SQL SELECT Statement

In SQL, the SELECT statement is used to query or retrieve data from a table in the database. The returns data is stored in a table, and the result table is known as result-set.

**Syntax**

1. SELECT column1, column2, ...
2. FROM table\_name;

Here, the expression is the field name of the table that you want to select data from.

Use the following syntax to select all the fields available in the table:

1. SELECT  \*  FROM table\_name;

**Example:**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **PHONE\_NO** | **SALARY** |
| 1 | Kristen | Chicago | 9737287378 | 150000 |
| 2 | Russell | Austin | 9262738271 | 200000 |
| 3 | Angelina | Denver | 9232673822 | 600000 |
| 4 | Robert | Washington | 9367238263 | 350000 |
| 5 | Christian | Los angels | 7253847382 | 260000 |

To fetch the EMP\_ID of all the employees, use the following query:

1. SELECT EMP\_ID FROM EMPLOYEE;

**Output**

|  |
| --- |
| **EMP\_ID** |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

To fetch the EMP\_NAME and SALARY, use the following query:

1. SELECT EMP\_NAME, SALARY FROM EMPLOYEE;

|  |  |
| --- | --- |
| **EMP\_NAME** | **SALARY** |
| Kristen | 150000 |
| Russell | 200000 |
| Angelina | 600000 |
| Robert | 350000 |
| Christian | 260000 |

To fetch all the fields from the EMPLOYEE table, use the following query:

1. SELECT \* FROM EMPLOYEE

**Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **PHONE\_NO** | **SALARY** |
| 1 | Kristen | Chicago | 9737287378 | 150000 |
| 2 | Russell | Austin | 9262738271 | 200000 |
| 3 | Angelina | Denver | 9232673822 | 600000 |
| 4 | Robert | Washington | 9367238263 | 350000 |
| 5 | Christian | Los angels | 7253847382 | 260000 |

# SQL INSERT Statement

The SQL INSERT statement is used to insert a single or multiple data in a table. In SQL, You can insert the data in two ways:

1. Without specifying column name
2. By specifying column name

### **Sample Table**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Kristen | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |

### **1. Without specifying column name**

If you want to specify all column values, you can specify or ignore the column values.

**Syntax**

1. INSERT INTO TABLE\_NAME
2. VALUES (value1, value2, value 3, .... Value N);

**Query**

1. INSERT INTO EMPLOYEE VALUES (6, 'Marry', 'Canada', 600000, 48);

**Output:** After executing this query, the EMPLOYEE table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Kristen | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

### **2. By specifying column name**

To insert partial column values, you must have to specify the column names.

**Syntax**

1. INSERT INTO TABLE\_NAME
2. [(col1, col2, col3,.... col N)]
3. VALUES (value1, value2, value 3, .... Value N);

**Query**

1. INSERT INTO EMPLOYEE (EMP\_ID, EMP\_NAME, AGE) VALUES (7, 'Jack', 40);

**Output:** After executing this query, the table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Kristen | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |
| 7 | Jack | Null | null | 40 |

# SQL Update Statement

The SQL UPDATE statement is used to modify the data that is already in the database. The condition in the WHERE clause decides that which row is to be updated.

**Syntax**

1. UPDATE table\_name
2. SET column1 = value1, column2 = value2, ...
3. WHERE condition;

### **Sample Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Kristen | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

**EMPLOYEE**

## **Updating single record**

Update the column EMP\_NAME and set the value to 'Emma' in the row where SALARY is 500000.

**Syntax**

1. UPDATE table\_name
2. SET column\_name = value
3. WHERE condition;

**Query**

1. UPDATE EMPLOYEE
2. SET EMP\_NAME = 'Emma'
3. WHERE SALARY = 500000;

**Output:** After executing this query, the EMPLOYEE table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Emma | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

# SQL DELETE Statement

The SQL DELETE statement is used to delete rows from a table. Generally, DELETE statement removes one or more records form a table.

**Syntax**

1. DELETE FROM table\_name WHERE some\_condition;

### **Sample Table**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 4 | Kristen | Washington | 500000 | 29 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

## **Deleting Single Record**

Delete the row from the table EMPLOYEE where EMP\_NAME = 'Kristen'. This will delete only the fourth row.

**Query**

1. DELETE FROM EMPLOYEE
2. WHERE EMP\_NAME = 'Kristen';

**Output:** After executing this query, the EMPLOYEE table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 1 | Angelina | Chicago | 200000 | 30 |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

## **Deleting Multiple Record**

Delete the row from the EMPLOYEE table where AGE is 30. This will delete two rows(first and third row).

**Query**

1. DELETE FROM EMPLOYEE WHERE AGE= 30;

**Output:** After executing this query, the EMPLOYEE table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |
| 2 | Robert | Austin | 300000 | 26 |
| 3 | Christian | Denver | 100000 | 42 |
| 5 | Russell | Los angels | 200000 | 36 |
| 6 | Marry | Canada | 600000 | 48 |

## **Delete all of the records**

Delete all the row from the EMPLOYEE table. After this, no records left to display. The EMPLOYEE table will become empty.

**Syntax**

1. DELETE \* FROM table\_name;
2. or
3. DELETE FROM table\_name;

**Query**

1. DELETE FROM EMPLOYEE;

**Output:** After executing this query, the EMPLOYEE table will look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **CITY** | **SALARY** | **AGE** |

salesman\_id | name | city | commission

-------------+------------+----------+------------

5001 | James Hoog | New York | 0.15

5002 | Nail Knite | Paris | 0.13

5005 | Pit Alex | London | 0.11

5006 | Mc Lyon | Paris | 0.14

5007 | Paul Adam | Rome | 0.13

5003 | Lauson Hen | San Jose | 0.12

customer\_id | cust\_name | city | grade | salesman\_id

-------------+----------------+------------+-------+-------------

3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5005

salesman\_id | name | city | commission

-------------+------------+----------+------------

5001 | James Hoog | New York | 0.15

5002 | Nail Knite | Paris | 0.13

5005 | Pit Alex | London | 0.11

5006 | Mc Lyon | Paris | 0.14

5007 | Paul Adam | Rome | 0.13

5003 | Lauson Hen | San Jose | 0.12

customer\_id | cust\_name | city | grade | salesman\_id

-------------+----------------+------------+-------+-------------

3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5005

From the following tables write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.

ord\_no purch\_amt ord\_date customer\_id salesman\_id

---------- ---------- ---------- ----------- -----------

70001 150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007

70013 3045.6 2012-04-25 3002 5001

customer\_id | cust\_name | city | grade | salesman\_id

-------------+----------------+------------+-------+-------------

3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5005

From the following tables write a SQL query to find those orders where order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.

From the following tables write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission.

From the following tables write a SQL query to display the cust\_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending on customer\_id.

From the following tables write a SQL query to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission.

Write a SQL query to combine each row of salesman table with each row of customer table.