# Assignment 1-3

# DBMS LAB

# IT552


Aditya Badayalya

510819056

Fifth Semester

Information Technology (HY)

**18. Count the number of Customers**

Using MySQL

```
select count(cust_id) as 'Number of Customers' from cust;
```

| | Number of Customers |
|---|---|
| ▶ | 7 |

**19. Calculate total price of all the movies**

Using MySQL

```
select sum(price) as 'Total Price' from movie;
```

| | Total Price |
|---|---|
| ▶ | 1643.84 |

**20. Calculate the average price of all movies**

Using MySQL

```
select sum(price)/count(mv_no) as 'Average Price' from
movie;
```

| | Average Price |
|---|---|
| ▶ | 273.973333 |

**21. Calculate the maximum and minimum movie prices. Rename the title as max_price and min_price respectively**

Using MySQL

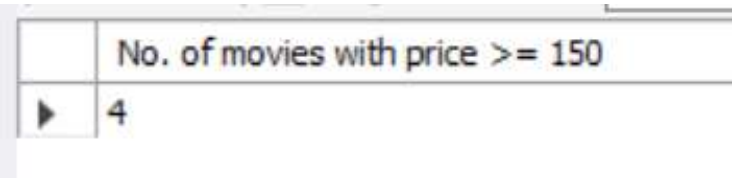
```
select max(price) as 'Max_Price', min(Price) as
'Min_Price' from movie;
```

| | Max_Price | Min_Price |
|---|---|---|
| ▶ | 502.00 | 100.22 |

**22. Count the number of movies having a price greater than or equal to 150**

Using MySQL

```
select count(price) as 'No. of movies with price >= 150'
from movie where price>=150;
```

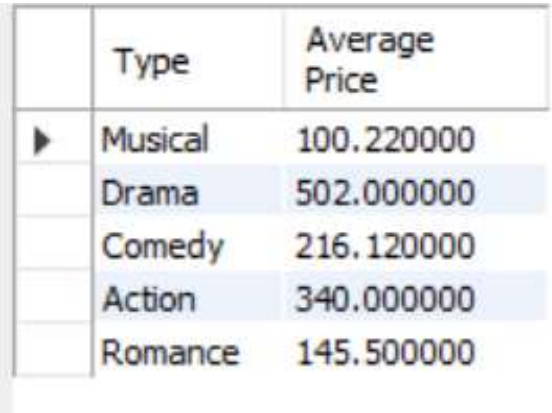


| | No. of movies with price >= 150 |
|---|---|
| ▶ | 4 |

**24. Print type and average price of each movie**

Using MySQL

```
select Type, sum(price)/count(mv_no) as 'Average Price'
from movie group by type;
```



| | Type | Average Price |
|---|---|---|
| ▶ | Musical | 100.220000 |
| | Drama | 502.000000 |
| | Comedy | 216.120000 |
| | Action | 340.000000 |
| | Romance | 145.500000 |

**25. Find the number of movies in each type**

Using MySQL

```
select Type, count(Type) as 'Number of Movies' from movie group by type;
```

| | Type | Number of Movies |
|---|---|---|
| ▶ | Musical | 1 |
| | Drama | 1 |
| | Comedy | 1 |
| | Action | 2 |
| | Romance | 1 |

**26. Count separately the number of movies in 'Comedy' and 'Drama' types**

Using MySQL

```
select type, count(type) as 'Number of Movies' from movie where type='Comedy' or type='Drama' group by type;
```

| | type | Number of Movies |
|---|---|---|
| ▶ | Drama | 1 |
| | Comedy | 1 |

**27. Calculate the average price of each type that has the maximum price of 150**

Using MySQL

```
select Type, count(type) as 'Number of Movies', sum(price) as 'Total Sum', sum(price)/count(type) as 'Average Cost' from movie where price <=150 group by type;
```

| | Type | Number of Movies | Total Sum | Average Cost |
|---|---|---|---|---|
| ▶ | Musical | 1 | 100.22 | 100.220000 |
| | Romance | 1 | 145.50 | 145.500000 |

**28. Calculate the average price of all the movies where the type is 'Comedy' or 'Action' and price is greater than or equal to 150**

Using MySQL

```
select Type, count(type) as 'Number of Movies', min(price) as 'Minimum Cost', sum(price) as 'Total Cost (each>=150)', sum(price)/count(type) as 'Average Cost' from movie where price >= 150 and (type = 'Comedy' or type ='Action') group by type;
```

| | Type | Number of Movies | Minimum Cost | Total Cost (each>=150) | Average Cost |
|---|---|---|---|---|---|
| ▶ | Comedy | 1 | 216.12 | 216.12 | 216.120000 |
| | Action | 2 | 180.00 | 680.00 | 340.000000 |

**29. Find out the movie number which has been issued to 'Raymond'**

Using MySQL

```
select invoice.Inv_no , cust.Fname, cust.Lname, invoice.Mv_no from invoice inner join cust on invoice.cust_id =  cust.cust_id and cust.Fname='Raymond';
```

| | Inv_no | Fname | Lname | Mv_no |
|---|---|---|---|---|
| ▶ | C1 | Raymond | Holt | 1 |

**30. Find the names and the movie numbers of all the customers who have been issued a movie**

Using MySQL

```
select invoice.Inv_no , cust.Fname, cust.Lname, invoice.Mv_no from invoice inner join cust on invoice.cust_id = cust.cust_id;
```

| | Inv_no | Fname | Lname | Mv_no |
|---|---|---|---|---|
| ▶ | AA1 | John | Doe | 6 |
| | B1 | Peter | Parker | 3 |
| | AD1 | Bruce | Wayne | 4 |
| | A1 | Bruce | Wayne | 2 |
| | Z1D | Barbara | Gordon | 5 |
| | C1 | Raymond | Holt | 1 |

**31. Select the title, cust_id, mv_no for all the movies that are issued**

Using MySQL

```
select invoice.Cust_id, invoice.Mv_no, movie.Title from
invoice inner join movie on movie.Mv_no = invoice.Mv_no;
```

| Cust_id | Mv_no | Title |
|---|---|---|
| 005 | 1 | La La Land |
| 003 | 2 | Lost in Translation |
| 002 | 3 | Deadpool |
| 003 | 4 | Mission Impossible |
| 004 | 5 | Casino Royale |
| 001 | 6 | Pride and Prejudice |

**32. Find the Title and Types of the movies that have been issued to 'Bruce'**

Using MySQL

```
select invoice.Inv_no, cust.Fname, movie.Title, movie.Type
from invoice join cust on invoice.Cust_id = cust.cust_id
join movie on invoice.Mv_no = movie.Mv_no
where cust.Fname = 'Bruce';
```

| Inv_no | Fname | Title | Type |
|---|---|---|---|
| A1 | Bruce | Lost in Translation | Drama |
| AD1 | Bruce | Mission Impossible | Action |

**33. Find the names of the customers who have issued the movie of type 'Drama'**

Using MySQL

```
select invoice.Inv_no, cust.Fname, cust.Lname,
movie.Title, movie.Type from invoice
join cust on invoice.Cust_id = cust.cust_id
join movie on invoice.Mv_no = movie.Mv_no
where movie.Type='Drama';
```

| Inv_no | Fname | Lname | Title | Type |
|---|---|---|---|---|
| A1 | Bruce | Wayne | Lost in Translation | Drama |

**34. Display the title, lname, fname for customers having movie number greater or equal to 3 in the following format**

**'The movie taken by {fname} {lname} is {title}'**

Using MySQL

```
select concat('The movie taken by ', cust.Fname, ' ',
cust.Lname, ' is ', movie.title) as 'Output' from invoice
join cust on invoice.Cust_id = cust.cust_id
join movie on movie.Mv_no = invoice.Mv_no
where invoice.Mv_no >= 3;
```

| Output |
|---|
| The movie taken by Peter Parker is Deadpool |
| The movie taken by Bruce Wayne is Mission Impossible |
| The movie taken by Barbara Gordon is Casino Royale |
| The movie taken by John Doe is Pride and Prejudice |