INSERT INTO `sflix`.`Account`

(`Name`,

`Email`,

`UserName`,

`Password`,

`Birthday`,

`Account\_Status`,

`Payment\_Method`)

VALUES

("Shim",

"Shim@sjsu.edu",

"ShimShim",

"Shim@123",

"1969-12-1",

1,

"CREDIT\_CARD");

INSERT INTO `sflix`.`Account`

(`Name`,

`Email`,

`UserName`,

`Password`,

`Birthday`,

`Account\_Status`,

`Payment\_Method`)

VALUES

("Kamal",

"Kamal@sjsu.edu",

"KamalP",

"Kamal@123",

"1995-09-12",

0,

"CHECK");

INSERT INTO `sflix`.`Account`

(`Name`,

`Email`,

`UserName`,

`Password`,

`Birthday`,

`Account\_Status`,

`Payment\_Method`)

VALUES

("KuldipP",

"Kuldip@sjsu.edu",

"KuldipP",

"Kuldip@123",

"1999-10-1",

1,

"DEBIT\_CARD");

INSERT INTO `sflix`.`Plan`

(`Name`,

`Price`,

`Features`,

`Profile\_Limit`)

VALUES

("Premium",

19.95,

"4K HDR + Full HD",

5);

INSERT INTO `sflix`.`Billing`

(`Bill\_Amount`,

`Payment\_Status`,

`Statement\_Date`)

VALUES

(19.95,

1,

"2022-03-1");

INSERT INTO `sflix`.`Billing`

(`Bill\_Amount`,

`Payment\_Status`,

`Statement\_Date`)

VALUES

(19.95,

1,

"2022-03-2");

INSERT INTO `sflix`.`Billing`

(`Bill\_Amount`,

`Payment\_Status`,

`Statement\_Date`)

VALUES

(19.95,

0,

"2022-03-3");

INSERT INTO `sflix`.`Subscription`

(`Account\_ID`,

`Bill\_ID`,

`Plan\_ID`,

`Start\_Date`,

`End\_Date`,

`isActive`)

VALUES

(1,

1,

1,

"2022-02-01",

"2022-02-28",

0);

INSERT INTO `sflix`.`Subscription`

(`Account\_ID`,

`Bill\_ID`,

`Plan\_ID`,

`Start\_Date`,

`End\_Date`,

`isActive`)

VALUES

(3,

2,

1,

"2022-03-02",

"2022-04-01",

1);

INSERT INTO `sflix`.`Subscription`

(`Account\_ID`,

`Bill\_ID`,

`Plan\_ID`,

`Start\_Date`,

`End\_Date`,

`isActive`)

VALUES

(1,

4,

1,

"2022-03-01",

"2022-03-31",

1);

INSERT INTO `sflix`.`Profiles`

(`Name`,

`Number`,

`Account\_ID`,

`isAgeRestricted`)

VALUES

("Shim1",

1,

1,

0);

INSERT INTO `sflix`.`Profiles`

(`Name`,

`Number`,

`Account\_ID`,

`isAgeRestricted`)

VALUES

("Shim2",

2,

1,

0);

INSERT INTO `sflix`.`Profiles`

(`Name`,

`Number`,

`Account\_ID`,

`isAgeRestricted`)

VALUES

("Kuldip1",

1,

3,

0);

INSERT INTO `sflix`.`Profiles`

(`Name`,

`Number`,

`Account\_ID`,

`isAgeRestricted`)

VALUES

("Kuldip2",

2,

3,

1);

INSERT INTO `sflix`.`Service\_Request`

(`Service\_Num`,

`Account\_ID`,

`Complains`,

`isResolved`)

VALUES

(101,

2,

"I am not able to activate my account",

0);

INSERT INTO `sflix`.`Service\_Request`

(`Service\_Num`,

`Account\_ID`,

`Complains`,

`isResolved`)

VALUES

(306,

3,

"My watch history is incorrect",

1);

INSERT INTO `sflix`.`Watch\_History`

(`Profile\_ID`,

`Movie\_ID`,

`Rating`,

`Likes`,

`Comment`,

`Timeline`,

`Last\_Watched`)

VALUES

(1,

7,

2.1,

0,

"Not that good Movie",

205,

"2022-02-14");

INSERT INTO `sflix`.`Watch\_History`

(`Profile\_ID`,

`Movie\_ID`,

`Rating`,

`Likes`,

`Comment`,

`Timeline`,

`Last\_Watched`)

VALUES

(3,

10,

7.4,

0,

"",

1200,

"2022-03-02");

INSERT INTO `sflix`.`Watch\_History`

(`Profile\_ID`,

`Movie\_ID`,

`Rating`,

`Likes`,

`Comment`,

`Timeline`,

`Last\_Watched`)

VALUES

(4,

33,

6.5,

2,

"Good Movie",

8640,

"2022-03-02");

INSERT INTO `sflix`.`Watch\_History`

(`Profile\_ID`,

`Movie\_ID`,

`Rating`,

`Likes`,

`Comment`,

`Timeline`,

`Last\_Watched`)

VALUES

(2,

57,

8.7,

10,

"Awesome",

5220,

"2022-03-05");

DELIMITER $$

CREATE PROCEDURE fetchTop50Movies()

BEGIN

SELECT Movie\_ID, Title, Genre, Description, Director, Actors, Year, Runtime, Rating, isAgeRestricted FROM Movies LIMIT 50;

END $$

call fetchTop50Movies();

DELIMITER $$

CREATE PROCEDURE findMovie(Search VARCHAR(50))

BEGIN

SELECT Movie\_ID, Title FROM Movies WHERE Title Like CONCAT("%",Search,"%");

END $$

call findMovie("Land");

DELIMITER $$

CREATE PROCEDURE filterMovie(SGenre VARCHAR(50), SRating INT)

BEGIN

SELECT Movie\_ID, Title FROM Movies WHERE Rating >= SRating AND Genre Like CONCAT("%",SGenre,"%");

END $$

call filterMovie("Action", 1.0);

SELECT Title, Genre, Description, Director, Actors, Year, Runtime, Rating, isAgeRestricted FROM Movies WHERE Movie\_ID = 2;

SELECT Content\_URL FROM Movies WHERE Movie\_ID = 1;

SELECT Name, Email, UserName, Birthday, Account\_Status FROM Account WHERE Account\_ID = 1;

SELECT Name, Number, isAgeRestricted FROM Profiles WHERE Account\_ID = 1;

SELECT P.Name, P.Price, P.Features, P.Profile\_Limit FROM Plan P JOIN Subscription S on S.Plan\_ID = P.Plan\_ID JOIN Account A ON A.Account\_ID = S.Account\_ID WHERE A.Account\_ID = 1 ORDER BY S.Subscription\_ID LIMIT 1;

SELECT B.Bill\_id, B.Bill\_Amount, A.Payment\_Method FROM Account A JOIN Subscription S ON A.Account\_ID = S.Account\_ID JOIN Billing B ON S.Bill\_ID = B.Bill\_ID WHERE A.Account\_ID = 1 ORDER BY B.Statement\_Date LIMIT 1;

SELECT B.Bill\_id, B.Bill\_Amount, A.Payment\_Method, B.Payment\_Status, B.Statement\_Date FROM Account A JOIN Subscription S ON A.Account\_ID = S.Account\_ID JOIN Billing B ON S.Bill\_ID = B.Bill\_ID WHERE S.isActive = 1 AND A.Account\_ID = 1;

SELECT M.Title, M.Genre, W.Likes, W.Rating FROM Account A JOIN Profiles P ON P.Account\_ID = A.Account\_ID JOIN Watch\_History W ON W.Profile\_ID = P.Profile\_ID JOIN Movies M ON M.Movie\_ID = W.Movie\_Id WHERE A.Account\_ID = 1 AND M.Title Like "%Land%";

SELECT M.Genre AS "Genre", COUNT(M.Movie\_ID) AS "Movie Count", AVG(M.Runtime) AS "Average Runtime" FROM Movies M GROUP BY M.Genre ORDER BY COUNT(M.Movie\_ID) DESC;

DELIMITER $$

CREATE PROCEDURE genreStats(SGenre VARCHAR(50))

BEGIN

SELECT COUNT(M.Movie\_ID) AS "Movie Count", AVG(M.Runtime) AS "Average Runtime" FROM Movies M WHERE M.Genre Like CONCAT("%",SGenre,"%");

END $$

call genreStats ("Action")

DELIMITER $$

CREATE PROCEDURE movieRatings(SMovie VARCHAR(50))

BEGIN

SELECT M.Rating AS "IMDB Rating", W.Rating AS "Viewer's Rating", ROUND((M.Rating + W.Rating) / 2, 1) AS "Average Rating" FROM Movies M JOIN Watch\_History W ON W.Movie\_Id = M.Movie\_ID WHERE M.Title Like CONCAT("%",SMovie,"%");

END $$

call movieRatings("La La");

DELIMITER $$

CREATE PROCEDURE watchHistory(SProfile INT)

BEGIN

SELECT M.Title, M.Genre, W.Likes, W.Rating, W.Comment, W.Timeline, W.Last\_Watched FROM Profiles P JOIN Watch\_History W ON W.Profile\_ID = P.Profile\_ID JOIN Movies M ON M.Movie\_ID = W.Movie\_Id WHERE P.Profile\_ID = SProfile;

END $$

call watchHistory(1);

DELIMITER $$

CREATE PROCEDURE sendComplain(SNumber INT, SAccount INT, SComplain VARCHAR(200))

BEGIN

INSERT INTO Service\_Request (`Service\_Num`, `Account\_ID`, `Complains`, `isResolved`) VALUES (SNumber, SAccount, SComplain, 0);

END $$

call sendComplain(306, 3, "Issue of Movie Page");

SELECT Service\_Num, Complains, isResolved FROM Service\_Request WHERE Service\_Num = 101;