

2022

Precise Management Production

IST 659 FINAL PROJECT

**MARIANNE LYNN SANTOS GARBO, KWASI MENSAH,
AMANDA NORWOOD**

SYRACUSE UNIVERSITY | PROFESSOR DR. GREGORY BLOCK

TABLE OF CONTENTS

BACKGROUND	1
ISSUES WITH CURRENT SYSTEM	2
STAKEHOLDERS	2
BUSINESS RULES	2
GLOSSARY	2
DATA QUESTIONS	3
CONCEPTUAL MODEL	3
LOGICAL MODEL	4
RAW DATA SAMPLE	4
PHYSICAL DATABASE CREATION	5
ALL DROPS	5
TABLE CREATIONS	5
TABLE INSERTIONS	8
VIEW CREATIONS	19
PROCEDURES	21
DATA QUESTIONS & VALIDATION	27
IMPLEMENTATION OF APPLICATION	34
REVIEW ACTORS' MOVIE HISTORY AND RATES	34
SEARCH FOR MOVIE DETAILS	35
SET MOVIE BUDGET	35
REFLECTION	37
SUMMARY	37
TEAM LOG	37
MARIANNE LYNNE SANTOS GARBO	37
KWASI MENSAH	37
AMANDA NORWOOD	38

BACKGROUND

Precise Management Production is a production management company that has been managing production companies for years and retained their movie production information and post release performance on a spreadsheet. They have found it increasingly difficult to manage and maintain their data as their production company has seen exponential growth in the past few years. With the recent growth, they noticed a rise of clerical errors, data inconsistencies, and poor version control across the datasets. They have reached out to DataModels to build a database to store the information on their movies and to provide functionality and ensure data integrity. The

goal with the new database is for users to be able to search for movie data, actors, directors, movie performance, and movie casting by year.

ISSUES WITH CURRENT SYSTEM

Precise Management Production currently warehouses their data in Excel spreadsheets. They do not have an efficient way of researching which actors and directors are currently assigned to a movie. They are also unable to an organized spreadsheet of movie castings, ratings, and revenue. They are also unable to perform statistical calculations of their movie performances. Another issue is version control within the company as multiple departments want to update the spreadsheet at one time.

STAKEHOLDERS

The production companies, movie casters, budget department, directors, and actors will greatly benefit from the database. This will allow the production company to manage their decisions in choosing film locations and castings, to determine which genres perform better, and which actors or directors to pursue in the future based on revenue and/or ratings. In addition, the new database will assist the production company with reporting on the movies.

BUSINESS RULES

- A movie is classified as a genre
- A movie films at one or more locations
- A movie films at a location during a span of many days
- A movie produced by a production company.
- A movie employs people as an actor/s and a director
- A movie critic rates the movies
- A fan rates the movies
- A person is given a role
- A person is casted in one or more movies
- A person directs one or more movies
- A person is paid their set rate
- A film location is paid their set rate
- A location is the destination where the movie was filmed

GLOSSARY

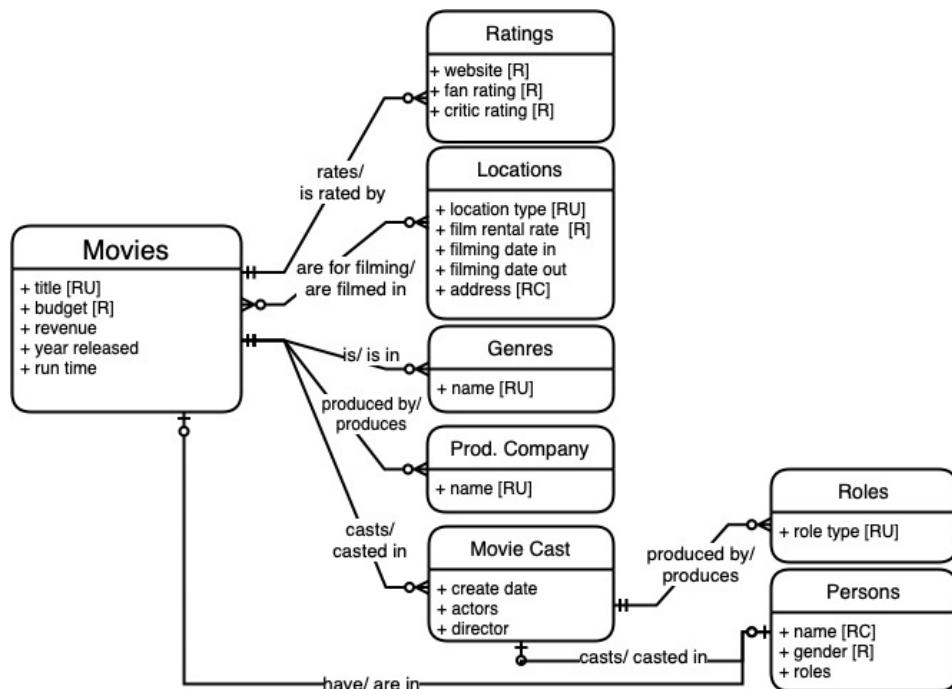
- A **person** is someone who can be a director, main actor, or support actor that is casted to the movie
- A **genre** is a classification of the movie type
- A **movie** is a film that contains casted members, an assigned director, budget, and run time
- A **location** is a particular place where something occurs.
- **Film locations** are places where a movie is filmed

- **Budget** includes the initial budget, the actual budget, the movie release revenue, and the total revenue
- A **rating** is a 0-100 score on how good a movie is. The rating is given by a fan or critic.
- A **production company** is a business that produces different types of subject matter.
- A **role** is a part assigned to an actor.

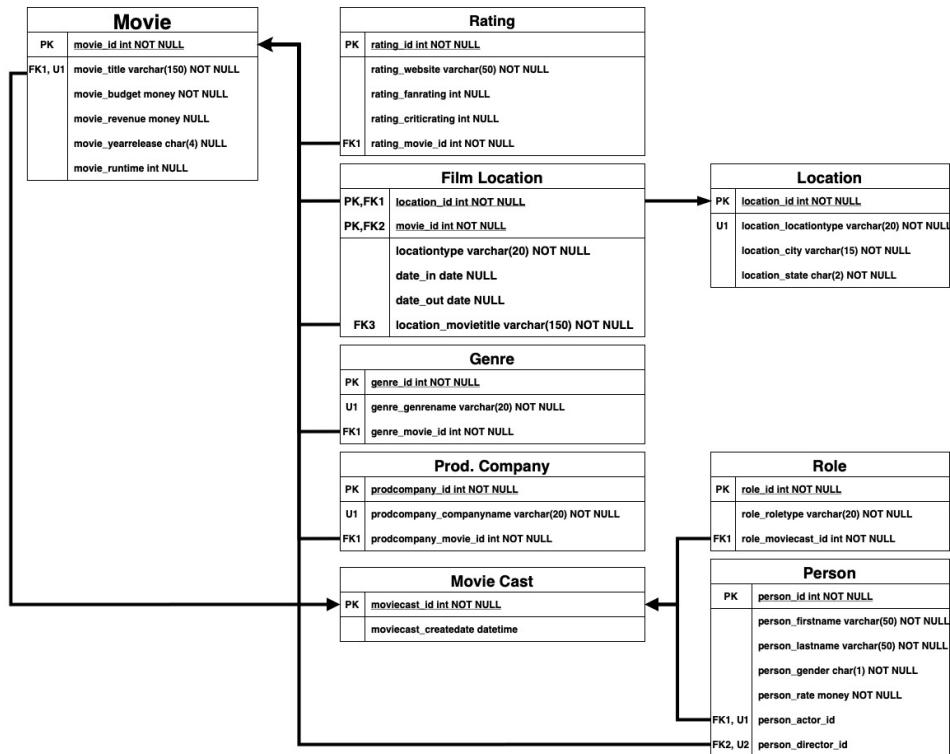
DATA QUESTIONS

1. What is the highest rated movie?
 - a. Who directed it?
 - b. Who was the main actor?
 - c. Is it the most profitable movie? What is its rank?
 - d. Did they spend more than their budget? How much did they save or how much did they go over their budget?
2. What is the lowest rated movie?
 - a. Who directed it?
 - b. Who was the main actor?
 - c. Did the movie earn any profit? How does it rank?
3. Which are the cheapest vs most expensive filming locations?
4. How many movies have filmed per location?
5. How many movies have the directors worked on?
6. How many movies have the actors starred in?
7. Which directors have the highest rated movies?
8. Which actors have the highest rated movies?
9. Who acts and directs the same movie?

CONCEPTUAL MODEL



LOGICAL MODEL



RAW DATA SAMPLE

The company currently houses their data in Excel, and it is broken down into 3 tabs: movies, locations, and persons. The tabs currently do not “talk” to one another. And the staff manually calculates the movie production costs on a calculator. The team members manually enter the data; however, it has caused multiple issues when more than one person is inputting data.

Movie Title	budget	revenue	year	run time	Actors	Director	Locations
New York Lights	200,000	525,690	2022	125	Gabby Smith, Kane Swift	Gabby Smith	Coffee Shop, Studio
Singing at Night	175,000	429,150	2020	165	Kane Swift, Alexa Payne	Luke Brown	Coffee Shop, Studio
Dog's Asleep	275,000	918,510	2014	95	Taylor Bryan, Luke Brown	James Conner	Beach, Studio
Painting on the Wall	150,000	329,430	2011	190	Grace Brady, Sam Combs	Cory Jordan	Beach, Park
Knock on the Door	300,000	639,180	2010	102	Taylor Bryan, Gabby Smith	Marie Styles	Park, Studio

Name	Pay
Luke Brown	15000
Gabby Smith	80000
Sam Combs	50000
Taylor Bryan	75000
Kane Swift	60000
James Conner	25000
Grace Brady	40000
Cory Jordan	20000
Alexa Payne	55000
Marie Styles	30000

locationtype	city	state	rentalrate
Beach	Malibu	CA	25000
Café	Los Angeles	CA	5000
Studio	Las Vegas	NV	20000
Park	Sacramento	CA	10000

PHYSICAL DATABASE CREATION

ALL DROPS

DROP PROCEDURE

```
drop procedure if exists p_search_movies
drop procedure if exists p_search_person
drop procedure if exists p_search_flr
drop procedure if exists p_upsert_newmovie
drop procedure if exists p_upsert_updatemovies
drop procedure if exists p_upsert_criticrating
drop procedure if exists p_upsert_fanrating
drop procedure if exists p_upsert_filmlocationrentalrate
drop procedure if exists p_upsert_personsrate
GO
```

DROP VIEWS

```
drop view if exists budget
drop view if exists totalRentalRatePerLocation
drop view if exists budgetinfo
drop view if exists filmschedule
GO
```

DROP TABLES

```
drop table if exists filmlocations
drop table if exists moviecast
drop table if exists ratings
drop table if exists locations
drop table if exists movies
drop table if exists prodcompany
drop table if exists genres
drop table if exists persons
drop table if exists role
GO
```

TABLE CREATIONS

CREATE TABLES

--CREATE ROLE TYPE TABLE

```
create table role (
```

```

roleid int primary key identity,
roletype varchar(50) not null,
constraint u_role_roletype unique (roletype),
)
GO

---CREATE PERSONS TABLE
create table persons (
    personid int primary key identity not null,
    firstname varchar(50) not null,
    lastname varchar(50) not null,
    gender char(1) not null,
    rate money not null,
    constraint ck_persons_gender_m_or_f check (gender = 'M' OR gender = 'F'),
    constraint ck_persons_firstname_length_gt_1 check (LEN(firstname)>1),
    constraint ck_persons_lastname_length_gt_1 check (LEN(lastname)>1)
)
GO

---CREATE GENRES TABLE
create table genres (
    genreid int primary key identity not null,
    genrename varchar(20) not null,
    constraint u_genres_genrename unique (genrename),
)
GO

---CREATE LOCATION TABLE
create table locations (
    locationid int primary key identity not null,
    locationtype varchar(20) not null,
    city varchar(15) not null,
    state char(2) not null,
    constraint u_locations_locationtype unique (locationtype),
    constraint ck_locations_state_length check (LEN(state)=2)
)
GO

---CREATE PRODUCTION COMPANY
create table prodcompany (
    companyid int primary key identity not null,
    companyname varchar(20) not null,
    constraint u_prod_company_companyname unique (companyname)
)
GO

---CREATE MOVIES TABLE
create table movies(
    movieid int primary key identity not null,
    movietitle varchar(150) not null,
    genre varchar(20) not null,
    budget money not null,

```

```

revenue money null,
yearrelease char(4) null,
runtime int null,
prodcompany varchar(20) not null,
director int not null,
constraint u_movies_movietitle unique (movieid,movietitle),
constraint u_movies_movietitle_prodcompany unique (movietitle,prodcompany),
constraint ck_movies_yearrelease_length check (LEN(yearrelease)=4 OR yearrele
ase = NULL),
constraint ck_movies_runtime check (runtime>0),
constraint fk_movies_prodcompany foreign key (prodcompany)
    references prodcompany(companyname),
constraint fk_movies_dir foreign key (director)
    references persons(personid),
constraint fk_movie_genre foreign key (genre)
    References genres(genrename)
)
GO

---CREATE RATINGS TABLE
create table ratings (
    ratingid int primary key identity not null,
    movieid int not null,
    website varchar(50) not null,
    fanrating int null,
    criticrating int null,
    check(fanrating >= 0 and fanrating <= 100),
    check(criticrating >= 0 and criticrating <= 100),
    constraint fk_ratings_movieid foreign key (movieid)
        references movies(movieid)
)
go
---CREATE MOVIECAST TABLE
create table moviecast (
    castid int primary key identity not null,
    role varchar(50) not null,
    actorid int not null,
    movietitle varchar(150) not null,
    prodcompany varchar(20) not null,
    createdate DATETIME DEFAULT getdate() ,
    constraint pk_moviecast_unique unique (role,actorid,movietitle),
    constraint pk_moviecast_unique_actor unique (actorid,movietitle),
    constraint fk_moviecast_role foreign key (role)
        references role(roletype),
    constraint fk_movie_movietitle foreign key (movietitle, prodcompany)
        references movies(movietitle, prodcompany),
    constraint fk_movie_cast_actor foreign key (actorid)
        references persons(personid)
)
GO

---CREATE TABLE FILMLOCATIONS

```

```

create table filmlocations (
    locationid int primary key identity not NULL,
    locationtype varchar(20) not null,
    movieid int not null,
    movietitle varchar(150) not null,
    rentalrate money not null,
    date_in date null,
    date_out date null,
    constraint pk_filmlocations_dates CHECK(date_in < date_out),
    constraint fk_filmlocations_locationtype foreign key (locationtype)
        references locations(locationtype),
    constraint fk_filmlocations_movietitle foreign key (movieid, movietitle)
        references movies(movieid,movietitle)
)
GO

```

TABLE INSERTIONS

-----TABLE INSERTS-----

```

---insert into role
insert into role (roletype)
values
    ('Main Actor'),
    ('Support Actor'),
    ('Director')
GO

--Read and verify role table
select * from role
GO

--insert into persons table
insert into persons (firstname, lastname, gender, rate)
values
    ('Luke', 'Brown', 'M', 15000),
    ('Gabby', 'Smith', 'F', 75000),
    ('Sam', 'Combs', 'M', 100000),
    ('Taylor', 'Bryan', 'F', 75000),
    ('Kane', 'Swift', 'M', 60000),
    ('James', 'Conner', 'M', 85000),
    ('Grace', 'Brady', 'F', 120000),
    ('Cory', 'Jordan', 'M', 90000),
    ('Alexa', 'Payne', 'F', 75000),
    ('Marie', 'Styles', 'F', 80000),
    ('Lionel', 'Brock', 'M', 91000),
    ('Gillian', 'Song', 'F', 65000),
    ('Johnnu', 'Skpe', 'M', 10000),
    ('Bryanna', 'Kobe', 'F', 75100),
    ('Joseph', 'Fox', 'M', 80000),
    ('Erica', 'Connel', 'F', 85000),
    ('Jennifer', 'Joestar', 'F', 95000),

```

```
('Tommy', 'Peaches', 'M', 50000),
('Ashley', 'Paul', 'F', 60350),
('Helena', 'Constintine', 'F', 57670),
('Yessin', 'Bizzare', 'M', 100000),
('Tiffany', 'Fortune', 'F', 98000),
('Tristen', 'Demar', 'M', 81000),
('Taeler', 'Taunts', 'F', 43050),
('Lisa', 'Dotty', 'F', 50250),
('Carmelo', 'Anthony', 'M', 87000),
('Christina', 'Betencourt', 'F', 67000),
('Ivan', 'Lopez', 'M', 93000),
('Camron', 'Booker', 'M', 96700),
('Lebron', 'James', 'M', 175060)
```

GO

```
--Read and verify persons table
select * from persons
GO
```

```
--insert into genres table
```

```
insert into genres (genrename)
values
('Horror'),
('Love'),
('Action'),
('Family'),
('Comedy'),
('Drama'),
('Children'),
('Documentary')
```

GO

```
--Read and verify genres table
select * from genres
GO
```

```
-- insert into location table
```

```
insert into locations (locationtype, city, state)
values
('Beach', 'Malibu', 'CA'),
('Coffee Shop', 'Los Angeles', 'CA'),
('Studio', 'Las Vegas', 'NV'),
('Park', 'Sacramento', 'CA'),
('Apartment', 'Manahttan', 'NY'),
('Loft ', 'San Francisco', 'CA'),
('City Streets', 'Philiadelphia', 'PE'),
('Amusement Park', 'Orlando', 'FL'),
```

```
('Wedding Hall','Los Vegas','NV'),
('Dojo','Sacramento','CA'),
('Skyscrapers','Chicago','IL'),
('Cruise Line','Honolulu','HI'),
('Barren Field','Austin','TX'),
('Restaurant 1','Santa Barbara','CA'),
('Restaurant 2','Newport Beach.','CA'),
('Restaurant 3','New York City','NY'),
('Prison','Troy','NY'),
('Bedroom','Hollywood','CA'),
('Animation Studio','San Antonio','TX'),
('Custom Themed Studio','Hollywood','CA')
```

GO

--Read and verify locations table

```
select * from locations
```

GO

--insert into prodcompany

```
insert into prodcompany (companynamne)
```

values

```
('Olive Butter'),
('Apple Sauce'),
('Visa Direct'),
('Three Kings'),
('Tiger Stripe'),
('Victory'),
('Atlantic Films'),
('Rest Well'),
('Film Club'),
('Rocket Movies'),
('MovieTime'),
('Popcorn Films'),
('Buttery PopCorn'),
('JawDrop'),
('IndyFilms'),
('Independent Network'),
('Happy Days'),
('Movies for All'))
```

GO

--Read and verify prodcompany table

```
select * from prodcompany
```

GO

--insert into movies table

```

insert into movies (movietitle, genre, budget, revenue, yearrelease,
runtime,prodcompany,director)
VALUES
    ('New York Lights', 'Horror', 200000, 525690, 2022, 125, 'Olive Butter', 2),
    ('Singing at Night', 'Comedy', 175000, 429150, 2020, 165, 'Apple Sauce', 1),
    ('Dogs Asleep', 'Family', 275000, 918510, 2014, 95, 'Visa Direct', 6),
    ('Painting on the Wall', 'Action', 150000, 329430, 2011, 190, 'Apple Sauce', 8),
    ('Knock on the Door', 'Drama', 300000, 639180, 2010, 102, 'Visa Direct', 10),
    ('First Kiss', 'Love', 400000, 5256900, 2022, 105, 'Three Kings', 2),
    ('Adventures at Night', 'Children', 375000, 909150, 2020, 162, 'Three Kings', 1),
    ('Ottors', 'Documentary', 275000, 918510, 2014, 95, 'Tiger Stripe', 6),
    ('Wall of China', 'Documentary', 150000, 329430, 2011, 180, 'Tiger Stripe', 8),
    ('Knockdown Love', 'Love', 300000, 639180, 2010, 102, 'Film Club', 10),
    ('Young Bucks', 'Children', 28000, 825690, 1999, 115, 'Victory', 2),
    ('Desert Nights', 'Documentary', 175000, 523150, 2016, 190, 'Atlantic Films', 1),
    ('Cats Asleep', 'Documentary', 275000, 318510, 2015, 95, 'Atlantic Films', 6),
    ('Painting with Kat', 'Children', 98000, 629430, 2014, 100, 'Rest Well', 8),
    ('Knockdown Love, The Reunion', 'Love', 400000, 939180, 2013, 137, 'Film Club', 10),
    ('The Journal', 'Love', 1300000, 225690, 2012, 120, 'Rocket Movies', 2),
    ('NightWing', 'Action', 175000, 529330, 2010, 85, 'MovieTime', 1),
    ('The Killing', 'Horror', 275000, 1918510, 2014, 100, 'MovieTime', 6),
    ('Bachelors Party', 'Comedy', 358000, 784330, 2019, 110, 'MovieTime', 8),
    ('The Court', 'Drama', 30000, 983380, 2000, 80, 'Popcorn Films', 10),
    ('Love Hurts', 'Love', 264080, 7840320, 2020, 105, 'Buttery PopCorn', 2),
    ('Sing Bee Sing!', 'Children', 95000, 672915, 2012, 115, 'Buttery PopCorn', 1),
    ('Baseball', 'Documentary', 275000, 1918510, 2022, 135, 'Buttery PopCorn', 6),
    ('Paris', 'Documentary', 80000, 129430, 2011, 190, 'JawDrop', 8),
    ('Love at First Arrest', 'Love', 630000, 2334900, 2015, 102, 'JawDrop', 10),
    ('Red Light Blue Light', 'Children', 50000, 1025690, 2002, 45, 'Victory', 2),
    ('Space', 'Documentary', 175000, 4290980, 2020, 120, 'Atlantic Films', 1),
    ('MARS ROVERS', 'Documentary', 275000, 318450, 2021, 125, 'Atlantic Films', 6),
    ('Card Fight', 'Children', 550000, 3329430, 2018, 85, 'Rest Well', 8),
    ('Family Love', 'Love', 430000, 635430, 2010, 60, 'IndyFilms', 10),
    ('Divorce Again?!!', 'Love', 200000, 225690, 2016, 95, 'IndyFilms', 2),
    ('DayWing', 'Action', 205000, 320150, 2001, 95, 'IndyFilms', 1),
    ('Dont Fail', 'Horror', 475000, 4918510, 2004, 65, 'IndyFilms', 6),
    ('Family Guy', 'Comedy', 150000, 1329430, 2011, 90, 'MovieTime', 8),
    ('The Challenge', 'Drama', 309200, 2639180, 2016, 102, 'Movies for All', 10),
    ('Family Reunion', 'Family', 267000, 3525690, 2017, 55, 'Movies for All', 2),
    ('Santa Escape', 'Children', 575000, 5429150, 2020, 65, 'Movies for All', 1),
    ('IT Today', 'Documentary', 235000, 718000, 2018, 85, 'Happy Days', 6),
    ('City Paint', 'Documentary', 950000, 4299430, 2009, 90, 'Happy Days', 8),
    ('Love Knocks Twice', 'Love', 100000, 239180, 2019, 120, 'Independent Network', 10),
    ('New York Christmas', 'Family', 200000, 520090, 2022, 80, 'Happy Days', 2),
    ('American Pre-School', 'Family', 175000, 425000, 2000, 95, 'Independent
Network', 1),
    ('Elements', 'Family', 275000, 780010, 2021, 100, 'Atlantic Films', 6),
    ('Teddy', 'Children', 150000, 809800, 2010, 85, 'Rest Well', 8),
    ('Supergirl', 'Action', 300000, 534180, 2003, 125, 'Film Club', 10),
    ('The Hill', 'Action', 200000, 1280090, 2022, 130, 'Rocket Movies', 2),
    ('Rangers', 'Action', 175000, 329150, 2020, 65, 'MovieTime', 1),
    ('Death by Sleep', 'Horror', 275000, 3318510, 2014, 105, 'Olive Butter', 6),

```

```

('LOL', 'Comedy', 850000, 3570000, 2011, 100, 'Independent Network', 8),
('The Election Campaign', 'Drama', 200000, 560080, 2018, 100, 'Olive Butter', 10),
('Love Lost', 'Love', 600000, 1525690, 2009, 50, 'Apple Sauce', 2),
('1st Grade', 'Children', 375000, 429150, 2015, 65, 'Apple Sauce', 1),
('Runaway', 'Drama', 275000, 604510, 2004, 75, 'Apple Sauce', 6),
('Family Vacation', 'Family', 150000, 390520, 2022, 80, 'Visa Direct', 8),
('Lovers', 'Love', 600000, 639180, 2022, 55, 'Visa Direct', 10),
('Dora', 'Children', 2000000, 5925690, 2021, 60, 'Visa Direct', 2),
('Minions', 'Family', 1750000, 7429150, 2021, 65, 'Visa Direct', 1),
('Apples of the World', 'Documentary', 975000, 1851000, 2013, 85, 'Atlantic
Films', 6),
('Avatar', 'Children', 850000, 5329430, 2012, 110, 'Rest Well', 8),
('Kevin Hart', 'Comedy', 300000, 450000, 2017, 100, 'Film Club', 10),
('George Lopez', 'Comedy', 700000, 5200000, 202, 95, 'Rocket Movies', 2),
('Dragonball Super', 'Action', 170000, 4250000, 2000, 105, 'MovieTime', 1),
('Hades', 'Horror', 275000, 780000, 2014, 95, 'MovieTime', 6),
('Bullet Train', 'Comedy', 155000, 379000, 2011, 60, 'MovieTime', 8),
('The Last Will', 'Drama', 500000, 709090, 2010, 90, 'Popcorn Films', 10)

```

GO

--Read and verify movies table

```
select * from movies
```

GO

-- insert into ratings table

---insert into ratings table

```
insert into ratings (movieid,website, fanrating,criticrating)
values
(1, 'Rotten Apples', 70, 76),(1, 'Movie Nation', 77, 82),(1, 'Critic
Corner', 90, 81),(1, 'Movie Monster', 68, 74),(2, 'Rotten Apples', 90, 71),
(2, 'Movie Nation', 81, 73),(2, 'Critic Corner', 84, 78),(2, 'Movie
Monster', 93, 70),(3, 'Rotten Apples', 100, 83),(3, 'Movie Nation', 92, 70),
(3, 'Critic Corner', 96, 79),(3, 'Movie Monster', 90, 71),(4, 'Rotten
Apples', 72, 90),(4, 'Movie Nation', 80, 92),(4, 'Critic Corner', 78, 87),
(4, 'Movie Monster', 71, 81),(5, 'Rotten Apples', 90, 73),(5, 'Movie
Nation', 84, 80),(5, 'Movie Monster', 86, 77),(5, 'Critic Corner', 87, 75),
(6, 'Rotten Apples', 97, 64),(7, 'Movie Nation', 99, 53),(8, 'Critic
Corner', 37, 70),(9, 'Movie Monster', 53, 73),(10, 'Rotten Apples', 35, 35),
(11, 'Movie Nation', 42, 59),(12, 'Critic Corner', 76, 34),(13, 'Movie
Monster', 30, 49),(14, 'Rotten Apples', 52, 65),(15, 'Movie Nation', 96, 72),
(16, 'Critic Corner', 94, 98),(17, 'Movie Monster', 97, 60),(18, 'Rotten
Apples', 67, 44),(19, 'Movie Nation', 55, 67),(20, 'Critic Corner', 88, 48),
(21, 'Movie Monster', 42, 96),(22, 'Rotten Apples', 68, 56),(23, 'Movie
Nation', 34, 49),(24, 'Critic Corner', 66, 83),(25, 'Movie Monster', 39, 84),
(26, 'Rotten Apples', 60, 66),(27, 'Movie Nation', 87, 74),(28, 'Critic
Corner', 95, 39),(29, 'Movie Monster', 33, 95),(30, 'Rotten Apples', 92, 75),
(31, 'Movie Nation', 52, 60),(32, 'Critic Corner', 94, 40),(33, 'Movie
Monster', 77, 81),(34, 'Rotten Apples', 90, 65),(35, 'Movie Nation', 84, 87),
(36, 'Critic Corner', 98, 81),(37, 'Movie Monster', 94, 99),(38, 'Rotten
Apples', 37, 63),(39, 'Movie Nation', 62, 70),(40, 'Critic Corner', 34, 52),
```

```
(41, 'Movie Monster', 47, 96), (42, 'Rotten Apples', 60, 63), (43, 'Movie Nation', 45, 84), (44, 'Critic Corner', 61, 45), (45, 'Movie Monster', 64, 36),
(46, 'Rotten Apples', 38, 47), (47, 'Movie Nation', 41, 75), (48, 'Critic Corner', 35, 77), (49, 'Movie Monster', 82, 50), (50, 'Rotten Apples', 95, 47),
(51, 'Movie Nation', 84, 51), (52, 'Critic Corner', 95, 33), (53, 'Movie Monster', 45, 89), (54, 'Rotten Apples', 67, 64), (55, 'Movie Nation', 82, 96),
(56, 'Critic Corner', 82, 94), (57, 'Movie Monster', 96, 77), (58, 'Rotten Apples', 88, 86), (59, 'Movie Nation', 53, 46), (60, 'Critic Corner', 75, 90),
(61, 'Movie Monster', 91, 93), (62, 'Rotten Apples', 44, 65), (63, 'Movie Nation', 63, 47), (64, 'Critic Corner', 95, 80), (65, 'Movie Monster', 88, 65)
GO
```

```
--Read and verify ratings table
select * from ratings
GO
```

```
---insert into moviecast
insert into moviecast (role, actorid,movietitle,prodcompany)
values
('Main Actor',2,'New York Lights','Olive Butter'),('Support Actor',5,'New York Lights','Olive Butter'),
('Main Actor',5,'Singing at Night','Apple Sauce'),('Support Actor',9,'Singing at Night','Apple Sauce'),
('Main Actor',4,'Dogs Asleep','Visa Direct'),('Support Actor',1,'Dogs Asleep','Visa Direct'),
('Main Actor',7,'Painting on the Wall','Apple Sauce'),('Support Actor',3,'Painting on the Wall','Apple Sauce'),
('Main Actor',4,'Knock on the Door','Visa Direct'),('Support Actor',2,'Knock on the Door','Visa Direct'),
('Main Actor',29,'First Kiss','Three Kings'),('Support Actor',13,'First Kiss','Three Kings'),
('Main Actor',9,'Adventures at Night','Three Kings'),('Support Actor',27,'Adventures at Night','Three Kings'),
('Main Actor',12,'Ottors','Tiger Stripe'),('Support Actor',25,'Ottors','Tiger Stripe'),
('Main Actor',15,'Wall of China','Tiger Stripe'),('Support Actor',27,'Wall of China','Tiger Stripe'),
('Main Actor',13,'Knockdown Love','Film Club'),('Support Actor',3,'Knockdown Love','Film Club'),
('Main Actor',7,'Young Bucks','Victory'),('Support Actor',9,'Young Bucks','Victory'),
('Main Actor',9,'Desert Nights','Atlantic Films'),('Support Actor',7,'Desert Nights','Atlantic Films'),
('Main Actor',8,'Cats Asleep','Atlantic Films'),('Support Actor',10,'Cats Asleep','Atlantic Films'),
('Main Actor',22,'Painting with Kat','Rest Well'),('Support Actor',15,'Painting with Kat','Rest Well'),
('Main Actor',21,'Knockdown Love, The Reunion','Film Club'),('Support Actor',23,'Knockdown Love, The Reunion','Film Club'),
```

('Main Actor', 20, 'The Journal', 'Rocket Movies'), ('Support Actor', 3, 'The Journal', 'Rocket Movies'),
 ('Main Actor', 29, 'NightWing', 'MovieTime'), ('Support Actor', 9, 'NightWing', 'MovieTime'),
 ('Main Actor', 8, 'The Killing', 'MovieTime'), ('Support Actor', 20, 'The Killing', 'MovieTime'),
 ('Main Actor', 7, 'Bachelors Party', 'MovieTime'), ('Support Actor', 21, 'Bachelors Party', 'MovieTime'),
 ('Main Actor', 29, 'The Court', 'Popcorn Films'), ('Support Actor', 22, 'The Court', 'Popcorn Films'),
 ('Main Actor', 24, 'Love Hurts', 'Buttery PopCorn'), ('Support Actor', 29, 'Love Hurts', 'Buttery PopCorn'),
 ('Main Actor', 1, 'Sing Bee Sing!', 'Buttery PopCorn'), ('Support Actor', 7, 'Sing Bee Sing!', 'Buttery PopCorn'),
 ('Main Actor', 28, 'Baseball', 'Buttery PopCorn'), ('Support Actor', 17, 'Baseball', 'Buttery PopCorn'),
 ('Main Actor', 1, 'Paris', 'JawDrop'), ('Support Actor', 21, 'Paris', 'JawDrop'),
 ('Main Actor', 12, 'Love at First Arrest', 'JawDrop'), ('Support Actor', 18, 'Love at First Arrest', 'JawDrop'),
 ('Main Actor', 25, 'Red Light Blue Light', 'Victory'), ('Support Actor', 28, 'Red Light Blue Light', 'Victory'),
 ('Main Actor', 29, 'Space', 'Atlantic Films'), ('Support Actor', 5, 'Space', 'Atlantic Films'),
 ('Main Actor', 10, 'MARS ROVERS', 'Atlantic Films'), ('Support Actor', 5, 'MARS ROVERS', 'Atlantic Films'),
 ('Main Actor', 19, 'Card Fight', 'Rest Well'), ('Support Actor', 10, 'Card Fight', 'Rest Well'),
 ('Main Actor', 25, 'Family Love', 'IndyFilms'), ('Support Actor', 16, 'Family Love', 'IndyFilms'),
 ('Main Actor', 17, 'Divorce Again?!', 'IndyFilms'), ('Support Actor', 10, 'Divorce Again?!', 'IndyFilms'),
 ('Main Actor', 12, 'DayWing', 'IndyFilms'), ('Support Actor', 13, 'DayWing', 'IndyFilms'),
 ('Main Actor', 24, 'Dont Fail', 'IndyFilms'), ('Support Actor', 2, 'Dont Fail', 'IndyFilms'),
 ('Main Actor', 18, 'Family Guy', 'MovieTime'), ('Support Actor', 17, 'Family Guy', 'MovieTime'),
 ('Main Actor', 1, 'The Challenge', 'Movies for All'), ('Support Actor', 24, 'The Challenge', 'Movies for All'),
 ('Main Actor', 3, 'Family Reunion', 'Movies for All'), ('Support Actor', 25, 'Family Reunion', 'Movies for All'),
 ('Main Actor', 3, 'Santa Escape', 'Movies for All'), ('Support Actor', 22, 'Santa Escape', 'Movies for All'),
 ('Main Actor', 16, 'IT Today', 'Happy Days'), ('Support Actor', 28, 'IT Today', 'Happy Days'),
 ('Main Actor', 21, 'City Paint', 'Happy Days'), ('Support Actor', 11, 'City Paint', 'Happy Days'),
 ('Main Actor', 8, 'Love Knocks Twice', 'Independent Network'), ('Support Actor', 10, 'Love Knocks Twice', 'Independent Network'),
 ('Main Actor', 27, 'New York Christmas', 'Happy Days'), ('Support Actor', 10, 'New York Christmas', 'Happy Days'),

```

        ('Main Actor',19,'American Pre-School','Independent Network'),('Support
Actor',13,'American Pre-School','Independent Network'),
        ('Main Actor',25,'Elements','Atlantic Films'),('Support
Actor',12,'Elements','Atlantic Films'),
        ('Main Actor',11,'Teddy','Rest Well'),('Support Actor',20,'Teddy','Rest
Well'),
        ('Main Actor',14,'Supergirl','Film Club'),('Support
Actor',24,'Supergirl','Film Club'),
        ('Main Actor',15,'The Hill','Rocket Movies'),('Support Actor',3,'The
Hill','Rocket Movies'),
        ('Main Actor',6,'Rangers','MovieTime'),('Support
Actor',5,'Rangers','MovieTime'),
        ('Main Actor',2,'Death by Sleep','Olive Butter'),('Support Actor',26,'Death
by Sleep','Olive Butter'),
        ('Main Actor',16,'L0L','Independent Network'),('Support
Actor',23,'L0L','Independent Network'),
        ('Main Actor',29,'The Election Campaign','Olive Butter'),('Support
Actor',2,'The Election Campaign','Olive Butter'),
        ('Main Actor',21,'Love Lost','Apple Sauce'),('Support Actor',27,'Love
Lost','Apple Sauce'),
        ('Main Actor',7,'1st Grade','Apple Sauce'),('Support Actor',1,'1st
Grade','Apple Sauce'),
        ('Main Actor',9,'Runaway','Apple Sauce'),('Support Actor',3,'Runaway','Apple
Sauce'),
        ('Main Actor',12,'Family Vacation','Visa Direct'),('Support Actor',14,'Family
Vacation','Visa Direct'),
        ('Main Actor',25,'Lovers','Visa Direct'),('Support Actor',26,'Lovers','Visa
Direct'),
        ('Main Actor',9,'Dora','Visa Direct'),('Support Actor',2,'Dora','Visa
Direct'),
        ('Main Actor',10,'Minions','Visa Direct'),('Support Actor',21,'Minions','Visa
Direct'),
        ('Main Actor',22,'Apples of the World','Atlantic Films'),('Support
Actor',14,'Apples of the World','Atlantic Films'),
        ('Main Actor',15,'Avatar','Rest Well'),('Support Actor',1,'Avatar','Rest
Well'),
        ('Main Actor',11,'Kevin Hart','Film Club'),('Support Actor',26,'Kevin
Hart','Film Club'),
        ('Main Actor',13,'George Lopez','Rocket Movies'),('Support Actor',25,'George
Lopez','Rocket Movies'),
        ('Main Actor',9,'Dragonball Super','MovieTime'),('Support
Actor',29,'Dragonball Super','MovieTime'),
        ('Main Actor',25,'Hades','MovieTime'),('Support
Actor',7,'Hades','MovieTime'),
        ('Main Actor',12,'Bullet Train','MovieTime'),('Support Actor',10,'Bullet
Train','MovieTime'),
        ('Main Actor',9,'The Last Will','Popcorn Films'),('Support Actor',26,'The
Last Will','Popcorn Films')
GO

```

```
--Read and verify moviecast table
select * from moviecast
```

GO

```
--insert into filmlocations
insert into filmlocations (locationtype, movieid,
movietitle,rentalrate,date_in,date_out)
values
    ('Beach',3,(select movietitle from movies where movieid=3),20000,'2012-04-
06','2012-06-01'),
    ('Coffee Shop',2,(select movietitle from movies where movieid=2),15000,'2018-
04-06','2018-05-02'),
    ('Park',5,(select movietitle from movies where movieid=5),10000,'2008-12-
04','2008-12-30'),
    ('Coffee Shop',1,(select movietitle from movies where movieid=1),15000,'2020-
08-27','2020-09-01'),
    ('Park',4,(select movietitle from movies where movieid=4),15000,'2010-04-
29','2010-05-16'),
    ('Coffee
Shop',20,(select movietitle from movies where movieid=20),17500,'1998-07-
16','1998-10-05'),
    ('Studio',42,(select movietitle from movies where movieid=42),13000,'2000-
01-17','2000-02-15'),
    ('Park',62,(select movietitle from movies where movieid=62),13000,'1999-02-
14','1999-04-10'),
    ('Apartment',32,(select movietitle from movies where movieid=32),12000,'1
999-11-08 ','1999-12-06'),
    ('Loft',26,(select movietitle from movies where movieid=26),112000,'2000-04-
17','2000-12-14'),
    ('City Streets',45,
(select movietitle from movies where movieid=45),12000,'2001-09-05 ','2001-10-
16'),
    ('Amusement
Park',33,(select movietitle from movies where movieid=33),12000,'2002-06-
27','2002-12-18'),
    ('Wedding Hall',53,
(select movietitle from movies where movieid=53),14000,'2002-07-16','2002-09-
16'),
    ('Dojo',39,(select movietitle from movies where movieid=39),12000,'2007-07-
17','2007-11-06'),
    ('Skyscrapers',51,(select movietitle from movies where movieid=51),19000,'20
07-09-03','2007-11-28'),
    ('Cruise
Line',10,(select movietitle from movies where movieid=10),12000,'2008-05-
04','2008-08-11'),
    ('Barren
Field',17,(select movietitle from movies where movieid=17),41000,'2008-06-
16','2008-08-24'),
    ('Restaurant
1',30,(select movietitle from movies where movieid=30),17500,'2008-04-30','2008-
11-08'),
```

```

        ('Restaurant
2',44,(select movietitle from movies where movieid=44),19200,'2008-10-29','2008-
12-06'),
        ('Restaurant
3',65,(select movietitle from movies where movieid=65),18700,'2008-09-29','2008-
12-03'),
        ('Prison',9,(select movietitle from movies where movieid=9),15000,'2009-01-
19','2009-03-21'),
        ('Bedroom',24,(select movietitle from movies where movieid=24),12000,'2009-
10-31','2009-12-11'),
        ('Animation
Studio',34,(select movietitle from movies where movieid=34),11000,'2009-08-
08','2009-10-20'),
        ('Custom Themed
Studio',49,(select movietitle from movies where movieid=49),15000,'2009-04-
11','2009-07-13'),
        ('Studio',64,(select movietitle from movies where movieid=64),12000,'2009-09-
13','2010-02-10'),
        ('Park',16,(select movietitle from movies where movieid=16),11000,'2010-10-
13','2010-11-26'),
        ('Apartment',22,(select movietitle from movies where movieid=22),19200,'2010-
10-30','2011-01-30'),
        ('Loft',59,(select movietitle from movies where movieid=59),11200,'2010-03-
02','2010-05-31'),
        ('City
Streets',15,(select movietitle from movies where movieid=15),16000,'2011-07-
11','2011-08-29'),
        ('Amusement
Park',58,(select movietitle from movies where movieid=58),11500,'2011-04-
10','2011-5-24'),
        ('Wedding
Hall',8,(select movietitle from movies where movieid=8),14000,'2012-03-05','2012-
04-19'),
        ('Dojo',14,(select movietitle from movies where movieid=14),12000,'2012-02-
16','2012-10-11'),
        ('Skyscrapers',18,(select movietitle from movies where movieid=18),19000,'201
2-03-31','2012-10-11'),
        ('Cruise
Line',48,(select movietitle from movies where movieid=48),12000,'2012-02-
13','2012-12-07'),
        ('Barren
Field',63,(select movietitle from movies where movieid=63),11100,'2012-01-
08','2012-02-13'),
        ('Restaurant
1',13,(select movietitle from movies where movieid=13),17500,'2013-08-16','2013-
10-05'),
        ('Restaurant
2',25,(select movietitle from movies where movieid=25),19200,'2013-07-30','2013-
11-18'),
        ('Restaurant
3',52,(select movietitle from movies where movieid=52),18700,'2013-09-24','2013-
11-18'),

```

```
('Prison',12,(select movietitle from movies where movieid=12),15000,'2014-01-15','2014-03-25'),
('Bedroom',31,(select movietitle from movies where movieid=31),12000,'2014-03-24','2015-01-20'),
('Animation Studio',35,(select movietitle from movies where movieid=35),11800,'2014-06-26','2014-09-16'),
('Custom Themed Studio',36,(select movietitle from movies where movieid=36),21500,'2015-08-24','2015-10-20'),
('Studio',60,(select movietitle from movies where movieid=60),12000,'2015-07-25','2015-08-21'),
('Park',29,(select movietitle from movies where movieid=29),11000,'2016-04-28','2016-09-19'),
('Apartment',38,(select movietitle from movies where movieid=38),19200,'2016-02-20','2016-03-13'),
('Loft',50,(select movietitle from movies where movieid=50),12000,'2016-05-09','2017-01-19'),
('City Streets',19,(select movietitle from movies where movieid=19),16000,'2018-06-26','2018-10-15'),
('Amusement Park',40,(select movietitle from movies where movieid=40),11500,'2018-10-18','2018-12-31'),
('Wedding Hall',61,(select movietitle from movies where movieid=61),14000,'2018-05-28','2018-09-21'),
('Dojo',7,(select movietitle from movies where movieid=7),12000,'2018-02-02','2018-07-21'),
('Skyscrapers',21,(select movietitle from movies where movieid=21),19000,'2018-02-28','2018-10-19'),
('Cruise Line',27,(select movietitle from movies where movieid=27),12000,'2019-07-01','2019-08-17'),
('Beach',37,(select movietitle from movies where movieid=37),11000,'2019-04-09','2019-09-16'),
('Coffee Shop',47,(select movietitle from movies where movieid=47),17500,'2019-03-31','2019-11-02'),
('Studio',28,(select movietitle from movies where movieid=28),19200,'2019-02-13','2019-06-26'),
('Park',43,(select movietitle from movies where movieid=43),18700,'2020-02-26','2020-04-28'),
('Apartment',56,(select movietitle from movies where movieid=56),15000,'2020-07-12','2020-12-12'),
('Loft',57,(select movietitle from movies where movieid=57),12000,'2020-01-19','2020-03-24'),
('City Streets',6,(select movietitle from movies where movieid=6),11800,'2020-03-08','2020-07-24'),
```

```

('Amusement
Park',23,(select movietitle from movies where movieid=23),11500,'2021-07-
13','2021-08-13'),
('Wedding
Hall',41,(select movietitle from movies where movieid=41),14000,'2021-07-
25','2021-09-26'),
('Dojo',46,(select movietitle from movies where movieid=46),12000,'2021-06-
30','2021-09-04'),
('Skyscrapers',54,(select movietitle from movies where movieid=54),91000,'202
1-02-17','2021-07-06'),
('Cruise
Line',55,(select movietitle from movies where movieid=55),14000,'2021-06-
16','2021-08-17')
GO

--Read and verify filmlocations table
select * from filmlocations
GO

```

VIEW CREATIONS

```

-----BUDGET VIEW-----
---This view allows users to see the movies financial performance and its total
revenue
--calculations include: initial budget, actual budget (actor cost, director cost,
location cost), theater revenue, and total revenue
create view budget as

with moviebudget as (
select distinct mc.movietitle, m.budget,
sum(p.rate) over (partition by mc.movietitle) as actorcost,
d.rate as directorcost,
fl.rentalrate as locationcost
from moviecast mc , movies m , persons p, filmlocations fl, persons d
where mc.movietitle = m.movietitle
and p.personid = mc.actorid
and d.personid = m.director
and fl.movieid=m.movieid
group by mc.movietitle , m.budget , mc.actorid, p.rate , d.rate , fl.rentalrate
)
select distinct mb.movietitle, m.genre, actorcost, directorcost, locationcost,
mb.budget as initial_budget,
sum(actorcost + directorcost + locationcost) over (partition by m.movietitle,
mb.movietitle) as actual_budget,
m.revenue as theater_revenue,
m.revenue - sum(actorcost + directorcost + locationcost) over (partition by m.mov
ietitle, mb.movietitle) total_revenue
from moviebudget mb
left join movies m on mb.movietitle = m.movietitle
GO

```

```
select movietitle, initial_budget, actual_budget,
(case
    when actual_budget > 0 then 'Under Budget'
    when actual_budget < 0 then 'Over Budget'
    else 'On Budget Target'
    end) 'Budget Classification'
from budget
GO
```

```
-----BUDGETINFO VIEW-----
create view budgetinfo as (
select movietitle, initial_budget, actual_budget, total_revenue,
(case
    when initial_budget > actual_budget then 'Under Budget'
    when initial_budget < actual_budget then 'Over Budget'
    else 'On Budget Target'
    end) 'Budget_Category',
(case
    when total_revenue < 0 then 'No Profit'
    when total_revenue > 0 then 'Profit'
    else 'Breakeven'
    end) 'Profit'
from budget
)
GO
```

```
-----SUM OF PERSON RATES VIEW-----
---This view generates the cumulative amount of how much each person has earned
create or alter view totalPersonsRate as
    select personid as personid, (firstname+ ' '+lastname) as 'fullName',
    sum(rate) as totalPersonsRate
        from persons
    innerjoin
    order by totalPersonsRate
GO
```

```
-----SUM OF PERSON RATES VIEW-----
--This view generates a cumulative sum of how much each location has earned
create or alter view totalRentalRatePerLocation as
    select locationid as location, sum(rate) as totalRentalRatePerLocation
        from filmlocations
    order by totalRentalRatePerLocation
GO
```

PROCEDURES

--UPDATE PERSONS RATE-----

```
CREATE PROCEDURE p_upsert_personsrate(
    @personid int,
    @rate money)
AS
if exists (select * from persons where personid=@personid)
    update persons
        set rate = @rate
        where personid = @personid
else
    insert persons(personid, rate)
        values(@personid, @rate)
GO
```

--EXECUTE p_upsert_personsrate

```
select * from persons where personid = 2
go
exec p_upsert_personsrate @personid=2, @rate=90000
GO
select * from persons where personid = 2
go
```

Results					
	personid	firstname	lastname	gender	rate
1	2	Gabby	Smith	F	80000.00

Messages					
	personid	firstname	lastname	gender	rate
1	2	Gabby	Smith	F	90000.00

--UPDATE FILMLOCATION RENTALRATE-----

```
CREATE PROCEDURE p_upsert_filmlocationrentalrate(
    @locationid int,
    @rentalrate money
)
AS
if exists (select * from filmlocations where locationid=@locationid )
    update filmlocations
        set rentalrate = @rentalrate
        where locationid = @locationid
else
    insert filmlocations(locationid,rentalrate)
        values(@locationid, @rentalrate)
GO
```

--EXECUTE p_upsert_filmlocationrentalrate

```

select * from filmlocations where locationid = 1
GO
exec p_upsert_filmlocationrentalrate @locationid=1, @rentalrate=250000
GO
select * from filmlocations where locationid = 1
GO

```

Results Messages

	locationid	locationtype	movieid	movietitle	rentalrate	date_in	date_out
1	1	Beach	3	Dogs Asleep	30000.00	2012-04-06	2012-06-01

	locationid	locationtype	movieid	movietitle	rentalrate	date_in	date_out
1	1	Beach	3	Dogs Asleep	250000.00	2012-04-06	2012-06-01

-----UPDATE MOVIE FAN RATING-----

```

CREATE PROCEDURE p_upsert_fanrating(
    @ratingid int,
    @movieid int,
    @fanrating int
)
AS
if exists (select * from ratings where ratingid=@ratingid and movieid = @movieid )
    update ratings
        set fanrating = @fanrating
        where ratingid = @ratingid and movieid = @movieid
else
    insert ratings(ratingid,movieid, fanrating)
        values(@ratingid,@movieid, @fanrating)
GO

--EXECUTE p_upsert_fanrating
select * from ratings where movieid = 1 and ratingid =3
GO
exec p_upsert_fanrating @ratingid=3, @movieid=1, @fanrating=89
GO
select * from ratings where movieid = 1 and ratingid =3
GO

```

Results Messages

	ratingid	movieid	website	fanrating	criticrating
1	3		Critic Corner	88	80
			Show Filter		

	ratingid	movieid	website	fanrating	criticrating
1	3	1	Critic Corner	89	80

-----UPDATE MOVIE/CRITIC RATING-----

```

CREATE PROCEDURE p_upsert_criticrating(
    @ratingid int,
    @movieid int,
    @criticrating int
)
AS
if exists (select * from ratings where ratingid=@ratingid and movieid = @movieid )
    update ratings
        set criticrating = @criticrating
        where ratingid = @ratingid and movieid = @movieid
else
    insert into ratings(ratingid, movieid, criticrating)
    values(@ratingid, @movieid, @criticrating)
GO

```

```

select * from ratings where movieid=1 and ratingid = 3
go
exec p_upsert_criticrating @ratingid=3, @movieid=1, @criticrating=79
GO
select * from ratings where movieid=1 and ratingid = 3
GO

```

Results Messages

	ratingid	movieid	website	fanrating	criticrating
1	3	1	Critic Corner	89	80
			Show Filter		

	ratingid	movieid	website	fanrating	criticrating
1	3	1	Critic Corner	89	79

```
--SEARCH FOR RENTAL RATES-----
CREATE proc p_search_flr
( @rentalrate money)
as BEGIN
select
    [locationtype]
    ,[movieid]
    ,[movietitle]
    ,[rentalrate]
    , [date_in]
    ,[date_out]
from filmlocations
where rentalrate = @rentalrate or rentalrate < @rentalrate
END
GO

--EXECUTE p_search_flr
EXEC p_search_flr 6000
GO
```

	locationtype	movieid	movietitle	rentalrate	date_in	date_out
1	Coffee Shop	2	Singing at Night	5000.00	2018-04-06	2018-05-02
2	Coffee Shop	1	New York Lights	5000.00	2020-08-27	2020-09-01
3	Park	4	Painting on the Wall	5000.00	2010-04-29	2010-05-16

```
--SEARCH FOR PERSONS-----
CREATE proc p_search_person
(@name varchar(MAX))
as BEGIN
select *
from persons
where firstname like @name+'%' or lastname like @name+'%'
or gender like @name+'%'
END
GO

--EXECUTE p_search_person
EXEC p_search_person 'j'
GO
```

	personid	firstname	lastname	gender	rate
1	6	James	Conner	M	25000.00
2	8	Cory	Jordan	M	20000.00
3	13	johnnu	Skpe	M	5000.00
4	15	Joseph	Fox	M	8000.00
5	17	Jennifer	Joestar	F	4500.00
6	30	Lebron	James	M	4560.00

----SEARCH FOR MOVIES BY TITLE, RELEASE YEAR, GENRE, OR PRODUCTION COMPANY----

```
CREATE proc p_search_movies
(@name varchar(MAX))
as BEGIN
select *
from movies
where movietitle like @name+'%' or genre like @name+'%'
or yearrelease like @name+'%' or prodcompany like @name+'%'
END
GO
```

--EXECUTE p_search_movies

```
EXEC p_search_movies 2020
GO
```

	movieid	movietitle	genre	budget	revenue	yearrelease	runtime	prodcompany	director
1	2	Singing at Night	Comedy	175000.00	429150.00	2020	165	Apple Sauce	1
2	7	Adventures at Night	Children	375000.00	909150.00	2020	162	Three Kings	1
3	21	Love Hurts	Love	264080.00	7840320.00	2020	105	Buttery PopCorn	2
4	27	Space	Documentary	175000.00	4290980.00	2020	120	Atlantic Films	1
5	37	Santa Escape	Children	575000.00	5429150.00	2020	65	Movies for All	1
6	47	Rangers	Action	175000.00	329150.00	2020	65	MovieTime	1

-----ADD NEW MOVIE-----

```
CREATE PROCEDURE p_upsert_newmovie
(@movietitle varchar(150),
@genre varchar(20),
@budget money,
@prodcompany varchar(20),
@director int) as
BEGIN
insert into movies(movietitle, genre, budget, prodcompany, director)
values (@movietitle, @genre, @budget, @prodcompany, @director)
END
GO
```

```

select * from movies where movietitle = 'Ghosts in the Wild'
GO
exec p_upsert_newmovie 'Ghosts in the Wild','Horror','250000','Olive Butter',2
GO
select * from movies where movietitle = 'Ghosts in the Wild'
GO

```

Results Messages									
	movieid	movietitle	genre	budget	revenue	yearrelease	runtime	prodcompany	director
1	66	Ghosts in the Wild	Horror	250000.00	NULL	NULL	NULL	Olive Butter	2

-----UPDATE MOVIES TABLE-----

```

CREATE PROCEDURE p_upsert_updatemoviestable
    (@movieid int,
     @revenue money,
     @yearrelease char(4),
     @runtime int) as
if exists (select * from movies where movieid=@movieid)
BEGIN
    UPDATE movies
    set revenue = @revenue,
    yearrelease = @yearrelease,
    runtime = @runtime
    where movieid = @movieid
    and revenue is NULL
    and yearrelease is null
    and runtime is null
END
go

```

```

--EXECUTE p_upsert_updatemoviestable
select * from movies where movieid = 66
GO
exec p_upsert_updatemoviestable 66,582018,2022,164
GO
select * from movies where movieid = 66
GO

```

Results Messages										
	movieid	movietitle	genre	budget	revenue	yearrelease	runtime	prodcompany	director	
1	66	Ghosts in the Wild	Horror	250000.00	NULL	NULL	NULL	Olive Butter	2	
	1	66	Ghosts in the Wild	Horror	250000.00	582018.00	2022	164	Olive Butter	2

DATA QUESTIONS & VALIDATION

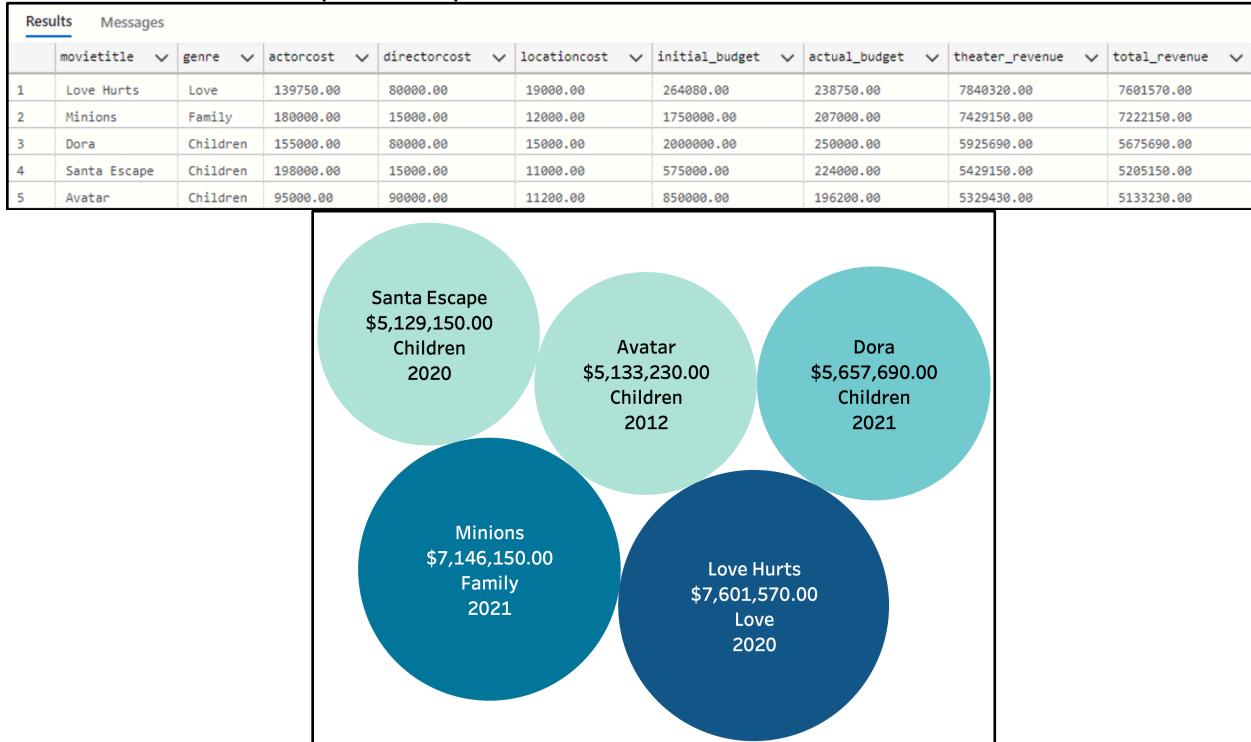
- What is the highest critic rated movie?
 - Who directed it?
 - Who was the main actor?

	movietitle	genre	main_actor	director	criticrating	fanrating
1	Santa Escape	Children	Sam Combs	Luke Brown	99	94

- What is the lowest critic rated movie?
 - Who directed it?
 - Who was the main actor?

	movietitle	genre	main_actor	director	criticrating	fanrating
1	1st Grade	Children	Grace Brady	Luke Brown	33	95

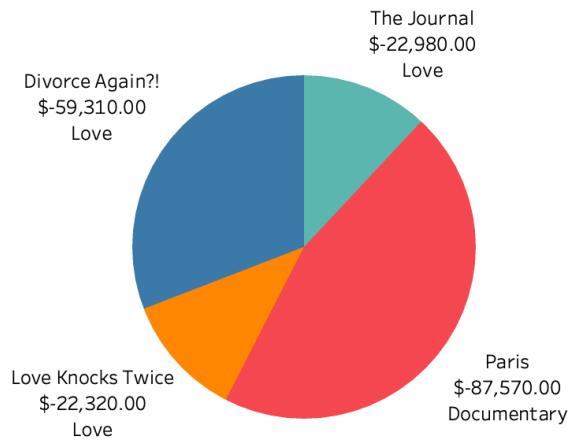
- What are the top 5 most profitable movies?



- Which movies made no profit?

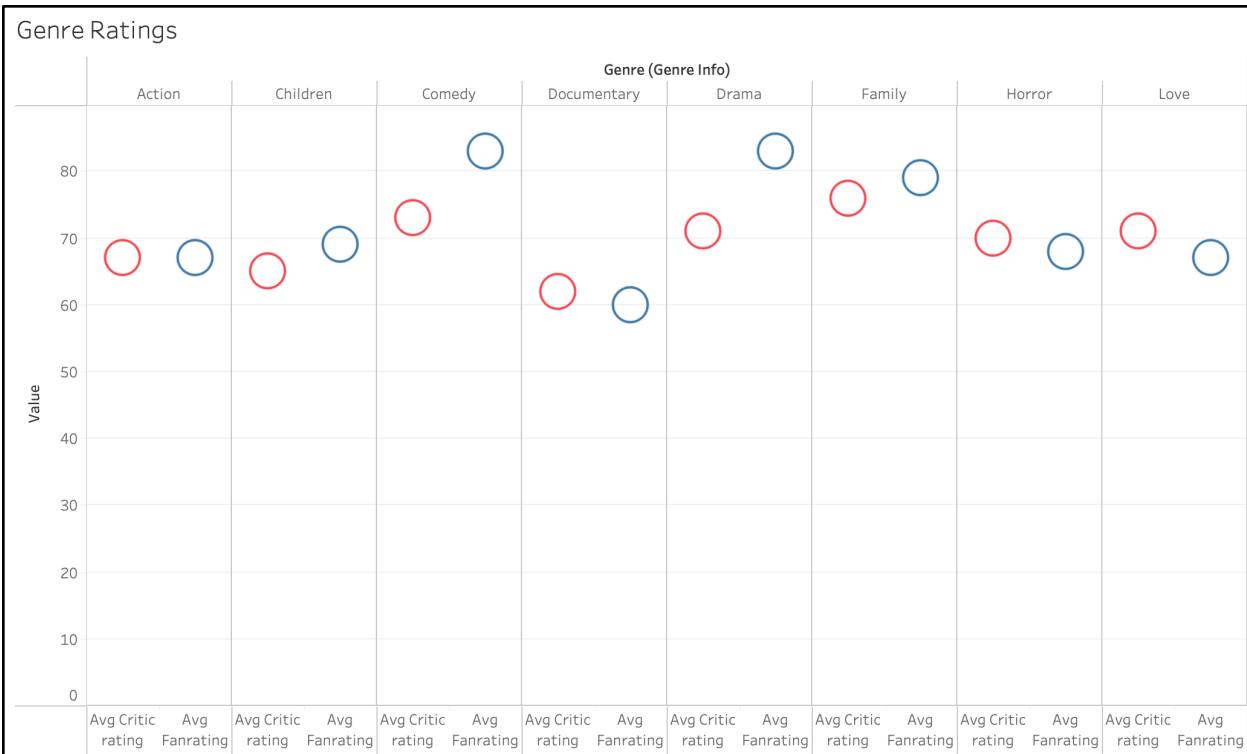
	moviedid	movietitle	genre	initial_budget	actual_budget	budget_diff	budget_category	total_revenue	profit	fanrating	criticrating
1	31	Divorce Again?!	Love	200000.00	267000.00	-67000.00	Over Budget	-41310.00	No Profit	52	60
2	40	Love Knocks Twice	Love	100000.00	261500.00	-161500.00	Over Budget	-22320.00	No Profit	34	52
3	24	Paris	Documentary	80000.00	217000.00	-137000.00	Over Budget	-87570.00	No Profit	66	83
4	16	The Journal	Love	1300000.00	248670.00	1051330.00	Under Budget	-22980.00	No Profit	94	98

Movies with No Profit



- What are the overall statistics by genre?

	genre	avg_fanrating	avg_criticrating	avg_location_rentalrate	avg_filming_days	avg_runtime	avg_director_cost
1	Action	67	67	16750.00	54	136	58000.00
2	Children	69	65	24130.00	138	89	58000.00
3	Comedy	83	73	14000.00	67	121	58000.00
4	Documentary	60	62	14445.4545	59	126	73636.3636
5	Drama	83	71	12666.6666	72	95	80555.5555
6	Family	79	76	29020.00	63	85	70500.00
7	Horror	68	70	14262.50	90	108	82500.00
8	Love	67	71	14818.1818	126	95	80000.00



- Which movies took over 200 days to film?

Results Messages

	movietitle	total_location_rentalrate	total_filming_days
1	Divorce Again?!	12000.00	302
2	Death by Sleep	12000.00	298
3	The Election Campaign	12000.00	255
4	Red Light Blue Light	112000.00	241
5	Painting with Kat	12000.00	238
6	Love Hurts	19000.00	233
7	Rangers	17500.00	216

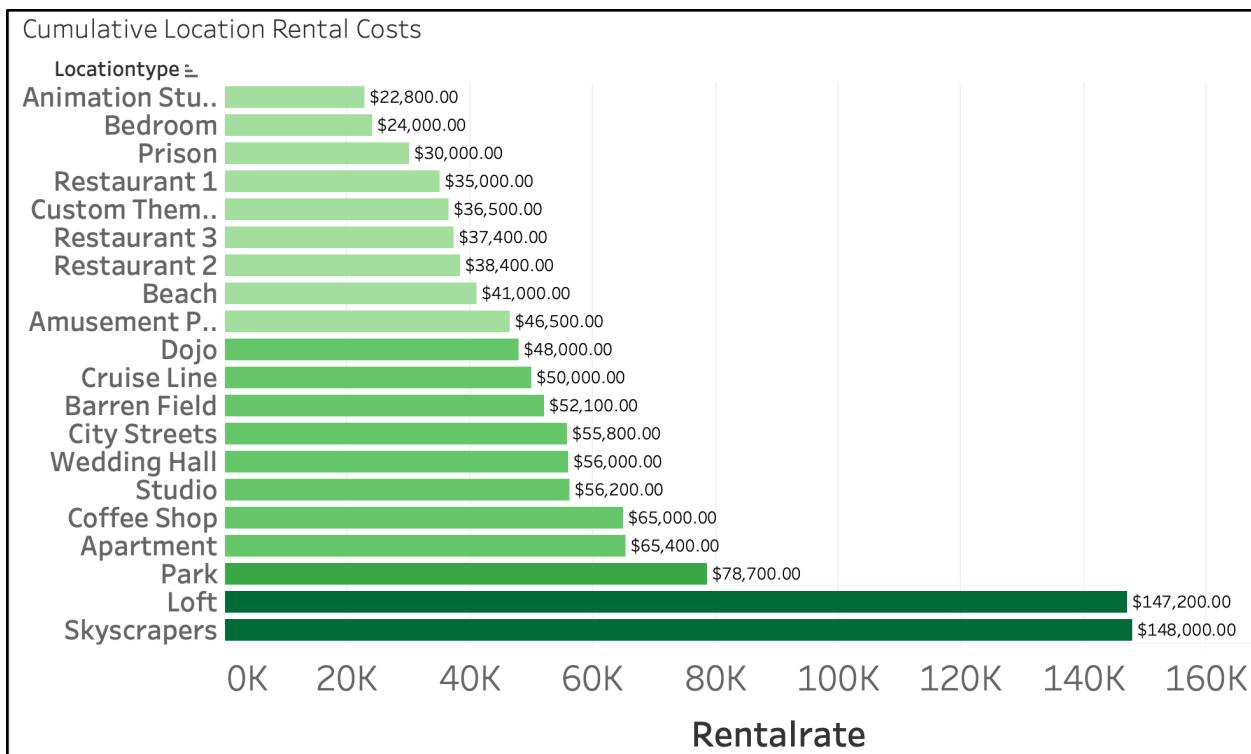
Movies with over 200 Days of Filming

Movietitle (Filmschedule) ▾

Rangers	216
Love Hurts	233
Painting with Kat	238
Red Light Blue Light	241
The Election Campaign	255
Death by Sleep	298
Divorce Again?!	302

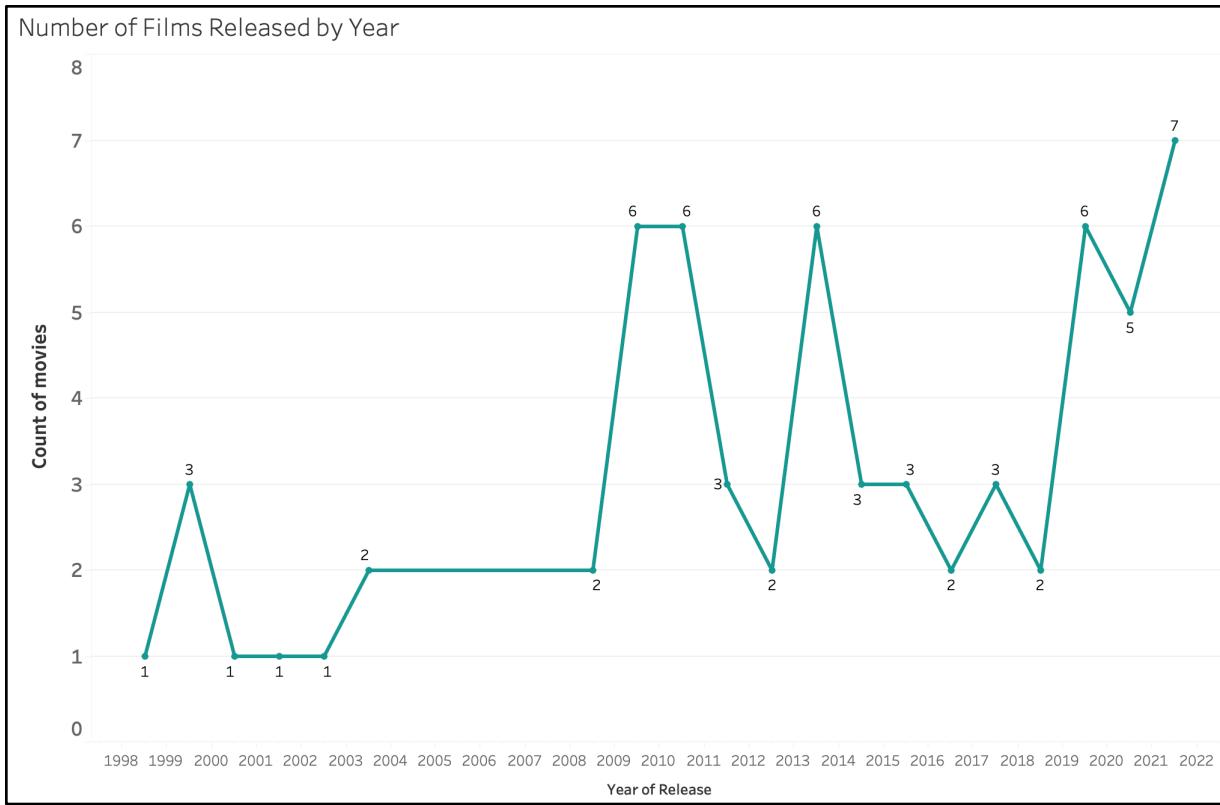
- How much has each film location earned over time?

	locationtype ▾	total_rentalrate ▾
1	Skyscrapers	148000.00
2	Loft	147200.00
3	Park	78700.00
4	Apartment	65400.00
5	Coffee Shop	65000.00
6	Studio	56200.00
7	Wedding Hall	56000.00
8	City Streets	55800.00
9	Barren Field	52100.00
10	Cruise Line	50000.00
11	Dojo	48000.00
12	Amusement Park	46500.00
13	Beach	41000.00
14	Restaurant 2	38400.00
15	Restaurant 3	37400.00
16	Custom Themed ...	36500.00
17	Restaurant 1	35000.00
18	Prison	30000.00
19	Bedroom	24000.00
20	Animation Stud...	22800.00



- How many movies were released per year?

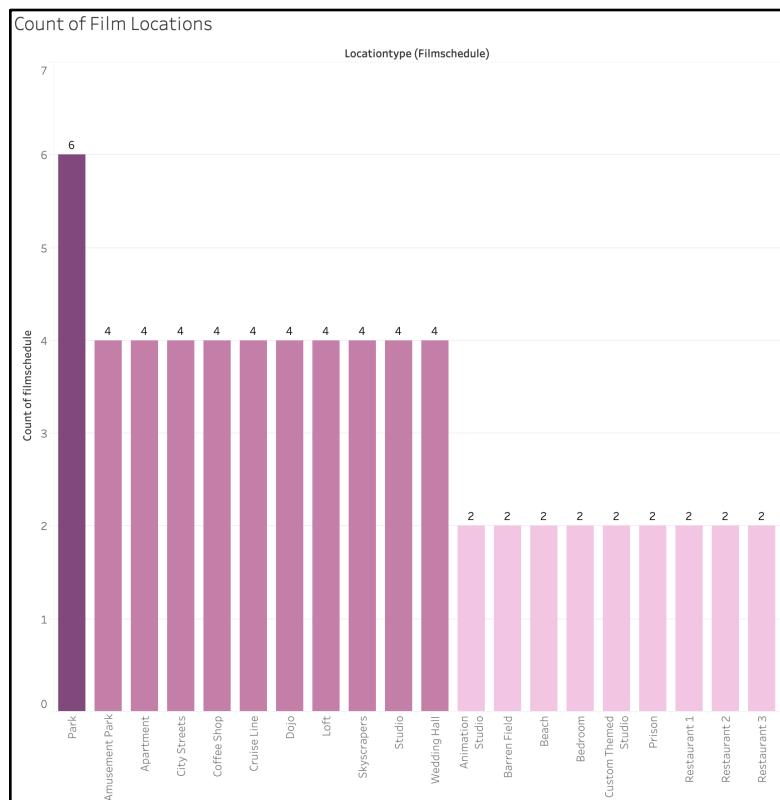
Results		Messages
	yearrelease	movies_per_year
1	1999	1
2	2000	3
3	2001	1
4	2002	1
5	2003	1
6	2004	2
7	2009	2
8	2010	6
9	2011	6
10	2012	3
11	2013	2
12	2014	6
13	2015	3
14	2016	3
15	2017	2
16	2018	3
17	2019	2
18	2020	6
19	2021	5
20	2022	7



- How many movies have filmed per location?

Results Messages

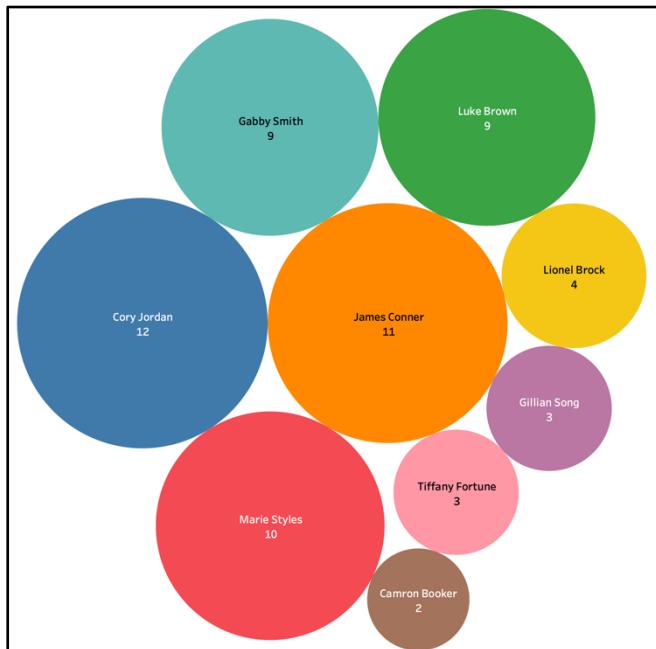
	locationtype	movieCount
1	Park	6
2	Amusement Park	4
3	Dojo	4
4	Loft	4
5	Apartment	4
6	City Streets	4
7	Coffee Shop	4
8	Cruise Line	4
9	Skyscrapers	4
10	Studio	4
11	Wedding Hall	4
12	Custom Themed Studio	2
13	Barren Field	2
14	Beach	2
15	Bedroom	2
16	Animation Studio	2
17	Prison	2
18	Restaurant 1	2
19	Restaurant 2	2
20	Restaurant 3	2



- How many movies have the directors worked on?

Results **Messages**

	Director_Name	Number_of_Movies
1	Cory Jordan	12
2	James Conner	11
3	Marie Styles	10
4	Gabby Smith	9
5	Luke Brown	9
6	Lionel Brock	4
7	Tiffany Fortune	3
8	Gillian Song	3
9	Camron Booker	2
10	Erica Connel	1
11	Tommy Peaches	1



- Which states have the most movies filmed?

Results **Messages**

	state	movieCount
1	CA	9
2	NY	3
3	NV	2
4	TX	2
5	PE	1
6	FL	1
7	HI	1
8	IL	1



- Who acts and directs the same movie?

	firstname	lastname
1	Camron	Booker
2	Cory	Jordan
3	Erica	Connel
4	Gabby	Smith
5	Gillian	Song
6	James	Conner
7	Lionel	Brock
8	Luke	Brown
9	Marie	Styles
10	Tiffany	Fortune
11	Tommy	Peaches

IMPLEMENTATION OF APPLICATION

REVIEW ACTORS' MOVIE HISTORY AND RATES

- Those with access to the database can look up an actor's movie history and how much they were paid for each role.

Precise Management Production

 Accounting
 Actors
 Movies

Select Actor to Look Up Actor's History & Rates!!!



Gabby Smith



Marie Styles



Grace Brady



Cory Jordan



Luke Brown



Gillian Song



Alexa Payne



Lionel Brock

SEARCH FOR MOVIE DETAILS

- The production company can search for individual movies and look at the details of the movies. The user can also click on the names of the director/actor to review more information about their pay and other details.

Precise Management Production



Movie Search Results

Movie Title: Knockdown Love
Genre: Love
Budget: 300000
Revenue: 639180
Year Release: 2010
Runtime: 102 mins
Production Company: Film Club

Movie Cast

Director

Grace Brady

Cast

Luke Brown
Gabby Smith
Sam Combs
Taylor Bryan
Kane Swift
James Conne
Cory Jordan
Alexa Payne
Marie Styles Lionel

Actors



Knockdown Love

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SET MOVIE BUDGET

- The accounting team can set the budget for each category, and it will calculate the overall movie budget.

Precise Management
Production

Accounting Actors Movies

Set Project Budget

Movie Project

Select Director/ Directors \$.
Select Actors/ Support Actors \$.
Select Scene Locations \$.
Select Production Company \$.

Add New Category

Submit!

SEARCH FOR RENTAL LOCATIONS

- User can search and select for rental locations by price or image

Precision Management
Production

Location 

Filters

Location Type V	Rental Price Range	Min	Max
			

<Location Name>
<Location Type Description>
<Location Rental Range Description>

REFLECTION

Upon starting this project, we initially did not realize how beneficial the conceptual and logical models were to create a database. As we constructed these visual models, we were able to identify different underlying relationships between entities. However, we had a difficult time identifying the foreign keys. We could only imagine how much longer it would take us if we were to make a database when more entities. It would require us to make more models to get better and the project was a good first way to get better.

If we had more time to do this project, we would optimize the database by expanding its structure. We could either add more entities such as websites, advertising, etc. or add more attributes to existing tables. For example, we could add date of birth to the person's table. Small additions to the database overtime widens the usability of our database.

SUMMARY

By properly following the database lifecycle, we were able to create & validate a working movie database. We first identified the scope of our project—(Precise Management Production) does not have a proper record keeping system. We then analyzed the problem and its requirements by creating a conceptual model. This model was sequentially transformed into a logical model, where the specifications for creating the movie database is defined. After implementing these models into a new database, we were able to answer our business questions. To aid our answers, we used Microsoft Excel and Tableau to make visual representations of our findings. The database can be used by any production company that's looking for more ways to their expenses and maximize their profits by choose the best actors, directors, and location.

TEAM LOG

MARIANNE LYNNE SANTOS GARBO

- Conceptual and logical model
- Constraints in tables
- Implementation of application graphics
- SQL data answers
- Data questions
- Business rules
- Glossary
- Reflection/summary

KWASI MENSAH

- Logical model
- Table creation and inserts
- Constraints in tables
- Implementation of application graphics

- Data questions
- Business rules
- Views
- Procedures
- SQL data answers

AMANDA NORWOOD

- Table creation and inserts
- Views
- Procedures
- Data questions, business rules, background, glossary
- Raw data sample
- SQL data answers
- SQL data answers graphics
- Constraints in tables
- Background