

Assignment 1 Netflix Originals Data Exploration and Analysis

Task 1:

Retrieve all Netflix Originals with an IMDb score greater than 7, runtime greater than 100 minutes, and the language is either English or Spanish

The screenshot displays two instances of MySQL Workbench. Both instances have the title bar "Assignment 1 SQL NetflixOrgina...".

Session 1 (Top):

- Query: `use netflixoriginals;`
- Query: `select * from netflix_originals;`
- Result Grid (Table: netflix_originals):

Title	GenreID	Runtime	IMDBScore	Language	Premiere_Date
Enter the Anime	G1	58	2.5	English	05-08-2019
Dark Forces	G2	81	2.6	Spanish	21-08-2020
The App	G3	79	2.6	Italian	26-12-2019
The Open House	G9	94	3.2	English	19-01-2018
Kaali Khuli	G4	90	3.4	Hindi	30-10-2020
Drive	G5	147	3.5	Hindi	01-11-2019
Leyla Everlasting	G6	112	3.7	Turkish	04-12-2020

Session 2 (Bottom):

 - Query: `use netflixoriginals;`
 - Query: `select * from netflix_originals;`
 - Query: `select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100 and Language in('english','spanish');`
 - Result Grid (Table: netflix_originals 2):

IMDBScore	Runtime	language
7.1	110	English
7.1	126	English
7.1	138	English
7.1	112	English
7.1	114	English
7.1	125	English
7.2	107	English

QUERY:

```
select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100  
and Language in('english','spanish');
```

Task 2:

Find the total number of Netflix Originals in each language, but only show those languages that have more than 5 titles

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **netflixoriginals** schema selected.
- Query Editor:** The query is:use netflixoriginals;
select * from netflix_originals;
select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100 and Language in('english','spanish');
select Language , count(*) as total_titles from netflix_originals
group by Language having total_titles>5;
- Result Grid:** Displays the results of the query:| Language | total_titles |
| --- | --- |
| English | 420 |
| Spanish | 34 |
| Italian | 14 |
| Hindi | 33 |
| Korean | 6 |
| Indonesian | 9 |
| French | 20 |
| Portuguese | 12 |
- Action Output:** Shows two log entries:| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 17 | 22:08:57 | select Language , count(*) as total_titles from netflix_originals group by Language | 23 row(s) returned | 0.000 sec / 0.000 sec |
| 18 | 22:09:30 | select Language . count() as total_titles from netflix_originals group by Language h... | 9 row(s) returned | 0.000 sec / 0.000 sec |

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **netflixoriginals** schema selected.
- Query Editor:** The query is identical to the one above:use netflixoriginals;
select * from netflix_originals;
select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100 and Language in('english','spanish');
select Language , count(*) as total_titles from netflix_originals
group by Language having total_titles>5;
- Result Grid:** Displays the results of the query:| Language | total_titles |
| --- | --- |
| English | 420 |
| Spanish | 34 |
| Italian | 14 |
| Hindi | 33 |
| Korean | 6 |
| Indonesian | 9 |
| French | 20 |
| Portuguese | 12 |
| Japanese | 6 |
- Action Output:** Shows two log entries (identical to the first run):| # | Time | Action | Message | Duration / Fetch |
| --- | --- | --- | --- | --- |
| 17 | 22:08:57 | select Language , count(*) as total_titles from netflix_originals group by Language | 23 row(s) returned | 0.000 sec / 0.000 sec |
| 18 | 22:09:30 | select Language . count() as total_titles from netflix_originals group by Language h... | 9 row(s) returned | 0.000 sec / 0.000 sec |

QUERY:

```
select Language , count(*) as total_titles from netflix_originals
group by Language having total_titles>5;
```

Task 3:

Get the top 3 longest-running movies in Hindi language sorted by IMDb score in descending order

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80, Assignment 1 SQL NetflixOrgina.
- Query Editor:** Contains the following SQL query:

```
use netflixoriginals;
select * from netflix_originals;
select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100 and Language in('english','spanish');
select Language , count(*) as total_titles from netflix_originals
group by Language having total_titles>5;
select title, runtime,IMDBScore from netflix_originals
where Language = 'Hindi' order by runtime desc,IMDBScore desc limit 3;
```
- Result Grid:** Displays the results of the query:

title	runtime	IMDBScore
Ludo	149	7.6
Raat Akeli Hai	149	7.3
Drive	147	3.5
- Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
6	13:58:57	select * from netflix_originals where Language = 'Hindi' order by runtime desc,IMDBScore desc limit 3;	3 row(s) returned	0.000 sec / 0.000 sec
7	13:59:21	select title, runtime,IMDBScore from netflix_originals where Language = 'Hindi' order... 3 row(s) returned		0.000 sec / 0.000 sec

QUERY:

```
select * from netflix_originals where Language = 'Hindi' order by runtime desc,IMDBScore desc limit 3;
```

Task 4:

Retrieve all titles that contain the word "House" in their name and have an IMDb score greater than 6

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80, Assignment 1 SQL NetflixOrgina.
- Query Editor:** Contains the following SQL query:

```
select * from netflix_originals;
select IMDBScore,Runtime,language from netflix_originals where imdbscore>7 and runtime>100 and Language in('english','spanish');
select Language , count(*) as total_titles from netflix_originals
group by Language having total_titles>5;
select title, runtime,IMDBScore from netflix_originals
where Language = 'Hindi' order by runtime desc,IMDBScore desc limit 3;
select title, imdbscore from netflix_originals where title like '%House%' and imdbscore >6;
```
- Result Grid:** Displays the results of the query:

title	imdbscore
His House	6.5
Knock Down the House	7.1
- Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
7	13:59:21	select title, runtime,IMDBScore from netflix_originals where Language = 'Hindi' order... 3 row(s) returned		0.000 sec / 0.000 sec
8	14:01:38	select title, imdbscore from netflix_originals where title like '%House%' and imdbscore... 2 row(s) returned		0.015 sec / 0.000 sec

QUERY:

```
select title, imdbscore from netflix_originals where title like '%House%' and imdbscore >6;
```

Task 5:

Find all Netflix Originals released between the years 2018 and 2020 that are in either English, Spanish, or Hindi

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema structure, including the **netflixoriginals** database and its **netflix_originals** table.
- SQL Editor:** Contains the following SQL code:

```
8 • select title, imdbscore from netflix_originals where title like '%House%' and imdbscore >6;
9 • UPDATE netflix_originals
10 SET Premiere_Date = STR_TO_DATE(Premiere_Date, '%d-%m-%Y');
11 • select * from netflix_originals;
12 • SELECT * FROM netflix_originals
13 WHERE language IN ('English', 'Spanish', 'Hindi') and Premiere_Date BETWEEN '2018-01-01' AND '2020-12-31';
```
- Result Grid:** Displays the results of the query, showing 343 rows. The columns are Title, GenreID, Runtime, IMDBScore, Language, and Premiere_Date. The data includes titles like "Enter the Anime", "Dark Forces", "The Open House", etc., with various release dates and genres.
- Output:** Shows the message "343 row(s) returned" and the duration "0.000 sec / 0.000 sec".

QUERY:

```
UPDATE netflix_originals
```

```
SET Premiere_Date = STR_TO_DATE(Premiere_Date, '%d-%m-%Y'); Changed the data type from text to date
```

```
select * from netflix_originals;
```

```
SELECT * FROM netflix_originals WHERE language IN ('English', 'Spanish', 'Hindi') and Premiere_Date BETWEEN '2018-01-01' AND '2020-12-31';
```

Task 6:

Find all movies that either have a runtime less than 60 minutes or an IMDb score less than 5, sorted by Premiere Date

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **netflixoriginals** database selected.
- Query Editor:** Contains the following SQL code:


```

9 • UPDATE netflix_originals
10 SET Premiere_Date = STR_TO_DATE(Premiere_Date, '%d-%m-%Y');
11 • select * from netflix_originals;
12 • SELECT * FROM netflix_originals
13 WHERE language IN ('English', 'Spanish', 'Hindi') AND Premiere_Date BETWEEN '2018-01-01' AND '2020-12-31';
14 • select Title, runtime, IMDBScore from netflix_originals where runtime < 60 or IMDBScore < 5 order by Premiere_Date desc;
```
- Result Grid:** Displays the results of the query, showing 117 rows. The columns are Title, runtime, and IMDBScore. Some rows include additional context like genre and premiere date.
- Action Output:** Shows the execution log with the message: "14 16:09:48 select * from netflix_originals where runtime < 60 or IMDBScore < 5 order by Premier... 117 row(s) returned".

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **netflixoriginals** database selected.
- Query Editor:** Contains the same SQL code as the first screenshot.
- Result Grid:** Displays the results of the query, showing 343 rows. The columns are Title, GenreID, Runtime, IMDBScore, Language, and Premiere_Date.
- Action Output:** Shows the execution log with the message: "13 16:08:20 SELECT * FROM netflix_originals WHERE language IN ('English', 'Spanish', 'Hindi')... 343 row(s) returned".

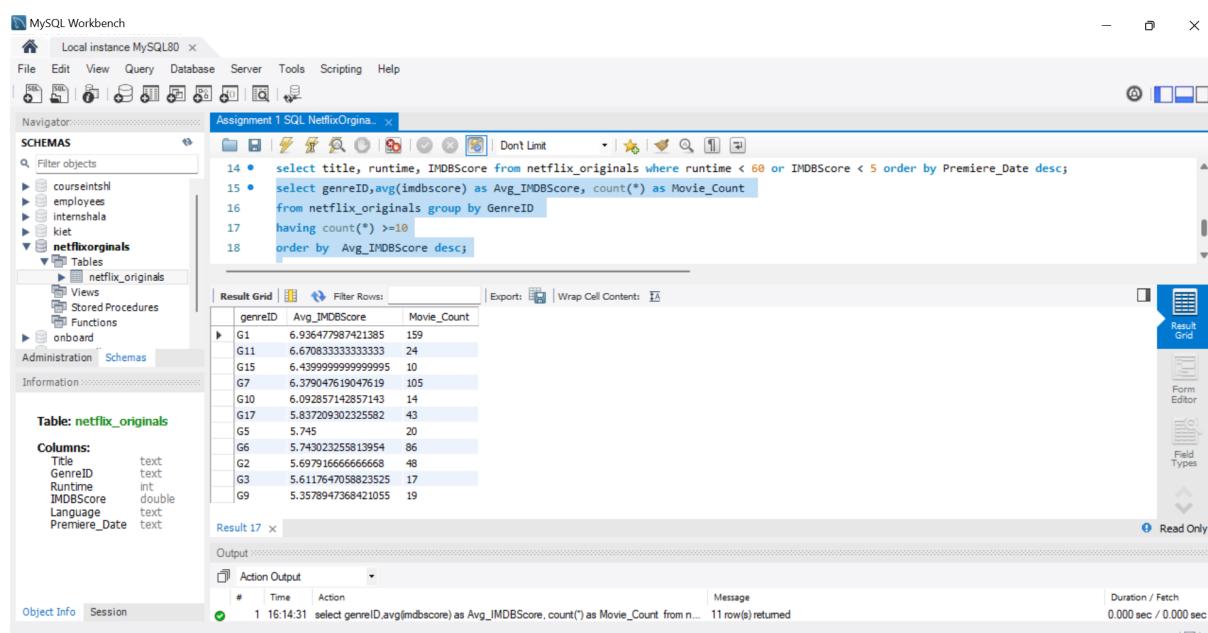
QUERY:

```
select title, runtime, IMDBScore from netflix_originals where runtime < 60 or IMDBScore < 5 order by Premiere_Date desc;
```

```
select * from netflix_originals where runtime < 60 or IMDBScore < 5 order by Premiere_Date desc;
```

Task 7:

Get the average IMDb score for each genre where the genre has at least 10 movies



The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **netflixoriginals** schema expanded, revealing the **netflix_originals** table.
- SQL Editor:** Contains the following SQL query:

```
14 • select title, runtime, IMDBScore from netflix_originals where runtime < 60 or IMDBScore < 5 order by Premiere_Date desc;
15 • select genreID,avg(imdbscore) as Avg_IMDBScore, count(*) as Movie_Count
16 from netflix_originals group by GenreID
17 having count(*) >=10
18 order by Avg_IMDBScore desc;
```
- Result Grid:** Displays the results of the query in a tabular format. The columns are **genreID**, **Avg_IMDBScore**, and **Movie_Count**. The data is as follows:

genreID	Avg_IMDBScore	Movie_Count
G1	6.936477987421385	159
G11	6.670833333333333	24
G15	6.439999999999995	10
G7	6.379047619047619	105
G10	6.092857142857143	14
G17	5.837209302325582	43
G5	5.745	20
G6	5.743023255813954	86
G2	5.697916666666668	48
G3	5.611764705882352	17
G9	5.3578947368421055	19

- Output:** Shows the execution message and duration.

Action Output	#	Time	Action	Message	Duration / Fetch
Object Info	Session	1	16:14:31	select genreID,avg(imdbscore) as Avg_IMDBScore, count(*) as Movie_Count from n...	11 row(s) returned 0.000 sec / 0.000 sec

QUERY:

```
select genreID,avg(imdbscore) as Avg_IMDBScore, count(*) as Movie_Count
from netflix_originals group by GenreID having count(*) >=10
order by Avg_IMDBScore desc;
```

Task 8:

Retrieve the top 5 most common runtimes for Netflix Originals

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The 'Query' tab is active. The 'Navigator' pane on the left shows the 'SCHEMAS' section with 'netflixoriginals' selected, revealing its tables: 'courseinsthi', 'employees', 'internshala', 'kiet', and 'netflix_originals'. The 'Tables' section under 'netflixoriginals' lists 'Views', 'Stored Procedures', and 'Functions'. The 'Information' section shows 'onboard' and 'Administration' tabs. The main workspace displays a query editor with two statements:

```
15 • select genreID,avg(imdbscore) as Avg_IMDBScore, count(*) as Movie_Count
16   from netflix_originals group by GenreID
17   having count(*) >=10
18   order by Avg_IMDBScore desc;
19 • SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;
```

The 'Result Grid' shows the following data:

Runtime	Count
97	24
94	19
98	19
95	18
100	17

The 'Result 18' tab in the bottom right shows the execution history:

#	Time	Action	Message	Duration / Fetch
1	16:14:31	select genreID,avg(imdbscore) as Avg_IMDBScore, count(*) as Movie_Count from netflix_originals group by GenreID having count(*) >=10 order by Avg_IMDBScore desc;	11 row(s) returned	0.000 sec / 0.000 sec
2	16:16:37	SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;	5 row(s) returned	0.000 sec / 0.000 sec

QUERY:

```
SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;
```

Task 9:

List all Netflix Originals that were released in 2020, grouped by language, and show the total count of titles for each language.

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to "Local instance MySQL80". The main area displays a query titled "Assignment 1 SQL NetflixOrginal...". The query is:

```

18   order by Avg_IMDBScore desc;
19 •   SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;
20 •   Select Language,Count(*) as Total_Titles from netflix_originals
21   where year(Premiere_Date) = '2020' group by Language order by Total_Titles desc;
22

```

The results grid shows the following data:

Language	Total_Titles
English	108
Spanish	18
Hindi	16
French	8
Italian	7
German	5
Portuguese	4
Korean	3
Indonesian	3

The "Object Info" tab is selected at the bottom left. The "Action Output" panel shows two log entries:

- # 2 16:16:37 SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime... 5 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec
- # 3 16:17:54 Select Language,Count(*) as Total_Titles from netflix_originals where year(Premiere... 18 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec

QUERY:

```
Select Language,Count(*) as Total_Titles from netflix_originals
where year(Premiere_Date) = '2020' group by Language order by Total_Titles desc;
```

Task 10:

Create a new table that enforces a constraint on the IMDb score to be between 0 and 10 and the runtime to be greater than 30 minutes.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrgina... x

SCHEMAS

- courseinstl
- employees
- internshala
- kiel
- netflixoriginals**
- Tables
- Views
- Stored Procedures
- Functions
- onboard

Administration Schemas

Information

Table: netflix_originals

Columns:

Title	text
GenreID	text
Runtime	int
IMDBScore	double
Language	text
Premiere_Date	text

Object Info Session

Query Completed

```

17 having count(*) >=10
18 order by Avg_IMDBScore desc;
19 • SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;
20 • Select Language, count(*) as Total_Titles from netflix_originals
21 where year(Premiere_Date) = '2020' group by Language order by Total_Titles desc;
22 • CREATE TABLE netflix_movies (
23     Movie_ID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,Genre VARCHAR(100),
24     Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes
25     IMDB_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be between 0 and 10
26     Language VARCHAR(50), Premiere_Date DATE);
27
28
29
30
31

```

Action Output

#	Time	Action	Message	Duration / Fetch
3	16:17:54	Select Language.count() as Total_Titles from netflix_originals where year(Premiere...	18 row(s) returned	0.000 sec / 0.000 sec
4	16:19:36	CREATE TABLE netflix_movies (Movie_ID INT PRIMARY KEY,Title VARCHAR(... 0 row(s) affected		0.032 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrgina... x

SCHEMAS

- courseinstl
- employees
- internshala
- kiel
- netflixoriginals**
- Tables
- Views
- Stored Procedures
- Functions
- onboard

Administration Schemas

Information

Table: netflix_originals

Columns:

Title	text
GenreID	text
Runtime	int
IMDBScore	double
Language	text
Premiere_Date	text

Object Info Session

Query Completed

```

17 having count(*) >=10
18 order by Avg_IMDBScore desc;
19 • SELECT Runtime, COUNT(*) AS Count FROM netflix_originals GROUP BY Runtime ORDER BY Count DESC LIMIT 5;
20 • Select Language, count(*) as Total_Titles from netflix_originals
21 where year(Premiere_Date) = '2020' group by Language order by Total_Titles desc;
22 • CREATE TABLE netflix_movies (
23     Movie_ID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,Genre VARCHAR(100),
24     Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes
25     IMDB_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be between 0 and 10
26     Language VARCHAR(50), Premiere_Date DATE);
27
28
29
30
31

```

Action Output

#	Time	Action	Message	Duration / Fetch
3	16:17:54	Select Language.count() as Total_Titles from netflix_originals where year(Premiere...	18 row(s) returned	0.000 sec / 0.000 sec
4	16:19:36	CREATE TABLE netflix_movies (Movie_ID INT PRIMARY KEY,Title VARCHAR(... 0 row(s) affected		0.032 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrginal... x

SCHEMAS

- courseinstl
- employees
- internshala
- kiet
- netflixoriginals**
- Tables
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

Table: netflix_movies

Columns:

Movie_ID	int PK
Title	varchar(2
Genre	varchar(1
Runtime	int
IMDb_Score	decimal(3
Language	varchar(5
Premiere_Date	date

Object Info Session

```

23 Movie_ID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,Genre VARCHAR(100),
24 Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes
25 IMDb_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be between 0 and 10
26 Language VARCHAR(50), Premiere_Date DATE;
27 • select * from netflix_movies;

```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: x

Movie_ID	Title	Genre	Runtime	IMDb_Score	Language	Premiere_Date
NULL	NULL	NULL	NULL	NULL	NULL	NULL

netflix_movies 20 x

Output:

#	Time	Action	Message	Duration / Fetch
4	16:19:36	CREATE TABLE netflix_movies (Movie_ID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,Genre VARCHAR(100),Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes,IMDb_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be between 0 and 10,Language VARCHAR(50), Premiere_Date DATE);	0 row(s) affected	0.032 sec
5	16:20:32	select * from netflix_movies	0 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrginal... x

SCHEMAS

- courseinstl
- employees
- internshala
- kiet
- netflixoriginals**
- Tables
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

Table: netflix_movies

Columns:

Movie_ID	int PK
Title	varchar(2
Genre	varchar(1
Runtime	int
IMDb_Score	decimal(3
Language	varchar(5
Premiere_Date	date

Object Info Session

```

20 • Select Language,count(*) as Total_Titles from netflix_originals
21 where year(Premiere_Date) = '2020' group by Language order by Total_Titles desc;
22 • CREATE TABLE netflix_movies (
23   Movie_ID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,Genre VARCHAR(100),
24   Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes
25   IMDb_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be between 0 and 10
26   Language VARCHAR(50), Premiere_Date DATE);
27 • select * from netflix_movies;
28 • INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,Premiere_Date)
29 VALUES (1, 'Sci-Fi', 148, 62, 10.8, 'English', '2010-07-16');
30
31
32
33
34

```

Output:

#	Time	Action	Message	Duration / Fetch
5	16:20:32	select * from netflix_movies	0 row(s) returned	0.000 sec / 0.000 sec
6	16:21:00	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Lang...	Error Code: 3819. Check constraint 'netflix_movies_chk_2' is violated.	0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrgina... Schemas

SCHEMAS Filter objects courseinstH employees internshala kiet netflixorginals Tables netflix_movies netflix_originals Views Stored Procedures Functions Administration Schemas Information

Table: netflix_movies

Columns:

Movie_ID	int PK
Title	varchar(2)
Genre	varchar(1)
Runtime	int
IMDb_Score	decimal(3)
Language	varchar(5)
Premiere_Date	date

Action Output

#	Time	Action	Message	Duration / Fetch
6	16:21:00	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Lang...	Error Code: 3819. Check constraint 'netflix_movies_chk_2' is violated.	0.000 sec
7	16:22:14	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Lang...	Error Code: 3819. Check constraint 'netflix_movies_chk_1' is violated.	0.000 sec

Object Info Session Query interrupted

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Assignment 1 SQL NetflixOrgina... Schemas

SCHEMAS Filter objects courseinstH employees internshala kiet netflixorginals Tables netflix_movies netflix_originals Views Stored Procedures Functions Administration Schemas Information

Table: netflix_movies

Columns:

Movie_ID	int PK
Title	varchar(2)
Genre	varchar(1)
Runtime	int
IMDb_Score	decimal(3)
Language	varchar(5)
Premiere_Date	date

Action Output

#	Time	Action	Message	Duration / Fetch
7	16:22:14	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Lang...	Error Code: 3819. Check constraint 'netflix_movies_chk_1' is violated.	0.000 sec
8	16:23:46	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Lang...	1row(s) affected	0.000 sec

Object Info Session Query Completed

The screenshot shows the MySQL Workbench interface. In the top-left corner, it says "Local instance MySQL80 x". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, Help. The Navigator pane on the left lists Schemas: courseinsthi, employees, internships, kiet, netflixoriginals (selected), netflix_movies, netflix_originals, Views, Stored Procedures, Functions. The main area has a tab titled "Assignment 1 SQL NetflixOrginala..". It contains the following SQL code:

```

30 • INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,Premiere_Date)
31 VALUES (2,'Sci-Fi',148,22,9.8,'English','2010-07-16');
32 • INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,Premiere_Date)
33 VALUES (1,'Sci-Fi',148,42,9.8,'English','2010-07-16');
34 • select * from netflix_movies;

```

The Result Grid shows the following data:

Movie_ID	Title	Genre	Runtime	IMDb_Score	Language	Premiere_Date
1	Sci-Fi	148	42	9.8	English	2010-07-16
•	NULL	NULL	NULL	NULL	NULL	NULL

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
8	16:23:46	INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,Premiere_Date)	1 row(s) affected	0.000 sec
9	16:24:35	select * from netflix_movies	1 row(s) returned	0.000 sec / 0.000 sec

QUERY:

```

CREATE TABLE netflix_movies (
    Movie_ID INT PRIMARY KEY, Title VARCHAR(255) NOT NULL, Genre VARCHAR(100),
    Runtime INT CHECK (Runtime > 30), -- Runtime must be greater than 30 minutes
    IMDb_Score DECIMAL(3,1) CHECK (IMDb_Score BETWEEN 0 AND 10), -- IMDb Score must be
    between 0 and 10 Language VARCHAR(50), Premiere_Date DATE); CREATED TABLE

```

```

INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,Premiere_Date)
VALUES (1, 'Sci-Fi', 148, 62, 10.8, 'English', '2010-07-16'); INSERTED IMDBSCORE MORETHAN 10
INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,
Premiere_Date) VALUES (2, 'Sci-Fi', 148, 22, 9.8, 'English', '2010-07-16'); INSERTED RUNTIME
LESSTHAN 30
INSERT INTO netflix_movies (Movie_ID,Title,Genre,Runtime,IMDb_Score,Language,
Premiere_Date) VALUES (1, 'Sci-Fi', 148, 42, 9.8, 'English', '2010-07-16'); INSERTED CORRECT VALUES

```

**Prepared By,
Utkarsh Anand**