

SQL PROJECT

STRUCTURED QUERY LANGUAGE

Under Guidance:
Mr.Kavi Bharathi
(Besant Tech)

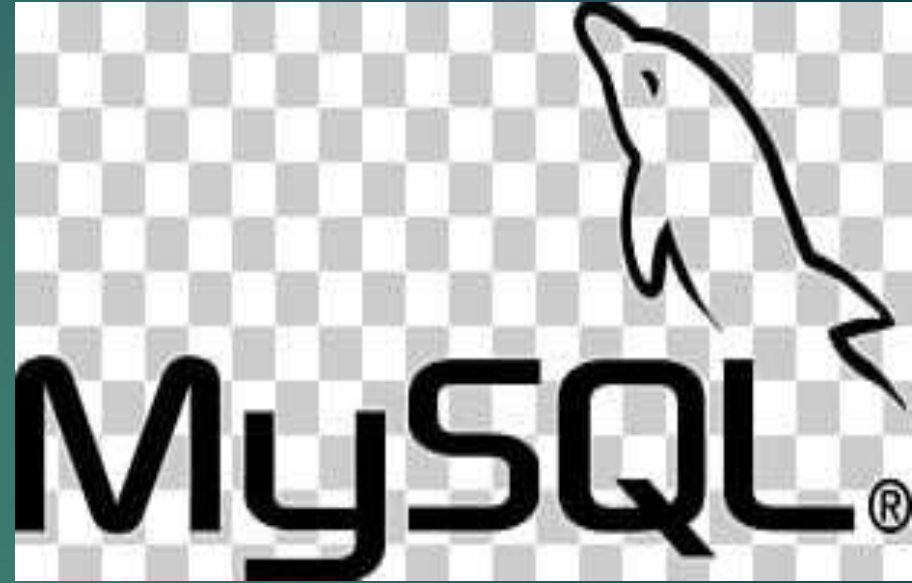
Presented By
Sujani Khan

SQL stands for Structured Query Language. It is a language used to communicate, manage or manipulate in a relational database management system (RDBMS). A relational database stores information in tabular form.



SQL commands are mainly categorized into five categories:

- DDL: Data Definition Language
- DQL: Data Query Language
- DML: Data Manipulation Language
- DCL: Data Control Language
- TCL: Transaction Control Language



Operators in SQL

An operator is a reserved word or character that is used to query our database in a SQL expression. To query a database using operators, we use WHERE clause.

Types of operators:

- 1.Arithmetic Operator
- 2.Concatenation Operator
- 3.Comparison Operator
- 4.Relational Operator
- 5.Logical Operator
- 6.Special Operator
- 7.Sub Query Operator



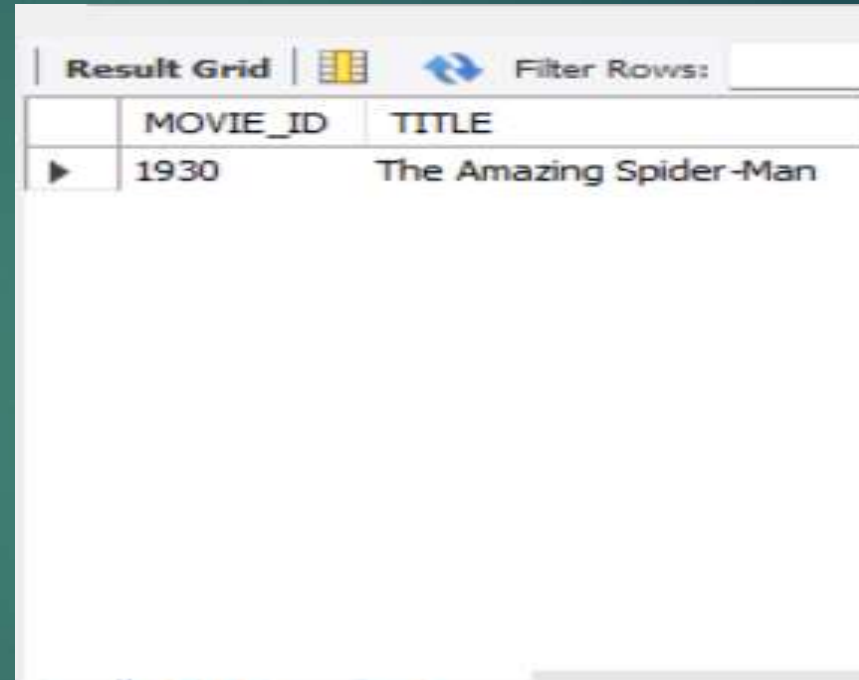
WAQTD budget and
original_title of the movies
where budget is greater
than 300000000.

```
SELECT BUDGET,  
ORIGINAL_TITLE  
FROM tmdb_5000_movies  
WHERE BUDGET>300000000;
```

Result Grid			Filter Rows:	Exp
	BUDGET	ORIGINAL_TITLE		
▶	237000000	Avatar		
	300000000	Pirates of the Caribbean: At World's End		
	245000000	Spectre		
	250000000	The Dark Knight Rises		
	260000000	John Carter		
	258000000	Spider-Man 3		
	260000000	Tangled		
	280000000	Avengers: Age of Ultron		
	250000000	Harry Potter and the Half-Blood Prince		
	250000000	Batman v Superman: Dawn of Justice		
	270000000	Superman Returns		
	200000000	Quantum of Solace		

WAQTD name of the movies whose movie_id is 1930.

```
SELECT MOVIE_ID, TITLE  
FROM tmdb_5000_credits  
WHERE MOVIE_ID=1930;
```

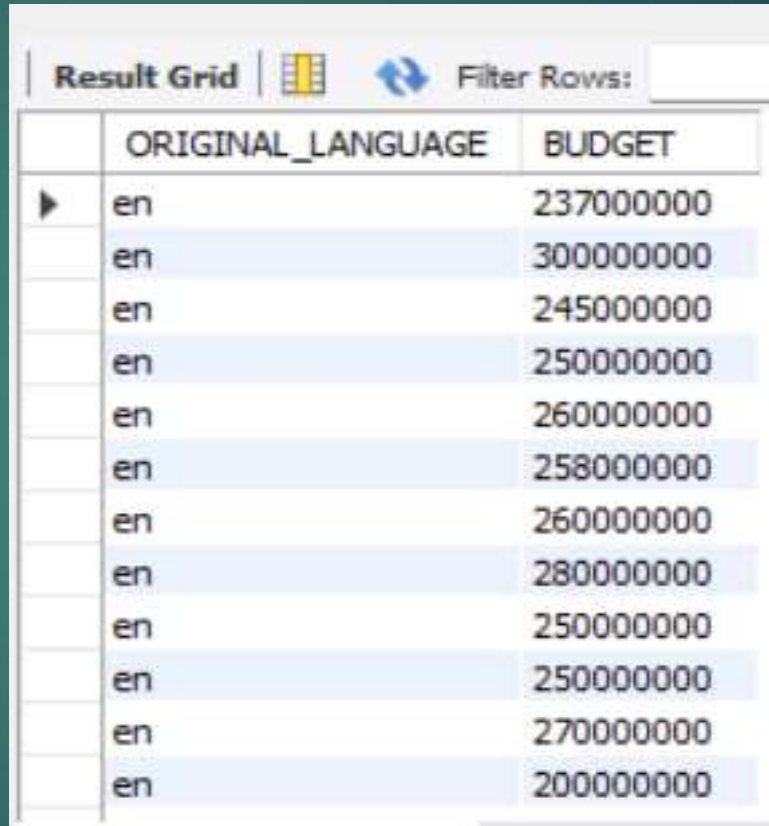


The screenshot shows a 'Result Grid' window with a toolbar containing a grid icon, a refresh icon, and a 'Filter Rows:' input field. The grid displays a single row of data with two columns: 'MOVIE_ID' and 'TITLE'. The value for 'MOVIE_ID' is 1930, and the value for 'TITLE' is 'The Amazing Spider-Man'.

	MOVIE_ID	TITLE
▶	1930	The Amazing Spider-Man

WAQTD movies that are in English language or budget is greater than 50000000.

```
SELECT  
ORIGINAL_LANGUAGE,  
BUDGET  
FROM tmdb_5000_movies  
WHERE  
ORIGINAL_LANGUAGE='EN'  
OR BUDGET>50000000;
```



The screenshot shows a 'Result Grid' window with a toolbar containing a film strip icon, a refresh icon, and a 'Filter Rows:' input field. The table below displays the results of a SQL query, showing 12 rows of data. The first column is 'ORIGINAL_LANGUAGE' and the second is 'BUDGET'.

	ORIGINAL_LANGUAGE	BUDGET
▶	en	237000000
	en	300000000
	en	245000000
	en	250000000
	en	260000000
	en	258000000
	en	260000000
	en	280000000
	en	250000000
	en	250000000
	en	270000000
	en	200000000

WAQTD original_title and
vote_average whose
vote_average between 7
and 9.

```
SELECT ORIGINAL_TITLE,  
VOTE_AVERAGE
```

```
FROM tmdb_5000_movies
```

```
WHERE VOTE_AVERAGE  
BETWEEN 7 AND 9;
```

Result Grid			Filter Rows:	Export:
	ORIGINAL_TITLE	VOTE_AVERAGE		
	Hugo	7		
	Edge of Tomorrow	7.6		
	Inside Out	8		
	Maleficent	7		
	Dawn of the Planet of the Apes	7.3		
	Captain America: The Winter Soldier	7.6		
	Big Hero 6	7.8		
	Wreck-It Ralph	7.1		
	How to Train Your Dragon	7.5		
	Guardians of the Galaxy	7.9		
	Interstellar	8.1		
	Inception	8.1		

WAQTD name of the
movies starting with T.

```
SELECT TITLE  
FROM tmdb_5000_credits  
Where title like 'T';
```



The screenshot shows a software window titled "Result Grid". It contains a table with a single column labeled "TITLE". The table lists 13 movie titles, each on a new row. The first row is "The Dark Knight Rises", followed by "Tangled", "The Lone Ranger", "The Chronicles of Narnia: Prince Caspian", "The Avengers", "The Hobbit: The Battle of the Five Armies", "The Amazing Spider-Man", "The Hobbit: The Desolation of Smaug", "The Golden Compass", "Titanic", "Transformers: Revenge of the Fallen", and "Transformers: Age of Extinction". The window also features a "Filter Rows:" input field and a refresh icon.

TITLE
The Dark Knight Rises
Tangled
The Lone Ranger
The Chronicles of Narnia: Prince Caspian
The Avengers
The Hobbit: The Battle of the Five Armies
The Amazing Spider-Man
The Hobbit: The Desolation of Smaug
The Golden Compass
Titanic
Transformers: Revenge of the Fallen
Transformers: Age of Extinction

WAQTD name of the movies in descending order.

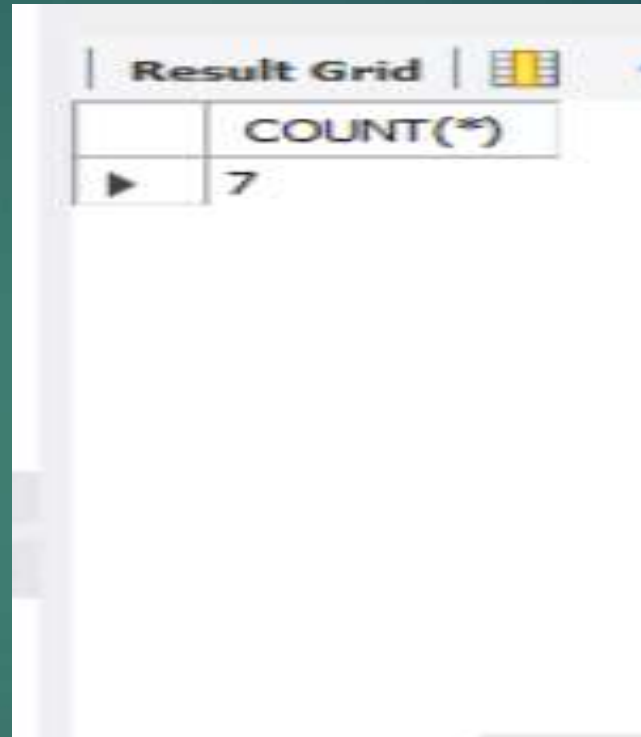
```
SELECT TITLE  
FROM tmdb_5000_credits  
ORDER BY TITLE DESC;
```



	TITLE
▶	Zulu
	Zoom
	Zoolander 2
	Zoolander
	Zookeeper
	Zombieland
	Zombie Hunter
	Zodiac
	ZMD: Zombies of Mass Destruction
	Zipper
	Zero Effect
	Zero Dark Thirty

WAQTD number of
movies having
vote_count greater than
9106.

```
SELECT COUNT(*)  
FROM tmdb_5000_movies  
WHERE  
VOTE_COUNT>9106;
```



The screenshot shows a 'Result Grid' window from a database application. It contains a single row of data. The first column is empty, and the second column is labeled 'COUNT(*)' and contains the value '7'. There is a small play button icon in the first column of the first row.

	COUNT(*)
▶	7

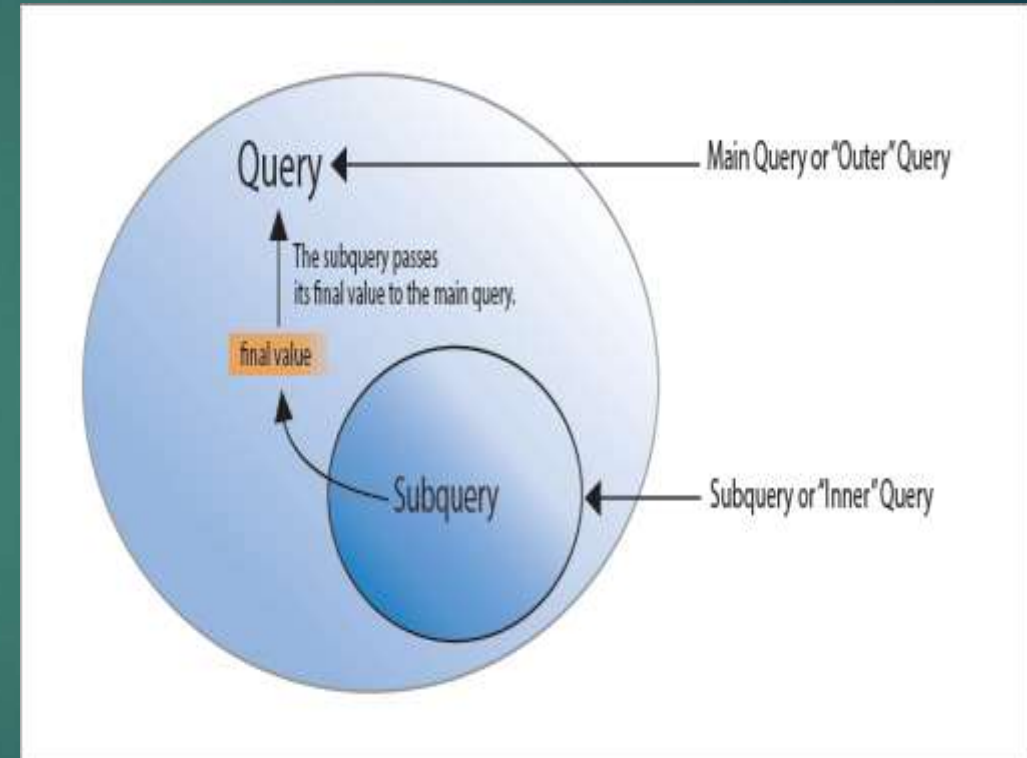
Sub Query in SQL

A query written inside another query is called sub query. In sub query the inner query will execute first and produce output. The output of inner query will be passed as a input to the outer query and then the outer query will produce the complete result. Therefore the outer query and inner query completely depends on each other

There are two cases of sub query

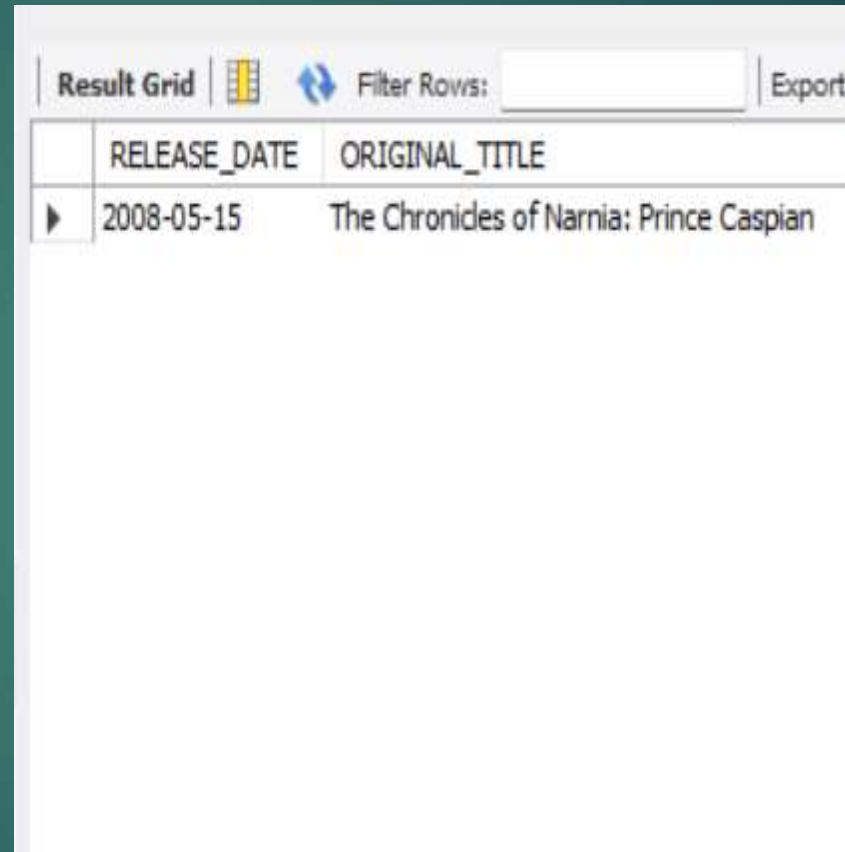
Case1:When we have unknown present in the query we go for sub query

Case2:Whenever the data to be selected and the condition are present in a different tables



WAQTD release date and original_title whose id is 2454.

```
SELECT  
RELEASE_DATE,ORIGINAL_  
TITLE  
FROM tmdb_5000_movies  
WHERE TITLE IN  
(SELECT TITLE  
FROM tmdb_5000_credits  
WHERE MOVIE_ID=2454);
```



The screenshot shows a database interface with a 'Result Grid' tab. It includes a 'Filter Rows' input field and an 'Export' button. The grid contains one row of data with two columns: 'RELEASE_DATE' and 'ORIGINAL_TITLE'.

	RELEASE_DATE	ORIGINAL_TITLE
▶	2008-05-15	The Chronicles of Narnia: Prince Caspian

WAQTD name of the movies and popularity whose popularity is less than The Avengers.

```
SELECT  
ORIGINAL_TITLE,POPULARITY  
FROM tmdb_5000_movies  
WHERE POPULARITY<  
(SELECT POPULARITY  
FROM tmdb_5000_movies  
WHERE ORIGINAL_TITLE='THE  
AVENGERS');
```

	ORIGINAL_TITLE	POPULARITY
►	Pirates of the Caribbean: At World's End	139.082615
	Spectre	107.376788
	The Dark Knight Rises	112.31295
	John Carter	43.926995
	Spider-Man 3	115.699814
	Tangled	48.681969
	Avengers: Age of Ultron	134.279229
	Harry Potter and the Half-Blood Prince	98.885637
	Superman Returns	57.925623
	Quantum of Solace	107.928811
	The Lone Ranger	49.046956
	Man of Steel	99.398009

Joins

It is a process of retrieve or fetch data from multiple table simultaneously.
There are five types of join

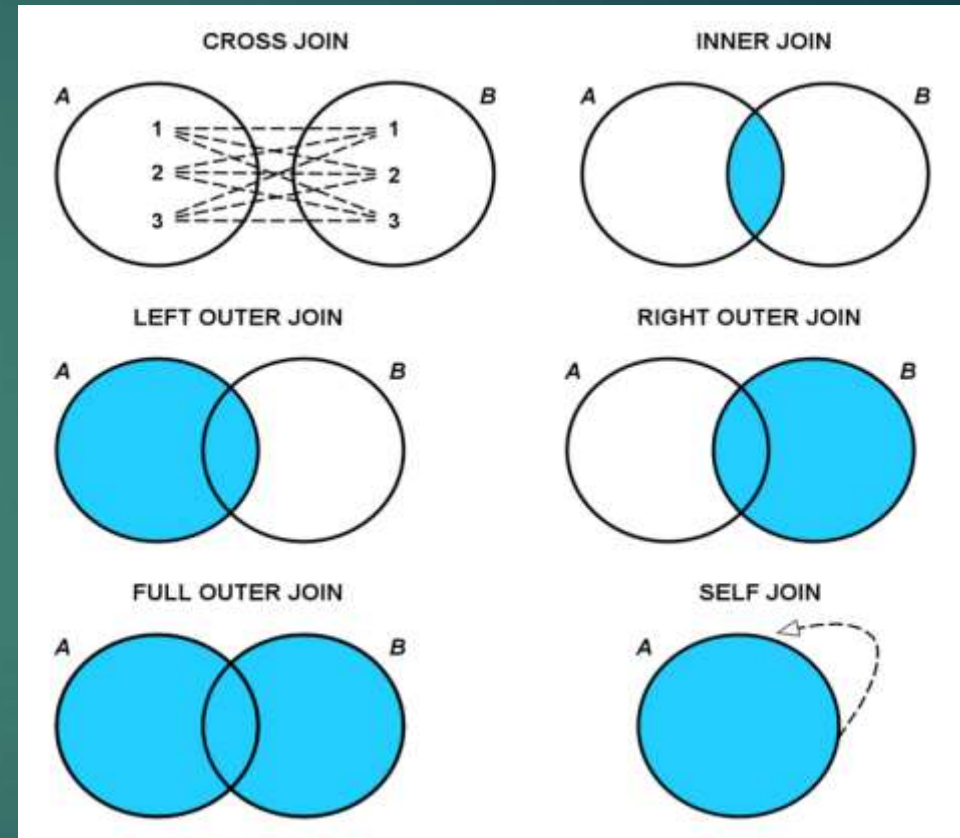
1. Cartesian Join or Cross Join

2. Inner Join

3. Outer Join (Left Outer, Right Outer and Full Outer)

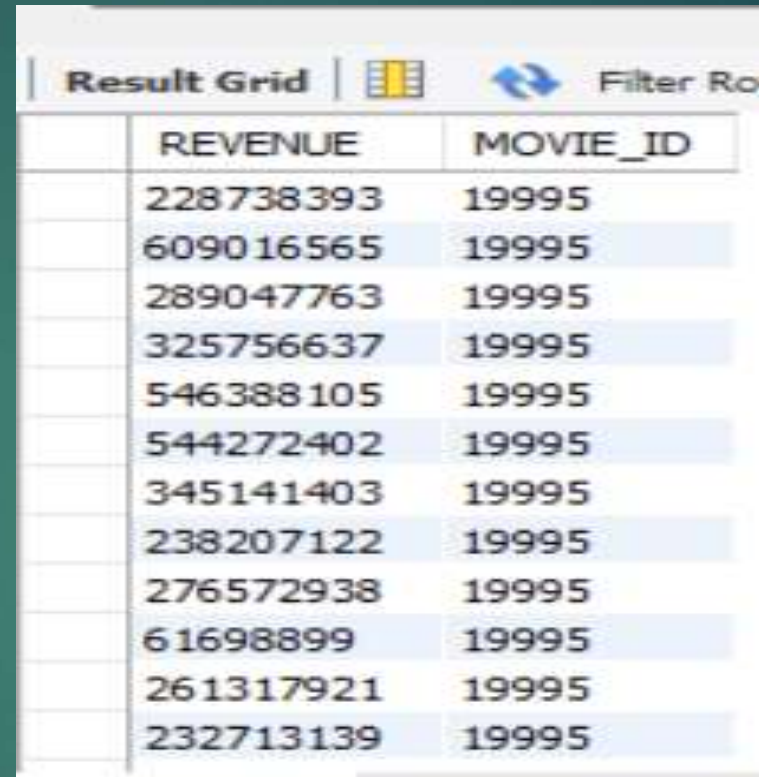
4. Natural Join

5. Self join



WAQTD revenue and
movie_id using cross join.

```
SELECT REVENUE,  
MOVIE_ID  
FROM tmdb_5000_movies  
CROSS JOIN  
tmdb_5000_credits;
```

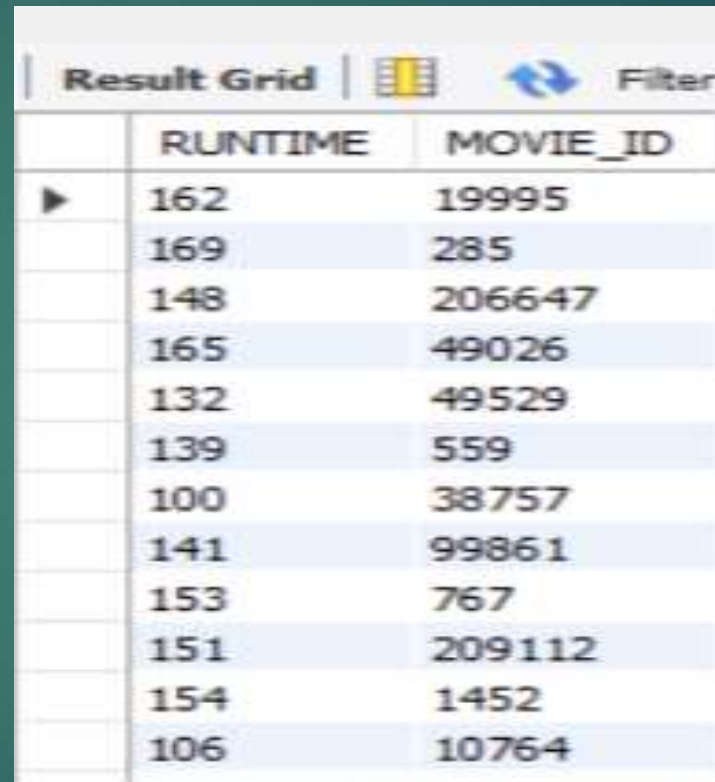


The screenshot shows a 'Result Grid' window with a table containing 12 rows of data. The table has two columns: 'REVENUE' and 'MOVIE_ID'. The 'MOVIE_ID' column contains the value '19995' for all rows, indicating a cross join. The 'REVENUE' column contains various numerical values. The window also features a 'Filter Rows' button and a grid icon.

	REVENUE	MOVIE_ID
	228738393	19995
	609016565	19995
	289047763	19995
	325756637	19995
	546388105	19995
	544272402	19995
	345141403	19995
	238207122	19995
	276572938	19995
	61698899	19995
	261317921	19995
	232713139	19995

WAQTD runtime and
movie_id using inner join.

```
SELECT RUNTIME,  
MOVIE_ID  
FROM tmdb_5000_movies  
INNER JOIN  
tmdb_5000_credits  
ON  
tmdb_5000_movies.TITLE=t  
mdb_5000_credits.TITLE;
```



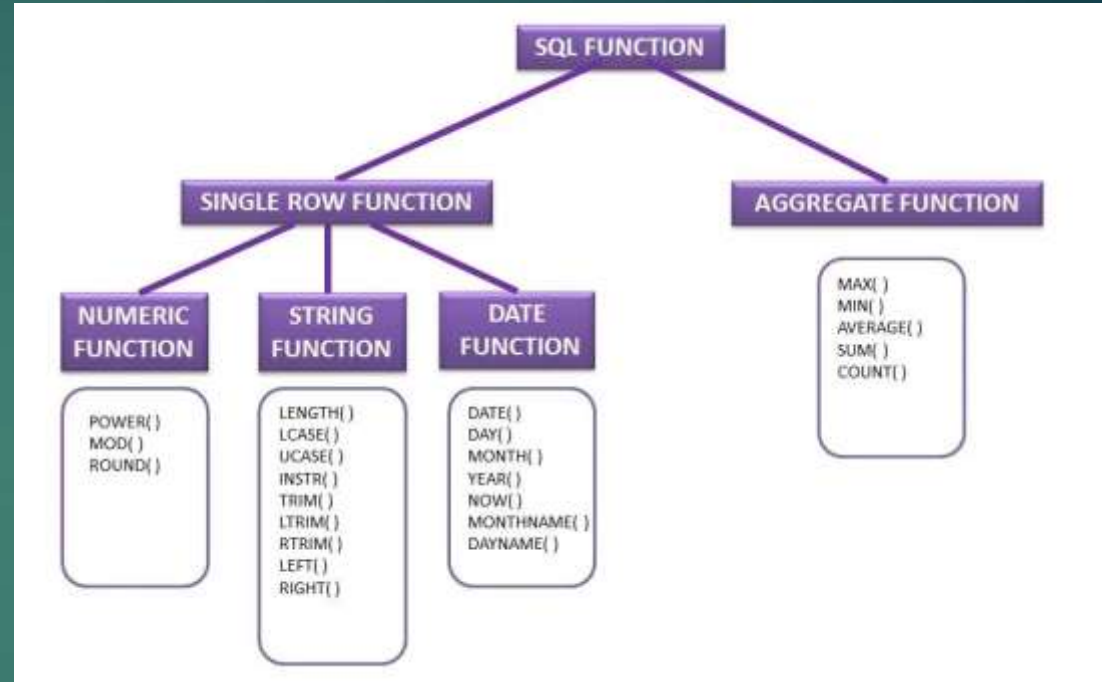
The image shows a screenshot of a database query result grid. The grid has two columns: 'RUNTIME' and 'MOVIE_ID'. There are 15 rows of data. The first row has a play button icon in the first column. The grid is titled 'Result Grid' and has a 'Filter' button.

	RUNTIME	MOVIE_ID
▶	162	19995
	169	285
	148	206647
	165	49026
	132	49529
	139	559
	100	38757
	141	99861
	153	767
	151	209112
	154	1452
	106	10764

Functions

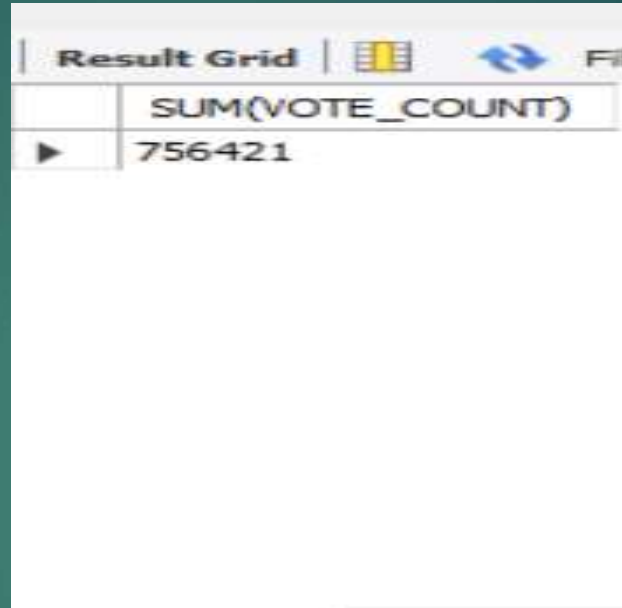
Functions are block of code or list of instruction which will do some specific task. There are two types of functions namely User Defined Function and Buildin Function (Aggregate Function). Buildin Function are of two types :

1. Single Row Function
2. Multi Row Function



WAQTD total vote_count.

```
SELECT  
SUM(VOTE_COUNT)  
FROM tmdb_5000_movies;
```



The image shows a screenshot of a SQL query result grid. The grid has a header row with the column name 'SUM(VOTE_COUNT)' and a single data row with the value '756421'. The grid is titled 'Result Grid' and has icons for refreshing and filtering.

	SUM(VOTE_COUNT)
▶	756421

WAQTD name of the movies whose budget is exactly nine digits.

```
SELECT  
ORIGINAL_TITLE,BUDGET  
FROM tmdb_5000_movies  
WHERE BUDGET  
LIKE'_____';
```

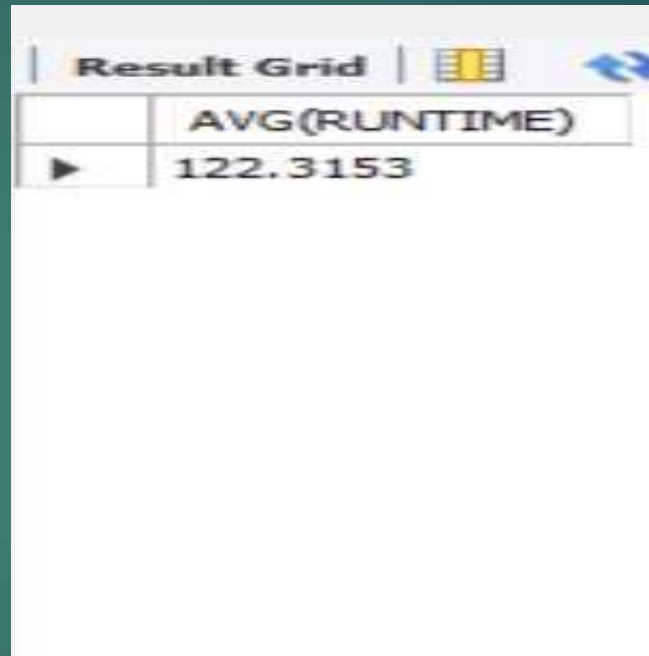


The screenshot shows a database query result grid with two columns: ORIGINAL_TITLE and BUDGET. The grid lists 12 movies with their respective budgets. The interface includes a 'Result Grid' header, a 'Filter Rows' input field, and an 'Exp' button.

	ORIGINAL_TITLE	BUDGET
▶	Avatar	237000000
	Pirates of the Caribbean: At World's End	300000000
	Spectre	245000000
	The Dark Knight Rises	250000000
	John Carter	260000000
	Spider-Man 3	258000000
	Tangled	260000000
	Avengers: Age of Ultron	280000000
	Harry Potter and the Half-Blood Prince	250000000
	Batman v Superman: Dawn of Justice	250000000
	Superman Returns	270000000
	Quantum of Solace	200000000

WAQTD average runtime.

```
SELECT AVG(RUNTIME)
FROM
tmdb_5000_movies;
```



A screenshot of a SQL query result grid. The grid has a title bar that says "Result Grid" with a yellow icon and a blue refresh icon. The grid contains one column with the header "AVG(RUNTIME)" and one row with the value "122.3153".

	AVG(RUNTIME)
▶	122.3153

THANK YOU