**What are CRUD operations?**

Any organization that tracks data *(like accounts, payment information, or other records)* needs systems that provide persistent storage, which is usually organized into a database. A relational database consists of data organized by rows and columns. They can be connected to other tables using primary and [foreign keys](https://www.educative.io/blog/what-is-foreign-key-database).

CRUD (create, read, update, delete) is an acronym that refers to the **four functions** we use to implement persistent storage applications and relational database applications, including the Oracle Database, Microsoft SQL Server, and [MySQL](https://www.educative.io/blog/mysql-tutorial" \t "_blank).

The table below summarizes what each CRUD operation means.

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

For SQL, CRUD maps to insert, select, update, and delete, respectively. Operations such as security control, transaction control, access and permission, and performance optimization are all based on CRUD.

### Why are CRUD operations so important?

CRUD is constantly used for anything related to databases and [database design](https://www.educative.io/blog/database-design-tutorial). Software developers can’t get anything done without CRUD operations. Website development, for example, uses REST (Representational State Transfer), which is a superset of CRUD used for HTTP resources. REST APIs are some of the most commonly used APIs in web development.

On the other end, CRUD is just as crucial for **end-users**. Without it, things like registering for websites, creating blogs, or bookmarks would be impossible. Most applications we use let us add or create new entries, search for existing ones, make changes to them or delete them.

**CRUD offers many benefits, including:**

* It facilitates security control by satisfying the various access requirements
* It simplifies and facilitates application design making it more scalable
* It performs better compared to ad-hoc SQL statements

## How to CREATE Operation:-

The CREATE operation lets you add new rows to a table. You can add new records using the command, INSERT INTO. The 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, we will add data to our bakery table.

INSERT INTO menu  
VALUES (1, 'croissant', 1, '2020-12-16');

If we want to add multiple rows, we do that using:

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

This will add new rows to the menu table, and each entry will have a unique id.

### Keep the learning going.

Learn SQL without scrubbing through videos or documentation. Educative’s text-based courses are easy to skim and feature live coding environments, making learning quick and efficient.

**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 take a look at the items we sell at our bakery. To do that, we have to display all the data in our menu table using:

SELECT \* FROM menu;

This will not make any changes to the menu table but will display all the records in that table.

Take a look at this example to see how SELECT retrieves desired data:

IF OBJECT\_ID('cusp\_CustomerRead') IS NOT NULL  
BEGIN   
    DROP PROC cusp\_CustomerRead  
END   
GO  
CREATE PROC cusp\_CustomerRead  
    @CustomerID int  
AS   
BEGIN   
   
    SELECT CustomerID, FirstName, LastName, Email, PhoneNumber  
    FROM   Customer    
    WHERE  (CustomerID = @CustomerID)   
END  
GO

**UPDATE operation**

The UPDATE operation is how we change an existing record in the table. We can use this to modify existing records that exist in the database. When performing UPDATE, you must define the target table and columns to be updated. You also need the associated values and sometimes the rows.

It is recommended to limit the number of rows, as this helps to avoid concurrency issues.

To update an existing record, use the following:

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

Let’s say we want to update the item name and price. We would use:

UPDATE menu  
 SET item\_name = 'chocolate croissant', price = 2.5  
 WHERE item\_id = 1;

This will update the table so that the previous record with id 1 will be replaced with chocolate croissant with an attribute of price 2.5.

## DELETE operation

DELETE **removes a record** from the table. SQL has a built-in delete function for deleting one or more records from the database. Some relational database applications may permit a hard delete (permanent delete) or soft delete (update row status).

The delete command is as follows:

DELETE FROM table\_name WHERE condition;

If You want to remove one item from the table, we use:

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;

**Here above all item explains How to Do :-**

1. **How to Insert Operations**
2. **How to Delete Operations**
3. **How to Update Operations**
4. **How to READ Operations**
5. **How to CREATE Operations**

**Here Above all operations are CRUD operations**