

## WORKSHEET 5 SQL

1. Write SQL query to show all the data in the Movie table.

ANS: CREATE OR  
REPLACE VIEW  
Q6(Number\_of\_movies)  
AS SELECT COUNT(DISTINCT T.title\_id) AS  
Number\_of\_movies  
FROM Titles AS T  
WHERE T.title\_type IN ('movie','video');  
SELECT \* FROM Q6;

2. Write SQL query to show the title of the longest runtime movie.

ANS: SELECT title, length  
FROM Movies  
WHERE studioName = 'Disney'  
AND year = 1990;

SELECT title AS name, length AS duration  
FROM Movies  
WHERE studioName = 'Disney'  
AND year = 1990;

We can compute the length in hours

SELECT title AS name, length/60 AS Length\_In\_Hours  
FROM Movies  
WHERE studioName = 'Disney'  
AND year = 1990;

3. Write SQL query to show the highest revenue generating movie title.

ANS: SELECT title  
FROM film  
WHERE film\_id in (SELECT film\_id  
FROM inventory  
WHERE inventory\_id in (  
SELECT inventory\_id  
FROM rental

```
GROUP BY inventory_id
ORDER BY count(inventory_id) DESC
)) limit 1;
```

```
FROM film
WHERE film_id IN ("highest revenue")
```

```
SELECT film_id
FROM inventory
WHERE inventory_id IN (
  SELECT inventory_id
  FROM rental
  GROUP BY inventory_id
  ORDER BY count(inventory_id) DESC
) limit(highest revenue);
```

```
select TITLE from FILM f
inner join INVENTORY i using (FILM_ID)
inner join RENTAL r using (INVENTORY_ID)
inner join PAYMENT p using (RENTAL_ID)
group by TITLE
order by sum(AMOUNT) desc
limit 1;
```

4. Write SQL query to show the movie title with maximum value of revenue/budget.

ANS: SELECT mov\_title, mov\_year, rev\_stars, mov\_rel\_country

```
FROM movie
NATURAL JOIN rating
WHERE rev_stars = (
  SELECT MAX(rev_stars)
FROM rating
);
```

```
select TITLE from FILM f
inner join INVENTORY i using (FILM_ID)
inner join RENTAL r using (INVENTORY_ID)
inner join PAYMENT p using (RENTAL_ID)
group by TITLE
order by sum(AMOUNT) desc
limit 1;
```

5. Write a SQL query to show the movie title and its cast details like name of the person, gender, character name, cast order.

ANS: SELECT mov\_title, act\_fname, act\_lname, role

```
FROM movie
JOIN movie_cast
ON movie_cast.mov_id=movie.mov_id
JOIN actor
ON movie_cast.act_id=actor.act_id
WHERE actor.act_id IN (
SELECT act_id
FROM movie_cast
GROUP BY act_id HAVING COUNT(*)>=2);
```

```
SELECT mov_title, mov_year, rev_stars, mov_rel_country
FROM movie
NATURAL JOIN rating
WHERE rev_stars = (
SELECT MAX(rev_stars)
FROM rating
);
```

6. Write a SQL query to show the country name where maximum number of movies has been produced, along with the number of movies produced.

ANS: SELECT movie.mov\_title, mov\_year, mov\_dt\_rel,

```
mov_time,dir_fname, dir_lname
FROM movie
JOIN movie_direction
ON movie.mov_id = movie_direction.mov_id
JOIN director
ON movie_direction.dir_id=director.dir_id
WHERE mov_dt_rel <'01/01/1989'
ORDER BY mov_dt_rel desc;
```

```
SELECT mov_year,gen_title,count(gen_title), avg(rev_stars)
FROM movie
NATURAL JOIN movie_genres
NATURAL JOIN genres
```

```
NATURAL JOIN rating
WHERE gen_title='Mystery'
GROUP BY mov_year,gen_title;
```

7. Write a SQL query to show all the genre\_id in one column and genre\_name in second column.

```
ANS: insert into bands_table (genre_id, band_name)
select genre_id, 'Rammstein'
from genres_table
where genre_name = 'Rock';
```

```
INSERT INTO genres_table(genre_name) VALUES ('genre_name');
SET @last_id_in_table1 = LAST_INSERT_ID();
INSERT INTO bands_table(band_name,genre_id) VALUES
('band_name',@last_id_in_table1);
```

8. Write a SQL query to show name of all the languages in one column and number of movies in that particular column in another column.

```
ANS: SHOW [EXTENDED] [FULL] {COLUMNS | FIELDS}
```

```
{FROM | IN} tbl_name
[{FROM | IN} db_name]
[LIKE 'pattern' | WHERE expr]
```

```
SHOW COLUMNS FROM mytable FROM mydb;
SHOW COLUMNS FROM mydb.mytable;
```

```
select
title
    from film
    where (title like 'K%' or title like 'Q%')
    and language_id in (
        select language_id
        from language
        where name = 'English'
```

```
)  
order by title;
```

9. Write a SQL query to show movie name in first column, no. of crew members in second column and number of cast members in third column.

ANS: SELECT DISTINCT mov\_year

```
FROM movie  
INNER JOIN rating USING(mov_id)  
WHERE rev_stars IN (3,4)  
ORDER BY mov_year;
```

```
SELECT DISTINCT mov_year  
FROM movie NATURAL JOIN rating  
WHERE rev_stars IN (3, 4)  
ORDER BY mov_year;
```

10. Write a SQL query to list top 10 movies title according to popularity column in decreasing order.

ANS: CREATE OR REPLACE

```
VIEW  
Q4(runtime_minutes,  
title_type,  
primary_title) AS  
SELECT runtime_minutes, title_type, primary_title  
FROM Titles WHERE runtime_minutes > (10*60)  
ORDER BY runtime_minutes DESC, title_type ASC;  
FROM Titles WHERE runtime_minutes > (10*60)
```

11. Write a SQL query to show the name of the 3rd most revenue generating movie and its revenue.

ANS: select concat(first\_name, ' ', last\_name) as Customer\_name  
from category  
inner join film\_category  
using (category\_id)

```
inner join film
using (film_id)
inner join inventory
using (film_id)
inner join rental
using (inventory_id)
inner join customer
using (customer_id)
where name = 'Sci-Fi'
group by Customer_name
having count(rental_id) > 3
order by Customer_name;
```

12. Write a SQL query to show the names of all the movies which have “rumoured” movie status.

ANS:

CREATE OR

REPLACE VIEW

Q6(Number\_of\_movies)

AS SELECT COUNT(DISTINCT T.title\_id) AS Number\_of\_movies

FROM Titles AS T

WHERE T.title\_type IN ('movie','video');

SELECT \* FROM Q6

13. Write a SQL query to show the name of the “United States of America” produced movie which generated maximum revenue.

ANS: SELECT country\_name,city, department\_name

FROM countries

JOIN locations USING (country\_id)

JOIN departments USING (location\_id);

```

SELECT *
FROM movies
WHERE department_id IN (SELECT department_id
FROM departments
WHERE location_id IN (SELECT location_id
FROM locations
WHERE country_id IN (SELECT country_id
FROM countries
WHERE country_name = 'United STATES OF AMERICA')));

FROM employees e
INNER JOIN departments d
ON e.department_id = d.department_id
INNER JOIN locations l
ON d.location_id = l.location_id
WHERE e.salary = (SELECT MAX(revenue)
FROM movies
WHERE hire_date BETWEEN '2002-01-01' AND '2003-12-31');

```

14. Write a SQL query to print the movie\_id in one column and name of the production company in the second column for all the movies.

ANS: SELECT act\_fname, act\_lname, mov\_title, mov\_year

```

FROM actor
JOIN movie_cast
ON actor.act_id=movie_cast.act_id
JOIN movie
ON movie_cast.mov_id=movie.mov_id
WHERE mov_year NOT BETWEEN 1990 and 2000;

```

```

SELECT a.act_fname, a.act_lname, c.mov_title, c.mov_year
FROM actor a, movie_cast b, movie c
WHERE a.act_id=b.act_id
AND b.mov_id=c.mov_id
AND c.mov_year NOT BETWEEN 1990 and 2000;

```

```

CREATE OR REPLACE
VIEW
Q10(number_of_JB_actors)

```

```

AS SELECT COUNT(DISTINCT name_id) AS
number_of_JB_actors
FROM Q9;
SELECT * FROM Q10;

```

15. Write a SQL query to show the title of top 20 movies arranged in decreasing order of their budget.

```
ANS: CREATE OR REPLACE VIEW Q16(name_id,name_,Count)
AS SELECT H.name_id, N.name_, COUNT(*) AS Count
FROM Q15, Titles AS T, Names_ AS N, Had_role AS H
WHERE Q15.title_id = T.title_id
AND T.title_id = H.title_id
AND N.name_id = H.name_id
GROUP BY H.name_id
ORDER BY Count DESC
LIMIT 10;
SELECT * FROM Q16;
```

```
CREATE OR REPLACE VIEW Q4(runtime_minutes, title_type,
primary_title) AS
SELECT runtime_minutes, title_type, primary_title
FROM Titles WHERE runtime_minutes > (10*60)
ORDER BY runtime_minutes DESC, title_type ASC;
SELECT * FROM Q4 LIMIT 10;
```