**CRUD operations explained: Create, read, update, and delete**

* CRUD stands for CREATE, READ, UPDATE and DELETE — the four essential functions of any persistent storage system, like a database.
* CRUD operations can be used to manipulate both SQL and NoSQL databases
* Structured Query Language (SQL)

## What are CRUD operations?

Any organization that tracks data (such as user accounts, payment information, or other records) needs systems that provide persistent storage, which is usually organized as a database.

A relational database consists of data organized in tables where each table comprises rows and columns. A table can be connected to other tables using primary and [foreign keys](https://www.educative.io/blog/what-is-foreign-key-database).

CRUD refers to the four operations we use to implement persistent storage applications like relational databases. Examples of relational databases include Oracle, Microsoft SQL Server, and [MySQL](https://www.educative.io/blog/mysql-tutorial" \t "_blank).

Operations such as security control, transaction control, access, and permission, and performance optimization are all based on CRUD.

| **Letter** | **Operation** | **Function** |
| --- | --- | --- |
| C | Create | Create, Insert |
| R | Read | Select |
| U | Update | Edit |
| D | Delete | Delete |

## Why are CRUD operations so important?

You can’t interact with databases without using CRUD. CRUD is essential for anything requiring persistent storage. Without CRUD, web developers wouldn’t be able to use REST (Representational State Transfer), which is a superset of CRUD used to access HTTP resources. REST APIs are some of the most commonly used in web development.

Basically, software developers wouldn’t be able to complete any project requiring persistent storage without CRUD operations.

On the other end, CRUD is just as crucial for end-users as it is for application developers. The end-users access CRUD operations through an application’s interface. Without CRUD, actions like registering for websites, creating blogs, or using bookmarks would be impossible.

**Benefits of CRUD**

* Facilitates security control by satisfying the various access requirements
* Simplifies application design and makes it more scalable
* Better performance compared to ad-hoc SQL statements

## CRUD with SQL

SQL is a standard language used to store, manipulate, and retrieve data from databases. Let’s go over a few different ways you can use CRUD SQL queries to create a database and table, insert data into tables, and perform basic operations on data within tables.

### CREATE operation

The CREATE operation lets you create a new database and its tables.

In SQL, a new database is created using the CREATE DATABASE keyword.

CREATE DATABASE db\_name  
Similarly, you can create a new table using the `CREATE TABLE` keyword.  
  
CREATE TABLE table\_name (col\_1, col\_2, ...)

You can add new records using the command INSERT INTO. This command starts with the INSERT INTO keyword, followed by the table name, column names, and the values to be inserted.

When using INSERT INTO, you have two options:

INSERT INTO table\_name  
 VALUES (value1, value2, value3, ...);

INSERT INTO table\_name (column1, column2, column3, ...)  
 VALUES (value1, value2, value3, ...);

In the example below, you can create a database bakery and table menu. Afterward, you can insert data into your menu table.

CREATE DATABASE bakery;  
  
CREATE TABLE menu (id Int, item varchar(10), date DATE);  
  
INSERT INTO menu  
VALUES (1, 'croissant', '2020-12-16');

If you want to insert **multiple rows**, you can do that using the following syntax:

INSERT INTO menu  
VALUES   
(2, 'bread', '2020-12-16' ),  
(3, 'eclairs', '2020-12-16' );

This will insert two new rows to the menu table, and give each entry a unique id.

### READ operation

The READ function is similar to a **search function**, allowing you to retrieve specific records and read their values. In SQL, the read function uses the SELECT keyword.

For example, let’s look at menu items sold in a bakery.

To view the complete list of baked goods, you need to display all the data in your menu table using:

SELECT \* FROM menu;

This will not change the menu table. Instead, it will display all of the records within that table.

Here’s an example of how the SELECT keyword can be used to retrieve more specific data:

 SELECT id, item, date  
    FROM   menu    
    WHERE  id = '3';

In this SQL query, a menu item is selected using the unique ID number assigned to it. This query will display the complete row from the menu table, where the id is 3.

### UPDATE operation

The UPDATE operation is used to modify existing records of a table. When using the UPDATE keyword, you **must** define the target table and columns to be updated. You may also need to know associated values or specific rows, depending on how specific you want your query to be.

**Note**: To avoid concurrency issues, limit the number of rows that are updated in each query.

Here is the basic syntax for updating a record:

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

Let’s say you want to update the item and date of a specific item in the menu table. You can use:

UPDATE menu  
 SET item = 'chocolate croissant', date = 2022-12-16  
 WHERE id = 2;

This will update the table so that the previous record with an ID number of 2 is assigned a new item name, chocolate croissant, and a new date, 2022-12-16.

### DELETE operation

The SQL DELETE operation can remove one or more records from the table of a database. Some relational database applications may permit a hard delete (permanent delete) or soft delete (update row status).

The syntax for the DELETE operation is as follows:

DELETE FROM table\_name WHERE condition;

If you want to remove only one item from the table, you can specify which row to delete. You can also use a combination of conditions if multiple rows need deletion.

DELETE FROM menu WHERE item\_name='bread';

This will remove the row with the item bread from the table. If you want to delete all the records from the table, you can use:

DELETE FROM menu;

Application on employee database

**Q1: create a test table named Employee**

CREATE TABLE Employee (EMPId Int Identity NOT NULL,

EmpNo varchar (10), SSN varchar (10), DOB DATE,

CreatedDt Datetime, CreatedBy varchar(10));

**O/P: SQL query successfully executed. However, the result set is empty.**

**Q2: INSERT some values into the Employee table created above with various possible scenarios as listed below. Default scenarios specifying values in the order columns are present excluding the Identity Datatype column EmpId.**

INSERT INTO Employee (EmpId, SSN, DOB, CreatedDt, CreatedBy)

VALUES (1,'1234567890','2000-01-01', '01-02-2023', 'system');

O/p: SQL query successfully executed. However, the result set is empty.

Q3: