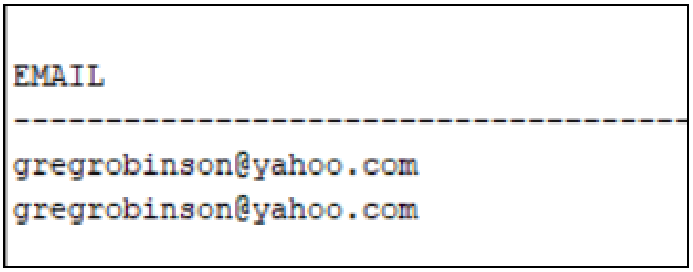
```
   MOVIEID BUYINGPRICE PURCHASEDA
---------- ----------- ----------
         1          20 2022-09-24
         2          20 2022-09-25
         3          20 2022-09-25
         4          20 2022-09-26
         7          20 2022-09-28
        11          20 2022-10-01
        13          20 2022-10-03
         6          10 2022-09-27
         5          10 2022-09-27
         8          10 2022-09-29
        10          10 2022-10-01

   MOVIEID BUYINGPRICE PURCHASEDA
---------- ----------- ----------
        12          10 2022-10-02
        15          10 2022-10-04
        14          10 2022-10-04
         9           7 2022-09-30

15 rows selected.
```



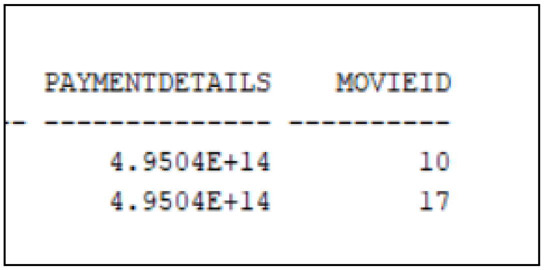
```
MOVIENAME                                                                                             RELEASEYEAR RUNTIME
----------------------------------------------------------------------------------------------------- ----------- --------------------
The Lion King                                                                                                2019 118 Minutes
Knives Out                                                                                                   2019 130 Minutes
Interstellar                                                                                                 2014 169 Minutes
Big Hero 6                                                                                                   2014 102 Minutes
Frozen                                                                                                       2013 102 Minutes
The Hunger Games                                                                                             2012 143 Minutes
Tangled                                                                                                      2010 100 Minutes
Despicable Me                                                                                                2010 95 Minutes
Avatar                                                                                                       2009 162 Minutes
Kung Fu Panda                                                                                                2008 92 Minutes
Night at the Museum                                                                                          2006 109 Minutes

MOVIENAME                                                                                             RELEASEYEAR RUNTIME
----------------------------------------------------------------------------------------------------- ----------- --------------------
Madagascar                                                                                                   2005 86 Minutes
Top Gun: Maverick                                                                                            2002 130 Minutes
Ice Age                                                                                                      2002 81 Minutes
Shrek                                                                                                        2001 90 Minutes
Harry Potter and the Philosopher's Stone                                                                     2001 152 Minutes
The Matrix                                                                                                   1999 136 Minutes
Men in Black                                                                                                 1997 98 Minutes
Mission: Impossible                                                                                          1996 110 Minutes
Independance Day                                                                                             1996 145 Minutes

20 rows selected.
```



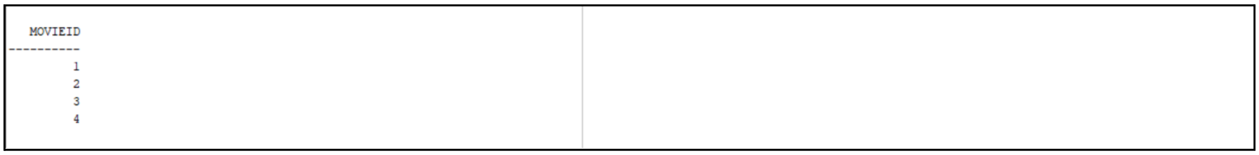
```
MOVIENAME                                                                                             GENRE
----------------------------------------------------------------------------------------------------- ------------------------------------------------------------------------------------
Kung Fu Panda                                                                                         Animated, Family
Ice Age                                                                                               Animated, Family
Shrek                                                                                                 Animated, Family


PAYMENTDETAILS
--------------
    4.3986E+15
    9.9895E+15
    7.8790E+15
    9.4585E+15
    5.8356E+15
    4.8948E+15
    5.3840E+15
    3.9404E+14
    9.0994E+15
    5.7345E+15
    5.4558E+15

11 rows selected.
```



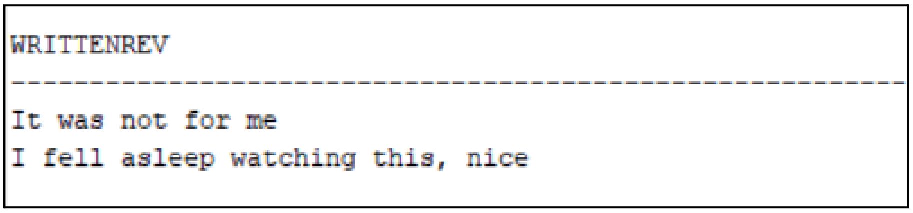
```
EMAIL
----------------------------------------------
gregrobinson@yahoo.com
gregrobinson@yahoo.com
```



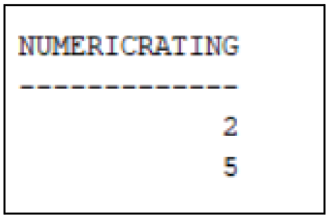
```
 PAYMENTDETAILS    MOVIEID
-- -------------- ----------
      4.9504E+14         10
      4.9504E+14         17
```



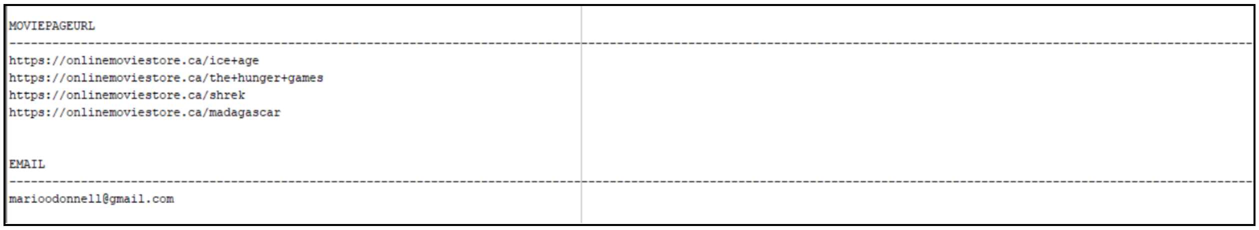
```
   MOVIEID
----------
         1
         2
         3
         4
```



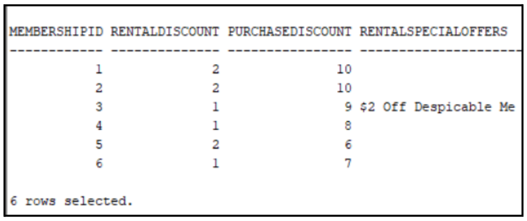
```
WRITTENREV
----------------------------------------------------------------
It was not for me
I fell asleep watching this, nice
```



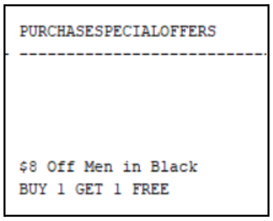
```
NUMERICRATING
-------------
            2
            5
```



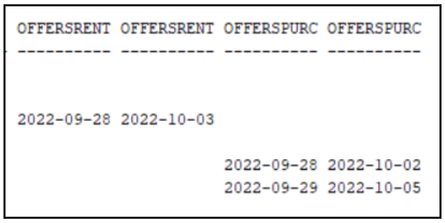
```
MOVIEPAGEURL
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
https://onlinemoviestore.ca/ice+age
https://onlinemoviestore.ca/the+hunger+games
https://onlinemoviestore.ca/shrek
https://onlinemoviestore.ca/madagascar


EMAIL
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
marioodonnell@gmail.com
```



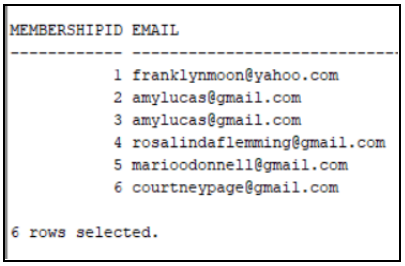
```
MEMBERSHIPID RENTALDISCOUNT PURCHASEDISCOUNT RENTALSPECIALOFFERS
------------ -------------- ---------------- ----------------------
           1              2               10
           2              2               10
           3              1                9 $2 Off Despicable Me
           4              1                8
           5              2                6
           6              1                7

6 rows selected.
```



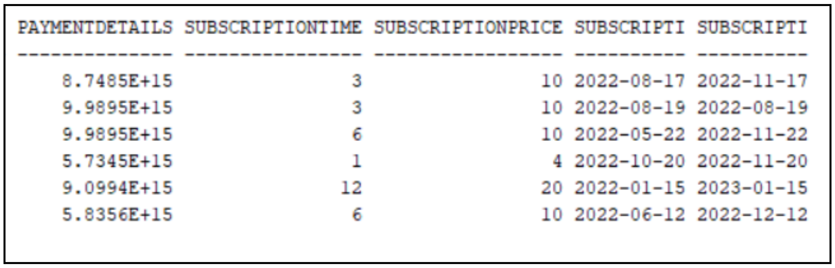
```
PURCHASESPECIALOFFERS
---------------------------



$8 Off Men in Black
BUY 1 GET 1 FREE
```



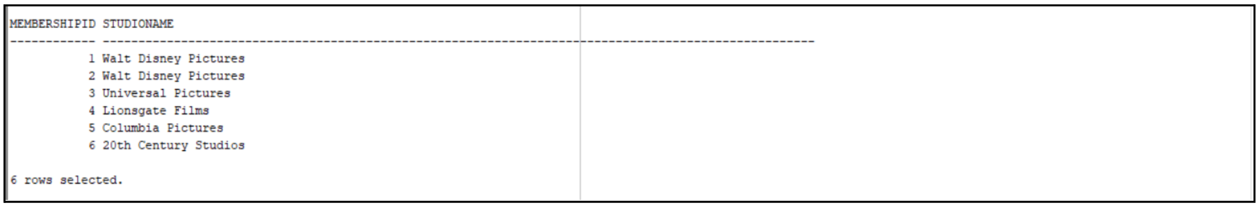
```
OFFERSRENT OFFERSRENT OFFERSPURC OFFERSPURC
---------- ---------- ---------- ----------


2022-09-28 2022-10-03

                      2022-09-28 2022-10-02
                      2022-09-29 2022-10-05
```



```
MEMBERSHIPID EMAIL
------------ ------------------------------
           1 franklynmoon@yahoo.com
           2 amylucas@gmail.com
           3 amylucas@gmail.com
           4 rosalindaflemming@gmail.com
           5 marioodonnell@gmail.com
           6 courtneypage@gmail.com

6 rows selected.
```



```
PAYMENTDETAILS SUBSCRIPTIONTIME SUBSCRIPTIONPRICE SUBSCRIPTI SUBSCRIPTI
-------------- ---------------- ----------------- ---------- ----------
    8.7485E+15                3                10 2022-08-17 2022-11-17
    9.9895E+15                3                10 2022-08-19 2022-08-19
    9.9895E+15                6                10 2022-05-22 2022-11-22
    5.7345E+15                1                 4 2022-10-20 2022-11-20
    9.0994E+15               12                20 2022-01-15 2023-01-15
    5.8356E+15                6                10 2022-06-12 2022-12-12
```



```
MEMBERSHIPID STUDIONAME
------------ ----------------------------------------------------------------------------------------------------
           1 Walt Disney Pictures
           2 Walt Disney Pictures
           3 Universal Pictures
           4 Lionsgate Films
           5 Columbia Pictures
           6 20th Century Studios

6 rows selected.
```

Part 2

Create Views - SQL Code

CREATE VIEW for_the_family AS (SELECT moviename FROM movie WHERE genre LIKE '%Family%');

CREATE VIEW avg_user_scores AS (SELECT l.movieid, AVG(numericrating) AS Avg_User_rating FROM movie_review m, links_to l WHERE l.revid = m.revid GROUP BY movieid);

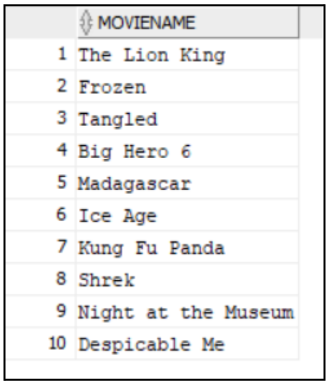
CREATE VIEW studios_and_movies AS (SELECT m.moviename, p.studioname FROM movie m, produces p WHERE m.movieid = p.movieid);

Drop Views - SQL Code

```
DROP VIEW for_the_family;
DROP VIEW avg_user_scores;
DROP VIEW studios_and_movies;
```

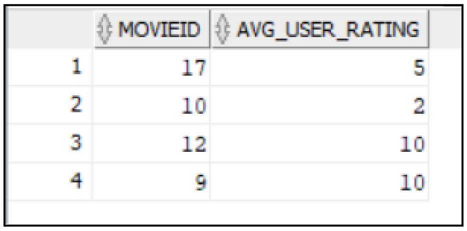
Screenshots of View Tables

For_the_family



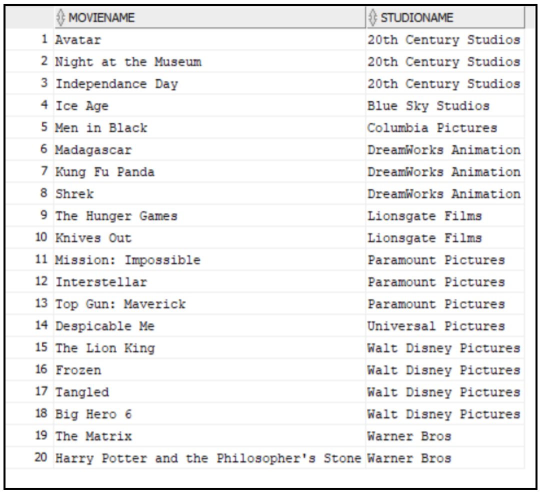
	MOVIENAME
1	The Lion King
2	Frozen
3	Tangled
4	Big Hero 6
5	Madagascar
6	Ice Age
7	Kung Fu Panda
8	Shrek
9	Night at the Museum
10	Despicable Me

Avg_user_scores



	MOVIEID	AVG_USER_RATING
1	17	5
2	10	2
3	12	10
4	9	10

Studios_and_movies

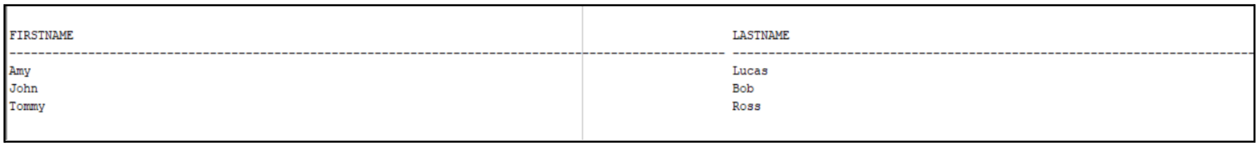


	MOVIENAME	STUDIONAME
1	Avatar	20th Century Studios
2	Night at the Museum	20th Century Studios
3	Independance Day	20th Century Studios
4	Ice Age	Blue Sky Studios
5	Men in Black	Columbia Pictures
6	Madagascar	DreamWorks Animation
7	Kung Fu Panda	DreamWorks Animation
8	Shrek	DreamWorks Animation
9	The Hunger Games	Lionsgate Films
10	Knives Out	Lionsgate Films
11	Mission: Impossible	Paramount Pictures
12	Interstellar	Paramount Pictures
13	Top Gun: Maverick	Paramount Pictures
14	Despicable Me	Universal Pictures
15	The Lion King	Walt Disney Pictures
16	Frozen	Walt Disney Pictures
17	Tangled	Walt Disney Pictures
18	Big Hero 6	Walt Disney Pictures
19	The Matrix	Warner Bros
20	Harry Potter and the Philosopher's Stone	Warner Bros

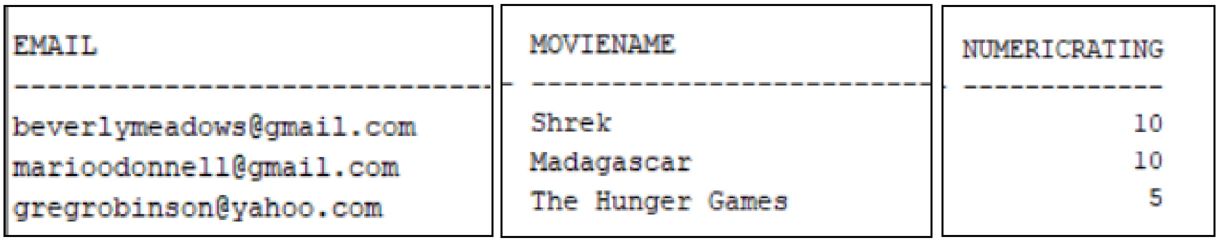
Join Queries - SQL Code

```
--List Names of people who have ordered the movie 'Frozen'
SELECT FirstName, LastName
FROM orders o, user_acc u
WHERE MovieID=2 AND u.Email = o.Email
ORDER BY FirstName ASC;

--List Emails, Movie Name and Movie Rating of people who gave movies a rating of more than or equal to 5.
SELECT u.Email, m.MovieName, mr.NumericRating
FROM user_acc u, writes w, movie_review mr, links_to lt, movie m
WHERE mr.NumericRating >=5 AND u.Email = w.Email AND w.RevID = mr.RevID AND mr.RevID = lt.RevID AND lt.MovieID = m.MovieID
ORDER BY mr.NumericRating DESC;

--List the Emails and Names of people who have subscribed to the Studio 'Walt Disney Animation'
SELECT u.Email, u.FirstName, u.LastName
FROM user_acc u, subscribes_to st, Fanclub_membership fm, has h
WHERE h.StudioName = 'Walt Disney Pictures' AND u.Email = st.Email AND st.MembershipID = fm.MembershipID AND fm.MembershipID = h.MembershipID
ORDER BY Email ASC;
```

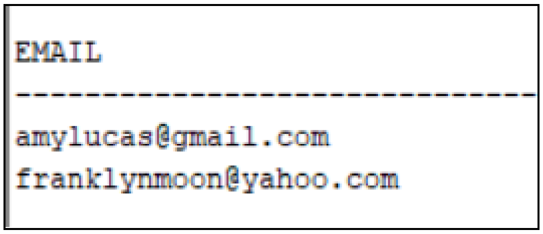
Output for Join Queries



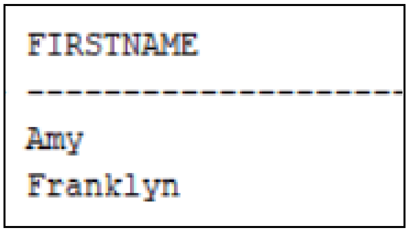
```
FIRSTNAME                                   LASTNAME
-----------------------------------------   ------------------------------------------
Amy                                         Lucas
John                                        Bob
Tommy                                       Ross
```



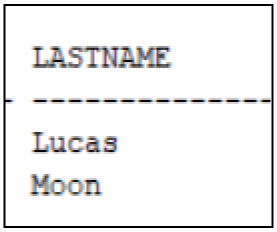
```
EMAIL                          MOVIENAME                  NUMERICRATING
------------------------------ -------------------------- -------------
beverlymeadows@gmail.com       Shrek                                 10
mariodonnell@gmail.com         Madagascar                            10
gregrobinson@yahoo.com         The Hunger Games                       5
```



```
EMAIL
------------------------------
amylucas@gmail.com
franklynmoon@yahoo.com
```



```
FIRSTNAME
--------------------
Amy
Franklyn
```



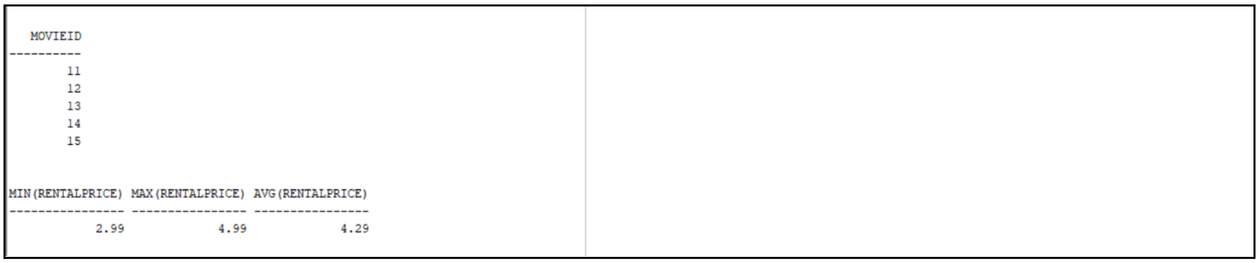
```
LASTNAME
-------------
Lucas
Moon
```

Assignment 5 - Demonstration of Advanced Queries by Unix shell Implementation

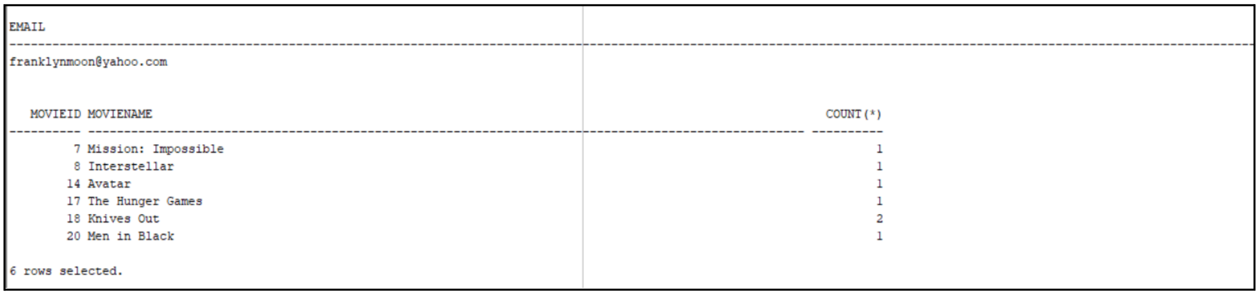
Advanced Queries - SQL Code

```
--Advanced Queries
--List MovieID, MovieName which can be purchased but not rented.
SELECT m.MovieID
FROM movie m, purchase p
WHERE m.MovieID = p.MovieID AND m.MovieID NOT IN (SELECT  r.MovieID FROM rental r);

--List maximum, minimum, average price for renting a movie.
SELECT MIN(rentalprice), MAX(rentalprice), AVG(rentalprice)
FROM rental;

--List user info whose payment method is only credit.
SELECT u.email, u.firstname, u.lastname
FROM user_acc u
WHERE NOT EXISTS (SELECT u.email, u.firstname, u.lastname FROM payment_method p,
uses s WHERE u.email = s.email AND s.email = p.email AND p.methodtype = 'Debit');

--List movieID and name for PG-13 movies along with number of times it has been ordered.
SELECT o.movieID, m.movieName, count(*)
FROM orders o, movie m WHERE o.movieId = m.movieID AND m.movieRating = 'PG-13'
GROUP BY o.movieID, m.movieName ORDER BY movieID;

--List of movies produced by Walt Disney Pictures or have a resolution of 4K HDR.
SELECT m.movieName
FROM movie m
WHERE EXISTS (SELECT m.movieName FROM produces p WHERE m.movieId = p.movieId
AND p.studioName = 'Walt Disney Pictures')
UNION (SELECT m.movieName FROM movie m WHERE m.resolution = '4K HDR');
```

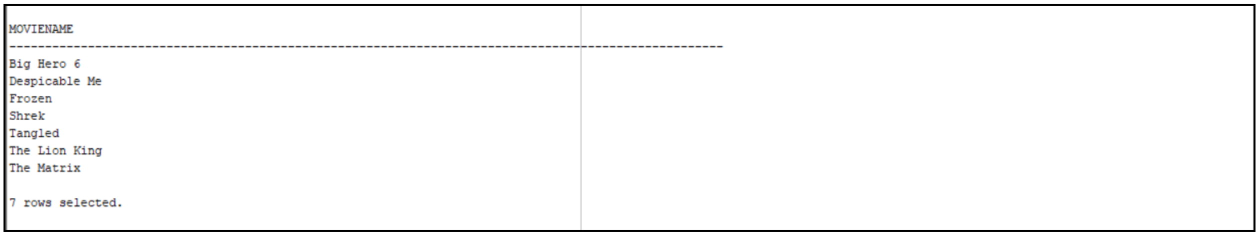
Output for Advanced Queries



```
   MOVIEID
----------
        11
        12
        13
        14
        15


MIN(RENTALPRICE) MAX(RENTALPRICE) AVG(RENTALPRICE)
---------------- ---------------- ----------------
            2.99             4.99             4.29
```

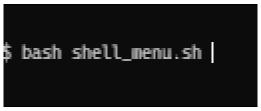


```
EMAIL
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
franklynmoon@yahoo.com


   MOVIEID MOVIENAME                                                                                              COUNT(*)
---------- ---------------------------------------------------------------------------------------------------- ----------
         7 Mission: Impossible                                                                                           1
         8 Interstellar                                                                                                  1
        14 Avatar                                                                                                        1
        17 The Hunger Games                                                                                              1
        18 Knives Out                                                                                                    2
        20 Men in Black                                                                                                  1

6 rows selected.
```

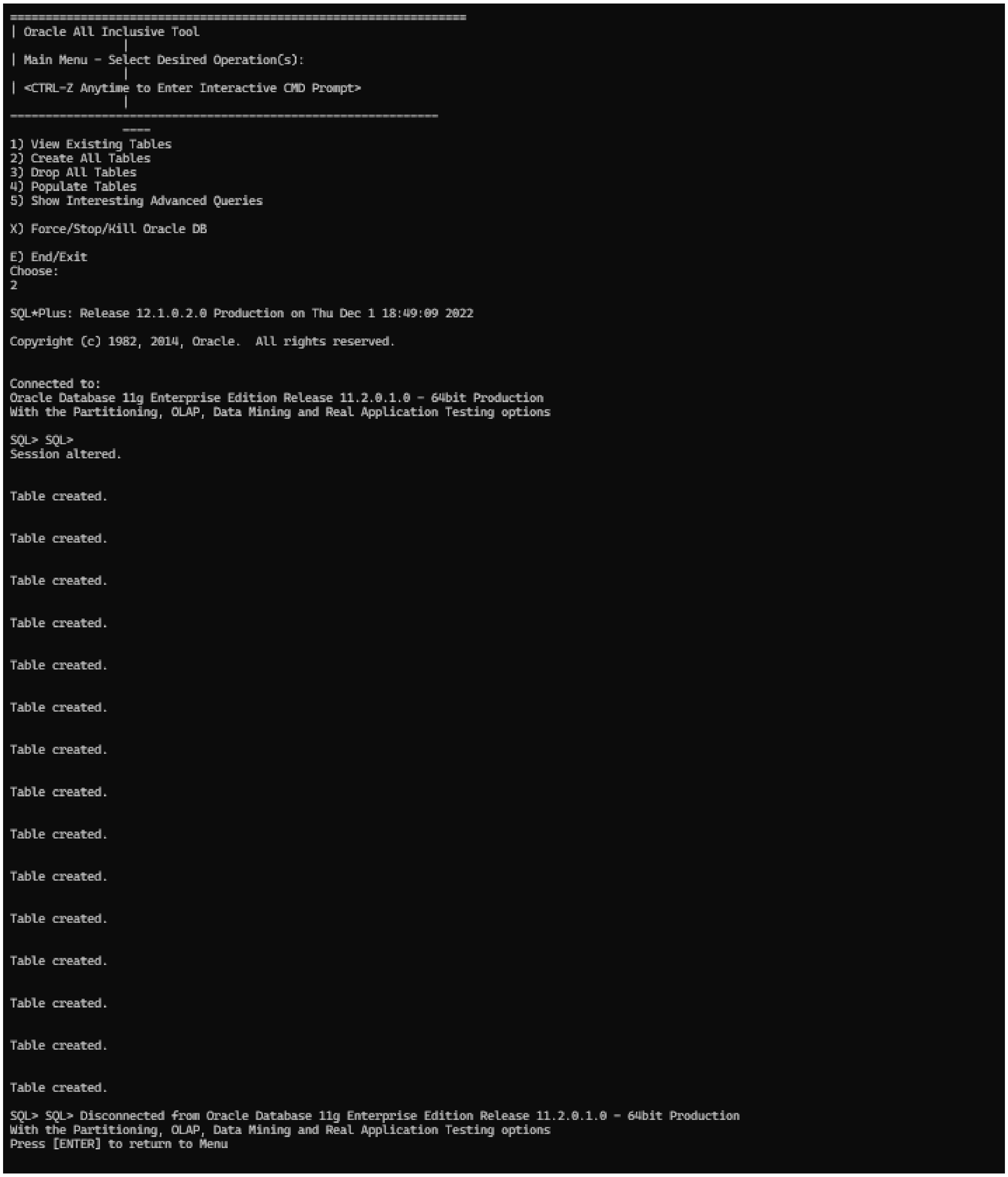


```
MOVIENAME
----------------------------------------------------------------------------------------------------
Big Hero 6
Despicable Me
Frozen
Shrek
Tangled
The Lion King
The Matrix

7 rows selected.
```

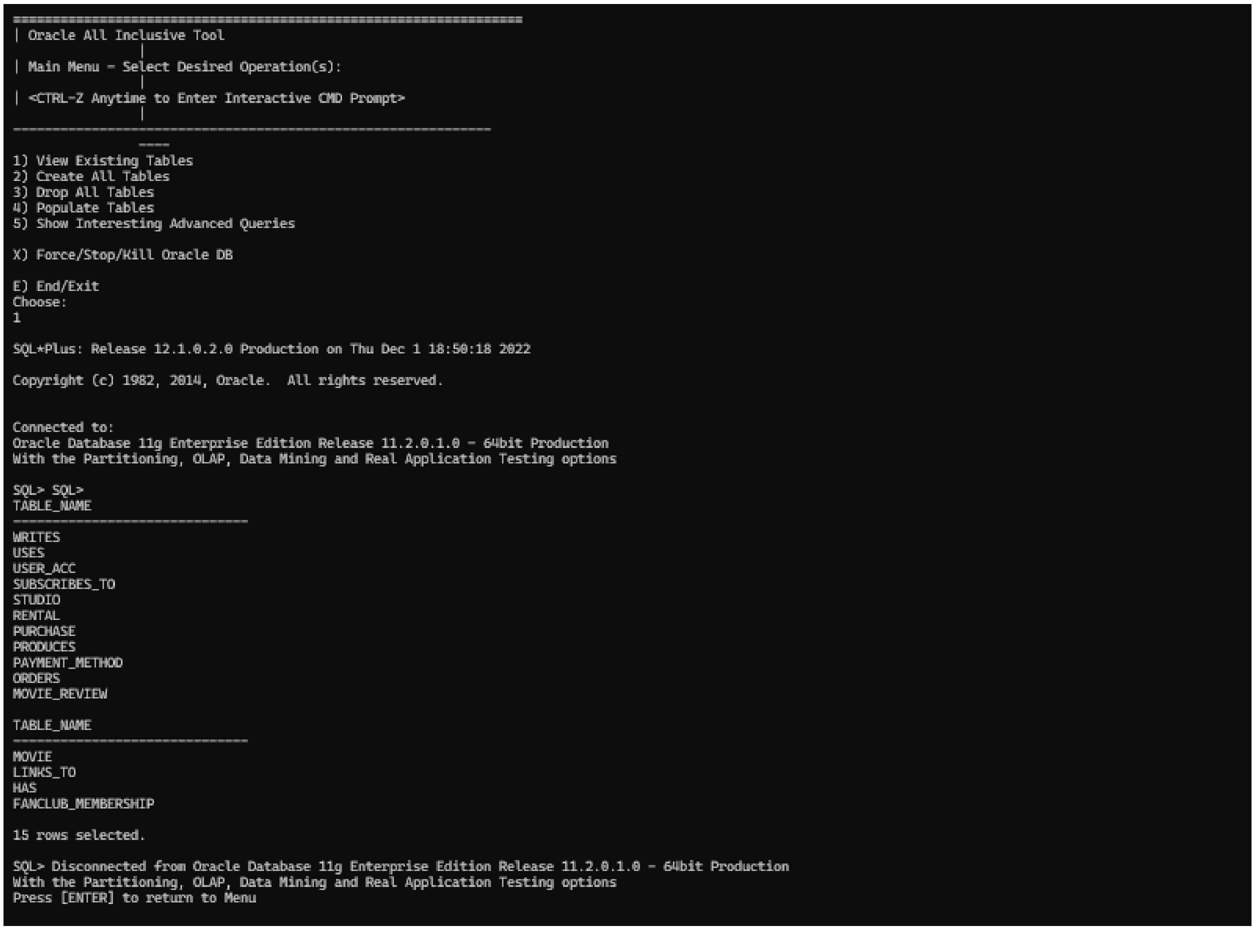
Shell Menu Demo



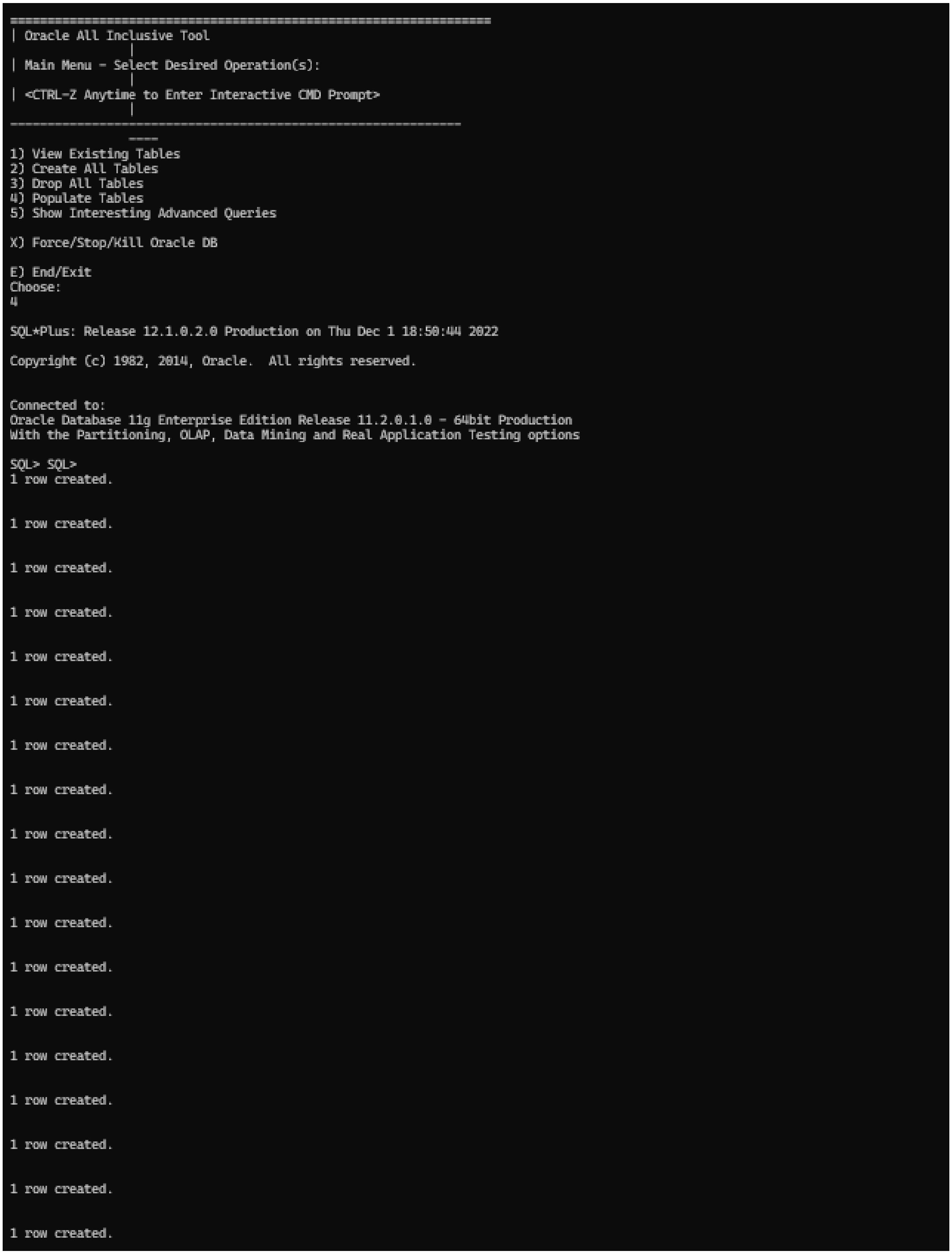
```
$ bash shell_menu.sh |
```



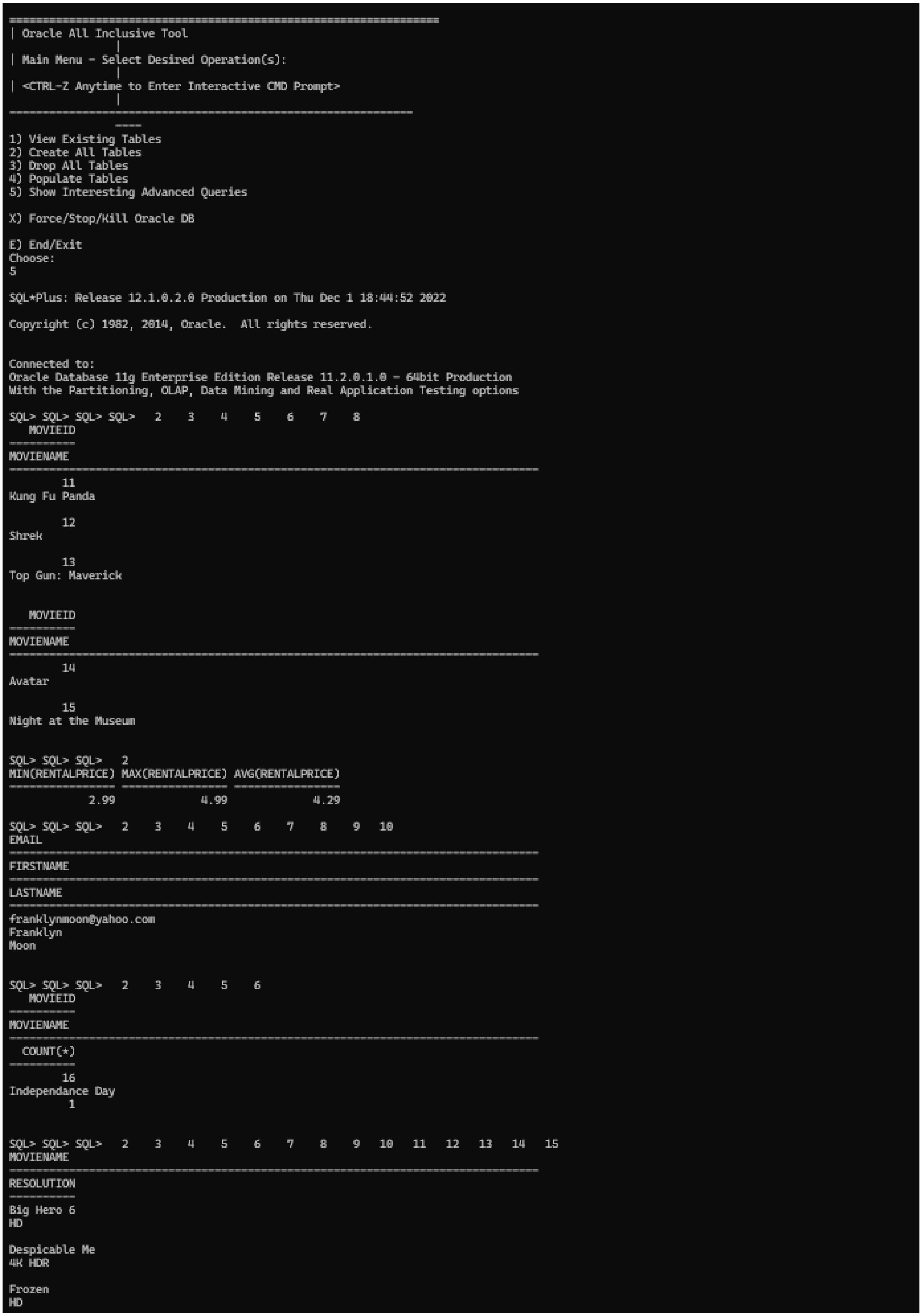
```
==============================================================
| Oracle All Inclusive Tool
                |
| Main Menu - Select Desired Operation(s):
                |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |
-----------------------------------------------------------
               ----
1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries

X) Force/Stop/Kill Oracle DB

E) End/Exit
Choose:
2

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 1 18:49:09 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>
Session altered.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.


Table created.

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
Press [ENTER] to return to Menu
```



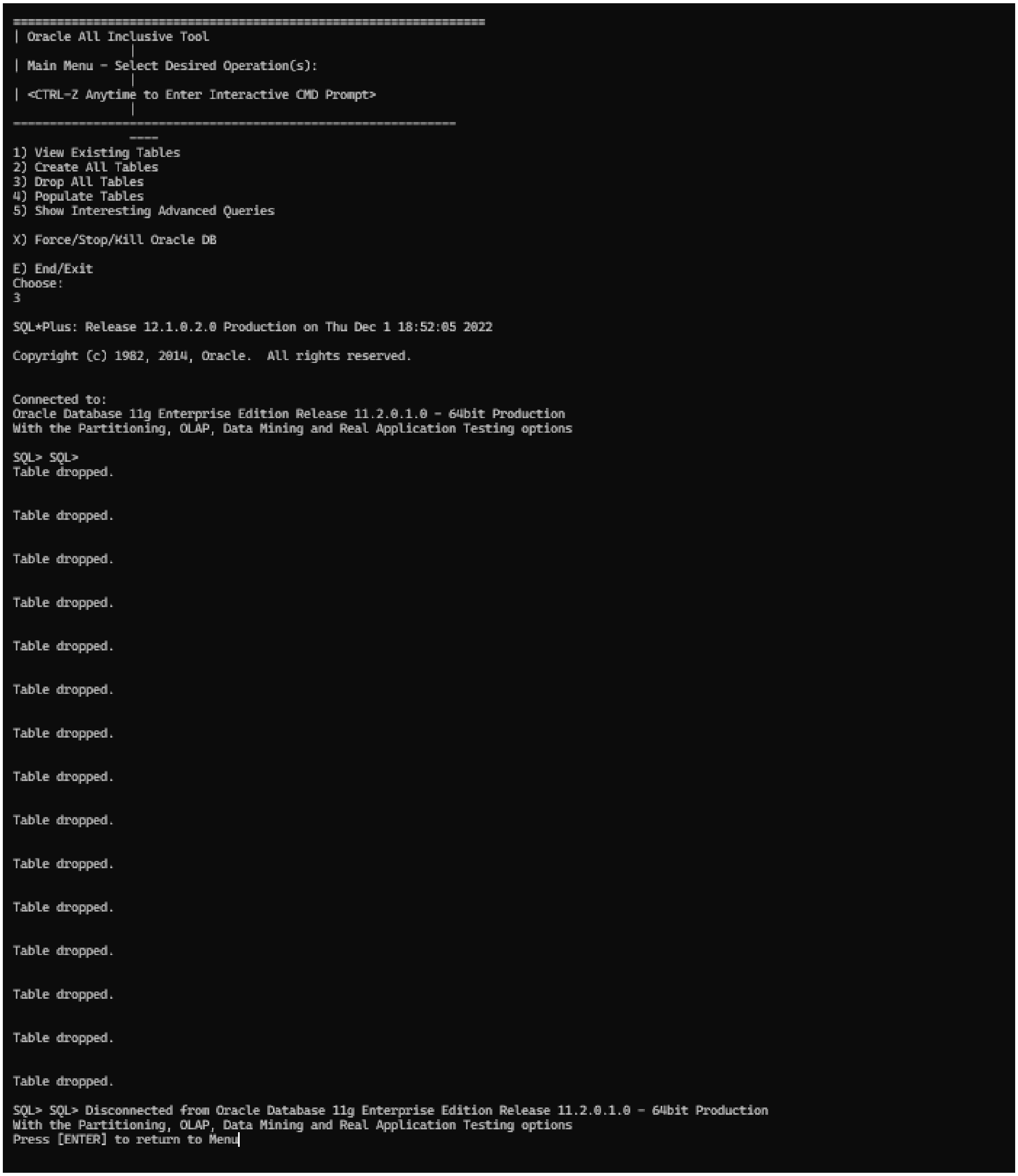
```
=================================================================
| Oracle All Inclusive Tool
                |
| Main Menu - Select Desired Operation(s):
                |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |
-------------------------------------------------------------
                ----
1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries

X) Force/Stop/Kill Oracle DB

E) End/Exit
Choose:
1

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 1 18:50:18 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_NAME
------------------------------
WRITES
USES
USER_ACC
SUBSCRIBES_TO
STUDIO
RENTAL
PURCHASE
PRODUCES
PAYMENT_METHOD
ORDERS
MOVIE_REVIEW

TABLE_NAME
------------------------------
MOVIE
LINKS_TO
HAS
FANCLUB_MEMBERSHIP

15 rows selected.

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
Press [ENTER] to return to Menu
```



```
=================================================================
| Oracle All Inclusive Tool
                |
| Main Menu - Select Desired Operation(s):
                |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |
-------------------------------------------------------------
                ----
1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries

X) Force/Stop/Kill Oracle DB

E) End/Exit
Choose:
4

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 1 18:50:44 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>
1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.
```



```
=================================================================
| Oracle All Inclusive Tool
                |
| Main Menu - Select Desired Operation(s):
                |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |
-------------------------------------------------------------
                ----
1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries

X) Force/Stop/Kill Oracle DB

E) End/Exit
Choose:
5

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 1 18:44:52 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    7    8
   MOVIEID
----------
MOVIENAME
--------------------------------------------------------------------------------
        11
Kung Fu Panda

        12
Shrek

        13
Top Gun: Maverick


   MOVIEID
----------
MOVIENAME
--------------------------------------------------------------------------------
        14
Avatar

        15
Night at the Museum


SQL> SQL> SQL>   2
MIN(RENTALPRICE) MAX(RENTALPRICE) AVG(RENTALPRICE)
---------------- ---------------- ----------------
            2.99             4.99             4.29

SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10
EMAIL
--------------------------------------------------------------------------------
FIRSTNAME
--------------------------------------------------------------------------------
LASTNAME
--------------------------------------------------------------------------------
franklynmoon@yahoo.com
Franklyn
Moon


SQL> SQL> SQL>   2    3    4    5    6
   MOVIEID
----------
MOVIENAME
--------------------------------------------------------------------------------
  COUNT(*)
----------
        16
Independance Day
         1


SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13   14   15
MOVIENAME
--------------------------------------------------------------------------------
RESOLUTION
----------
Big Hero 6
HD

Despicable Me
4K HDR

Frozen
HD
```



```
=================================================================
| Oracle All Inclusive Tool
                |
| Main Menu - Select Desired Operation(s):
                |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |
-------------------------------------------------------------
                ----
1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries

X) Force/Stop/Kill Oracle DB

E) End/Exit
Choose:
3

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 1 18:52:05 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.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


Table dropped.


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
Press [ENTER] to return to Menu
```

Shell Menu Code

“create_tables.sh”

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
username=$(head -1 "credentials")
password=$(tail -1 "credentials")
sqlplus64
"$username/$password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca) (Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

@../sql/createTables;

exit;

EOF
```

“drop_tables.sh”

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
username=$(head -1 "credentials")
password=$(tail -1 "credentials")
sqlplus64
"$username/$password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca) (Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

@../sql/dropTables;

exit;


EOF
```

“interesting_advqueries.sh”

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
username=$(head -1 "credentials")
password=$(tail -1 "credentials")
stuff=$(tail -53 "../sql/queries.sql")
sqlplus64
"$username/$password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca) (Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
```

```
$stuff

exit;


EOF
```

“populate_tables.sh”

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
username=$(head -1 "credentials")
password=$(tail -1 "credentials")
sqlplus64
"$username/$password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ry
erson.ca) (Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

@../sql/insertMovies;
@../sql/insert;

exit;


EOF
```

“shell_menu.sh”

```
#!/bin/sh
MainMenu()
{
        while [ "$CHOICE" != "START" ]
        do
                clear
                echo
"======================================================================"
                echo "| Oracle All Inclusive Tool
                |"
                echo "| Main Menu - Select Desired Operation(s):
                |"
                echo "| <CTRL-Z Anytime to Enter Interactive CMD Prompt>
                |"
                echo "------------------------------------------------------------------
                ----"
                echo "1) View Existing Tables"
                echo "2) Create All Tables"
                echo "3) Drop All Tables"
        echo "4) Populate Tables"
        echo "5) Show Interesting Advanced Queries"
        echo " "
        echo "X) Force/Stop/Kill Oracle DB"
        echo " "
        echo "E) End/Exit"
        echo "Choose: "
        read CHOICE
        if [ "$CHOICE" == "0" ]
        then
            echo "Nothing Here"
        elif [ "$CHOICE" == "1" ]
        then
            bash view_tables.sh
            read -p "Press [ENTER] to return to Menu"
            pause
        elif [ "$CHOICE" == "2" ]
        then
            bash create_tables.sh
            read -p "Press [ENTER] to return to Menu"
            pause
        elif [ "$CHOICE" == "3" ]
        then
            bash drop_tables.sh
            read -p "Press [ENTER] to return to Menu"
            pause
        elif [ "$CHOICE" == "4" ]
        then
            bash populate_tables.sh
            read -p "Press [ENTER] to return to Menu"
            pause
        elif [ "$CHOICE" == "5" ]
        then
            bash interesting_advqueries.sh
            read -p "Press [ENTER] to return to Menu"
            pause
        elif [ "$CHOICE" == "E" ]
        then
            exit
        fi
    done
}

#--COMMENTS BLOCK--
```

```
# Main Program
#--COMMENTS BLOCK--
ProgramStart()
      {
            StartMessage
            while [ 1 ]
            do
                  MainMenu
            done
      }

ProgramStart
```

“view_tables.sh”

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
username=$(head -1 "credentials")
password=$(tail -1 "credentials")
sqlplus64
"$username/$password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca) (Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

select table_name FROM user_tables;
exit;

EOF
```

Assignment 6 - Functional Dependencies

user_acc(Email, FirstName, LastName, DateOfBirth, Password)
{ Email } → FirstName, LastName, DateOfBirth, Password

payment_method(Email, PaymentDetails, MethodType)
{ Email, PaymentDetails } → MethodType

uses(Email, PaymentDetails)
{ PaymentDetails } → Email

movie(MovieID, MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast)
{ MovieID } → MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast

orders(Email, MovieID, PaymentDetails)

studio(StudioName, ProducedMovies)
{ StudioName } → ProducedMovies

produces(MovieID, StudioName)
{ MovieID } → StudioName

rental(MovieID, RentalPrice, RentalDate, RentalExpiry)
{ MovieID } → RentalPrice, RentalDate, RentalExpiry

purchase(MovieID, BuyingPrice, PurchaseDate)
{ MovieID } → BuyingPrice, PurchaseDate

movie_review(RevID, WrittenRev, NumericRating, Email)
{ RevID } → WrittenRev, NumericRating, Email

links_to(RevID, MovieID, MoviePageUrl)
{ RevID } → MovieID, MoviePageUrl

writes(RevID, Email)
{ RevID } → Email
{ Email } → RevID

fanclub_membership(MembershipID, RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate)

{ MembershipID } → RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate

has(MembershipID, StudioName)
{ MembershipID } → StudioName

Assignment 7 - Normalization/3NF

user_acc(Email, FirstName, LastName, DateOfBirth, Password)
{ Email } → FirstName, LastName, DateOfBirth, Password

This relation is in 3NF because all non-primary keys (FirstName, LastName, DateofBirth, and Password) are fully functionally and non-transitively dependent on the primary key Email.

payment_method(Email, PaymentDetails, MethodType)
{ Email, PaymentDetails } → MethodType

This relation is not in 3NF because the attribute MethodType is partially dependent on the compound primary key i.e, {PaymentDetails} -> MethodType.

We fix this be decomposing the relation into two relations:

uses(Email, PaymentDetails)
{Email} -> PaymentDetails
{Payment} -> Email

payment_method(PaymentDetails, MethodType)
{PaymentDetails} -> MethodType

movie(MovieID, MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast)
{ MovieID } → MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast

This relation is in 3NF because all non-primary keys (MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis and MovieCast) are fully functionally and non-transitively dependent on the primary key MovieID.

orders(Email, MovieID, PaymentDetails)

This is a many-to-many relationship, hence, it has no functional dependencies.

studio(StudioName, ProducedMovies)
{ StudioName } → ProducedMovies

This relation is in 3NF because the non-primary key ProducedMovies is fully functionally and non-transitively dependent on the primary key StudioName.

produces(<u>MovieID</u>, <u>StudioName</u>)

This is a many-to-many relationship, hence, it has no functional dependencies.

rental(<u>MovieID</u>, RentalPrice, RentalDate, RentalExpiry)
{ MovieID } → RentalPrice, RentalDate, RentalExpiry

This relation is in 3NF because all non-primary keys (RentalPrice, RentalDate and Rental Expiry) are fully functionally and non-transitively dependent on the primary key MovieID.

purchase(<u>MovieID</u>, BuyingPrice, PurchaseDate)
{ MovieID } → BuyingPrice, PurchaseDate

This relation is in 3NF because all non-primary keys (BuyingPrice and PurchaseDate) are fully functionally and non-transitively dependent on the primary key MovieID.

movie_review(<u>RevID</u>, WrittenRev, NumericRating, Email, FirstName, LastName)
{ RevID } → WrittenRev, NumericRating, Email, FirstName, LastName
{ Email } -> FirstName, LastName

This relation is not in 3NF because the attributes (FirstName and LastName) are transitively dependent on Email.

To fix this we decompose the relation into two relations:

movie_review(<u>RevID</u>, WrittenRev, NumericRating, Email)
{RevID} -> WrittenRev, NumericRating, Email

user_details(<u>Email</u>, FirstName, LastName)
{Email} -> FirstName, LastName

links_to(<u>RevID, MovieID</u>, MoviePageUrl)
{RevID} -> MovieID
{MovieID} ->RevID
{ RevID, MovieID } → MoviePageUrl

This relation is not in 3NF because the non-key attribute (MoviePageUrl) is partially dependent on the compound primary key i.e, MovieID -> MoviePageUrl.

We fix this by decomposing the relation into two relations:

links_to(RevID,MovieID)
{ RevID } -> MovieID
{ MovieID } -> RevID

movie_url(MovieID, MoviePageUrl)
{MovieID} -> MoviePageUrl

fanclub_membership(MembershipID, RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate)
{ MembershipID } → RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate

This relation is in 3NF because all non-primary keys (RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate and OffersPurchaseEndDate) are fully functionally and non-transitively dependent on the primary key MembershipID.

has(MembershipID, StudioName)

This is a many-to-many relationship, hence, it has no functional dependencies.

Assignment 8 - Normalization/BCNF

user_acc(Email, FirstName, LastName, DateOfBirth, Password)
{ Email } → FirstName, LastName, DateOfBirth, Password

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

payment_method(Email, PaymentDetails, MethodType)
Step 1: Determining the FDs
FDs:
{ Email, PaymentDetails } → MethodType
{PaymentDetails} → MethodType

Step 2: Breaking RHS (in this case none) and finding redundancies
EP → M: EP+ = {E,P,M}   We get M here, therefore the FD: EP → M is redundant.
P → M: P+ = {p}   Not redundant.

So our FDs after removing EP → M are : FD = {P → M}.

Step 3: Finding the keys
E is not in the LHS or RHS therefore, it is part of the key.
P is in the LHS but not in the RSH therefore, it is part of the key.
M is not in the LHS but in the RHS therefore, it is not part of the key.
This implies that our key is {E,P}

Step 4: Making the tables
So now we finally have the FD: {P → M}
We make a relation for P → M: R1(P,M).
Since the relation R1 does not have the key, we add another relation for the key.
Therefore, we get 2 relations:
R1(P,M) with FD: P→M
R2(E,P) with no FDs.

These relations are now in 3NF according to Bernstein's algorithm.
These relations are also in BCNF because the left hand side of all these non-trivial, irreducible FDs are super keys.

movie(MovieID, MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast)
{ MovieID } → MovieName, Genre, Director, MovieRating, ReleaseYear, Resolution, Runtime, Synopsis, MovieCast

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

orders(Email, MovieID, PaymentDetails)

This is a many-to-many relationship, hence, it has no functional dependencies.

studio(StudioName, ProducedMovies)
{ StudioName } → ProducedMovies

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

produces(MovieID, StudioName)

This is a many-to-many relationship, hence, it has no functional dependencies.

rental(MovieID, RentalPrice, RentalDate, RentalExpiry)
{ MovieID } → RentalPrice, RentalDate, RentalExpiry

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

purchase(MovieID, BuyingPrice, PurchaseDate)
{ MovieID } → BuyingPrice, PurchaseDate

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

movie_review(RevID, WrittenRev, NumericRating, Email, FirstName, LastName)
Step 1: Determining the FDs
FDs:
{RevID} → WrittenRev, NumericRating, Email, FirstName, LastName
{Email} → FirstName, LastName

Step 2: Breaking RHS into single attributes and finding redundancies

FD = {RevID → WrittenRev, RevID → NumericRating, RevID → Email, RevID → FirstName, RevID → LastName, Email → FirstName, Email → LastName}

RevID → WrittenRev: RevID+ = {RevID, NumericRating, Email, FirstName, LastName}
Not redundant
RevID → NumericRating: RevID+ = {RevID, WrittenRev, Email, FirstName, LastName}
Not redundant
RevID → Email: RevID+ = {RevID, WrittenRev, NumericRating, FirstName, LastName}
Not redundant
RevID → FirstName: RevID+ = {RevID, WrittenRev, NumericRating, Email, FirstName, LastName}
We get FirstName, therefore the FD RevID → FirstName is redundant.
RevID → LastName: RevID+ = {RevID, WrittenRev, NumericRating, Email, FirstName, LastName}
We get LastName, therefore the FD RevID → LastName is redundant.
Email → FirstName: Email+ = {Email, LastName}
Not redundant
Email → LastName: Email+ = {Email, FirstName}
Not redundant

We remove the 2 redundant FDs which leaves
FD = {RevID → WrittenRev, RevID → NumericRating, RevID → Email, Email → FirstName, Email → LastName}

Step 3: Finding the keys
RevID is in the LHS but not in the RHS therefore, it is part of the key.
WrittenRev, NumericRating, FirstName and LastName are not in the LHS but in the RHS therefore, they are not part of the key.
Therefore {RevID} should be part of any key of relations R.
We now test keys by combining the rest of the attributes.

RevID+ = {RevID, WrittenRev, NumericRating, Email}
This is not the key since we do not get all the attributes.
{RevID, Email}+ = {RevID, WrittenRev, NumericRating, Email, FirstName, LastName}
This is the key since we get all the attributes.

Step 4: Making the tables
We combine the FDs with same LHS':
RevID → WrittenRev, NumericRating, Email
Email → FirstName, LastName

We now form the following relations:
R1(RevID, WrittenRev, NumericRating, Email)

R2(Email, FirstName, LastName)
We do not need to add another relation for the key since R1 already has RevID,Email.

These relations are now in 3NF according to Bernstein's algorithm.
These relations are also in BCNF because the left hand side of all these non-trivial, irreducible FDs are super keys.

fanclub_membership(<u>MembershipID</u>, RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate)
{ MembershipID } → RentalDiscount, PurchaseDiscount, RentalSpecialOffers, PurchaseSpecialOffers, OffersRentalStartDate, OffersRentalEndDate, OffersPurchaseStartDate, OffersPurchaseEndDate

This relation is in BCNF because the left hand side of the non-trivial, irreducible FD is a super key.

has(<u>MembershipID, StudioName</u>)

This is a many-to-many relationship, hence, it has no functional dependencies.

Assignment 9 - Java UI

Java UI Code

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
import java.util.Properties;
import java.sql.SQLException;
import java.sql.ResultSetMetaData;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.FileReader;
import java.io.Reader;
import java.io.File;
import java.util.Scanner;


public class CPS510_Assignment9_JavaUI {

	public static String[] parseFile (String filename) throws IOException
	{
		String create = "";
		String line = "";
		Scanner in = new Scanner(new File(filename));
		while (in.hasNextLine())
		{
			line = in.nextLine();
			line = line.trim();
			if (!line.equals("\n") && !line.equals("") && line != null)
			{
				if (!line.substring(0,1).equals("-"))
				{
					create += line + " ";
				}
			}
		}
		in.close();
		String[] sql_commands = create.split(";");

		for (int j = 0; j < sql_commands.length; j++)
```

```
            {
                  sql_commands[j] = sql_commands[j].trim();
            }
            return sql_commands;
      }

      public static String parseCommand() throws IOException
      {
            String[] sql_commands = null;

            Scanner in = new Scanner(System.in);
            System.out.print("Please enter your sql command: ");
            String command = in.nextLine();
            command = command.trim();

            if (!command.equals("") && command != null)
            {
                  sql_commands = command.split(";");
            }
            else
            {
                  return "";
            }

            if (sql_commands.length == 1)
            {
                  return sql_commands[0];
            }
            else if (sql_commands.length > 1)
            {
                  return "1";
            }
            else
            {
                  return "";
            }

      }

  public static void main(String[] args) {

    Connection conn1 = null;

    try {
        // registers Oracle JDBC driver - though this is no longer required
        // since JDBC 4.0, but added here for backward compatibility
        Class.forName("oracle.jdbc.OracleDriver");


        String dbURL1 = "jdbc:oracle:thin:@oracle.scs.ryerson.ca:1521:orcl";  // that is school
Oracle database and you can only use it in the labs



         //String DBURL = "jdbc:oracle:thin:@localhost:1521:xe";
                        /* This XE or local database that you installed on your laptop. 1521 is the
default port for database, change according to what you used during installation.
                        xe is the sid, change according to what you setup during installation. */
                        conn1 = DriverManager.getConnection(dbURL1, "sisyed","12167695");
        if (conn1 != null) {
           System.out.println("Connected with connection #1");
        }

                        String tableName = "";
                        int i = 0;
                        String command = "";
                        String[] tokens = null;
                        Scanner input = null;
                        String word = "";
                        String c = "";

                        while (!word.equals("E") && !word.equals("X"))
                        {
                                try (Statement stmt =
conn1.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY))
                                {

                                                System.out.println("\n");
                                                System.out.println("1) View Existing Tables");
                                                System.out.println("2) Create All Tables");
                                                System.out.println("3) Drop All Tables");
                                                System.out.println("4) Populate Tables");
                                                System.out.println("5) Show Interesting Advanced
Queries");

                                                System.out.println("6) Create Views");
                                                System.out.println("7) Drop Views");
                                                System.out.println("8) Insert Record");
                                                System.out.println("9) Update Record");
                                                System.out.println("10) Delete Record");
                                                System.out.println("11) Search Record");
                                                System.out.println("E) End/Exit");
                                                System.out.println("\n");
                                                System.out.print("Choose: ");
                                                input = new Scanner(System.in);
                                                word = input.next();

                                                try
                                                {
                                                    System.out.println();
                                                    if (word.equals("0"))
                                                    {
                                                        System.out.println("Nothing Here");
                                                    }
                                                    else if (word.equals("1"))
                                                    {
                                                        String query = "select table_name
FROM user_tables";
                                                        ResultSet rs =
stmt.executeQuery(query);
                                                        while (rs.next())
                                                        {
                                                            String table =
rs.getString("table_name");
                                                            System.out.println(table);
                                                        }
                                                    }
                                                    else if (word.equals("2"))
                                                    {

                                                        String[] sql_commands =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\createTables.s
ql");
                                                        if (sql_commands[0] != null &&
!sql_commands[0].equals(""))
                                                        {

stmt.executeUpdate(sql_commands[0]);
                                                        }
```

```
                                                                for (i = 1; i < sql_commands.length;
i++)
                                                                {
                                                                        command =
sql_commands[i];

                                                                        if (command != null &&
!command.equals(""))
                                                                        {

stmt.executeUpdate(command);
                                                                                tokens =
command.split(" ");
                                                                                tableName =
tokens[2];

System.out.println("Successfully created table" + " " + tableName);
                                                                        }
                                                                }
                                                        }
                                                        else if (word.equals("3"))
                                                        {
                                                                String[] sql_commands =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\dropTables.sql"
);

                                                                for (i = 0; i < sql_commands.length;
i++)
                                                                {
                                                                        command =
sql_commands[i];

                                                                        if (command != null &&
!command.equals(""))
                                                                        {

stmt.executeUpdate(command);
                                                                                tokens =
command.split(" ");
                                                                                tableName =
tokens[2];
```

```
System.out.println("Successfully dropped table" + " " + tableName);
                                    }

                                }
                            }
                            else if (word.equals("4"))
                            {
                                String[] sql_commands1 =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\insertMovies.sq
l");
                                String[] sql_commands2 =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\insert.sql");
                                for (i = 0; i < sql_commands1.length;
i++)
                                {
                                    command =
sql_commands1[i];

                                    if (command != null &&
!command.equals(""))
                                    {

stmt.executeUpdate(command);
                                        System.out.println("1
Row Inserted");
                                    }
                                }
                                for (i = 0; i <
sql_commands2.length-1; i++)
                                {
                                    command =
sql_commands2[i];

                                    if (command != null &&
!command.equals(""))
                                    {

stmt.executeUpdate(command);
```

```
                                        System.out.println("1
Row Inserted");
                                    }
                                }
                            }
                            else if (word.equals("5"))
                            {
                                String[] queries =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\queries.sql");

                                for (i = 0; i < queries.length; i++)
                                {
                                    command = queries[i];

                                    if (command != null &&
!command.equals(""))
                                    {
                                        ResultSet rs =
stmt.executeQuery(command);
                                        if (!rs.next())
                                        {

System.out.println("No rows selected");
                                        }
                                        else
                                        {

rs.beforeFirst();

ResultSetMetaData rsmd = rs.getMetaData();

                                            int
numColumns = rsmd.getColumnCount();

                                            while
(rs.next())
                                            {
                                                String
row = "";
                                                for (int
j = 0; j < numColumns; ++j)
                                                {
```

```
Object val = rs.getObject(j+1);

row += val + ", ";
                                                }

System.out.println(row.substring(0,row.length()-2));
                                            }

System.out.println();
                                        }
                                    }
                                }
                            }
                            else if (word.equals("6"))
                            {
                                String[] sql_commands =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\viewTables.sql"
);


                                for (i = 0; i < sql_commands.length;
i++)
                                {
                                    command =
sql_commands[i];

                                    if (command != null &&
!command.equals(""))
                                    {
                                        ResultSet rs =
stmt.executeQuery(command);
                                        tokens =
command.split(" ");
                                        tableName =
tokens[2];
                                        ResultSetMetaData
rsmd = rs.getMetaData();
                                        int numColumns =
rsmd.getColumnCount();

System.out.println("Successfully created view" + " " + tableName);
```

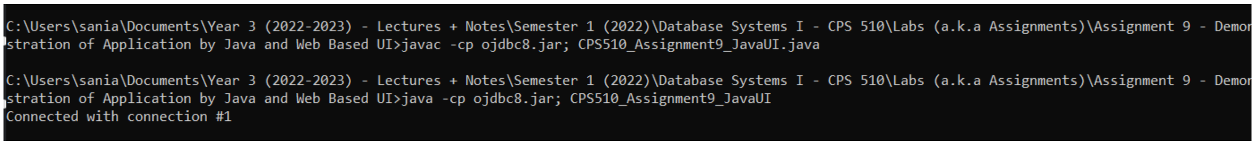
```
                                    }

                                }
                            }
                            else if (word.equals("7"))
                            {
                                String[] sql_commands =
parseFile("C:\\Users\\sania\\Documents\\Year 3 (2022-2023) - Lectures + Notes\\Semester 1
(2022)\\Database Systems I - CPS 510\\Labs (a.k.a
Assignments)\\Database_Application_Github\\cps510_databaseApplication\\sql\\dropViews.sql")
;

                                for (i = 0; i < sql_commands.length;
i++)
                                {
                                    command =
sql_commands[i];

                                    if (command != null &&
!command.equals(""))
                                    {

stmt.executeUpdate(command);
                                        tokens =
command.split(" ");
                                        tableName =
tokens[2];

System.out.println("Successfully dropped view" + " " + tableName);
                                    }
                                }
                            }
                            else if (word.equals("8") || word.equals("9")
|| word.equals("10") || word.equals("11"))
                            {
                                String whatCommand = "";
                                String sql_command =
parseCommand();
                                if (sql_command.equals(""))
                                {
                                    System.out.println("\nInvalid
Command");
                                }
                                else if (sql_command.equals("1"))
                                {
                                    System.out.println("\nPlease
enter only one command");
                                }
                                else
                                {
                                    tokens = sql_command.split("
");
                                    whatCommand = tokens[0];
                                    if
(whatCommand.equalsIgnoreCase("insert") && word.equals("8"))
                                    {

stmt.executeUpdate(sql_command);

System.out.println("\nInserted 1 row");
                                    }
                                    else if
(whatCommand.equalsIgnoreCase("update") && word.equals("9"))
                                    {

stmt.executeUpdate(sql_command);

System.out.println("\nUpdated 1 row");
                                    }
                                    else if
(whatCommand.equalsIgnoreCase("delete") && word.equals("10"))
                                    {
                                        int num =
stmt.executeUpdate(sql_command);
                                        if (num > 0)
                                        {

System.out.println("\nDeleted 1 row");

                                        }
                                        else
                                        {

System.out.println("\nThis row does not exist");
                                        }

                                    }
                                    else if
(whatCommand.equalsIgnoreCase("select") && word.equals("11"))
```

```
                                                            {
                                                                  ResultSet rs =
stmt.executeQuery(sql_command);
                                                                  if (!rs.next())
                                                                  {

System.out.println("\nNo rows selected");
                                                                  }
                                                                  else
                                                                  {

rs.beforeFirst();

ResultSetMetaData rsmd = rs.getMetaData();

                                                                        int
numColumns = rsmd.getColumnCount();


System.out.println();


                                                                        while
(rs.next())
                                                                        {
                                                                              String
row = "";
                                                                              for (int
j = 0; j < numColumns; ++j)
                                                                              {

Object val = rs.getObject(j+1);

row += val + ", ";
                                                                              }

System.out.println(row.substring(0,row.length()-2));
                                                                        }

System.out.println();
                                                                  }
                                                            }
                                                            else
                                                            {
```

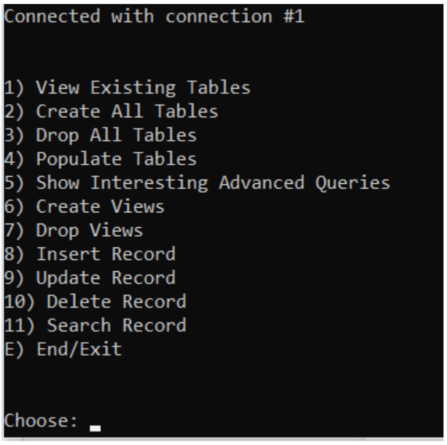
```
System.out.println("\nInvalid Command");
                                    }
                                }
                            }
                            else if (word.equals("E"))
                            {
                                conn1.close();
                            }
                            else
                            {
                                System.out.println("Invalid Input");
                            }
                        }
                        catch (FileNotFoundException e)
                        {
                            System.out.println("File not found: " +
e.getMessage());
                        }
                        catch (IOException e)
                        {
                            System.out.println(e.getMessage());
                        }
                    }

                catch (SQLException e)
                {
                    System.out.println(e.getMessage());
                }
            }



    } catch (ClassNotFoundException ex) {
      ex.printStackTrace();
    } catch (SQLException ex) {
      ex.printStackTrace();
    } finally {
      try {
        if (conn1 != null && !conn1.isClosed()) {
          conn1.close();
        }

      } catch (SQLException ex) {
```

```
                ex.printStackTrace();
            }
        }


    }
}
```

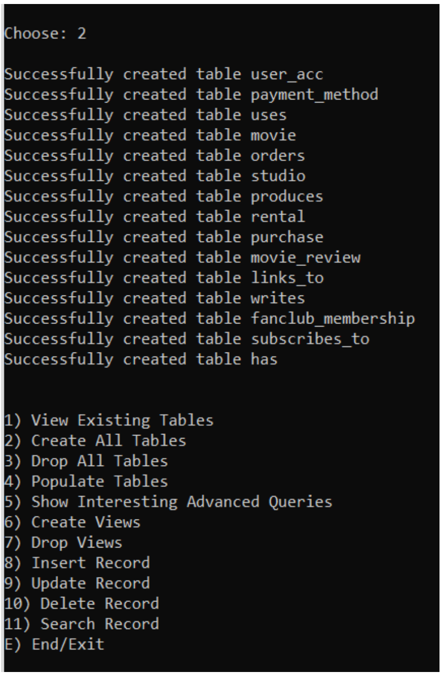
Java UI Demo



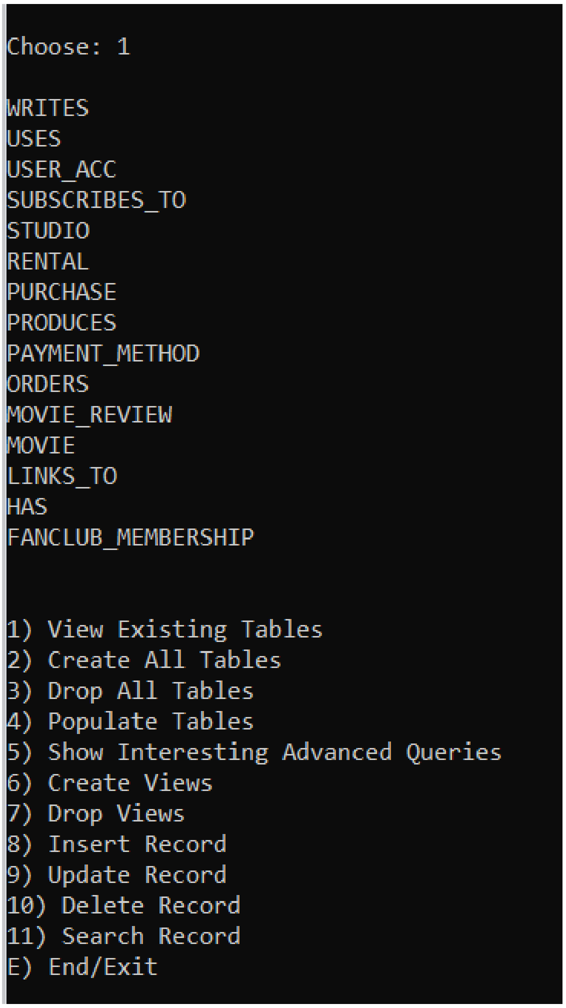
```
C:\Users\sania\Documents\Year 3 (2022-2023) - Lectures + Notes\Semester 1 (2022)\Database Systems I - CPS 510\Labs (a.k.a Assignments)\Assignment 9 - Demon
stration of Application by Java and Web Based UI>javac -cp ojdbc8.jar; CPS510_Assignment9_JavaUI.java

C:\Users\sania\Documents\Year 3 (2022-2023) - Lectures + Notes\Semester 1 (2022)\Database Systems I - CPS 510\Labs (a.k.a Assignments)\Assignment 9 - Demon
stration of Application by Java and Web Based UI>java -cp ojdbc8.jar; CPS510_Assignment9_JavaUI
Connected with connection #1
```



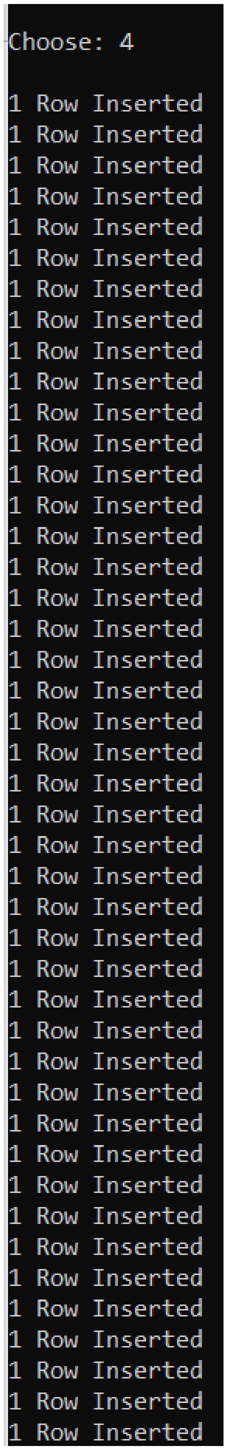
```
Connected with connection #1


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose: _
```



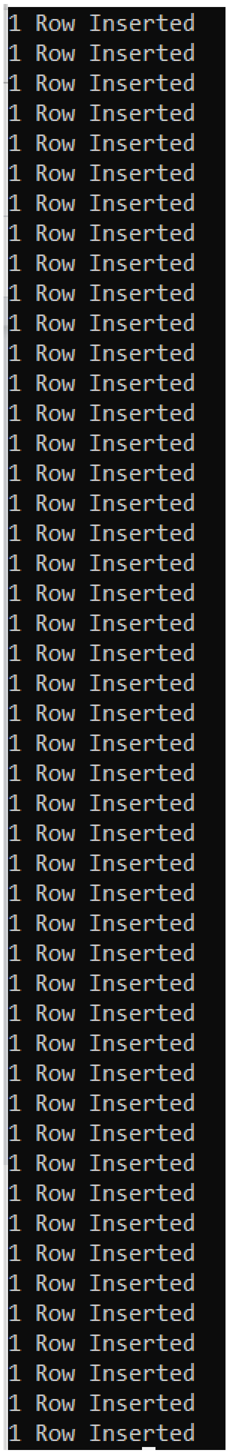
```
Choose: 2

Successfully created table user_acc
Successfully created table payment_method
Successfully created table uses
Successfully created table movie
Successfully created table orders
Successfully created table studio
Successfully created table produces
Successfully created table rental
Successfully created table purchase
Successfully created table movie_review
Successfully created table links_to
Successfully created table writes
Successfully created table fanclub_membership
Successfully created table subscribes_to
Successfully created table has


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit
```



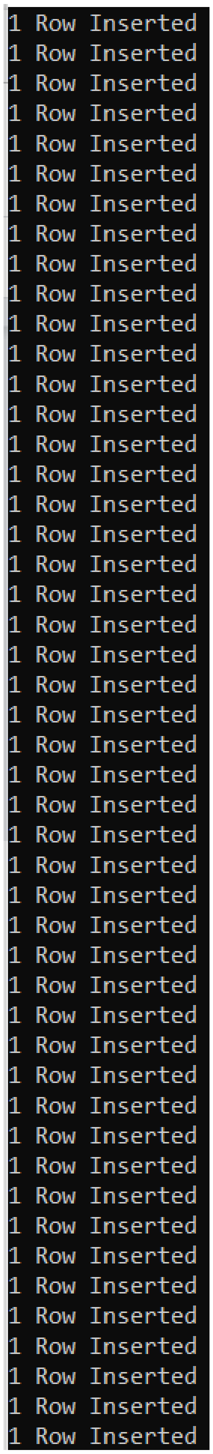
```
Choose: 1

WRITES
USES
USER_ACC
SUBSCRIBES_TO
STUDIO
RENTAL
PURCHASE
PRODUCES
PAYMENT_METHOD
ORDERS
MOVIE_REVIEW
MOVIE
LINKS_TO
HAS
FANCLUB_MEMBERSHIP


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit
```



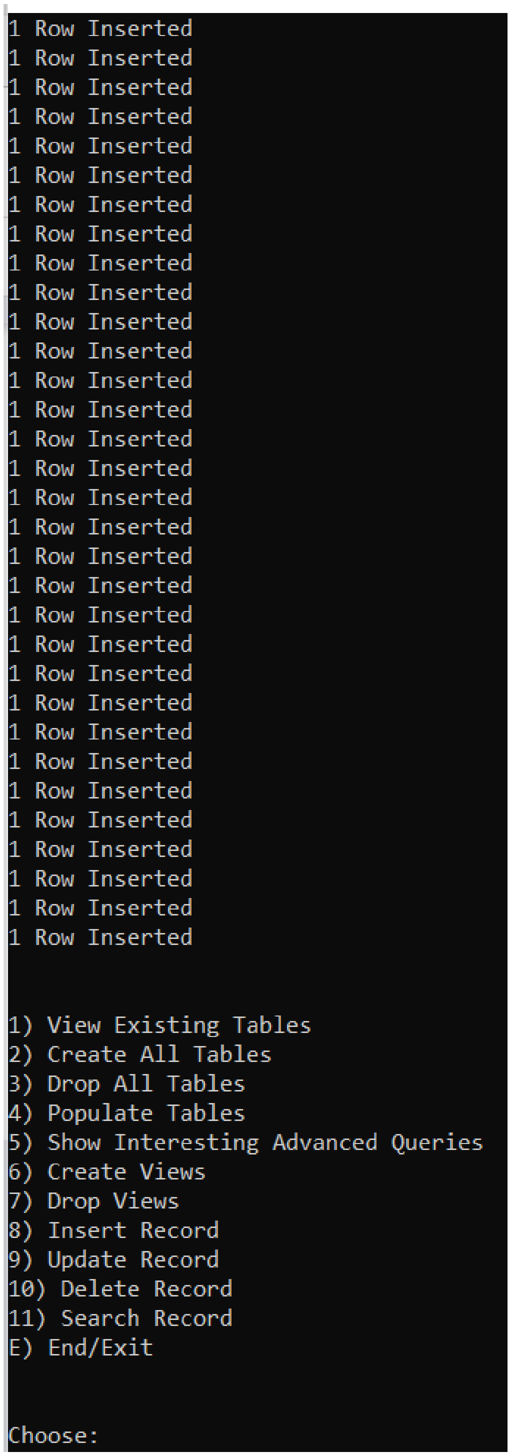
```
Choose: 4

1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
```



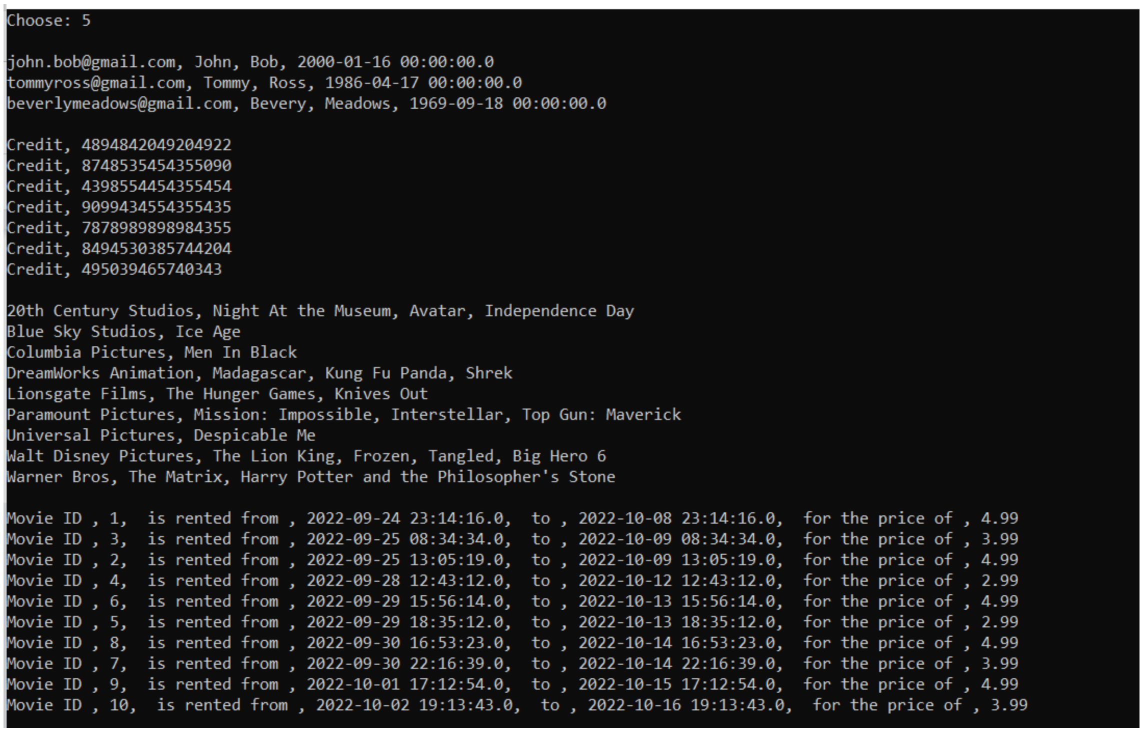
```
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
```



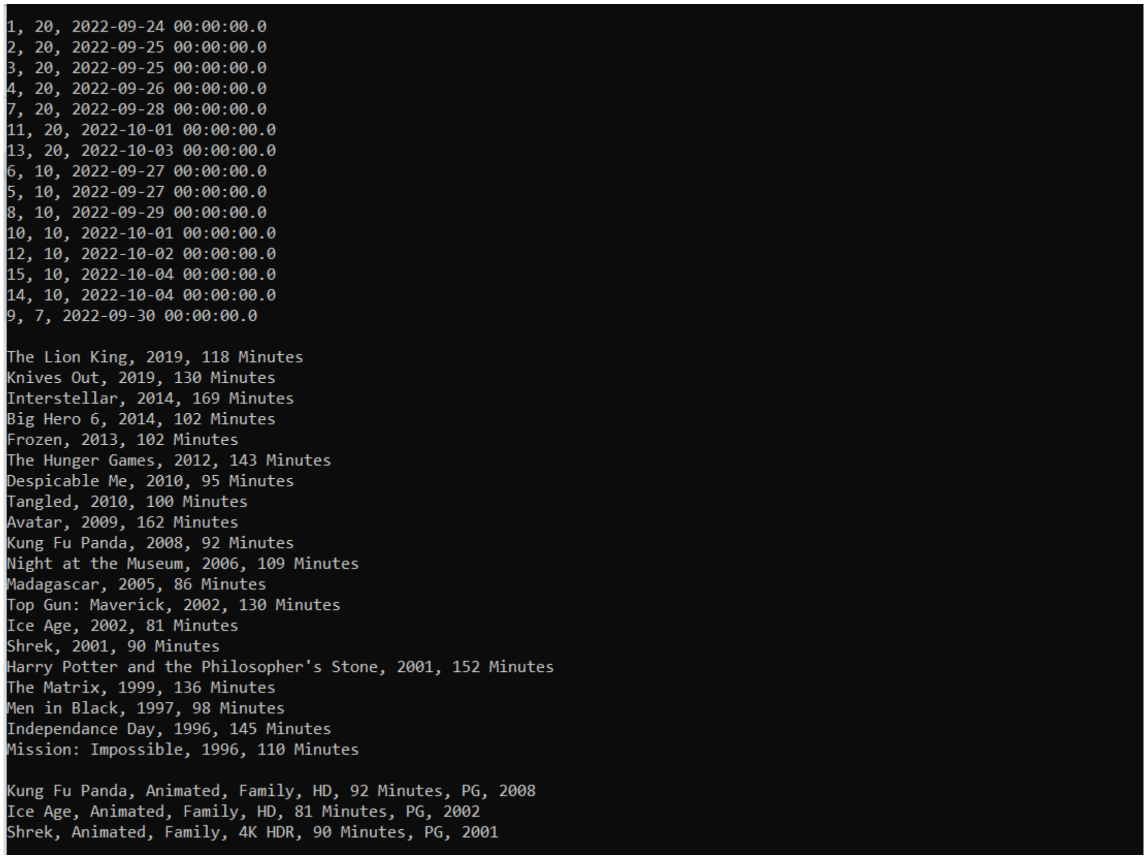
```
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
```



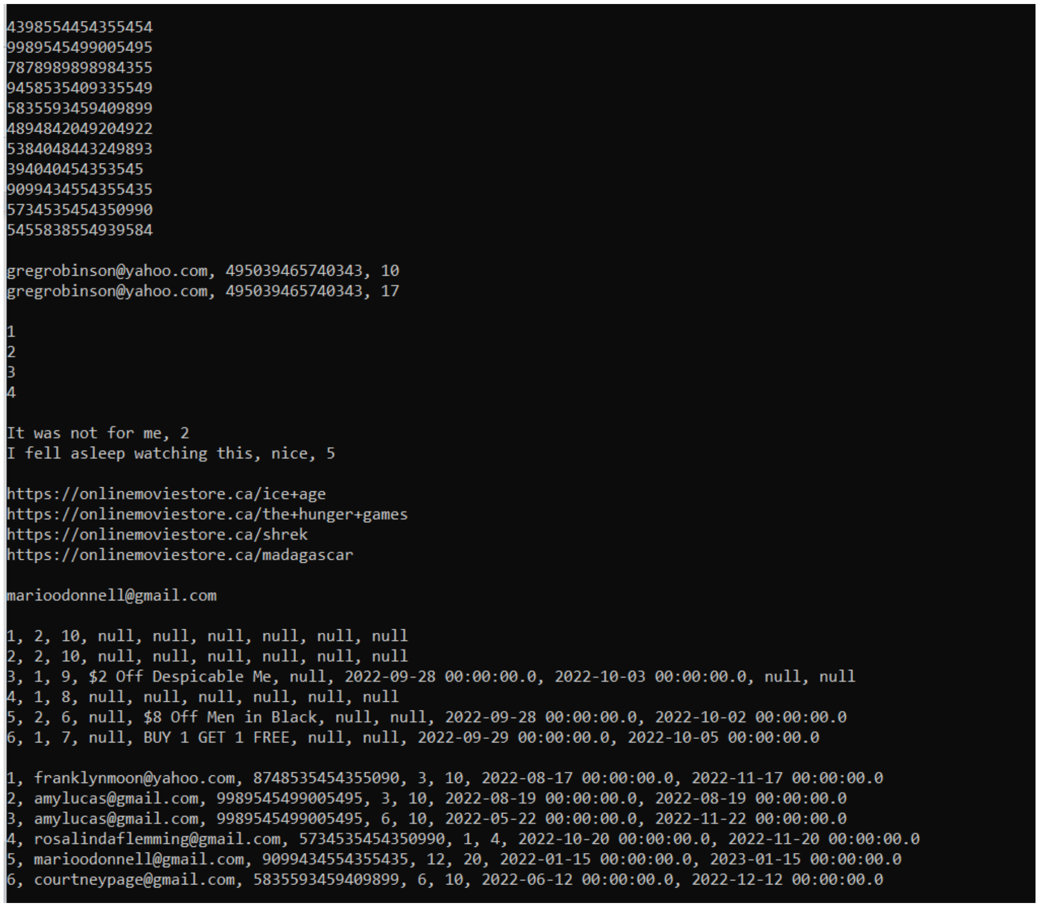
```
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted
1 Row Inserted


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



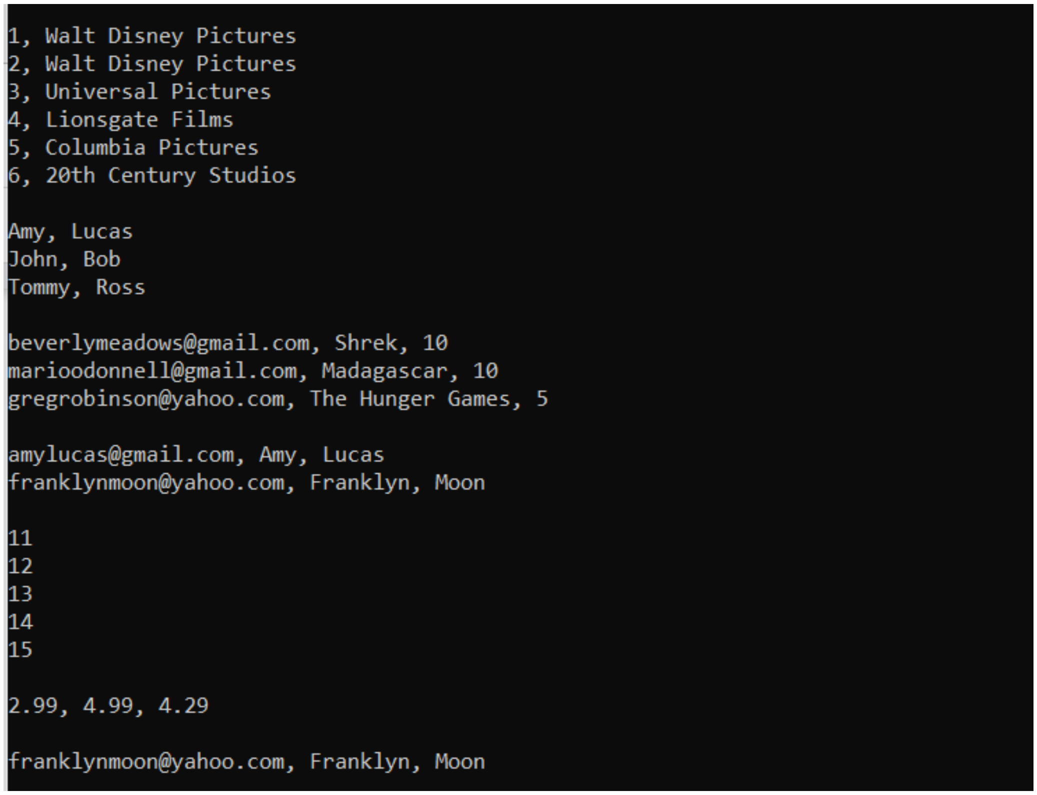
```
Choose: 5

john.bob@gmail.com, John, Bob, 2000-01-16 00:00:00.0
tommyross@gmail.com, Tommy, Ross, 1986-04-17 00:00:00.0
beverlymeadows@gmail.com, Bevery, Meadows, 1969-09-18 00:00:00.0

Credit, 4894842049204922
Credit, 8748535454355090
Credit, 4398554454355454
Credit, 9099434554355435
Credit, 7878989898984355
Credit, 8494530385744204
Credit, 495039465740343

20th Century Studios, Night At the Museum, Avatar, Independence Day
Blue Sky Studios, Ice Age
Columbia Pictures, Men In Black
DreamWorks Animation, Madagascar, Kung Fu Panda, Shrek
Lionsgate Films, The Hunger Games, Knives Out
Paramount Pictures, Mission: Impossible, Interstellar, Top Gun: Maverick
Universal Pictures, Despicable Me
Walt Disney Pictures, The Lion King, Frozen, Tangled, Big Hero 6
Warner Bros, The Matrix, Harry Potter and the Philosopher's Stone

Movie ID , 1,  is rented from , 2022-09-24 23:14:16.0,  to , 2022-10-08 23:14:16.0,  for the price of , 4.99
Movie ID , 3,  is rented from , 2022-09-25 08:34:34.0,  to , 2022-10-09 08:34:34.0,  for the price of , 3.99
Movie ID , 2,  is rented from , 2022-09-25 13:05:19.0,  to , 2022-10-09 13:05:19.0,  for the price of , 4.99
Movie ID , 4,  is rented from , 2022-09-28 12:43:12.0,  to , 2022-10-12 12:43:12.0,  for the price of , 2.99
Movie ID , 6,  is rented from , 2022-09-29 15:56:14.0,  to , 2022-10-13 15:56:14.0,  for the price of , 4.99
Movie ID , 5,  is rented from , 2022-09-29 18:35:12.0,  to , 2022-10-13 18:35:12.0,  for the price of , 2.99
Movie ID , 8,  is rented from , 2022-09-30 16:53:23.0,  to , 2022-10-14 16:53:23.0,  for the price of , 4.99
Movie ID , 7,  is rented from , 2022-09-30 22:16:39.0,  to , 2022-10-14 22:16:39.0,  for the price of , 3.99
Movie ID , 9,  is rented from , 2022-10-01 17:12:54.0,  to , 2022-10-15 17:12:54.0,  for the price of , 4.99
Movie ID , 10,  is rented from , 2022-10-02 19:13:43.0,  to , 2022-10-16 19:13:43.0,  for the price of , 3.99
```



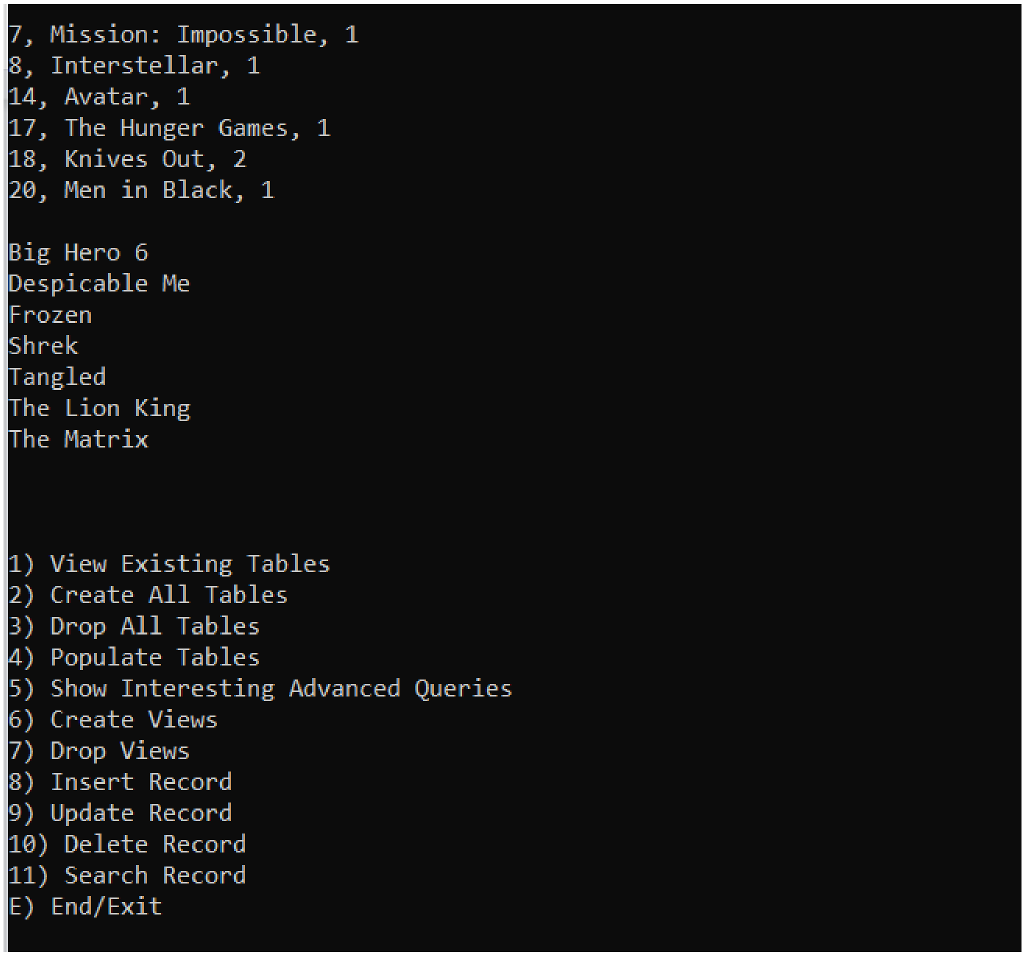
```
1, 20, 2022-09-24 00:00:00.0
2, 20, 2022-09-25 00:00:00.0
3, 20, 2022-09-25 00:00:00.0
4, 20, 2022-09-26 00:00:00.0
7, 20, 2022-09-28 00:00:00.0
11, 20, 2022-10-01 00:00:00.0
13, 20, 2022-10-03 00:00:00.0
6, 10, 2022-09-27 00:00:00.0
5, 10, 2022-09-27 00:00:00.0
8, 10, 2022-09-29 00:00:00.0
10, 10, 2022-10-01 00:00:00.0
12, 10, 2022-10-02 00:00:00.0
15, 10, 2022-10-04 00:00:00.0
14, 10, 2022-10-04 00:00:00.0
9, 7, 2022-09-30 00:00:00.0

The Lion King, 2019, 118 Minutes
Knives Out, 2019, 130 Minutes
Interstellar, 2014, 169 Minutes
Big Hero 6, 2014, 102 Minutes
Frozen, 2013, 102 Minutes
The Hunger Games, 2012, 143 Minutes
Despicable Me, 2010, 95 Minutes
Tangled, 2010, 100 Minutes
Avatar, 2009, 162 Minutes
Kung Fu Panda, 2008, 92 Minutes
Night at the Museum, 2006, 109 Minutes
Madagascar, 2005, 86 Minutes
Top Gun: Maverick, 2002, 130 Minutes
Ice Age, 2002, 81 Minutes
Shrek, 2001, 90 Minutes
Harry Potter and the Philosopher's Stone, 2001, 152 Minutes
The Matrix, 1999, 136 Minutes
Men in Black, 1997, 98 Minutes
Independance Day, 1996, 145 Minutes
Mission: Impossible, 1996, 110 Minutes

Kung Fu Panda, Animated, Family, HD, 92 Minutes, PG, 2008
Ice Age, Animated, Family, HD, 81 Minutes, PG, 2002
Shrek, Animated, Family, 4K HDR, 90 Minutes, PG, 2001
```



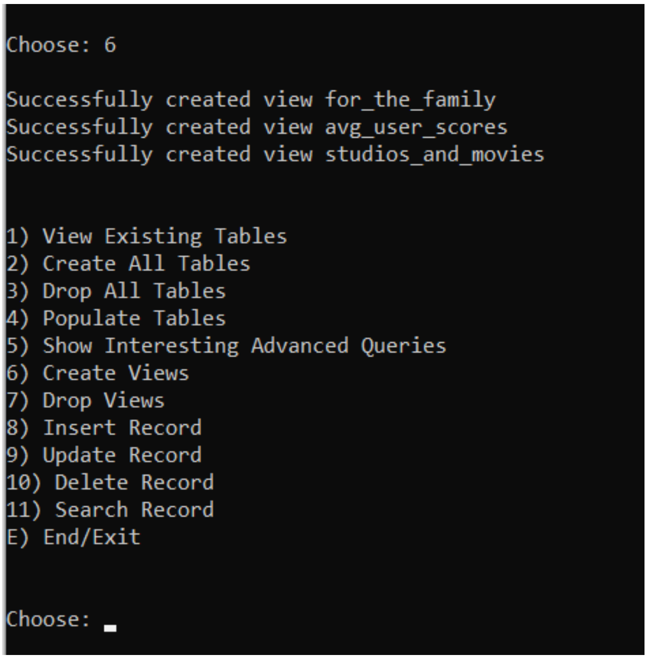
```
4398554454355454
9989545499005495
7878989898984355
9458535409335549
5835593459409899
4894842049204922
5384048443249893
394040454353545
9099434554355435
5734535454350990
5455838554939584

gregrobinson@yahoo.com, 495039465740343, 10
gregrobinson@yahoo.com, 495039465740343, 17

1
2
3
4

It was not for me, 2
I fell asleep watching this, nice, 5

https://onlinemoviestore.ca/ice+age
https://onlinemoviestore.ca/the+hunger+games
https://onlinemoviestore.ca/shrek
https://onlinemoviestore.ca/madagascar

marioodonnell@gmail.com

1, 2, 10, null, null, null, null, null, null
2, 2, 10, null, null, null, null, null, null
3, 1, 9, $2 Off Despicable Me, null, 2022-09-28 00:00:00.0, 2022-10-03 00:00:00.0, null, null
4, 1, 8, null, null, null, null, null, null
5, 2, 6, null, $8 Off Men in Black, null, null, 2022-09-28 00:00:00.0, 2022-10-02 00:00:00.0
6, 1, 7, null, BUY 1 GET 1 FREE, null, null, 2022-09-29 00:00:00.0, 2022-10-05 00:00:00.0

1, franklynmoon@yahoo.com, 8748535454355090, 3, 10, 2022-08-17 00:00:00.0, 2022-11-17 00:00:00.0
2, amylucas@gmail.com, 9989545499005495, 3, 10, 2022-08-19 00:00:00.0, 2022-08-19 00:00:00.0
3, amylucas@gmail.com, 9989545499005495, 6, 10, 2022-05-22 00:00:00.0, 2022-11-22 00:00:00.0
4, rosalindaflemming@gmail.com, 5734535454350990, 1, 4, 2022-10-20 00:00:00.0, 2022-11-20 00:00:00.0
5, marioodonnell@gmail.com, 9099434554355435, 12, 20, 2022-01-15 00:00:00.0, 2023-01-15 00:00:00.0
6, courtneypage@gmail.com, 5835593459409899, 6, 10, 2022-06-12 00:00:00.0, 2022-12-12 00:00:00.0
```



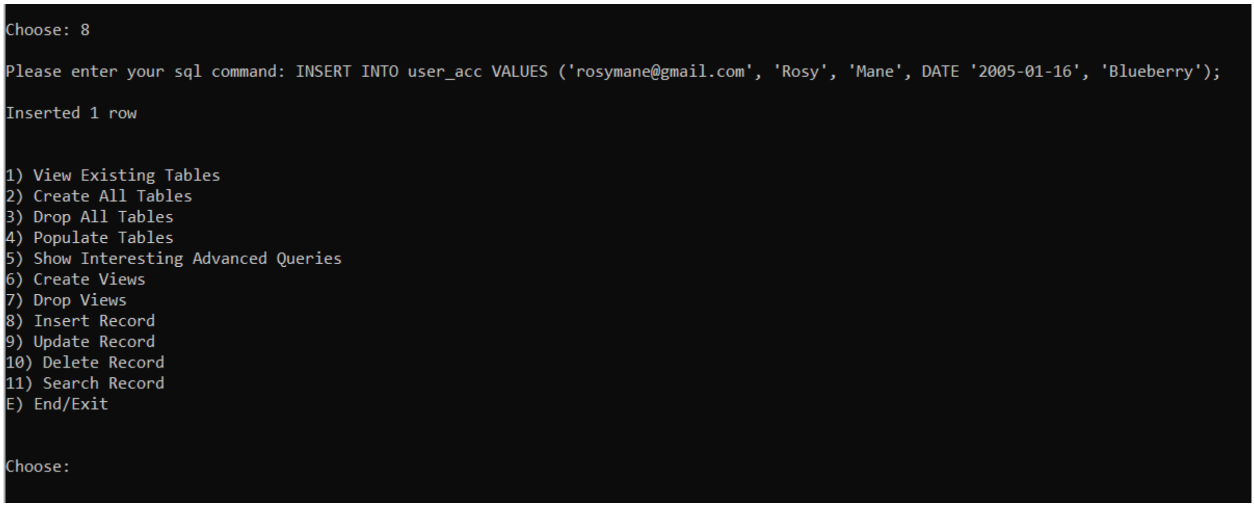
```
1, Walt Disney Pictures
2, Walt Disney Pictures
3, Universal Pictures
4, Lionsgate Films
5, Columbia Pictures
6, 20th Century Studios

Amy, Lucas
John, Bob
Tommy, Ross

beverlymeadows@gmail.com, Shrek, 10
marioodonnell@gmail.com, Madagascar, 10
gregrobinson@yahoo.com, The Hunger Games, 5

amylucas@gmail.com, Amy, Lucas
franklynmoon@yahoo.com, Franklyn, Moon

11
12
13
14
15

2.99, 4.99, 4.29

franklynmoon@yahoo.com, Franklyn, Moon
```



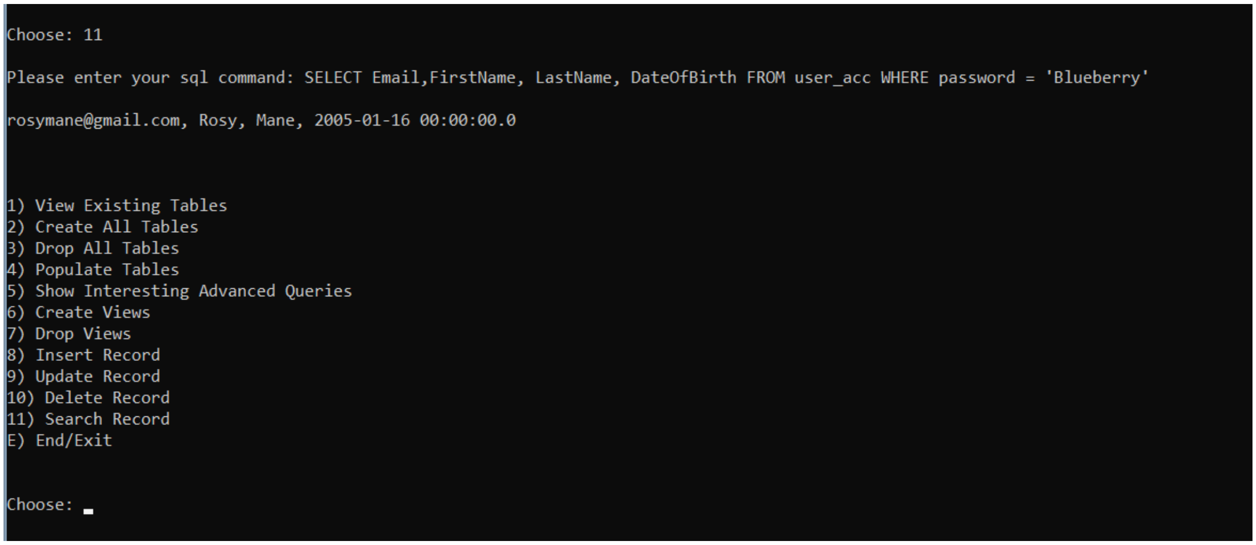
```
7, Mission: Impossible, 1
8, Interstellar, 1
14, Avatar, 1
17, The Hunger Games, 1
18, Knives Out, 2
20, Men in Black, 1

Big Hero 6
Despicable Me
Frozen
Shrek
Tangled
The Lion King
The Matrix



1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit
```



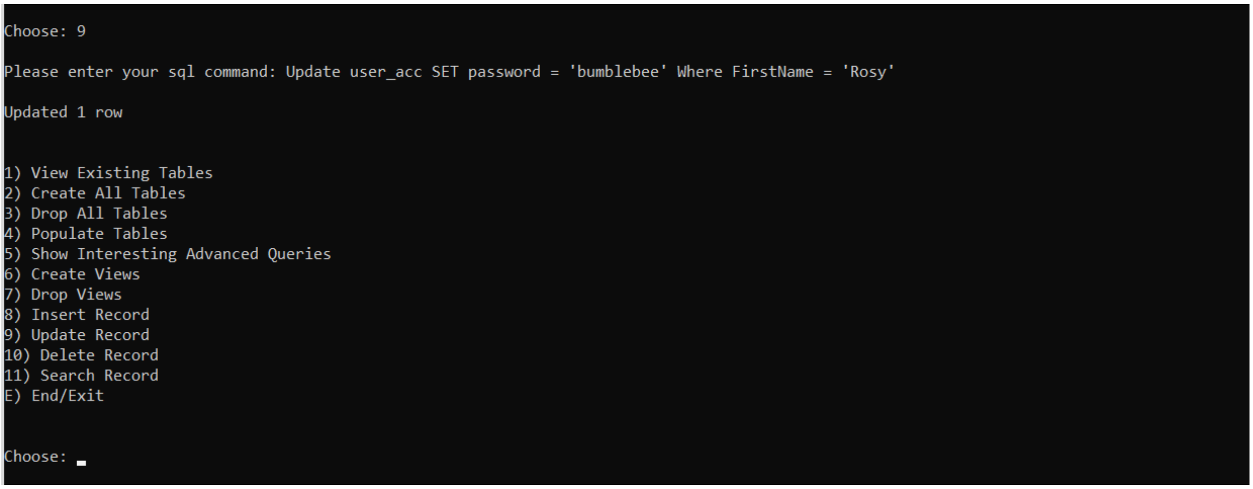
```
Choose: 6

Successfully created view for_the_family
Successfully created view avg_user_scores
Successfully created view studios_and_movies


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose: _
```



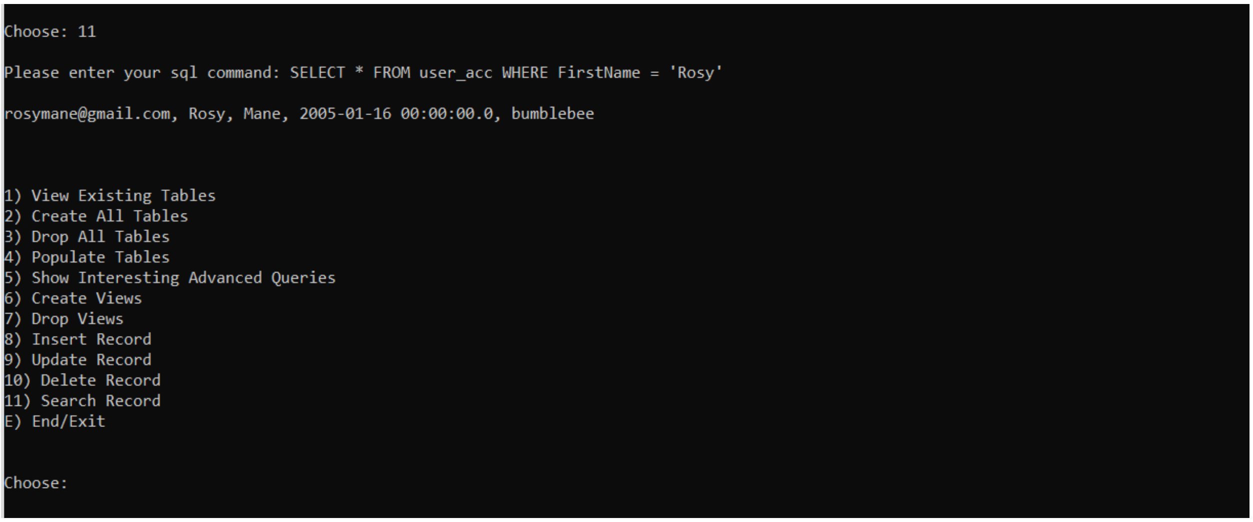
```
Choose: 8

Please enter your sql command: INSERT INTO user_acc VALUES ('rosymane@gmail.com', 'Rosy', 'Mane', DATE '2005-01-16', 'Blueberry');

Inserted 1 row


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



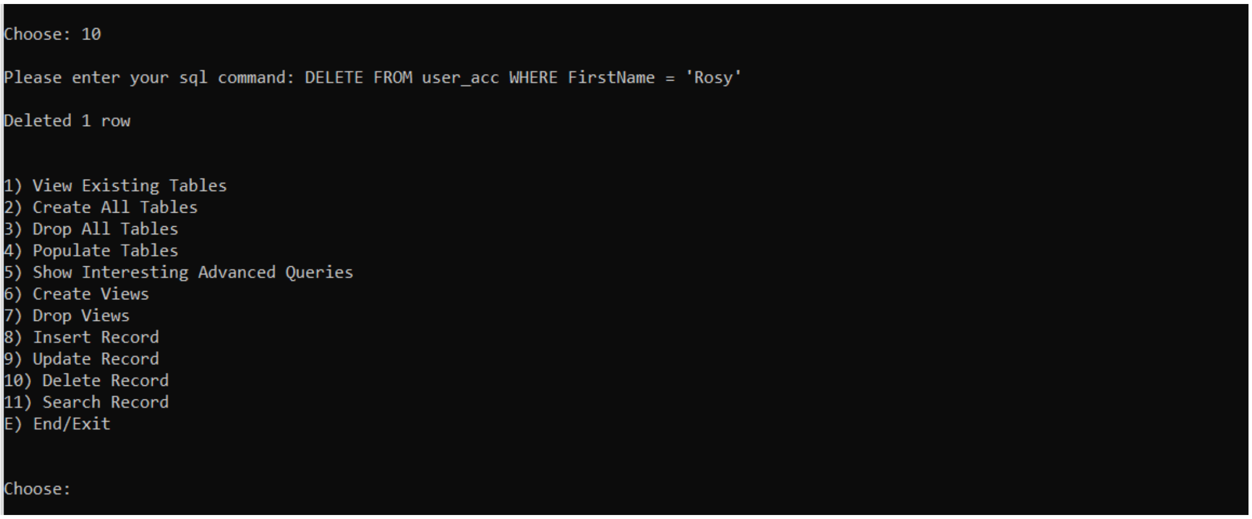
```
Choose: 11

Please enter your sql command: SELECT Email,FirstName, LastName, DateOfBirth FROM user_acc WHERE password = 'Blueberry'

rosymane@gmail.com, Rosy, Mane, 2005-01-16 00:00:00.0



1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose: _
```



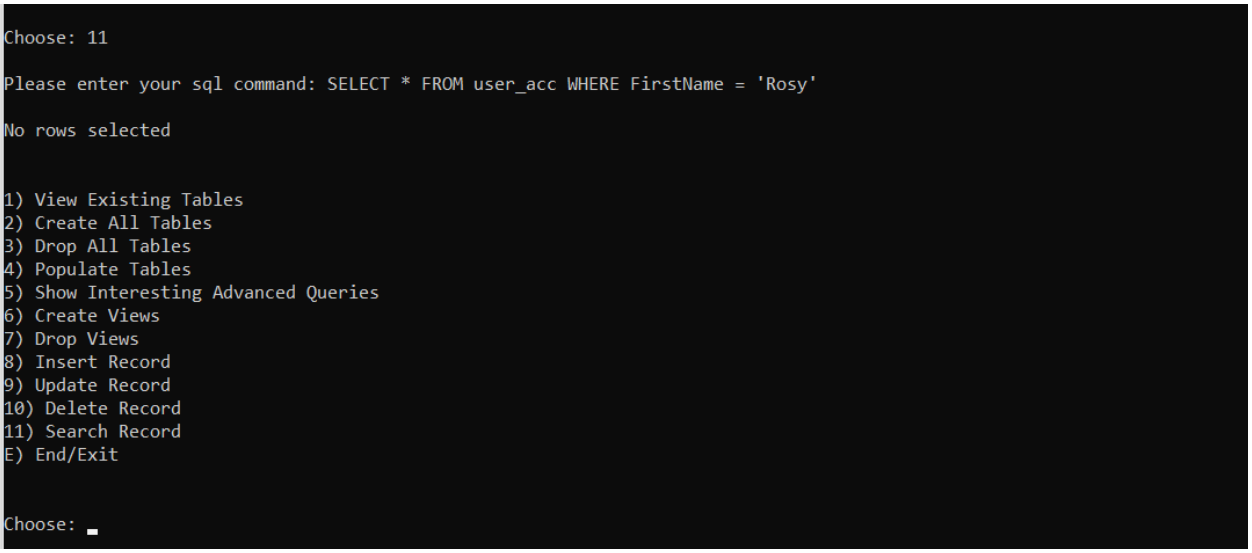
```
Choose: 9

Please enter your sql command: Update user_acc SET password = 'bumblebee' Where FirstName = 'Rosy'

Updated 1 row


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose: _
```



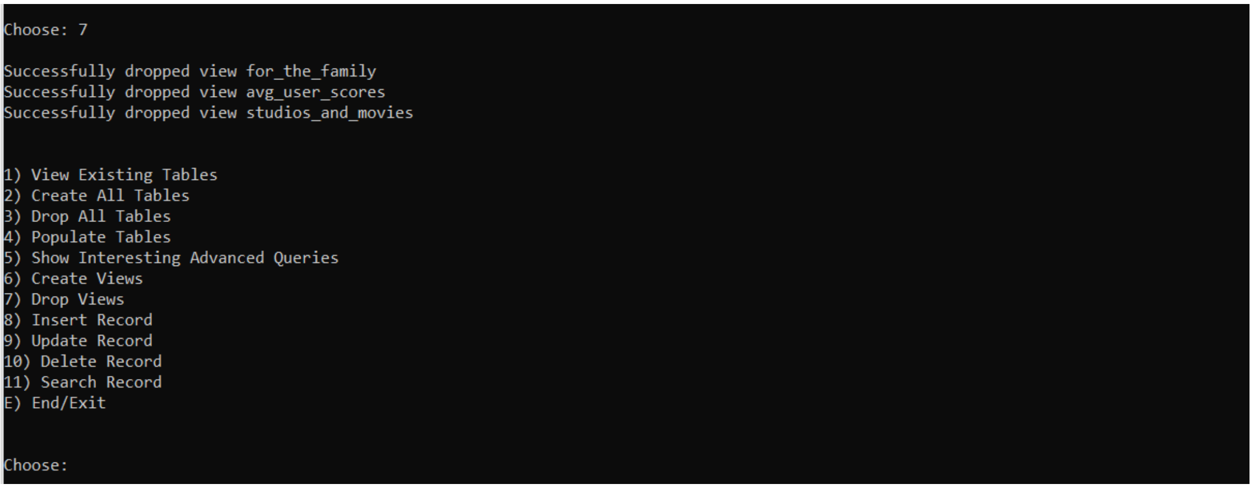
```
Choose: 11

Please enter your sql command: SELECT * FROM user_acc WHERE FirstName = 'Rosy'

rosymane@gmail.com, Rosy, Mane, 2005-01-16 00:00:00.0, bumblebee



1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



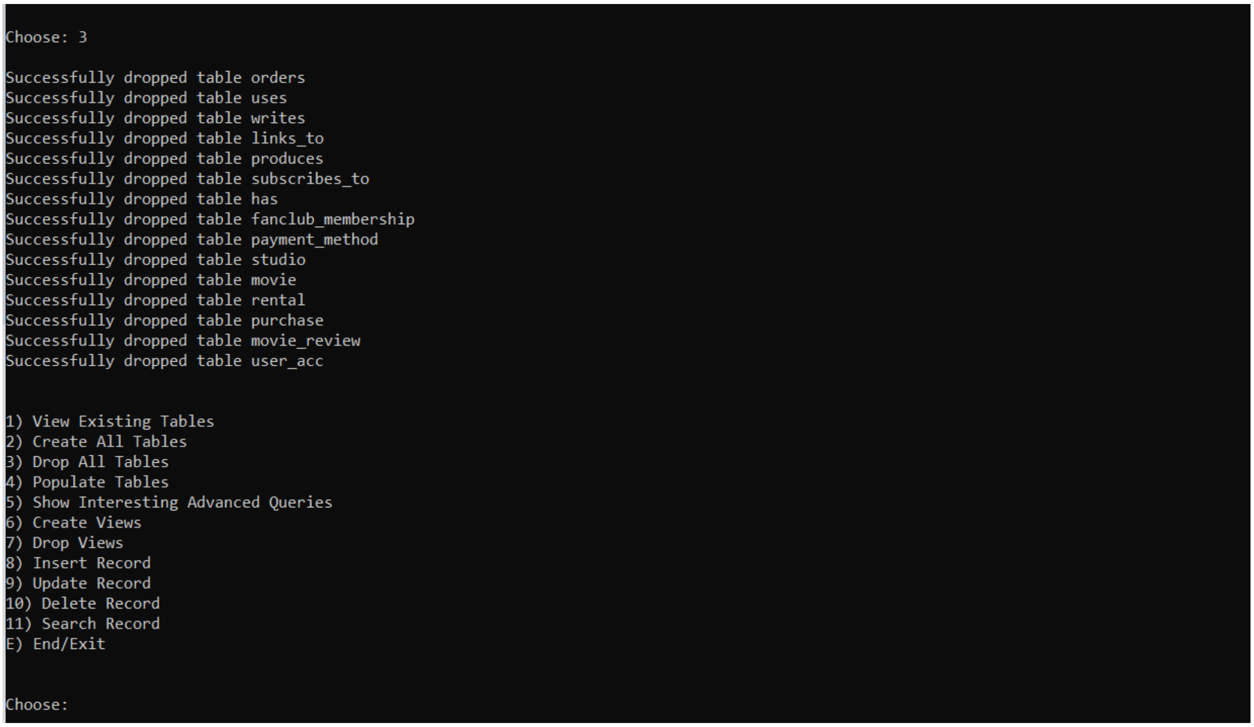
```
Choose: 10

Please enter your sql command: DELETE FROM user_acc WHERE FirstName = 'Rosy'

Deleted 1 row


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



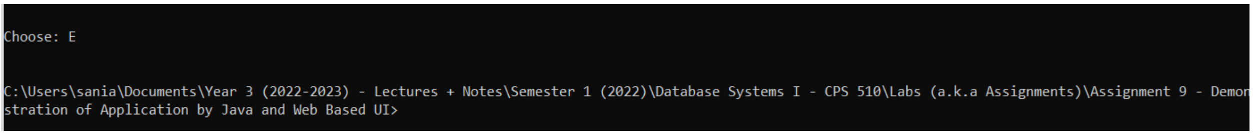
```
Choose: 11

Please enter your sql command: SELECT * FROM user_acc WHERE FirstName = 'Rosy'

No rows selected


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



```
Choose: 7

Successfully dropped view for_the_family
Successfully dropped view avg_user_scores
Successfully dropped view studios_and_movies


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



```
Choose: 3

Successfully dropped table orders
Successfully dropped table uses
Successfully dropped table writes
Successfully dropped table links_to
Successfully dropped table produces
Successfully dropped table subscribes_to
Successfully dropped table has
Successfully dropped table fanclub_membership
Successfully dropped table payment_method
Successfully dropped table studio
Successfully dropped table movie
Successfully dropped table rental
Successfully dropped table purchase
Successfully dropped table movie_review
Successfully dropped table user_acc


1) View Existing Tables
2) Create All Tables
3) Drop All Tables
4) Populate Tables
5) Show Interesting Advanced Queries
6) Create Views
7) Drop Views
8) Insert Record
9) Update Record
10) Delete Record
11) Search Record
E) End/Exit


Choose:
```



```
Choose: E


C:\Users\sania\Documents\Year 3 (2022-2023) - Lectures + Notes\Semester 1 (2022)\Database Systems I - CPS 510\Labs (a.k.a Assignments)\Assignment 9 - Demon
stration of Application by Java and Web Based UI>
```

Assignment 10 - Relational Algebra

Query for user_acc

```
SELECT Email,FirstName, LastName, DateOfBirth
FROM user_acc
WHERE DateOfBirth LIKE '%6%';
```

$$\pi_{\text{Email, FirstName, LastName, DateOfBirth}}(\sigma_{\text{DateOfBirth LIKE ‘\%6\%’}}\ (\text{user\_account}))$$

Query for payment_method

```
SELECT MethodType, PaymentDetails
FROM payment_method
WHERE MethodType = 'Credit';
```

$$\pi_{\text{MethodType, PaymentDetails}}(\sigma_{\text{MethodType = ‘Credit’}}\ (\text{payment\_method}))$$

Query for studio

```
SELECT  *
FROM studio
ORDER BY StudioName;
```

studio

Query for rental

```
SELECT 'Movie ID ', MovieID, ' is rented from ', RentalDate, ' to ', RentalExpiry, ' for the price of
', RentalPrice
FROM rental
ORDER BY RentalDate ASC;
```

$$\pi_{\text{’Movie ID ’, MovieID, ‘is rented from ’, RentalDate, ‘ to ’, RentalExpiry, ‘ for the price of’, RentalPrice}}\ (\text{rental})$$

Query for purchase

```
SELECT *
FROM purchase
ORDER BY BuyingPrice DESC, PurchaseDate ASC;
```

purchase

Query for movie

```
SELECT MovieName, ReleaseYear, Runtime
FROM movie
ORDER BY ReleaseYear DESC;
```

$\pi_{\text{MovieName, ReleaseYear, Runtime}}(\text{movie})$

Query for movie

```
SELECT MovieName, Genre, Resolution, Runtime, MovieRating, ReleaseYear
FROM movie
WHERE Genre = 'Animated, Family'
ORDER BY ReleaseYear DESC;
```

$\pi_{\text{MovieName, Genre, Resolution, Runtime, MovieRating, ReleaseYear}}(\sigma_{\text{Genre = 'Animated, Family'}}(\text{movie}))$

Query for uses

```
SELECT PaymentDetails
FROM uses
WHERE Email LIKE '%gmail.com';
```

$\pi_{\text{PaymentDetails}}(\sigma_{\text{Email LIKE '\%gmail.com'}}(\text{uses}))$

Query for orders

```
SELECT Email, PaymentDetails, MovieID
FROM orders
WHERE Email = 'gregrobinson@yahoo.com';
```

$\pi_{\text{Email, PaymentDetails, MovieID}}(\sigma_{\text{Email = 'gregrobinson@yahoo.com'}}(\text{orders}))$

Query for produces

```
SELECT MovieID
FROM produces
```

```
WHERE StudioName = 'Walt Disney Pictures';
```

$$\boldsymbol{\pi}_{\text{MovieID}}\ (\boldsymbol{\sigma}_{\text{StudioName = ‘Walt Disney Pictures’}}\ (\textbf{produces}))$$

Query for movie_review

```
SELECT WrittenRev, NumericRating
FROM movie_review
WHERE Email = 'gregrobinson@yahoo.com';
```

$$\boldsymbol{\pi}_{\text{WrittenRev, NumericRating}}\ (\boldsymbol{\sigma}_{\text{Email=’gregrobinson@yahoo.com’}}\ (\textbf{movie\_review}))$$

Query for links_to

```
SELECT MoviePageUrl
FROM links_to;
```

$$\boldsymbol{\pi}_{\text{MoviePageURL}}\ (\textbf{links\_to})$$

Query for writes

```
SELECT Email
FROM writes
WHERE RevID = 34;
```

$$\boldsymbol{\pi}_{\text{Email}}\ (\boldsymbol{\sigma}_{\text{RevID = 34}}\ (\textbf{writes}))$$

Query for fanclub_membership

```
SELECT *
FROM Fanclub_membership;
```

Fanclub_membership

Query for subscribes_to

```
SELECT *
FROM subscribes_to;
```

subscribes_to

Query for has

```
SELECT *
FROM has;
```

has

List names of people who have ordered the movie 'Frozen'

```
SELECT FirstName, LastName
FROM orders o, user_acc u
WHERE MovieID=2 AND u.Email = o.Email
ORDER BY FirstName ASC;
```

$$\pi_{\text{FirstName, LastName}}(\sigma_{\text{MovieID = 2}}(\text{user\_acc} \bowtie \text{orders}))$$

List Emails, Movie Name and Movie Rating of people who gave movies a rating of more than or equal to 5.

```
SELECT u.Email, m.MovieName, mr.NumericRating
FROM user_acc u, writes w, movie_review mr, links_to lt, movie m
WHERE mr.NumericRating >=5 AND u.Email = w.Email AND w.RevID = mr.RevID AND
mr.RevID = lt.RevID AND lt.MovieID = m.MovieID
ORDER BY mr.NumericRating DESC;
```

$$\pi_{\text{Email, MovieName, NumericRating}}(\sigma_{\text{NumericRating >= 5}}(\text{users} \bowtie \text{writes} \bowtie \text{movie\_review} \bowtie \text{links\_to} \bowtie \text{movie}))$$

List the Emails and Names of people who have subscribed to the Studio 'Walt Disney Animation'

```
SELECT u.Email, u.FirstName, u.LastName
FROM user_acc u, subscribes_to st, Fanclub_membership fm, has h
WHERE h.StudioName = 'Walt Disney Pictures' AND u.Email = st.Email AND st.MembershipID
= fm.MembershipID AND fm.MembershipID = h.MembershipID
ORDER BY Email ASC;
```

$$\pi_{\text{Email, FirstName, LastName}}(\sigma_{\text{StudioName = 'Walt Disney Pictures'}}(\text{users} \bowtie \text{subscribes\_to} \bowtie \text{fanclub\_membership} \bowtie \text{has}))$$

List MovieID, MovieName which can be purchased but not rented.

```
SELECT m.MovieID
FROM movie m, purchase p
WHERE m.MovieID = p.MovieID AND m.MovieID NOT IN (SELECT  r.MovieID FROM rental r);
```

$\text{rent} \leftarrow \pi_{\text{MovieID}}\ (\text{rental})$

$\text{purchase} \leftarrow \pi_{\text{MovieID}}\ (\text{movie} \bowtie \text{purchase})$

$\text{purchaseNoRent} \leftarrow \text{purchase - rent}$

List maximum, minimum, average price for renting a movie.

```
SELECT MIN(rentalprice), MAX(rentalprice), AVG(rentalprice)
FROM rental;
```

$F_{\text{MIN rentalPrice, Max rentalPrice, AVG rentalPrice}}\ (\text{rental})$

List user info whose payment method is only credit

```
SELECT u.email, u.firstname, u.lastname
FROM user_acc u
WHERE NOT EXISTS (SELECT u.email, u.firstname, u.lastname FROM payment_method p,
uses s WHERE u.email = s.email AND s.email = p.email AND p.methodtype = 'Debit');
```

$\text{users} \leftarrow \pi_{\text{Email, FirstName, LastName}}\ (\text{user\_acc})$

$\text{debit} \leftarrow \pi_{\text{Email, FirstName, LastName}}\ (\sigma_{\text{methodtype = 'debit'}}$

$\quad (\text{user\_acc} \bowtie \text{uses} \bowtie \text{payment\_method}))$

$\text{credit} \leftarrow \text{users - debit}$

List movieID and name for PG-13 movies along with number of times it has been ordered.

```
SELECT o.movieID, m.movieName, count(*)
FROM orders o, movie m WHERE o.movieId = m.movieID AND m.movieRating = 'PG-13'
GROUP BY o.movieID, m.movieName ORDER BY movieID;
```

$\pi_{\text{movieID, movieName, (movieID, movieName}}F_{\text{COUNT)}}\ (\sigma_{\text{movieRating = 'PG-13'}}$

$\quad (\text{orders} \bowtie \text{movies}))$

List of movies produced by Walt Disney Pictures or have a resolution of 4K HDR

```
SELECT m.movieName
FROM movie m
WHERE EXISTS (SELECT m.movieName FROM produces p WHERE m.movieId = p.movieId
AND p.studioName = 'Walt Disney Pictures')
UNION (SELECT m.movieName FROM movie m WHERE m.resolution = '4K HDR');
```

$$\textbf{produced} \leftarrow \pi_{\text{movieName}} (\sigma_{\text{studioName = ‘Walt Disney Pictures’}} (\textbf{movie} \bowtie \textbf{produces}))$$
$$\textbf{resolution} \leftarrow \pi_{\text{movieName}} (\sigma_{\text{resolution = ‘4K HDR’}} (\textbf{movie}))$$
$$\textbf{result} \leftarrow \textbf{produced} \cup \textbf{resolution}$$

Installation and Operation Instructions

Shell UI

1) Download shell_menu folder and sql folder
2) cd into shell_menu
3) Put oracle username in first line of “credentials” file and oracle password in the second line of that file
4) Use the bash commanf to run “shell_menu.sh”
5) Use option 2, to create the tables
6) Use option 4, to run the sql code for inserting rows
7) Use option 5, to display the queries
8) Use option E, to end session

Java UI

1) Download all the sql files in the sql folder and CPS510_Assignment9_JavaUI.java.
2) Download ojdbc8.jar from the oracle website.
3) Place the jar file and the java file in the same folder on your computer.
4) Inside the java file, connect to the school database by entering your oracle username and password inside the line: **conn1 = DriverManager.getConnection(dbURL1, “username”, “password”);**, which is located at the top of the main method.
5) In the java file, change the file paths in all of the parseFile method calls to where the sql files are located on your computer.
6) In command prompt, cd into the folder where the jar file and java file are located on PC.
7) Compile the files by entering the following command: **javac -cp ojdbc8.jar; CPS510_Assignment9_JavaUI.java**
8) Run the files by entering the following command: **java -cp ojdbc8.jar; CPS510_Assignment9_JavaUI**. Make sure you are connected to the school VPN.
9) To interact with the UI:
    a) Use option 1 to view all the tables that have been created.
    b) Use option 2 to create the tables in the createTables.sql file.
    c) Use option 3 to drop all the tables.
    d) Use option 4 to populate all the tables with the data in the insertMovies.sql and insert.sql files.
    e) Use option 5 to run all the queries in the queries.sql file.
    f) Use option 6 to create the views in the viewTables.sql file.
    g) Use option 7 to drop all the views.
    h) Use option 8 to insert a record.
    i) Use option 9 to update a record.
    j) Use option 10 to delete a record.
    k) Use option 11 to query records.
    l) Use option E to end the program and close the connection.

Concluding Remarks

By the end of this project, we have managed to make the database in 3NF or BCNF. We have accomplished this through the normalization process associated with converting relations to 3NF and BCNF. These normalization processes can be seen in the Assignment 7 and 8 parts of the project where we converted the functional dependencies of our database relations. After accomplishing the aforementioned we modified our SQL code to reflect the effects of normalizing the relations. Overall, the experience of completing this database project was both challenging and enjoyable. Individually we can all say that we have picked up valuable skills related to database management.