What is SQL? How should it be used?

> SQL (Structured Query Language) is a programming language used in relational database management systems (RDBMS) for database access and data manipulation.

> SQL is used to create queries in databases, define database schemas, manage database objects and manipulate data in the database.

Basic Components of SQL:

1. **Databases:** SQL is used to manage databases, which include data elements such as tables, indexes, and relationships.
2. **Tables:** The fundamental structure where data is stored. Each table consists of one or more columns, and each column represents a specific data type.
3. **Queries:** Used to retrieve information from the database, and the `SELECT` command forms the basis of queries.
4. **Relational Database Management Systems (RDBMS):** Software systems that process SQL commands and manage the database. Examples include MySQL, PostgreSQL, Microsoft SQL Server, and Oracle.

SQL Commands:

1. **SELECT:** Used to retrieve data. You can select specific columns, filter, and sort.

```
SELECT column1, column2 FROM table WHERE condition;
```

2. **INSERT:** Used to insert data.

```
INSERT INTO table (column1, column2) VALUES (value1, value2);
```

3. **UPDATE:** Used to update existing data.

```
UPDATE table SET column1 = value1 WHERE condition;
```

4. **DELETE:** Used to delete data.

```
DELETE FROM table WHERE condition;
```

5. **CREATE:** Used to create new databases, tables, or indexes.

```
CREATE DATABASE database_name;
CREATE TABLE table_name (column1 datatype, column2 datatype, ...);
```

6. **ALTER:** Used to modify or edit existing tables.

```
ALTER TABLE table_name ADD column_name datatype;
```

7. **DROP:** Used to delete databases, tables, or indexes.

```
DROP DATABASE database_name;
DROP TABLE table_name;
```

These basic commands illustrate the general use of SQL. SQL also includes more advanced features such as more complex queries, functions, grouping and joining. Each database management system (e.g. MySQL, PostgreSQL, Microsoft SQL Server) may support different SQL features, so it is important to refer to the documentation for the specific system used.

Functions and Expressions in SQL:

1. **WHERE:** Used to specify conditions in queries.
2. **ORDER BY:** Used to sort results in a specific order.
3. **GROUP BY:** Used to group data based on a specific column.

4. **JOIN:** Used to combine two or more tables.
5. **HAVING:** Used with GROUP BY to specify conditions on groups.
6. **COUNT, SUM, AVG, MIN, MAX:** Aggregate functions used to perform operations like counting, summing, averaging, finding the minimum and maximum values on data.

- **COUNT:** Used to count rows that meet a specific condition.

```
SELECT COUNT(*) FROM table_name WHERE condition;
```

- **SUM:** Used to find the total of numerical values in a specific column.

```
SELECT SUM(column_name) FROM table_name WHERE condition;
```

- **AVG:** Used to find the average of numerical values in a specific column.

```
SELECT AVG(column_name) FROM table_name WHERE condition;
```

- **MIN:** Used to find the smallest value in a specific column.

```
SELECT MIN(column_name) FROM table_name WHERE condition;
```

- **MAX:** Used to find the largest value in a specific column.

```
SELECT MAX(column_name) FROM table_name WHERE condition;
```

Examples

Below, you can find some examples that demonstrate how SQL commands are written and executed, assuming I am using a customer table as follows:

customer_id	first_name	last_name	city	gender	score
1	Robert	Johnson	New York	M	64
2	Emily	Anderson	Chicago	F	55
3	Alexander	Turner	Dallas	M	45
4	Olivia	White	New York	F	23
5	William	Miller	Miami	M	85

Example 1: You can write the following SQL command to select the first name, last name, and score columns from the customer table.

```
SELECT first_name, last_name, score FROM customer;
```

The output of this command will be as follows:

first_name	last_name	score
Robert	Johnson	64
Emily	Anderson	55
Alexander	Turner	45
Olivia	White	23
William	Miller	85

Example 2: You can write the following SQL command to select distinct values from the city column of the customer table.

```
SELECT DISTINCT city FROM customer;
```

The output of this command will be as follows:

city
New York
Chicago
Dallas

Miami

Example 3: You can write the following SQL command to select customers with gender 'M' and a score greater than 50.

```
SELECT * FROM customer WHERE gender = 'M' AND score > 50;
```

The output of this command will be as follows:

customer_id	first_name	last_name	city	gender	score
1	Robert	Johnson	New York	M	64
5	William	Miller	Miami	M	85

Example 4: You can write the following SQL command to select the top 3 customers with the highest scores.

```
SELECT * FROM customer ORDER BY score DESC LIMIT 3;
```

The output of this command will be as follows:

customer_id	first_name	last_name	city	gender	score
5	William	Miller	Miami	M	85
1	Robert	Johnson	New York	M	64
2	Emily	Anderson	Chicago	F	55

Important Explanation

💡 SQL is basically a standard and is supported by many relational database management systems (RDBMS). Therefore, the basic SQL commands I have provided are generally applicable to most RDBMSs. However, each RDBMS may have its own unique features, language extensions and some syntax rules that may differ.

For example, while popular RDBMSs such as MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, etc. generally follow the SQL standard, they may differ in some specific functions, data types or query structures. Therefore, as I mentioned earlier, it is always useful to review the documentation for the specific RDBMS being used.

The basic SQL commands are usually largely similar, but it is important to learn the specific details for a particular RDBMS. This can be done by reading the official documentation of the respective RDBMS or by studying the learning resources provided.