# BASIC SQL STATEMENTS

## SELECT

1. Retrieve all records with all columns from the “FilmLocations” table.
   ```
   SELECT * FROM FilmLocations;
   ```

2. Retrieve the names of all films with director names and writer names.
   ```
   SELECT Title, Director, Writer FROM FilmLocations;
   ```

3. Retrieve the names of all films released in the 21st century and onwards (release years after 2001 including 2001), along with filming locations and release years.
   ```
   SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE
   ReleaseYear>=2001;
   ```

4. Retrieve the fun facts and filming locations of all films.
   ```
   SELECT Locations, FunFacts FROM FilmLocations;
   ```

5. Retrieve the names of all films released in the 20th century and before (release years before 2000 including 2000) that, along with filming locations and release years.
   ```
   SELECT Title, ReleaseYear, Locations FROM FilmLocations WHERE
   ReleaseYear<=2000;
   ```

6. Retrieve the names, production company names, filming locations, and release years of the films which are not written by James Cameron.
   ```
   SELECT Title, ProductionCompany, Locations, ReleaseYear FROM
   FilmLocations WHERE Writer<>"James Cameron";
   ```

## COUNT

7. Retrieve the number of rows from the “FilmLocations” table.
   ```
   SELECT COUNT(*) FROM FilmLocations;
   ```

8. Retrieve the number of locations of the films which are written by James Cameron.
   ```
   SELECT COUNT(Locations) FROM FilmLocations WHERE Writer="James
   Cameron";
   ```

9. Retrieve the number of locations of the films which are directed by Woody Allen.
   ```
   SELECT COUNT(Locations) FROM FilmLocations WHERE Director="Woody
   Allen";
   ```

10. Retrieve the number of films shot at Russian Hill.
    ```
    SELECT Count(Title) FROM FilmLocations WHERE Locations="Russian Hill";
    ```

11. Retrieve the number of rows having a release year older than 1950 from the “FilmLocations” table.
    SELECT Count(*) FROM FilmLocations WHERE ReleaseYear<1950;

**DISTINCT**

12. Retrieve the name of all films without any repeated titles.
    SELECT DISTINCT Title FROM FilmLocations;

13. Retrieve the number of release years of the films distinctly, produced by Warner Bros. Pictures.
    SELECT COUNT(DISTINCT ReleaseYear) FROM FilmLocations WHERE ProductionCompany="Warner Bros. Pictures";

14. Retrieve the name of all unique films released in the 21st century and onwards, along with their release years.
    SELECT DISTINCT Title, ReleaseYear FROM FilmLocations WHERE ReleaseYear>=2001;

15. Retrieve the names of all the directors and their distinct films shot at City Hall.
    SELECT DISTINCT Title, Director FROM FilmLocations WHERE Locations="City Hall";

16. Retrieve the number of distributors distinctly who distributed films acted by Clint Eastwood as 1st actor.
    SELECT COUNT(DISTINCT Distributor) FROM FilmLocations WHERE Actor1="Clint Eastwood";

**LIMIT**

17. Retrieve the first 25 rows from the “FilmLocations” table.
    SELECT * FROM FilmLocations LIMIT 25;

18. Retrieve the first 15 rows from the “FilmLocations” table starting from row 11.
    SELECT * FROM FilmLocations LIMIT 15 OFFSET 10;

19. Retrieve the name of first 50 films distinctly.
    SELECT DISTINCT Title FROM FilmLocations LIMIT 50;

20. Retrieve first 10 film names distinctly released in 2015.
    SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 10;

21. Retrieve the next 3 film names distinctly after first 5 films released in 2015.

SELECT DISTINCT Title FROM FilmLocations WHERE ReleaseYear=2015 LIMIT 3 OFFSET 5;

**INSERT: used to insert new rows into a table**

**INSERT INTO table_name (column1, column2, ... )**

**VALUES (value1, value2, ... )**

**;**

1. Insert a new instructor record with id 4 for Sandip Saha who lives in Edmonton, CA into the “Instructor” table.
   INSERT INTO Instructor(ins_id, lastname, firstname, city, country)
   VALUES(10, 'Saha', 'Sandip', 'Edmonton', 'CA');

2. Insert two new instructor records into the “Instructor” table. First record with id 5 for John Doe who lives in Sydney, AU. Second record with id 6 for Jane Doe who lives in Dhaka, BD.
   INSERT INTO Instructor(ins_id, lastname, firstname, city, country)
   VALUES(5, 'Doe', 'John', 'Sydney', 'AU'), (6, 'Doe', 'Jane', 'Dhaka', 'BD');

3. Insert a new instructor record with id 3 for Antonio Cangiano who lives in Vancouver, CA into the “Instructor” table.
   INSERT INTO Instructor(ins_id, lastname, firstname, city, country)
   VALUES(3, ' Cangiano ', ' Antonio', 'Vancouver', 'CA');

4. Insert two new instructor records into the “Instructor” table. First record with id 11 for Selim Topcu who lives in Sneek, NL. Second record with id 12 for Oguzhan Topcu who lives in Blois, FR.
   INSERT INTO Instructor(ins_id, lastname, firstname, city, country)
   VALUES(11, ‘Topcu’, ‘Selim’, ‘Sneek’, ‘NL’), (12, ‘Topcu’, ‘Oguzhan’, ‘Blois’, ‘FR’);

   SELECT * FROM Instructor;

**UPDATE: used to update the data in existing rows in the table**

**UPDATE table_name**

**SET column1 = value1, column2 = value2, ...**

**WHERE condition**

**;**

1. Update the country for Selim to the Netherlands.
   UPDATE Instructor
   SET country=’NL’

WHERE firstname='Selim';

2. Update the city and country for Doe with id 5 to Dubai and AE respectively.
   UPDATE Instructor
   SET city='Dubai', country='AE'
   WHERE ins_id=5;
   SELECT * FROM Instructor;

3. Update the city of the instructor record to Haydarabad whose id is 1.
   UPDATE Instructor
   SET city='Haydarabad'
   WHERE ins_id=1;
   SELECT * FROM Instructor;

4. Update the city and country for Sandip with id 4 to Dhaka and BD respectively.
   UPDATE Instructor
   SET city='Dhaka', country='BD'
   WHERE ins_id=4;
   SELECT * FROM Instructor;

**DELETE: used to remove rows from a table**

**DELETE FROM table_name**

**WHERE condition**

**;**

1. Remove the instructor record of Doe whose id is 6
   DELETE FROM Instructor
   WHERE ins_id=6;
   SELECT * FROM Instructor;

2. Remove the instructor record of Hima.
   DELETE FROM Instructor
   WHERE ins_id=3;
   SELECT * FROM Instructor;