

**DEPARTMENT OF COMPUTER SCIENCE  
RAJAGIRI COLLEGE OF SOCIAL SCIENCES  
(Autonomous)  
KALAMASSERY - KOCHI - 683104**



**MASTER OF COMPUTER APPLICATIONS**

**DBMS LAB RECORD**

**NAME** : \_\_\_\_\_

**SEMESTER** : \_\_\_\_\_

**REGISTER NO.** : \_\_\_\_\_

**DEPARTMENT OF COMPUTER SCIENCE  
RAJAGIRI COLLEGE OF SOCIAL SCIENCES  
(Autonomous)  
KALAMASSERY - KOCHI - 683104**



**MASTER OF COMPUTER APPLICATIONS**

**DBMS  
LAB RECORD**

**NAME** : \_\_\_\_\_

**SEMESTER** : \_\_\_\_\_

**REGISTER NO.** : \_\_\_\_\_



**DEPARTMENT OF COMPUTER SCIENCE  
RAJAGIRI COLLEGE OF SOCIAL SCIENCES  
(Autonomous)  
KALAMASSERY - KOCHI - 683104**

**MASTER OF COMPUTER APPLICATIONS**

**CERTIFICATE**

**NAME** : \_\_\_\_\_

**SEMESTER** : \_\_\_\_\_

**REGISTER NO.** : \_\_\_\_\_

*Certified that this is a bonafide record of work done by the student in the Software Laboratory of Rajagiri Department of Computer Science, Kalamassery.*

Faculty in Charge

Dean, Computer Science

Internal Examiner

External Examiner

Place : Kalamassery

Date :

## Table of Contents

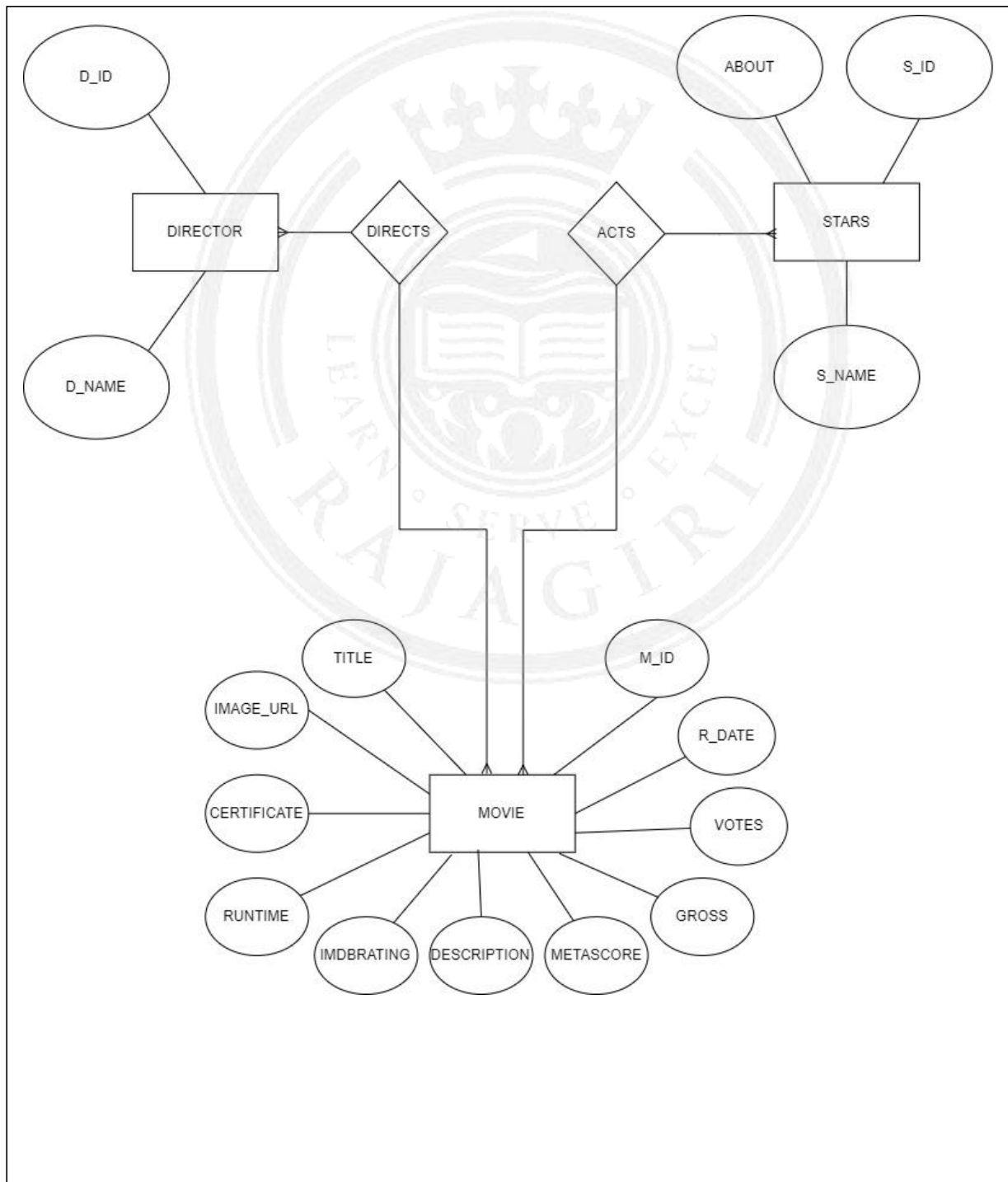
| Activity  | Page No |
|---|---------|
| <b>1. E-R Diagram &amp; Table Design</b>  |         |
| <b>2. Practice SQL Data Definition Language(DDL) commands</b>   |         |
| 2.1 Table creation and alteration   |         |
| <b>3. Practice SQL Data Manipulation Language (DML) commands</b>  |         |
| 3.1 Row insertion, deletion and updating  |         |
| 3.2 Retrieval of data (Simple select query and select with where options (include all relational and logical operators) |         |
| 3.3 Functions: Numeric Data, Character Conversion and   |         |
| 3.4 Group functions   |         |
| 3.5 Data manipulations using date functions   |         |
| 3.6 Set Operations  |         |
| 3.7 Illustration of Group by Having Clause  |         |
| 3.8 Sub Queries   |         |
| 3.9 Retrieving from multiple tables<br>(Illustrate with Join Clause also)   |         |
| <b>4. SQL Views</b>   |         |
| <b>5. Practice PL/SQL</b>   |         |
| 5.1 Introductory programs   |         |
| 5.2 Illustration of Cursors   |         |
| 5.3 Illustration of Procedures  |         |
| 5.4 Illustration of functions   |         |
| 5.5 Illustration of Triggers  |         |

## Activity #1

### E-R Diagram & Table Design

|             |  |
|-------------|--|
| Description | Creating ER Diagrams, Table designs and Table descriptions |
| Date        | 14/08/2023   |

### ER Diagram & Table Design



**TABLE : DIRECTORS**

| D_ID | D_NAME |
|------|--------|
|      |        |

**TABLE : STARS**

| S_ID | S_NAME | ABOUT |
|------|--------|-------|
|      |        |       |

**TABLE : MOVIES**

| M_ID | TITLE | IMAGE_URL | R_DATE | CERTIFICATE | RUNTIME | IMDBRATING | DESCRIPTION | METASCORE | GROSS | VOTES |
|------|-------|-----------|--------|-------------|---------|------------|-------------|-----------|-------|-------|
|      |       |           |        |             |         |            |             |           |       |       |

**TABLE : MOVIESDIRECTORS**

| MOVIESID | DIRECTORSID |
|----------|-------------|
|          |             |

**TABLE : MOVIESSTARS**

| MOVIESID | STARSID |
|----------|---------|
|          |         |

### **TABLE DESIGN:-**

**Table name: Directors**

**Description: Used to store Directors Information**

| Attribute | Data Type    | Constraints           |
|-----------|--------------|-----------------------|
| Id        | Int          | Primary Key/ Not Null |
| Name      | Varchar2(40) | Not Null              |

**Table name: Stars**

**Description: Used to store Stars Information**

| Attribute | Data Type     | Constraints           |
|-----------|---------------|-----------------------|
| Id        | Int           | Primary Key/ Not Null |
| Name      | Varchar2(40)  | Unique                |
| About     | Varchar2(100) |                       |

**Table name: Movies**

**Description: Used to store Movies Information**

| Attribute   | Data Type     | Constraints                               |
|-------------|---------------|---|
| Id          | Int           | Primary Key/ Not Null                     |
| Title       | Varchar2(40)  | Not Null                                  |
| R_date      | Date          |   |
| Image_url   | Varchar2(100) |   |
| Certificate | Varchar2(20)  |   |
| Runtime     | Number(3,2)   |   |
| ImdbRating  | Number (3,1)  | By default 0                              |
| Description | Text(100)     | By default Null                           |
| Metascore   | Number (3,1)  | By default 0                              |
| Votes       | Int           | By default 0                              |
| Gross       | Number(10,2)  | Gross amount should be greater than 10000 |

**Table name: MoviesDirectors**

**Description: Used to store Movie Directors Information**

| Attribute   | Data Type | Constraints   |             |
|-------------|-----------|---|-------------|
| MoviesId    | Int       | Foreign Key references Id of <b>Movies</b> table    | Primary Key |
| DirectorsId | Int       | Foreign Key references Id of <b>Directors</b> table |             |

**Table name: MoviesStars**

**Description: Used to store Movie Stars Information**

| Attribute | Data Type | Constraints   |             |
|-----------|-----------|---|-------------|
| MoviesId  | Int       | Foreign Key references<br>Id of <b>Movies</b> table | Primary Key |
| StarsId   | Int       | Foreign Key references<br>Id of <b>Stars</b> table  |             |





## Activity #2

### Practice SQL Data Definition Language(DDL) commands

|                    |  |
|--------------------|--|
| <b>Description</b> | <b>Table creation and alterations using CREATE and ALTER commands.</b> |
| <b>Date</b>        | 14/08/2023   |

- Create the tables(DIRECTORS,STARS,MOVIES,MOVIESDIRECTORS,MOVIESSTARS) based on the given description.

#### //CREATING TABLE : DIRECTORS

##### Query

SQL> create table directors(d\_id int,d\_name varchar2(40) not null,constraint prim\_of\_id primary key(d\_id));

Table created.

SQL> desc directors;

| Name   | Null?    | Type         |
|--------|----------|--------------|
| -----  | -----    | -----        |
| D_ID   | NOT NULL | NUMBER(38)   |
| D_NAME | NOT NULL | VARCHAR2(40) |

SQL>

SQL> select constraint\_name,constraint\_type from user\_constraints where table\_name='DIRECTORS';

|                 |      |
|-----------------|------|
| CONSTRAINT_NAME | C    |
| -----           | ---- |
| SYS_C0011410    | C    |
| PRIM_OF_ID      | P    |

#### //CREATING TABLE :STARS

##### Query

SQL> create table stars(s\_id int,s\_name varchar2(40) unique,about varchar2(100),constraint prime\_sid primary key(s\_id));

Table created.

SQL> desc stars;

| Name   | Null?    | Type          |
|--------|----------|---------------|
| -----  | -----    | -----         |
| S_ID   | NOT NULL | NUMBER(38)    |
| S_NAME |          | VARCHAR2(40)  |
| ABOUT  |          | VARCHAR2(100) |

```
SQL> select constraint_name,constraint_type from user_constraints where table_name='STARS';
```

| CONSTRAINT_NAME | C    |
|-----------------|------|
| -----           | ---- |
| PRIME_SID       | P    |
| SYS_C0011413    | U    |

## //CREATING TABLE : MOVIES

### Query

```
SQL> create table movies(m_id int,title varchar2(40) not null,r_date date,image_url
varchar2(100),certificate varchar2(20),runtime number(3,2),imdbrating number(3,1)
default(0),description varchar2(100) default(null),metascore number(3,1) default(0),votes int
default(0),gross number(10,2),constraint gross_check check(gross>10000),constraint prime_mid
primary key(m_id));
```

Table created.

```
SQL> desc movies;
```

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| -----       | -----    | -----         |
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| R_DATE      |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| GROSS       |          | NUMBER(10,2)  |

```
SQL> select constraint_name,constraint_type from user_constraints where table_name='MOVIES';
```

| CONSTRAINT_NAME | C    |
|-----------------|------|
| -----           | ---- |
| SYS_C0011414    | C    |
| GROSS_CHECK     | C    |
| PRIME_MID       | P    |

## //CREATING TABLE : MOVIESDIRECTORS

### Query

```
SQL> create table moviesdirectors(moviesid int,directorsid int,foreign key(moviesid) references
movies(m_id),foreign key(directorsid) references directors(d_id),primary
key(moviesid,directorsid));
```

Table created.

```
SQL> desc moviesdirectors;
```

| Name        | Null?    | Type       |
|-------------|----------|------------|
| MOVIESID    | NOT NULL | NUMBER(38) |
| DIRECTORSID | NOT NULL | NUMBER(38) |

```
SQL> select constraint_name,constraint_type from user_constraints where  
table_name='MOVIESDIRECTORS';
```

| CONSTRAINT_NAME | C |
|-----------------|---|
| SYS_C0011417    | P |
| SYS_C0011418    | R |
| SYS_C0011419    | R |

### //CREATING TABLE : MOVIESSTARS

#### Query

```
SQL> create table moviesstars(moviesid int,starsid int,foreign key(moviesid) references  
movies(m_id),foreign key(starsid) references stars(s_id),primary key(moviesid,starsid));
```

Table created.

```
SQL> desc moviesstars;
```

| Name     | Null?    | Type       |
|----------|----------|------------|
| MOVIESID | NOT NULL | NUMBER(38) |
| STARSID  | NOT NULL | NUMBER(38) |

```
SQL> select constraint_name,constraint_type from user_constraints where  
table_name='MOVIESSTARS';
```

| CONSTRAINT_NAME | C |
|-----------------|---|
| SYS_C0011420    | P |
| SYS_C0011421    | R |
| SYS_C0011422    | R |

#### ➤ Add a column 'DOB' to Stars table.

#### Query

```
SQL> alter table stars add dob date;
```

Table altered.

SQL> desc stars;

| Name   | Null?    | Type          |
|--------|----------|---------------|
| S_ID   | NOT NULL | NUMBER(38)    |
| S_NAME |          | VARCHAR2(40)  |
| ABOUT  |          | VARCHAR2(100) |
| DOB    |          | DATE          |

➤ **Drop the column 'Gross' in Movies table.**

**Query**

SQL> alter table movies drop column gross;

Table altered.

SQL> desc movies;

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| R_DATE      |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |

➤ **Add column 'Language' in Movies table.**

**Query**

SQL> alter table movies add language varchar2(20);

Table altered.

SQL> desc movies;

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| R_DATE      |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |

➤ **Add column Gross Number(10,2) in Movies table.**

**Query**

```
SQL> alter table movies add gross number(12,2);
```

Table altered.

```
SQL> desc movies;
```

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| R_DATE      |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |
| GROSS       |          | NUMBER(12,2)  |

➤ **Change the name of the column 'R\_date' in Movies table to Releasedate.**

**Query**

```
SQL> alter table movies rename column r_date to releasedate;
```

Table altered.

```
SQL> desc movies;
```

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| RELEASEDATE |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |
| GROSS       |          | NUMBER(12,2)  |

➤ **Add a column 'Age' in Directors table as Number. Age must be 7 years or above.**

**Query**

SQL> alter table directors add age int;

Table altered.

SQL> alter table directors add constraint age\_chk check(age >= 7);

Table altered.

SQL> desc directors;

| Name   | Null?    | Type         |
|--------|----------|--------------|
| -----  |          |              |
| D_ID   | NOT NULL | NUMBER(38)   |
| D_NAME | NOT NULL | VARCHAR2(40) |
| AGE    |          | NUMBER(38)   |

SQL> select constraint\_name,constraint\_type from user\_constraints where table\_name='DIRECTORS';

|                 |   |
|-----------------|---|
| CONSTRAINT_NAME | C |
| -----           |   |
| SYS_C0011410    | C |
| PRIM_OF_ID      | P |
| AGE_CHK         | C |

➤ **Add a new column 'Hit' in Movies table with datatype Number(1) and by default 0.**

**Query**

SQL> alter table movies add hit number(1) default 0;

Table altered.

SQL> desc movies;

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| -----       |          |               |
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| RELEASEDATE |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |
| GROSS       |          | NUMBER(12,2)  |
| HIT         |          | NUMBER(1)     |

- **Add a new column 'Entry\_date' in Movies table to record the date on which the movie details are entered in the data base.**

**Query**

SQL> alter table movies add entry\_date date;

Table altered.

SQL> desc movies;

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| RELEASEDATE |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |
| GROSS       |          | NUMBER(12,2)  |
| HIT         |          | NUMBER(1)     |
| ENTRY_DATE  |          | DATE          |

- **Destroy the table MoviesStars and recreate it.**

**Query**

SQL> drop table moviesstars;

Table dropped.

SQL> create table moviesstars(moviesid int,starsid int,foreign key(moviesid) references movies(m\_id),foreign key(starsid) references stars(s\_id),primary key(moviesid,starsid));

Table created.

SQL> desc moviesstars;

| Name     | Null?    | Type       |
|----------|----------|------------|
| MOVIESID | NOT NULL | NUMBER(38) |
| STARSID  | NOT NULL | NUMBER(38) |

➤ **Change the size of the Director's name to 30.**

**Query**

```
SQL> alter table directors modify d_name varchar2(30);
```

Table altered.

```
SQL> desc directors;
```

| Name   | Null?    | Type         |
|--------|----------|--------------|
| D_ID   | NOT NULL | NUMBER(38)   |
| D_NAME | NOT NULL | VARCHAR2(30) |
| AGE    |          | NUMBER(38)   |

➤ **Add the following check constraints:**

- **Releasedate should be less than the Entry\_date in the Movies table.**
- **Language of movies should be Malayalam, English, Tamil or Hindi.**

**Query**

```
SQL> alter table movies add constraint chk_entry_date check(releasedate<entry_date);
```

Table altered.

```
SQL> alter table movies add constraint chk_language check(language  
in('Malayalam','English','Tamil','Hindi'));
```

Table altered.

```
SQL> select constraint_name,constraint_type from user_constraints where table_name='MOVIES';
```

| CONSTRAINT_NAME | C |
|-----------------|---|
| SYS_C0011414    | C |
| PRIME_MID       | P |
| CHK_ENTRY_DATE  | C |
| CHK_LANGUAGE    | C |



### Activity #3

#### Practice SQL Data Manipulation Language (DML) commands

|                        |   |
|------------------------|---|
| <b>Description 3.1</b> | <b>Illustration of Row insertion, deletion and updating</b> |
| <b>Date</b>            | 14/08/2023  |

- **Insert the appropriate data (10 rows) for the tables with respect to defined datatypes, size and constraints.**

**//INSERTING VALUES TO DIRECTORS :**

#### Query

SQL> desc directors;

| Name   | Null?    | Type         |
|--------|----------|--------------|
| D_ID   | NOT NULL | NUMBER(38)   |
| D_NAME | NOT NULL | VARCHAR2(30) |
| AGE    |          | NUMBER(38)   |

SQL> insert into directors values('101','LAL JOSE',57);

1 row created.

SQL> insert into directors values('102','VINEETH SREENIVASAN',38);

1 row created.

SQL> insert into directors values('103','ANJALI MENON',44);

1 row created.

SQL> insert into directors values('104','S SANKAR',60);

1 row created.

SQL> insert into directors values('105','LOKESH KANAGARAJ',37);

1 row created.

SQL> insert into directors values('106','MANI RATNAM',67);

1 row created.

SQL> insert into directors values('107','RAJKUMAR HIRANI',60);

1 row created.

SQL> insert into directors values('108','NITESH TIWARI',51);

1 row created.

SQL> insert into directors values('109','JAMES CAMERON',69);

1 row created.

SQL> insert into directors values('110','CHRISTOPHER NOLAN',53);

1 row created.

SQL> select \* from directors;

| D_ID D_NAME             | AGE |
|-------------------------|-----|
| 101 LAL JOSE            | 57  |
| 102 VINEETH SREENIVASAN | 38  |
| 103 ANJALI MENON        | 44  |
| 104 S SANKAR            | 60  |
| 105 LOKESH KANAGARAJ    | 37  |
| 106 MANI RATNAM         | 67  |
| 107 RAJKUMAR HIRANI     | 60  |
| 108 NITESH TIWARI       | 51  |
| 109 JAMES CAMERON       | 69  |
| 110 CHRISTOPHER NOLAN   | 53  |

10 rows selected.

**//INSERTING VALUES TO STARS :**

**Query**

SQL> desc stars;

| Name   | Null?    | Type          |
|--------|----------|---------------|
| S_ID   | NOT NULL | NUMBER(38)    |
| S_NAME |          | VARCHAR2(40)  |
| ABOUT  |          | VARCHAR2(100) |
| DOB    |          | DATE          |

SQL>

SQL> insert into stars values(501,'PRANAV MOHANLAL','MALAYALAM ACTOR','13/jul/1990');

1 row created.

SQL> insert into stars values(502,'DULQUER SALMAAN','MALAYALAM ACTOR','28/jul/1986');

1 row created.

```
SQL> insert into stars values(503,'DILEEP','MALAYALAM ACTOR','27/oct/1967');
```

1 row created.

```
SQL> insert into stars values(504,'RAJINIKANTH','TAMIL ACTOR','12/dec/1950');
```

1 row created.

```
SQL> insert into stars values(505,'VIJAY','TAMIL ACTOR','22/jun/1974');
```

1 row created.

```
SQL> insert into stars values(506,'AISHWARYA RAI BACHCHAN','TAMIL ACTRESS','01/nov/1973');
```

1 row created.

```
SQL> insert into stars values(507,'AAMIR KHAN','BOLLYWOOD ACTOR','14/mar/1965');
```

1 row created.

```
SQL> insert into stars values(508,'SUSHANT SINGH RAJPUT','BOLLYWOOD ACTOR','21/jan/1986');
```

1 row created.

```
SQL> insert into stars values(509,'CILLIAN MURPHY','HOLLYWOOD ACTOR','25/may/1976');
```

1 row created.

```
SQL> insert into stars values(510,'ARNOLD SCHWARZENEGGER','HOLLYWOOD  
ACTOR','30/jul/1947');
```

1 row created.

```
SQL> insert into stars values(511,'ZOE SALDANA','HOLLYWOOD ACTRESS','19/jun/1979');
```

1 row created.

```
SQL> insert into stars values(512,'MATTHEW MCCONAUGHEY','HOLLYWOOD  
ACTOR','4/nov/1969');
```

1 row created.

```
SQL> insert into stars values(513,'PARVATHY THIRUVOTHU','MALAYALAM  
ACTRESS','7/apr/1988');
```

1 row created.

```
SQL> select * from stars;
```

| S_ID S_NAME                | ABOUT             | DOB       |
|----------------------------|-------------------|-----------|
| 501 PRANAV MOHANLAL        | MALAYALAM ACTOR   | 13-JUL-90 |
| 502 DULQUER SALMAAN        | MALAYALAM ACTOR   | 28-JUL-86 |
| 503 DILEEP                 | MALAYALAM ACTOR   | 27-OCT-67 |
| 504 RAJINIKANTH            | TAMIL ACTOR       | 12-DEC-50 |
| 505 VIJAY                  | TAMIL ACTOR       | 22-JUN-74 |
| 506 AISHWARYA RAI BACHCHAN | TAMIL ACTRESS     | 01-NOV-73 |
| 507 AAMIR KHAN             | BOLLYWOOD ACTOR   | 14-MAR-65 |
| 508 SUSHANT SINGH RAJPUT   | BOLLYWOOD ACTOR   | 21-JAN-86 |
| 509 CILLIAN MURPHY         | HOLLYWOOD ACTOR   | 25-MAY-76 |
| 510 ARNOLD SCHWARZENEGGER  | HOLLYWOOD ACTOR   | 30-JUL-47 |
| 511 ZOE SALDANA            | HOLLYWOOD ACTRESS | 19-JUN-79 |
| S_ID S_NAME                | ABOUT             | DOB       |
| 512 MATTHEW MCCONAUGHEY    | HOLLYWOOD ACTOR   | 04-NOV-69 |
| 513 PARVATHY THIRUVOTHU    | MALAYALAM ACTRESS | 07-APR-88 |

13 rows selected.

### //INSERTING VALUES TO MOVIES :

#### Query

SQL> desc movies;

| Name        | Null?    | Type          |
|-------------|----------|---------------|
| M_ID        | NOT NULL | NUMBER(38)    |
| TITLE       | NOT NULL | VARCHAR2(40)  |
| RELEASEDATE |          | DATE          |
| IMAGE_URL   |          | VARCHAR2(100) |
| CERTIFICATE |          | VARCHAR2(20)  |
| RUNTIME     |          | NUMBER(3,2)   |
| IMDBRATING  |          | NUMBER(3,1)   |
| DESCRIPTION |          | VARCHAR2(100) |
| METAScore   |          | NUMBER(3,1)   |
| VOTES       |          | NUMBER(38)    |
| LANGUAGE    |          | VARCHAR2(20)  |
| GROSS       |          | NUMBER(12,2)  |
| HIT         |          | NUMBER(1)     |
| ENTRY_DATE  |          | DATE          |

SQL>

SQL> insert into movies

values(1001,'Hridayam','16/jun/2020','https://www.movies.com/Hridayam.jpg','U/A',2.34,8.4,'The emotional journey of Arun',90,93,'Malayalam',1600000000,1,'28/aug/2023');

1 row created.

```
SQL> insert into movies values(1002,'Meesa Madhavan','20/aug/2002','https://www.movies.com/Meesamadhavan.jpg','U',2.45,8,'Story of madhavan who is forced into a thief',92,94,'Malayalam',190000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1003,'Wonder women','18/nov/2022','https://www.movies.com/wonderwomen.jpg','U/A',1.2,5.2,'story of six pregnant women',60,66,'Malayalam',50000000,0,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1004,'Enthiran','1/oct/2010','https://www.movies.com/enthiran.jpg','U/A',2.5,7.1,'Story of humanoid robot',70,78,'Tamil',375000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1005,'Master','13/jan/2021','https://www.movies.com/master.jpg','U/A',2.59,7.3,'A professor clashes with a gangster',80,87,'Tamil',220000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1006,'Ponniyin Selvan:1','30/sep/2022','https://www.movies.com/ponniyinselvan1.jpg','U/A',2.5,7.6,'Chola Raja story',80,86,'Tamil',350000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1007,'3 idiots','25/dec/2009','https://www.movies.com/3idiots.jpg','U/A',2.51,8.4,'Story of 3 friends',90,94,'Hindi',460000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1008,'Chichchore','6/sep/2019','https://www.movies.com/chichchore.jpg','U/A',2.23,8.3,'life of college friends',90,91,'Hindi',182000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies values(1009,'Avatar','18/dec/2009','https://www.movies.com/avatar.jpg','U/A',2.42,7.9,'Sci-fi epic',80,86,'English',293000000,1,'28/aug/2023');
```

1 row created.

```
SQL> insert into movies
values(1010,'Interstellar','7/nov/2014','https://www.movies.com/interstellar.jpg','U/A',2.49,8.7,'E
x-NASA pilot tasked to find new planet for humans',90,92,'English',7150000000,1,'28/aug/2023');
```

1 row created.

```
SQL> commit;
```

Commit complete.

```
SQL> select * from movies;
```

| M_ID | TITLE             | RELEASED  | DATE | IMAGE_URL                                  | CERTIFICATE | RUNTIME    | IMDB  | BRATING   | DESCRIPTION  | GROSS | HIT | ENTRY_DAT |
|------|-------------------|-----------|------|--|-------------|------------|-------|-----------|--|-------|-----|-----------|
|      |                   |           |      |  |             | METAScore  | VOTES | LANGUAGE  |  |       |     |           |
| 1001 | Hridayam          | 16-JUN-20 |      | https://www.movies.com/Hridayam.jpg        | U/A         | 2.34       | 8.4   |           | The emotional journey of Arun                      |       |     |           |
|      |                   |           | 90   | 93   | Malayalam   | 1600000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1002 | Meesa Madhavan    | 20-AUG-02 |      | https://www.movies.com/Meesamadhavan.jpg   | U           | 2.45       | 8     |           | Story of madhavan who is forced into a thief       |       |     |           |
|      |                   |           | 92   | 94   | Malayalam   | 1900000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1003 | Wonder women      | 18-NOV-22 |      | https://www.movies.com/wonderwomen.jpg     | U/A         | 1.2        | 5.2   |           | story of six pregnant women                        |       |     |           |
|      |                   |           | 60   | 66   | Malayalam   | 50000000   | 0     | 28-AUG-23 |  |       |     |           |
| 1004 | Enthiran          | 01-OCT-10 |      | https://www.movies.com/enthiran.jpg        | U/A         | 2.5        | 7.1   |           | Story of humanoid robot                            |       |     |           |
|      |                   |           | 70   | 78   | Tamil       | 3750000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1005 | Master            | 13-JAN-21 |      | https://www.movies.com/master.jpg          | U/A         | 2.59       | 7.3   |           | A professor clashes with a gangster                |       |     |           |
|      |                   |           | 80   | 87   | Tamil       | 2200000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1006 | Ponniyin Selvan:1 | 30-SEP-22 |      | https://www.movies.com/ponniyinselfan1.jpg | U/A         | 2.5        | 7.6   |           | Chola Raja story                                   |       |     |           |
|      |                   |           | 80   | 86   | Tamil       | 3500000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1007 | 3 idiots          | 25-DEC-09 |      | https://www.movies.com/3idiots.jpg         | U/A         | 2.51       | 8.4   |           | Story of 3 friends                                 |       |     |           |
|      |                   |           | 90   | 94   | Hindi       | 4600000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1008 | Chichchore        | 06-SEP-19 |      | https://www.movies.com/chichchore.jpg      | U/A         | 2.23       | 8.3   |           | life of college friends                            |       |     |           |
|      |                   |           | 90   | 91   | Hindi       | 1820000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1009 | Avatar            | 18-DEC-09 |      | https://www.movies.com/avatar.jpg          | U/A         | 2.42       | 7.9   |           | Sci-fi epic  |       |     |           |
|      |                   |           | 80   | 86   | English     | 2930000000 | 1     | 28-AUG-23 |  |       |     |           |
| 1010 | Interstellar      | 07-NOV-14 |      | https://www.movies.com/interstellar.jpg    | U/A         | 2.49       | 8.7   |           | Ex-NASA pilot tasked to find new planet for humans |       |     |           |
|      |                   |           | 90   | 92   | English     | 7150000000 | 1     | 28-AUG-23 |  |       |     |           |

10 rows selected.

**//INSERTING VALUES TO MOVIESDIRECTORS :**

**Query**

SQL> desc moviesdirectors;

| Name        | Null?    | Type       |
|-------------|----------|------------|
| -----       |          |            |
| MOVIESID    | NOT NULL | NUMBER(38) |
| DIRECTORSID | NOT NULL | NUMBER(38) |

SQL>

SQL> insert into moviesdirectors values(1001,102);

1 row created.

SQL> insert into moviesdirectors values(1002,101);

1 row created.

SQL> insert into moviesdirectors values(1003,103);

1 row created.

SQL> insert into moviesdirectors values(1004,104);

1 row created.

SQL> insert into moviesdirectors values(1005,105);

1 row created.

SQL> insert into moviesdirectors values(1006,106);

1 row created.

SQL> insert into moviesdirectors values(1007,107);

1 row created.

SQL> insert into moviesdirectors values(1008,108);

1 row created.

SQL> insert into moviesdirectors values(1009,109);

1 row created.

SQL> insert into moviesdirectors values(1010,110);

1 row created.

SQL> select \* from moviesdirectors;

MOVIESID DIRECTORSID

| MOVIESID | DIRECTORSID |
|----------|-------------|
| 1001     | 102         |
| 1002     | 101         |
| 1003     | 103         |
| 1004     | 104         |
| 1005     | 105         |
| 1006     | 106         |
| 1007     | 107         |
| 1008     | 108         |
| 1009     | 109         |
| 1010     | 110         |

10 rows selected.

//INSERTING VALUES TO MOVIESSTARS :

Query

SQL> desc moviesstars;

| Name     | Null?    | Type       |
|----------|----------|------------|
| MOVIESID | NOT NULL | NUMBER(38) |
| STARSID  | NOT NULL | NUMBER(38) |

SQL>

SQL> insert into moviesstars values(1001,501);

1 row created.

SQL> insert into moviesstars values(1002,503);

1 row created.

SQL> insert into moviesstars values(1003,513);

1 row created.

SQL> insert into moviesstars values(1004,504);

1 row created.

SQL> insert into moviesstars values(1005,505);

1 row created.

SQL> insert into moviesstars values(1006,506);



1 row created.

```
SQL> insert into moviesstars values(1007,507);
```

1 row created.

```
SQL> insert into moviesstars values(1008,508);
```

1 row created.

```
SQL> insert into moviesstars values(1009,511);
```

1 row created.

```
SQL> insert into moviesstars values(1010,512);
```

1 row created.

```
SQL> select * from moviesstars;
```

| MOVIESID | STARSID |
|----------|---------|
| 1001     | 501     |
| 1002     | 503     |
| 1003     | 513     |
| 1004     | 504     |
| 1005     | 505     |
| 1006     | 506     |
| 1007     | 507     |
| 1008     | 508     |
| 1009     | 511     |
| 1010     | 512     |

10 rows selected.

➤ **Change value of Hit to 1 where 'Votes' greater than or equal to 90.**

**Query**

```
SQL> update movies set hit=1 where (votes >= 90);
```

5 rows updated.

➤ **Create table IndustryHit with the following columns:**

**Id**  
**Title**  
**Releasedate**  
**Language**  
**Votes**  
**Gross**

**The data types and null characteristics for these columns should be the same as the corresponding columns in the Movies table described at the beginning of the lab exercise.**

### Query

```
SQL> create table industryhit(i_id number(38),i_title varchar2(38),i_releasedate  
varchar2(40),i_language varchar2(10),i_votes number(38),i_gross number(12,2),constraint  
prmky_iid primary key(i_id));
```

Table created.

```
SQL> desc industryhit;
```

| Name          | Null?    | Type         |
|---------------|----------|--------------|
| -----         |          |              |
| I_ID          | NOT NULL | NUMBER(38)   |
| I_TITLE       |          | VARCHAR2(38) |
| I_RELEASEDATE |          | VARCHAR2(40) |
| I_LANGUAGE    |          | VARCHAR2(10) |
| I_VOTES       |          | NUMBER(38)   |
| I_GROSS       |          | NUMBER(12,2) |

➤ **New movies hit the box office; their data is as follows:**

**Id: 1014, 1021, 1032**

**Title: 2018: Everyone is a Hero, Oppenheimer, Maamannan**

**Releasedate: 5 May 2023, 21 July 2023, 29 June 2023**

**Language: Malayalam, English, Tamil**

**Votes: 97, 96, 95**

**Gross: 750000000, 500000000, 505000000**

**Add the new employees to the IndustryHit table.**

➤ **Insert data into the new IndustryHit table.**

### Query

```
SQL> insert into industryhit values(1014,'2018:Everyone is a  
Hero','5/may/2023','Malayalam',97,750000000);
```

1 row created.

```
SQL> insert into industryhit values(1021,'Oppenheimer','21/jul/2023','English',96,500000000);
```

1 row created.

```
SQL> insert into industryhit values(1032,'Maamannan','29/jun/2023','Tamil',95,505000000);
```

1 row created.

SQL> select \* from industryhit;

| I_ID | I_TITLE                 | I_RELEASEDATE | I_LANGUAGE | I_VOTES | I_GROSS   |
|------|-------------------------|---------------|------------|---------|-----------|
| 1014 | 2018:Everyone is a Hero | 5/may/2023    | Malayalam  | 97      | 750000000 |
| 1021 | Oppenheimer             | 21/jul/2023   | English    | 96      | 500000000 |
| 1032 | Maamannan               | 29/jun/2023   | Tamil      | 95      | 505000000 |

- **Insert data into the IndustryHit table by copying the appropriate columns in the Movies table for those Movies that have Votes greater than or equal to 95.**

**Query**

SQL> insert into industryhit (i\_id,i\_title,i\_releasedate,i\_language,i\_votes,i\_gross) select m\_id,title,releasedate,language,votes,gross from movies where votes >= 90;

5 rows created.

SQL> select \* from industryhit;

| I_ID | I_TITLE                 | I_RELEASEDATE | I_LANGUAGE | I_VOTES | I_GROSS    |
|------|-------------------------|---------------|------------|---------|------------|
| 1014 | 2018:Everyone is a Hero | 5/may/2023    | Malayalam  | 97      | 750000000  |
| 1021 | Oppenheimer             | 21/jul/2023   | English    | 96      | 500000000  |
| 1032 | Maamannan               | 29/jun/2023   | Tamil      | 95      | 505000000  |
| 1001 | Hridayam                | 16-JUN-20     | Malayalam  | 93      | 1600000000 |
| 1002 | Meesa Madhavan          | 20-AUG-02     | Malayalam  | 94      | 190000000  |
| 1007 | 3 idiots                | 25-DEC-09     | Hindi      | 94      | 4600000000 |
| 1008 | Chichchore              | 06-SEP-19     | Hindi      | 91      | 1820000000 |
| 1010 | Interstellar            | 07-NOV-14     | English    | 92      | 7150000000 |

8 rows selected.

- **Movie Oppenheimer got a Metascore of 80. Make the appropriate data change.**

**Query**

-----[FIRST ADDING OPPENHEIMER TO TABLE : MOVIES]-----

SQL> insert into movies  
values(1021,'Oppenheimer','21/jul/2023','https://www.movies.com/oppenheimer.jpg','U/A',3,8.6,'  
Development of the atomic bomb.',75,96,'English',5500000000,1,'28/aug/2023');

1 row created.

//UPDATING METAScore TO 80:

SQL> update movies set metascore=80 where m\_id=1021;

1 row updated.

- **Delete all movies whose Metascore is less than 50.**

**Query**

SQL> delete from movies where metascore < 50;

0 rows deleted.

- **Movie 'Voice Of Sathyanathan' was released.**

**For 'Voice Of Sathyanathan' enter the following data:**

**Id: 1015**

**Title: Voice Of Sathyanathan**

**Releasedate: 28 July 2023**

**Image\_url: https://m.media-amazon.com/imak2M\_.jpg**

**Certificate: U**

**Runtime: 2.10**

**ImdbRating: 7.4**

**Description: A man's life becomes increasingly complicated after his neighbor is injured in a dispute over a fence.**

**Metascore: 60**

**Votes: 90**

**Gross: 109500000**

**Query**

SQL> insert into movies values('1015','Voice Of Sathyanathan','18/jul/2023','https://m.media-amazon.com/imak2M\_.jpg','U',2.10,7.4,'A man life becomes increasing complicated after his neighbor is injured in a dispute over a fense',60,90,'Malayalam',109500000,0,'28/aug/2023');

1 row created.

- **Delete all rows from IndustryHit and drop the IndustryHit table.**

**Query**

SQL> delete from industryhit;

SQL> drop table industryhit;

Table dropped.

|                        |   |
|------------------------|---|
| <b>Description 3.2</b> | <b>Retrieval of data (Simple select query and select with 'where' options (include all relational and logical operators))</b> |
| <b>Date</b>            | 14/08/2023  |

➤ **List details of all movies.**

**Query**

SQL> select \* from movies;

| M_ID | TITLE             | RELEASED  | IMAGE_URL   | CERTIFICATE | RUNTIME | IMDB | BRATING | DESCRIPTION  | METAScore | VOTES | LANGUAGE  | GROSS      | HIT | ENTRY_DAT |
|------|-------------------|-----------|---|-------------|---------|------|---------|--|-----------|-------|-----------|------------|-----|-----------|
| 1001 | Hridayam          | 16-JUN-20 | <a href="https://www.movies.com/Hridayam.jpg">https://www.movies.com/Hridayam.jpg</a>               | U/A         | 2.34    | 8.4  |         | The emotional journey of Arun                      | 90        | 93    | Malayalam | 1600000000 | 1   | 28-AUG-23 |
| 1002 | Meesa Madhavan    | 20-AUG-02 | <a href="https://www.movies.com/Meesamadhavan.jpg">https://www.movies.com/Meesamadhavan.jpg</a>     | U           | 2.45    | 8    |         | Story of madhavan who is forced into a thief       | 92        | 94    | Malayalam | 1900000000 | 1   | 28-AUG-23 |
| 1003 | Wonder women      | 18-NOV-22 | <a href="https://www.movies.com/wonderwomen.jpg">https://www.movies.com/wonderwomen.jpg</a>         | U/A         | 1.2     | 5.2  |         | story of six pregnant women                        | 60        | 66    | Malayalam | 500000000  | 0   | 28-AUG-23 |
| 1004 | Enthiran          | 01-OCT-10 | <a href="https://www.movies.com/enthiran.jpg">https://www.movies.com/enthiran.jpg</a>               | U/A         | 2.5     | 7.1  |         | Story of humanoid robot                            | 70        | 78    | Tamil     | 3750000000 | 1   | 28-AUG-23 |
| 1005 | Master            | 13-JAN-21 | <a href="https://www.movies.com/master.jpg">https://www.movies.com/master.jpg</a>                   | U/A         | 2.59    | 7.3  |         | A professor clashes with a gangster                | 80        | 87    | Tamil     | 2200000000 | 1   | 28-AUG-23 |
| 1006 | Ponniyin Selvan:1 | 30-SEP-22 | <a href="https://www.movies.com/ponniyinSelvan1.jpg">https://www.movies.com/ponniyinSelvan1.jpg</a> | U/A         | 2.5     | 7.6  |         | Chola Raja story                                   | 80        | 86    | Tamil     | 3500000000 | 1   | 28-AUG-23 |
| 1007 | 3 idiots          | 25-DEC-09 | <a href="https://www.movies.com/3idiots.jpg">https://www.movies.com/3idiots.jpg</a>                 | U/A         | 2.51    | 8.4  |         | Story of 3 friends                                 | 90        | 94    | Hindi     | 4600000000 | 1   | 28-AUG-23 |
| 1008 | Chichchore        | 06-SEP-19 | <a href="https://www.movies.com/chichchore.jpg">https://www.movies.com/chichchore.jpg</a>           | U/A         | 2.23    | 8.3  |         | life of college friends                            | 90        | 91    | Hindi     | 1820000000 | 1   | 28-AUG-23 |
| 1009 | Avatar            | 18-DEC-09 | <a href="https://www.movies.com/avatar.jpg">https://www.movies.com/avatar.jpg</a>                   | U/A         | 2.42    | 7.9  |         | Sci-fi epic  | 80        | 86    | English   | 2930000000 | 1   | 28-AUG-23 |
| 1010 | Interstellar      | 07-NOV-14 | <a href="https://www.movies.com/interstellar.jpg">https://www.movies.com/interstellar.jpg</a>       | U/A         | 2.49    | 8.7  |         | Ex-NASA pilot tasked to find new planet for humans | 90        | 92    | English   | 7150000000 | 1   | 28-AUG-23 |
| 1021 | Oppenheimer       | 21-JUL-23 | <a href="https://www.movies.com/oppenheimer.jpg">https://www.movies.com/oppenheimer.jpg</a>         | U/A         | 3       | 8.6  |         | Development of the atomic bomb.                    |           |       |           |            |     |           |

80 96 English 5500000000 1 28-AUG-23

| M_ID        | TITLE                 | RELEASEDA  | IMAGE_URL   |
|-------------|-----------------------|------------|---|
| CERTIFICATE | RUNTIME               | IMDBRATING | DESCRIPTION   |
|             | METAScore             | VOTES      | LANGUAGE  |
|             |                       | GROSS      | HIT ENTRY_DAT   |
| 1015        | Voice Of Sathyanathan | 18-JUL-23  | <a href="https://m.media-amazon.com/imak2M.jpg">https://m.media-amazon.com/imak2M.jpg</a>         |
| U           | 2.1                   | 7.4        | A man life becomes increasing complicated after his neighbor is injured in a dispute over a fence |
|             | 60                    | 90         | Malayalam   |
|             |                       | 109500000  | 0 28-AUG-23   |

12 rows selected.

- **List Title, Votes, Releasedate, Gross where Gross collection greater than 5000,000,00. Sequence the results in descending order by Gross.**

**Query**

SQL> select title,votes,releasedate,gross from movies where gross > 500000000 order by gross desc;

| TITLE             | VOTES | RELEASEDA | GROSS      |
|-------------------|-------|-----------|------------|
| Interstellar      | 92    | 07-NOV-14 | 7150000000 |
| Oppenheimer       | 96    | 21-JUL-23 | 5500000000 |
| 3 idiots          | 94    | 25-DEC-09 | 4600000000 |
| Enthiran          | 78    | 01-OCT-10 | 3750000000 |
| Ponniyin Selvan:1 | 86    | 30-SEP-22 | 3500000000 |
| Avatar            | 86    | 18-DEC-09 | 2930000000 |
| Master            | 87    | 13-JAN-21 | 2200000000 |
| Chichchore        | 91    | 06-SEP-19 | 1820000000 |
| Hridayam          | 93    | 16-JUN-20 | 1600000000 |

9 rows selected.

- **Retrieve the titles and years of Tamil movies released in 2022.**

**Query**

-----TWO POSSIBLE QUERIES-----

SQL> select title,extract(year from releasedate) as YEAR from movies where language='Tamil' and (releasedate between '1/jan/2022' and '31/dec/2022');

| TITLE             | YEAR |
|-------------------|------|
| Ponniyin Selvan:1 | 2022 |

>>OR

SQL> select title,extract(year from releasedate) as YEAR from movies where language='Tamil' and (extract(year from releasedate)='2022');

| TITLE             | YEAR |
|-------------------|------|
| Ponniyin Selvan:1 | 2022 |

➤ **Get the titles, years, and meta scores of movies sorted in descending order of meta scores.**  
**Query**

SQL> select title,extract(year from releasedate) as YEAR,metascore from movies order by metascore desc;

| TITLE             | YEAR | METAScore |
|-------------------|------|-----------|
| Meesa Madhavan    | 2002 | 92        |
| 3 idiots          | 2009 | 90        |
| Interstellar      | 2014 | 90        |
| Chichchore        | 2019 | 90        |
| Hridayam          | 2020 | 90        |
| Master            | 2021 | 80        |
| Ponniyin Selvan:1 | 2022 | 80        |
| Oppenheimer       | 2023 | 80        |
| Avatar            | 2009 | 80        |
| Enthiran          | 2010 | 70        |
| Wonder women      | 2022 | 60        |

| TITLE                 | YEAR | METAScore |
|-----------------------|------|-----------|
| Voice Of Sathyanathan | 2023 | 60        |

12 rows selected.

➤ **List titles, years, languages, dates and votes of all Malayalam and English movies released before 2022 and ImdbRating less than 7. The list should be ordered by Title.**

**Query**

-----[FIRST UPDATING ONE OF MOVIE VALUE TO BE LESS THAN 2022]-----

SQL> update movies set releasedate='18/nov/2021' where m\_id=1003;

1 row updated.

SQL> select title,extract(year from releasedate) as YEAR,language,releasedate,votes from movies where language in ('Malayalam','English') and extract(year from releasedate) < '2022' and imdbrating < 7 order by title;

| TITLE        | YEAR | LANGUAGE  | RELEASEDA | VOTES |
|--------------|------|-----------|-----------|-------|
| Wonder women | 2021 | Malayalam | 18-NOV-21 | 66    |

- **List all the movies whose title starts with 'Open'. Order the result by descending order of their id.**

**Query**

```
SQL> select m_id,title from movies where title like('Open%') order by m_id desc;
```

no rows selected

```
SQL> select m_id,title from movies where title like('Oppen%') order by m_id desc;
```

```
M_ID TITLE
-----
1021 Oppenheimer
```

- **List Hit movies released in 2022 and 2023. Order the result by ascending order of their Titles.**

**Query**

-----[TWO POSSIBLE QUERIES]-----

```
SQL> select title as MOVIE from movies where hit=1 and extract(year from releasedate)
in('2022','2023') order by title asc;
```

```
MOVIE
-----
Oppenheimer
Ponniyin Selvan:1
```

>>OR

```
SQL> select title as MOVIE from movies where hit=1 and extract(year from releasedate) between
'2022' and '2023' order by title asc;
```

```
MOVIE
-----
Oppenheimer
Ponniyin Selvan:1
```

- **Retrieve movies with a runtime between 1.5 and 2.5 hours.**

**Query**

```
SQL> select title as movie_name, runtime from movies where runtime between 1.5 and 2.5;
```

| MOVIE_NAME            | RUNTIME |
|-----------------------|---------|
| Hridayam              | 2.34    |
| Meesa Madhavan        | 2.45    |
| Enthiran              | 2.5     |
| Ponniyin Selvan:1     | 2.5     |
| Chichchore            | 2.23    |
| Avatar                | 2.42    |
| Interstellar          | 2.49    |
| Voice Of Sathyanathan | 2.1     |

8 rows selected.



➤ **Retrieve movies with Metascore ratings below 50 and IMDb ratings above 6.0.**

**Query**

-----[FIRST UPDATING ONE OF METAScore TO < 50]-----

SQL> update movies set metascore = 45 where m\_id=1004;

1 row updated.

SQL> select title as movie\_name,metascore from movies where metascore < 50 and imdbrating > 6.0;

| MOVIE_NAME | METAScore |
|------------|-----------|
| Enthiran   | 45        |

➤ **Retrieve movies with no description provided.**

**Query**

-----[FIRST UPDATING ONE OF DESCRIPTION TO NULL]-----

SQL> update movies set description = null where m\_id = 1006;

1 row updated.

SQL> select title as movie\_name from movies where description is null;

| MOVIE_NAME        |
|-------------------|
| Ponniyin Selvan:1 |

|                        |   |
|------------------------|---|
| <b>Description 3.3</b> | <b>Functions: Numeric Data, Character Conversion and Group functions.</b> |
| <b>Date</b>            | 14/08/2023  |

- **Illustrate the different numeric functions using dual table (power,round, ceil, floor, abs, exp, greatest, least, mod, trunc, round,sign, sqrt etc.)**

**Query**

**//POWER:-**

SQL> select power(2,3) from dual;

POWER(2,3)

-----  
8

**//ROUND:-**

SQL> select round(12.345,2) from dual;

ROUND(12.345,2)

-----  
12.35

**//CEIL:-**

SQL> select ceil(12.345) from dual;

CEIL(12.345)

-----  
13

**//FLOOR:-**

SQL> select floor(12.345) from dual;

FLOOR(12.345)

-----  
12

**//ABS:-**

SQL> select abs(-12.345) from dual;

ABS(-12.345)

-----  
12.345

**//EXP:-**

SQL> select exp(2) from dual;

EXP(2)

-----  
7.3890561

**//GREATEST:-**

SQL> select greatest(1,2,3) from dual;

GREATEST(1,2,3)

-----  
3

**//LEAST:-**

SQL> select least(1,2,3) from dual;

LEAST(1,2,3)

-----  
1

**//MOD:-**

SQL> select mod(10,3) from dual;

MOD(10,3)

-----  
1

**//TRUNC:-**

SQL> select trunc(12.345,1) from dual;

TRUNC(12.345,1)

-----  
12.3

**//SIGN:-**

SQL> select sign(-12.345) from dual;

SIGN(-12.345)

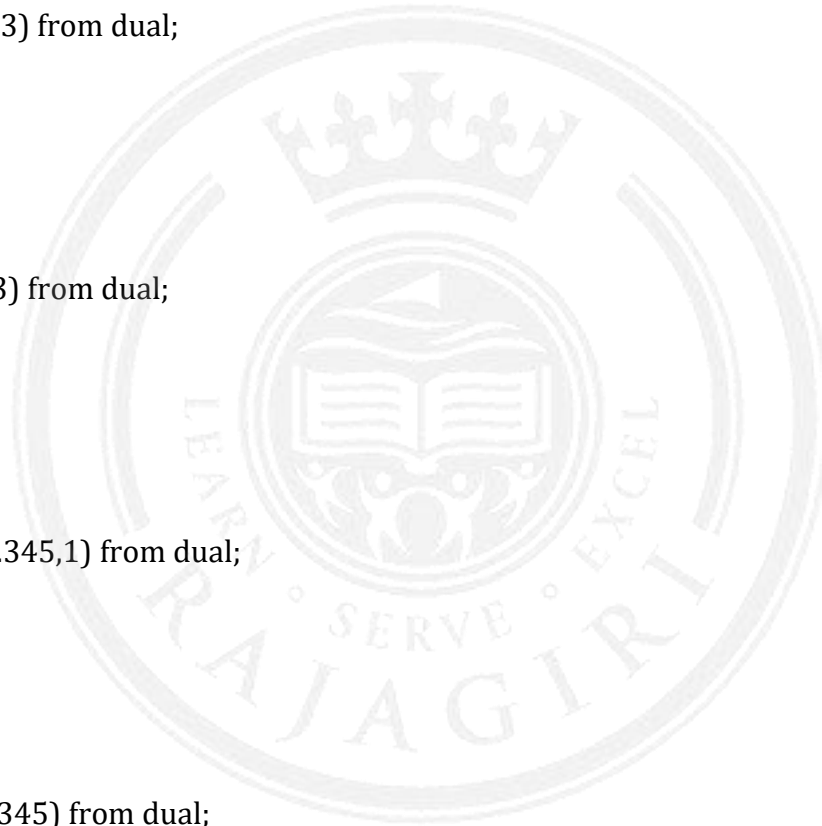
-----  
-1

**//SQRT:-**

SQL> select sqrt(16) from dual;

SQRT(16)

-----  
4



- Illustrate the character functions (upper, lower, initcap, length, concat, ascii, substr, ltrim, rtrim, trim, translate, instr, chr, Lpad, Rpad etc) using the table Movies.

//UPPER:-

**Query**

SQL> select title,upper(title) from movies;

| TITLE                 | UPPER(TITLE)          |
|-----------------------|-----------------------|
| Hridayam              | HRIDAYAM              |
| Meesa Madhavan        | MEESA MADHAVAN        |
| Wonder women          | WONDER WOMEN          |
| Enthiran              | ENTHIRAN              |
| Master                | MASTER                |
| Ponniyin Selvan:1     | PONNIYIN SELVAN:1     |
| 3 idiots              | 3 IDIOTS              |
| Chichchore            | CHICHCHORE            |
| Avatar                | AVATAR                |
| Interstellar          | INTERSTELLAR          |
| Oppenheimer           | OPPENHEIMER           |
| TITLE                 | UPPER(TITLE)          |
| Voice Of Sathyanathan | VOICE OF SATHYANATHAN |

12 rows selected.

//LOWER:-

**Query**

SQL> select title,lower(title) from movies;

| TITLE                 | LOWER(TITLE)          |
|-----------------------|-----------------------|
| Hridayam              | hridayam              |
| Meesa Madhavan        | meesa madhavan        |
| Wonder women          | wonder women          |
| Enthiran              | enthiran              |
| Master                | master                |
| Ponniyin Selvan:1     | ponniyin selvan:1     |
| 3 idiots              | 3 idiots              |
| Chichchore            | chichchore            |
| Avatar                | avatar                |
| Interstellar          | interstellar          |
| Oppenheimer           | oppenheimer           |
| TITLE                 | LOWER(TITLE)          |
| Voice Of Sathyanathan | voice of sathyanathan |

12 rows selected.

//INITCAP:-

**Query**

->The title of the movie with the first letter of each word capitalized

SQL> select title,initcap(title) from movies;

| TITLE                 | INITCAP(TITLE)        |
|-----------------------|-----------------------|
| Hridayam              | Hridayam              |
| Meesa Madhavan        | Meesa Madhavan        |
| Wonder women          | Wonder Women          |
| Enthiran              | Enthiran              |
| Master                | Master                |
| Ponniyin Selvan:1     | Ponniyin Selvan:1     |
| 3 idiots              | 3 Idiots              |
| Chichchore            | Chichchore            |
| Avatar                | Avatar                |
| Interstellar          | Interstellar          |
| Oppenheimer           | Oppenheimer           |
| Voice Of Sathyanathan | Voice Of Sathyanathan |

12 rows selected.

//LENGTH:-

**Query**

-> The length of the title of the movie:

SQL> select title,length(title) from movies;

| TITLE                 | LENGTH(TITLE) |
|-----------------------|---------------|
| Hridayam              | 8             |
| Meesa Madhavan        | 14            |
| Wonder women          | 12            |
| Enthiran              | 8             |
| Master                | 6             |
| Ponniyin Selvan:1     | 17            |
| 3 idiots              | 8             |
| Chichchore            | 10            |
| Avatar                | 6             |
| Interstellar          | 12            |
| Oppenheimer           | 11            |
| Voice Of Sathyanathan | 21            |

12 rows selected.

//CONCAT:-

**Query**

->The title of the movie concatenated with the language.

SQL> select title,concat(title,language) from movies;

| TITLE             | CONCAT(TITLE,LANGUAGE)  |
|-------------------|-------------------------|
| Hridayam          | HridayamMalayalam       |
| Meesa Madhavan    | Meesa MadhavanMalayalam |
| Wonder women      | Wonder womenMalayalam   |
| Enthiran          | EnthiranTamil           |
| Master            | MasterTamil             |
| Ponniyin Selvan:1 | Ponniyin Selvan:1Tamil  |
| 3 idiots          | 3 idiotsHindi           |
| Chichchore        | ChichchoreHindi         |
| Avatar            | AvatarEnglish           |
| Interstellar      | InterstellarEnglish     |
| Oppenheimer       | OppenheimerEnglish      |

| TITLE                 | CONCAT(TITLE,LANGUAGE)         |
|-----------------------|--------------------------------|
| Voice Of Sathyanathan | Voice Of SathyanathanMalayalam |

12 rows selected.

//ASCII:-

**Query**

->The ASCII code for the first letter is displayed:

SQL> select title,ASCII(title) from movies;

| TITLE             | ASCII(TITLE) |
|-------------------|--------------|
| Hridayam          | 72           |
| Meesa Madhavan    | 77           |
| Wonder women      | 87           |
| Enthiran          | 69           |
| Master            | 77           |
| Ponniyin Selvan:1 | 80           |
| 3 idiots          | 51           |
| Chichchore        | 67           |
| Avatar            | 65           |
| Interstellar      | 73           |
| Oppenheimer       | 79           |

| TITLE                 | ASCII(TITLE) |
|-----------------------|--------------|
| Voice Of Sathyanathan | 86           |

12 rows selected.

//SUBSTR:-

**Query**

->The first 3 characters of the title of the movie are:

SQL> select title,substr(title,1,3) from movies;

| TITLE             | SUB   |
|-------------------|-------|
| -----             | ----- |
| Hridayam          | Hri   |
| Meesa Madhavan    | Mee   |
| Wonder women      | Won   |
| Enthiran          | Ent   |
| Master            | Mas   |
| Ponniyin Selvan:1 | Pon   |
| 3 idiots          | 3 i   |
| Chichchore        | Chi   |
| Avatar            | Ava   |
| Interstellar      | Int   |
| Oppenheimer       | Opp   |

| TITLE                 | SUB   |
|-----------------------|-------|
| -----                 | ----- |
| Voice Of Sathyanathan | Voi   |

12 rows selected.

//LTRIM:-

**Query**

->The title of the movie with leading spaces trimmed:

SQL> select title,ltrim(title) from movies;

| TITLE             | LTRIM(TITLE)      |
|-------------------|-------------------|
| -----             | -----             |
| Hridayam          | Hridayam          |
| Meesa Madhavan    | Meesa Madhavan    |
| Wonder women      | Wonder women      |
| Enthiran          | Enthiran          |
| Master            | Master            |
| Ponniyin Selvan:1 | Ponniyin Selvan:1 |
| 3 idiots          | 3 idiots          |
| Chichchore        | Chichchore        |
| Avatar            | Avatar            |
| Interstellar      | Interstellar      |
| Oppenheimer       | Oppenheimer       |

| TITLE                 | LTRIM(TITLE)          |
|-----------------------|-----------------------|
| -----                 | -----                 |
| Voice Of Sathyanathan | Voice Of Sathyanathan |

12 rows selected.

```
SQL> select ltrim(' hello') from dual;
```

LTRIM

-----

hello

**//RTRIM:**

**Query**

->The title of the movie with trailing spaces trimmed:

```
SQL> select title,rtrim(title) from movies;
```

| TITLE             | RTRIM(TITLE)      |
|-------------------|-------------------|
| Hridayam          | Hridayam          |
| Meesa Madhavan    | Meesa Madhavan    |
| Wonder women      | Wonder women      |
| Enthiran          | Enthiran          |
| Master            | Master            |
| Ponniyin Selvan:1 | Ponniyin Selvan:1 |
| 3 idiots          | 3 idiots          |
| Chichchore        | Chichchore        |
| Avatar            | Avatar            |
| Interstellar      | Interstellar      |
| Oppenheimer       | Oppenheimer       |

| TITLE                 | RTRIM(TITLE)          |
|-----------------------|-----------------------|
| Voice Of Sathyanathan | Voice Of Sathyanathan |

12 rows selected.

```
SQL> select rtrim(' hello ') from dual;
```

RTRIM('H

-----

hello

**//TRIM :-**

**Query**

->The title of the movie with leading and trailing spaces trimmed:

```
SQL> select title,trim(title) from movies;
```



| TITLE | TRIM(TITLE) |
|-------|-------------|
|-------|-------------|

|                   |                   |
|-------------------|-------------------|
| Hridayam          | Hridayam          |
| Meesa Madhavan    | Meesa Madhavan    |
| Wonder women      | Wonder women      |
| Enthiran          | Enthiran          |
| Master            | Master            |
| Ponniyin Selvan:1 | Ponniyin Selvan:1 |
| 3 idiots          | 3 idiots          |
| Chichchore        | Chichchore        |
| Avatar            | Avatar            |
| Interstellar      | Interstellar      |
| Oppenheimer       | Oppenheimer       |

| TITLE | TRIM(TITLE) |
|-------|-------------|
|-------|-------------|

|                       |                       |
|-----------------------|-----------------------|
| Voice Of Sathyanathan | Voice Of Sathyanathan |
|-----------------------|-----------------------|

12 rows selected.

SQL> select trim(' hello ') from dual;

TRIM(

-----

hello

**//TRANSLATE :-**

**Query**

The title of the movie with all the letters "a" will be replaced by "z":

SQL> select title,translate(title,'a','z') from movies;

| TITLE | TRANSLATE(TITLE,'A','Z') |
|-------|--------------------------|
|-------|--------------------------|

|                   |                   |
|-------------------|-------------------|
| Hridayam          | Hridzyzm          |
| Meesa Madhavan    | Meesz Mzdzhvzn    |
| Wonder women      | Wonder women      |
| Enthiran          | Enthirzn          |
| Master            | Mzster            |
| Ponniyin Selvan:1 | Ponniyin Selvzn:1 |
| 3 idiots          | 3 idiots          |
| Chichchore        | Chichchore        |
| Avatar            | Avztzr            |
| Interstellar      | Interstellzr      |
| Oppenheimer       | Oppenheimer       |

| TITLE | TRANSLATE(TITLE,'A','Z') |
|-------|--------------------------|
|-------|--------------------------|

|                       |                      |
|-----------------------|----------------------|
| Voice Of Sathyanathan | Voice Of Szthyznzhzn |
|-----------------------|----------------------|

12 rows selected.

//INSTR:-

**Query**

->The position of the substring "a" in the title of the movie is :

SQL> select title,instr(title,'a') from movies;

| TITLE             | INSTR(TITLE,'A') |
|-------------------|------------------|
| -----             | -----            |
| Hridayam          | 5                |
| Meesa Madhavan    | 5                |
| Wonder women      | 0                |
| Enthiran          | 7                |
| Master            | 2                |
| Ponniyin Selvan:1 | 14               |
| 3 idiots          | 0                |
| Chichchore        | 0                |
| Avatar            | 3                |
| Interstellar      | 11               |
| Oppenheimer       | 0                |

| TITLE                 | INSTR(TITLE,'A') |
|-----------------------|------------------|
| -----                 | -----            |
| Voice Of Sathyanathan | 11               |

12 rows selected.

//CHR:-

**Query**

SQL> select votes,chr(votes) from movies;

| VOTES | C |
|-------|---|
| ----- | - |
| 93    | ] |
| 94    | ^ |
| 66    | B |
| 78    | N |
| 87    | W |
| 86    | V |
| 94    | ^ |
| 91    | [ |
| 86    | V |
| 92    | \ |
| 96    | ` |

| VOTES | C |
|-------|---|
| ----- | - |
| 90    | Z |

12 rows selected.

//LPAD:-

Query

-> The title of the movie padded with specific number of \* to the left:

SQL> select title,lpad(title,20,'\*') from movies;

| TITLE             | LPAD(TITLE,20,'*')   |
|-------------------|----------------------|
| Hridayam          | *****Hridayam        |
| Meesa Madhavan    | *****Meesa Madhavan  |
| Wonder women      | *****Wonder women    |
| Enthiran          | *****Enthiran        |
| Master            | *****Master          |
| Ponniyin Selvan:1 | ***Ponniyin Selvan:1 |
| 3 idiots          | *****3 idiots        |
| Chichchore        | *****Chichchore      |
| Avatar            | *****Avatar          |
| Interstellar      | *****Interstellar    |
| Oppenheimer       | *****Oppenheimer     |

| TITLE                 | LPAD(TITLE,20,'*')   |
|-----------------------|----------------------|
| Voice Of Sathyanathan | Voice Of Sathyanatha |

12 rows selected.

//RPAD:-

Query

->The title of the movie padded with specific number of \* to the right:

SQL> select title,rpad(title,20,'\*') from movies;

| TITLE             | RPAD(TITLE,20,'*')   |
|-------------------|----------------------|
| Hridayam          | Hridayam*****        |
| Meesa Madhavan    | Meesa Madhavan*****  |
| Wonder women      | Wonder women*****    |
| Enthiran          | Enthiran*****        |
| Master            | Master*****          |
| Ponniyin Selvan:1 | Ponniyin Selvan:1*** |
| 3 idiots          | 3 idiots*****        |
| Chichchore        | Chichchore*****      |
| Avatar            | Avatar*****          |
| Interstellar      | Interstellar*****    |
| Oppenheimer       | Oppenheimer*****     |

| TITLE                 | RPAD(TITLE,20,'*')   |
|-----------------------|----------------------|
| Voice Of Sathyanathan | Voice Of Sathyanatha |

12 rows selected.

➤ **Illustration of conversion functions- to\_number,to\_char(numberconversion), to\_char(dateconversion)**

**//TO\_NUMBER :-**

**Query**

->This code will first convert the string '12345' to a number. The result will be a number with the data type NUMBER.

SQL> select TO\_NUMBER('12345') from dual;

TO\_NUMBER('12345')

-----  
12345

**//TO\_CHAR (NUMBER CONVERSION):-**

**Query**

SQL> SELECT TO\_CHAR(75917.63,'\$99,999.99') from dual;

TO\_CHAR(759

-----  
\$75,917.63

SQL> select gross,TO\_CHAR(gross,'\$999,99,99,999.99') from movies;

GROSS TO\_CHAR(GROSS,'\$99

-----  
1600000000 \$160,00,00,000.00  
1900000000 \$19,00,00,000.00  
500000000 \$5,00,00,000.00  
3750000000 \$375,00,00,000.00  
2200000000 \$220,00,00,000.00  
3500000000 \$350,00,00,000.00  
4600000000 \$460,00,00,000.00  
1820000000 \$182,00,00,000.00  
2930000000 \$293,00,00,000.00  
7150000000 \$715,00,00,000.00  
5500000000 \$550,00,00,000.00

GROSS TO\_CHAR(GROSS,'\$99

-----  
1095000000 \$10,95,00,000.00

12 rows selected.

**//TO\_CHAR (DATE CONVERSION):-**

**Query**

SQL> select sysdate,TO\_CHAR(sysdate,'day') from dual;

SYSDATE TO\_CHAR(S

-----

29-AUG-23 tuesday

SQL> select releasedate,TO\_CHAR(releasedate,'ddth-mon-yy') as DAY from movies;

RELEASEDA DAY

-----

16-JUN-20 16th-jun-20  
20-AUG-02 20th-aug-02  
18-NOV-21 18th-nov-21  
01-OCT-10 01st-oct-10  
13-JAN-21 13th-jan-21  
30-SEP-22 30th-sep-22  
25-DEC-09 25th-dec-09  
06-SEP-19 06th-sep-19  
18-DEC-09 18th-dec-09  
07-NOV-14 07th-nov-14  
21-JUL-23 21st-jul-23

RELEASEDA DAY

-----

18-JUL-23 18th-jul-23

12 rows selected.



➤ **Count the total no. of Movies**

**Query**

SQL> select COUNT(\*) as Total\_Movies from movies;

TOTAL\_MOVIES

-----  
12

SQL> select COUNT(m\_id) as Total\_Movies from movies;

TOTAL\_MOVIES

-----  
12

➤ **Calculate the average votes of movies.**

**Query**

SQL> select AVG(votes) from movies;

AVG(VOTES)

-----  
87.75

➤ **Determine the maximum and minimum collection of movies. Rename the output as Max\_Coll and Min\_Coll respectively.**

**Query**

SQL> select MAX(gross) as Max\_Coll, MIN(gross) as Min\_Col from movies;

MAX\_COLL MIN\_COL

-----  
7150000000 50000000

➤ **Count the number of movies crossed the collection 50,00,00,000.**

**Query**

SQL> select COUNT(\*) as movies\_crossed from movies where gross > 500000000;

MOVIES\_CROSSED

-----  
9

➤ **Count the hit movies of 2021.**

**Query**

SQL> select COUNT(\*) as hit\_movies from movies where hit=1 and extract(year from releasedate) = 2021;

HIT\_MOVIES

-----

1



|                        |  |
|------------------------|--|
| <b>Description 3.4</b> | <b>Data manipulations using date functions</b> |
| <b>Date</b>            | 14/08/2023                                     |

- **Provide a list of all movies which were released on June 16, 2020. Display the year and month of the released date and the Id. Sort the result by Id. Name the derived columns YEAR and MONTH.**

**Query**

SQL> select m\_id,title,TO\_CHAR(releasedate,'yyyy') as YEAR,TO\_CHAR(releasedate,'month') as MONTH from movies where releasedate='16/jun/2020' order by m\_id;

| M_ID TITLE    | YEAR MONTH |
|---------------|------------|
| 1001 Hridayam | 2020 june  |

- **List the number of months between release date and entry date of each movie.**

**Query**

SQL> select m\_id,title,releasedate,entry\_date,MONTHS\_BETWEEN(entry\_date,releasedate) as NO\_OF\_MONTHS\_BETWEEN from movies;

| M_ID TITLE                 | RELEASEDA ENTRY_DAT | NO_OF_MONTHS_BETWEEN |
|----------------------------|---------------------|----------------------|
| 1001 Hridayam              | 16-JUN-20 28-AUG-23 | 38.3870968           |
| 1002 Meesa Madhavan        | 20-AUG-02 28-AUG-23 | 252.258065           |
| 1003 Wonder women          | 18-NOV-21 28-AUG-23 | 21.3225806           |
| 1004 Enthiran              | 01-OCT-10 28-AUG-23 | 154.870968           |
| 1005 Master                | 13-JAN-21 28-AUG-23 | 31.483871            |
| 1006 Ponniyin Selvan:1     | 30-SEP-22 28-AUG-23 | 10.9354839           |
| 1007 3 idiots              | 25-DEC-09 28-AUG-23 | 164.096774           |
| 1008 Chichchore            | 06-SEP-19 28-AUG-23 | 47.7096774           |
| 1009 Avatar                | 18-DEC-09 28-AUG-23 | 164.322581           |
| 1010 Interstellar          | 07-NOV-14 28-AUG-23 | 105.677419           |
| 1021 Oppenheimer           | 21-JUL-23 28-AUG-23 | 1.22580645           |
| M_ID TITLE                 | RELEASEDA ENTRY_DAT | NO_OF_MONTHS_BETWEEN |
| 1015 Voice Of Sathyanathan | 18-JUL-23 28-AUG-23 | 1.32258065           |

12 rows selected.



- **List the Entry\_date in the format 'DD-Month-YY'.**

**Query**

SQL> select m\_id,TO\_CHAR(entry\_date,'DD-month-YY') from movies;

M\_ID TO\_CHAR(ENTRY\_D

-----  
1001 28-august -23  
1002 28-august -23  
1003 28-august -23  
1004 28-august -23  
1005 28-august -23  
1006 28-august -23  
1007 28-august -23  
1008 28-august -23  
1009 28-august -23  
1010 28-august -23  
1021 28-august -23

M\_ID TO\_CHAR(ENTRY\_D

-----  
1015 28-august -23

12 rows selected.

- **List the date, 8 days after today's date.**

**Query**

SQL> select sysdate+8 from dual;

SYSDATE+8

-----  
06-SEP-23

- **List all the movies which were released in the month of February.**

**Query**

-----[FIRST UPDATING ONE OF MOVIES's RELEASED MONTH TO FEB]-----

SQL> update movies set releasedate='20/feb/2002' where m\_id=1002;

1 row updated.

SQL> select m\_id,title from movies where TO\_CHAR(releasedate,'MON') = 'FEB';

M\_ID TITLE

-----  
1002 Meesa Madhavan

- Illustrate the different date functions using dual table (to\_date, Add\_months, last\_day, months\_between, next\_day, round etc.)

//TO\_DATE :-

Query

SQL> select TO\_DATE('2023-08-29','YYYY-MM-DD') from dual;

TO\_DATE('2023-08-29',  
-----  
29-AUG-23

//ADD\_MONTHS:-

Query

SQL> select sysdate,ADD\_MONTHS(sysdate,4) from dual;

SYSDATE ADD\_MONTH  
-----  
29-AUG-23 29-DEC-23

SQL>

//LAST\_DAY:-

Query

SQL> select sysdate,LAST\_DAY(sysdate) from dual;

SYSDATE LAST\_DAY(  
-----  
29-AUG-23 31-AUG-23

//MONTHS\_BETWEEN:-

Query

SQL> select MONTHS\_BETWEEN('25-AUG-23','25-DEC-22') from dual;

MONTHS\_BETWEEN('25-AUG-23','25-DEC-22')  
-----  
8

//NEXT\_DAY:-

Query

SQL> select sysdate,NEXT\_DAY(sysdate,'FRIDAY') from dual;

SYSDATE NEXT\_DAY(  
-----  
29-AUG-23 01-SEP-23

//ROUND:-

Query

SQL> select sysdate,ROUND(sysdate,'MM') as nearest\_month from dual;

| SYSDATE   | NEAREST_M |
|-----------|-----------|
| 29-AUG-23 | 01-SEP-23 |

-----

29-AUG-23 01-SEP-23

➤ **Illustration of special date formats using to\_char function (use of th,sp,sph)**

//TO\_CHAR(TH):-

Query

SQL> select sysdate,TO\_CHAR(sysdate,'ddth-mon-yy') from dual;

| SYSDATE   | TO_CHAR(SYS |
|-----------|-------------|
| 29-AUG-23 | 29th-aug-23 |

-----

29-AUG-23

-----

29th-aug-23

SQL> select releasedate,TO\_CHAR(releasedate,'ddth-mon-yy') as DAY from movies;

| RELEASEDA | DAY         |
|-----------|-------------|
| 16-JUN-20 | 16th-jun-20 |
| 20-FEB-02 | 20th-feb-02 |
| 18-NOV-21 | 18th-nov-21 |
| 01-OCT-10 | 01st-oct-10 |
| 13-JAN-21 | 13th-jan-21 |
| 30-SEP-22 | 30th-sep-22 |
| 25-DEC-09 | 25th-dec-09 |
| 06-SEP-19 | 06th-sep-19 |
| 18-DEC-09 | 18th-dec-09 |
| 07-NOV-14 | 07th-nov-14 |
| 21-JUL-23 | 21st-jul-23 |

| RELEASEDA | DAY         |
|-----------|-------------|
| 18-JUL-23 | 18th-jul-23 |

12 rows selected.

//TO\_CHAR(SP):-

**Query**

SQL> select sysdate,TO\_CHAR(sysdate,'ddsp-mon-yy') from dual;

| SYSDATE   | TO_CHAR(SYSDATE,'DD |
|-----------|---------------------|
| -----     | -----               |
| 29-AUG-23 | twenty-nine-aug-23  |

SQL> select releasedate,TO\_CHAR(releasedate,'ddsp-mon-yy') as DAY from movies;

| RELEASEDA | DAY                |
|-----------|--------------------|
| -----     | -----              |
| 16-JUN-20 | sixteen-jun-20     |
| 20-FEB-02 | twenty-feb-02      |
| 18-NOV-21 | eighteen-nov-21    |
| 01-OCT-10 | one-oct-10         |
| 13-JAN-21 | thirteen-jan-21    |
| 30-SEP-22 | thirty-sep-22      |
| 25-DEC-09 | twenty-five-dec-09 |
| 06-SEP-19 | six-sep-19         |
| 18-DEC-09 | eighteen-dec-09    |
| 07-NOV-14 | seven-nov-14       |
| 21-JUL-23 | twenty-one-jul-23  |

| RELEASEDA | DAY             |
|-----------|-----------------|
| -----     | -----           |
| 18-JUL-23 | eighteen-jul-23 |

12 rows selected.

//TO\_CHAR(SPTH):-

**Query**

SQL> select sysdate,TO\_CHAR(sysdate,'ddspth-mon-yy') from dual;

| SYSDATE   | TO_CHAR(SYSDATE,'DDSP |
|-----------|-----------------------|
| -----     | -----                 |
| 29-AUG-23 | twenty-ninth-aug-23   |

SQL> select releasedate,TO\_CHAR(releasedate,'ddsph-mon-yy') as DAY from movies;

| RELEASEDA | DAY                 |
|-----------|---------------------|
| -----     | -----               |
| 16-JUN-20 | sixteenth-jun-20    |
| 20-FEB-02 | twentieth-feb-02    |
| 18-NOV-21 | eighteenth-nov-21   |
| 01-OCT-10 | first-oct-10        |
| 13-JAN-21 | thirteenth-jan-21   |
| 30-SEP-22 | thirtieth-sep-22    |
| 25-DEC-09 | twenty-fifth-dec-09 |
| 06-SEP-19 | sixth-sep-19        |
| 18-DEC-09 | eighteenth-dec-09   |
| 07-NOV-14 | seventh-nov-14      |
| 21-JUL-23 | twenty-first-jul-23 |

| RELEASEDA | DAY               |
|-----------|-------------------|
| -----     | -----             |
| 18-JUL-23 | eighteenth-jul-23 |

12 rows selected.

➤ **Calculate the total gross earnings for movies released after June 16, 2020.**

**Query**

SQL> select SUM(GROSS) from movies where releasedate>'16/jun/2020';

| SUM(GROSS) |
|------------|
| -----      |
| 1.1360E+10 |

|                        |                       |
|------------------------|-----------------------|
| <b>Description 3.5</b> | <b>Set Operations</b> |
| <b>Date</b>            | 14/08/2023            |

- **Create a new table IndustryHit (Id, title, genre, Certificate, Gross, Releasedate). Insert some movies from Movies table and some new movies in the new table IndustryHit.**

**Query**

```
SQL> create table industryhit(i_id int,i_title varchar2(40),genre varchar2(40),certificate
varchar2(20),gross number(12,2),releasedate date,constraint prky_iid primary key(i_id));
```

Table created.

```
SQL> desc industryhit;
```

| Name        | Null?    | Type         |
|-------------|----------|--------------|
| I_ID        | NOT NULL | NUMBER(38)   |
| I_TITLE     |          | VARCHAR2(40) |
| GENRE       |          | VARCHAR2(40) |
| CERTIFICATE |          | VARCHAR2(20) |
| GROSS       |          | NUMBER(12,2) |
| RELEASEDATE |          | DATE         |

```
SQL> select constraint_name,constraint_type from user_constraints where
table_name='INDUSTRYHIT';
```

| CONSTRAINT_NAME | C |
|-----------------|---|
| PRKY_IID        | P |

```
SQL>
```

**//INSERTING:**

```
SQL> insert into industryhit (select m_id,title,description,certificate,gross,releasedate from movies
where m_id=1001 or m_id=1004 or m_id=1007 or m_id=1009 or m_id=1021);
```

5 rows created.

SQL> select \* from industryhit;

| I_ID | I_TITLE     | GENRE                           | CERTIFICATE | GROSS      | RELEASEDA |
|------|-------------|---------------------------------|-------------|------------|-----------|
| 1001 | Hridayam    | The emotional journey of Arun   | U/A         | 1600000000 | 16-JUN-20 |
| 1004 | Enthiran    | Story of humanoid robot         | U/A         | 3750000000 | 01-OCT-10 |
| 1007 | 3 idiots    | Story of 3 friends              | U/A         | 4600000000 | 25-DEC-09 |
| 1009 | Avatar      | Sci-fi epic                     | U/A         | 2930000000 | 18-DEC-09 |
| 1021 | Oppenheimer | Development of the atomic bomb. | U/A         | 5500000000 | 21-JUL-23 |

SQL> insert into industryhit values(1031,'Mission Impossible - Fallout','Action Thriller','U/A',7910000000,'27/jul/2018');

1 row created.

SQL> insert into industryhit values(1032,'Premam','Romance/Drama','U',7600000000,'29/may/2015');

1 row created.

SQL> insert into industryhit values(1033,'Dangal','Action/Sport','U',5380000000,'23/Dec/2016');

1 row created.

SQL> select \* from industryhit;

| I_ID | I_TITLE                      | GENRE                           | CERTIFICATE | GROSS      | RELEASEDA |
|------|------------------------------|---------------------------------|-------------|------------|-----------|
| 1001 | Hridayam                     | The emotional journey of Arun   | U/A         | 1600000000 | 16-JUN-20 |
| 1004 | Enthiran                     | Story of humanoid robot         | U/A         | 3750000000 | 01-OCT-10 |
| 1007 | 3 idiots                     | Story of 3 friends              | U/A         | 4600000000 | 25-DEC-09 |
| 1009 | Avatar                       | Sci-fi epic                     | U/A         | 2930000000 | 18-DEC-09 |
| 1021 | Oppenheimer                  | Development of the atomic bomb. | U/A         | 5500000000 | 21-JUL-23 |
| 1031 | Mission Impossible - Fallout | Action Thriller                 | U/A         | 7910000000 | 27-JUL-18 |
| 1032 | Premam                       | Romance/Drama                   | U           | 7600000000 | 29-MAY-15 |
| 1033 | Dangal                       | Action/Sport                    | U           | 5380000000 | 23-DEC-16 |

8 rows selected.

➤ **Retrieve the titles of all movies and industry hits which are in the action thriller genre.**

**Query**

SQL> select title from movies UNION select i\_title from industryhit where genre='Action Thriller';

TITLE

-----  
3 idiots  
Avatar  
Chichchore  
Enthiran  
Hridayam  
Interstellar  
Master  
Meesa Madhavan  
Mission Impossible - Fallout  
Oppenheimer  
Ponniyin Selvan:1

TITLE

-----  
Voice Of Sathyanathan  
Wonder women

13 rows selected.

➤ **Retrieve the titles of all movies including industry hits.**

**Query**

SQL> select title from movies UNION select i\_title from industryhit;

TITLE

-----  
3 idiots  
Avatar  
Chichchore  
Dangal  
Enthiran  
Hridayam  
Interstellar  
Master  
Meesa Madhavan  
Mission Impossible - Fallout  
Oppenheimer

TITLE

-----  
Ponniyin Selvan:1  
Premam  
Voice Of Sathyanathan  
Wonder women

15 rows selected.



➤ **Retrieve the titles of all movies which are not industry hits.**

**Query**

SQL> select title from movies MINUS select i\_title from industryhit;

TITLE

-----  
Chichchore  
Interstellar  
Master  
Meesa Madhavan  
Ponniyin Selvan:1  
Voice Of Sathyanathan  
Wonder women

7 rows selected.



|                        |   |
|------------------------|---|
| <b>Description 3.6</b> | <b>Illustration of Group By having clause</b> |
| <b>Date</b>            | 14/08/2023                                    |

- **For all genres, display genre type and the sum of all Gross for each genre. Name the derived column SUM\_COLL.**

**Query**

-----[FIRST UPDATING VALUES OF GENRE TO GET SOME SIMILAR ONES]-----

SQL> update industryhit set genre='Action Thriller' where i\_id=1004;

1 row updated.

SQL> update industryhit set genre='Romance/Drama' where i\_id=1001;

1 row updated.

SQL> update industryhit set genre='Romance/Drama' where i\_id=1007;

1 row updated.

SQL> select genre,SUM(gross) as SUM\_COLL from industryhit group by genre;

| GENRE                           | SUM_COLL   |
|---------------------------------|------------|
| Development of the atomic bomb. | 5500000000 |
| Sci-fi epic                     | 2930000000 |
| Action/Sport                    | 5380000000 |
| Action Thriller                 | 1.1660E+10 |
| Romance/Drama                   | 6960000000 |

- **For all genres, display the genre type and the number of titles. Name the derived column TITLE\_COUNT.**

**Query**

SQL> select genre,COUNT(i\_title) as TITLE\_COUNT from industryhit group by genre;

| GENRE                           | TITLE_COUNT |
|---------------------------------|-------------|
| Development of the atomic bomb. | 1           |
| Sci-fi epic                     | 1           |
| Action/Sport                    | 1           |
| Action Thriller                 | 2           |
| Romance/Drama                   | 3           |

- **Display the genres which have more than 3 titles.**

**Query**

-----[FIRST INSERTING EXTRA ROW TO GET MORE THAN 3 COUNT FOR SAME GENRE]-----

```
SQL> insert into industryhit
values(1034,'Titanic','Romance/Drama','U/A',6740000000,'19/Dec/1997');
```

1 row created.

```
SQL> select genre,COUNT(i_title) as TITLE_COUNT from industryhit group by genre having
COUNT(i_title) > 3;
```

| GENRE         | TITLE_COUNT |
|---------------|-------------|
| Romance/Drama | 4           |

- **Retrieve the total number of movies released in each year, only for years with at least 5 movies.**

**Query**

```
SQL> select TO_CHAR(releasedate,'yyyy'),COUNT(i_id) from industryhit group by
TO_CHAR(releasedate,'yyyy') having COUNT(i_id) >= 5;
```

no rows selected

```
SQL> select TO_CHAR(releasedate,'yyyy'),COUNT(i_id) from industryhit group by
TO_CHAR(releasedate,'yyyy') having COUNT(i_id) >= 2;
```

| TO_C | COUNT(I_ID) |
|------|-------------|
| 2009 | 2           |

- **List the certificates along with the number of movies for each certificate, but only show certificates with more than 3 movies.**

**Query**

```
SQL> select certificate,COUNT(i_id) from industryhit group by certificate having COUNT(i_id) > 3;
```

| CERTIFICATE | COUNT(I_ID) |
|-------------|-------------|
| U/A         | 7           |

- **Show the total gross earnings for each certificate, but only for certificates with total gross greater than \$1 million.**

#### **Query**

SQL> select certificate,SUM(gross) from industryhit group by certificate having SUM(gross) > 1000000;

| CERTIFICATE | SUM(GROSS) |
|-------------|------------|
| U/A         | 3.3030E+10 |
| U           | 6140000000 |

- **List the release years with the highest number of movies and the corresponding movie count, limited to the top 3 years.**

#### **Query**

##### **//INNER QUERY**

SQL> select to\_char(releasedate,'yyyy') year,count(i\_id) count from industryhit group by to\_char(releasedate,'yyyy') order by count(i\_id) desc;

| YEAR | COUNT |
|------|-------|
| 2009 | 2     |
| 2016 | 1     |
| 1997 | 1     |
| 2018 | 1     |
| 2015 | 1     |
| 2020 | 1     |
| 2023 | 1     |
| 2010 | 1     |

8 rows selected.

##### **//FINAL QUERY**

SQL> select year,count from(select to\_char(releasedate,'yyyy') year,count(i\_id) count from industryhit group by to\_char(releasedate,'yyyy') order by count(i\_id) desc) where rownum<4;

| YEAR | COUNT |
|------|-------|
| 2009 | 2     |
| 2023 | 1     |
| 1997 | 1     |

|                        |                    |
|------------------------|--------------------|
| <b>Description 3.7</b> | <b>Sub queries</b> |
| <b>Date</b>            | 14/08/2023         |

- **Retrieve the titles and runtime of movies with the highest Metascore.**

**Query**

SQL> select title, runtime from movies where metascore = (select MAX(metascore) from movies);

| TITLE          | RUNTIME |
|----------------|---------|
| -----          | -----   |
| Meesa Madhavan | 2.45    |

- **List the titles of movies with a Gross amount greater than the average Gross amount of all movies.**

**Query**

//INNER QUERY

SQL> select AVG(gross) from movies;

| AVG(GROSS) |
|------------|
| -----      |
| 2783291667 |

//FINAL QUERY

SQL> select title, gross from movies where gross > (select AVG(gross) from movies);

| TITLE             | GROSS      |
|-------------------|------------|
| -----             | -----      |
| Enthiran          | 3750000000 |
| Ponniyin Selvan:1 | 3500000000 |
| 3 idiots          | 4600000000 |
| Avatar            | 2930000000 |
| Interstellar      | 7150000000 |
| Oppenheimer       | 5500000000 |

6 rows selected.

- **Retrieve the titles and descriptions of movies with a Metascore lower than the average Metascore.**

**Query**

SQL> select title, metascore, description from movies where metascore < (select AVG(metascore) from movies);

| TITLE                 | METAScore | DESCRIPTION   |
|-----------------------|-----------|---|
| Wonder women          | 60        | story of six pregnant women   |
| Enthiran              | 45        | Story of humanoid robot   |
| Voice Of Sathyanathan | 60        | A man life becomes increasing complicated after his neighbor is injured in a dispute over a fence |

- **List the movie titles and their IMDb ratings for movies released in the year with the highest average IMDb rating.**

**Query**

**//INNER QUERY:**

SQL> select max(avg(imdbrating)) from movies group by to\_char(releasedate,'yyyy');

MAX(AVG(IMDBRATING))

-----  
8.7

**//INNER QUERY:**

SQL> select to\_char(releasedate,'yyyy') from movies group by to\_char(releasedate,'yyyy') having avg(imdbrating)=(select max(avg(imdbrating)) from movies group by to\_char(releasedate,'yyyy'));

TO\_C

----

2014

**//FINAL QUERY:**

SQL> select title,imdbrating from movies where to\_char(releasedate,'yyyy')=(select to\_char(releasedate,'yyyy') from movies group by to\_char(releasedate,'yyyy') having avg(imdbrating)=(select max(avg(imdbrating)) from movies group by to\_char(releasedate,'yyyy')));

| TITLE        | IMDBRATING |
|--------------|------------|
| Interstellar | 8.7        |

- Retrieve the movie titles and their IMDb ratings for movies that have a Metascore greater than twice their IMDb rating.

**Query**

**//INNER QUERY:**

SQL> select 2\*imdbrating from movies;

2\*IMDBRATING

-----  
16.8  
16  
10.4  
14.2  
14.6  
15.2  
16.8  
16.6  
15.8  
17.4  
17.2

2\*IMDBRATING

-----  
14.8

12 rows selected.

**//FINAL QUERY:**

SQL> select title,imdbrating from movies m where metascore > (select 2\*imdbrating from movies h where m.title=h.title);

| TITLE             | IMDBRATING |
|-------------------|------------|
| -----             | -----      |
| Hridayam          | 8.4        |
| Meesa Madhavan    | 8          |
| Wonder women      | 5.2        |
| Enthiran          | 7.1        |
| Master            | 7.3        |
| Ponniyin Selvan:1 | 7.6        |
| 3 idiots          | 8.4        |
| Chichchore        | 8.3        |
| Avatar            | 7.9        |
| Interstellar      | 8.7        |
| Oppenheimer       | 8.6        |

| TITLE                 | IMDBRATING |
|-----------------------|------------|
| -----                 | -----      |
| Voice Of Sathyanathan | 7.4        |

12 rows selected.

➤ Find the title and gross amount of the top 3 highest-grossing movies.

**Query**

**//INNER QUERY:**

SQL> select max(gross) from movies;

MAX(GROSS)

-----

7150000000

**//INNER QUERY:**

SQL> select title,gross from movies m1 where gross=(select max(gross) from movies m2 where m1.title=m2.title)order by gross desc;

| TITLE                 | GROSS      |
|-----------------------|------------|
| -----                 | -----      |
| Interstellar          | 7150000000 |
| Oppenheimer           | 5500000000 |
| 3 idiots              | 4600000000 |
| Enthiran              | 3750000000 |
| Ponniyin Selvan:1     | 3500000000 |
| Avatar                | 2930000000 |
| Master                | 2200000000 |
| Chichchore            | 1820000000 |
| Hridayam              | 1600000000 |
| Meesa Madhavan        | 1900000000 |
| Voice Of Sathyanathan | 1095000000 |

| TITLE        | GROSS     |
|--------------|-----------|
| -----        | -----     |
| Wonder women | 500000000 |

12 rows selected.

**//FINAL QUERY:**

SQL> select \* from(select title,gross from movies m1 where gross=(select max(gross) from movies m2 where m1.title=m2.title)order by gross desc) where rownum <= 3;

| TITLE        | GROSS      |
|--------------|------------|
| -----        | -----      |
| Interstellar | 7150000000 |
| Oppenheimer  | 5500000000 |
| 3 idiots     | 4600000000 |



- **Calculate the total number of votes received by movies released in the year 2022.**

**Query**

-----[FIRST UPDATING ONE OF MOVIE YEAR TO 2022]-----

SQL> update movies set releasedate='6/sep/2022' where m\_id=1008;

1 row updated.

**//INNER QUERY:**

SQL> select votes,TO\_CHAR(releasedate,'yyyy') from movies where TO\_CHAR(releasedate,'yyyy')='2022';

| VOTES | TO_C |
|-------|------|
| 86    | 2022 |
| 91    | 2022 |

**//FINAL QUERY:**

SQL> select sum(votes) from movies m1 where TO\_CHAR(releasedate,'yyyy')=(select TO\_CHAR(releasedate,'yyyy') from movies m2 where TO\_CHAR(releasedate,'yyyy')='2022' and m1.title=m2.title);

| SUM(VOTES) |
|------------|
| 177        |

- **List the titles and certificate ratings of movies that have an IMDb rating below the average IMDb rating.**

**Query**

**//INNER QUERY:**

SQL> select AVG(imdbrating) from movies;

| AVG(IMDBRATING) |
|-----------------|
| 7.74166667      |

**//FINAL QUERY:**

SQL> select title,certificate,imdbrating from movies where imdbrating < (select AVG(imdbrating) from movies);

| TITLE                 | CERTIFICATE | IMDBRATING |
|-----------------------|-------------|------------|
| Wonder women          | U/A         | 5.2        |
| Enthiran              | U/A         | 7.1        |
| Master                | U/A         | 7.3        |
| Ponniyin Selvan:1     | U/A         | 7.6        |
| Voice Of Sathyanathan | U           | 7.4        |