

CRUD OPERATIONS

TechRacine Academy

https://techracine.com

https://academy.techracine.com

Table of Contents

1. Introduction to CRUD Operations 3
2. Why CRUD Operation Matters 3
3. CRUD and Database Tables 3
4. Create Operation 4
5. Read Operation 4
6. Update Operation 4
7. Delete Operation 5
8. CRUD Using SQL 5
9. CRUD Using MySQL 5
10. Using WHERE Conditions Safely 6
11. Primary Keys and CRUD 6
12. CRUD and Foreign Keys 6
13. CRUD in Application Development 7
14. Common CRUD Mistakes 7
15. CRUD Best Practices 7
16. Real World Usage 8
17. Summary 8
18. Exercises 8

1. Introduction to CRUD Operations

CRUD stands for Create, Read, Update, and Delete.

These are the four basic operations used to manage data in a database.

CRUD Operations allow applications to add new records, view existing records, modify stored information, and remove unwanted records.

CRUD is a fundamental concept in database development and is commonly implemented using SQL statements.

Almost every data-driven application uses CRUD operations in some form.

2. Why CRUD Operation Matters

CRUD provides a simple structure for managing database records.

It connects application features such as registration, search, editing, and deletion with database actions.

Understanding CRUD helps developers build practical database applications.

CRUD concepts are used in student systems, banking systems, e-commerce applications, hospital systems, and many other software products.

3. CRUD and Database Tables

A database table stores related data in rows and columns.

CRUD Operations work on these rows.

For example, A students table may contain student_id, name, department, and mark columns.

Each CRUD Operation changes or retrieves records from the table.

Example: Student Table

student_id	name	department	mark
1	Arun	CSE	85
2	Meena	IT	78
3	Kavin	ECE	91

4. Create Operation

Create means adding a new record to a database table.

In SQL, the INSERT statement is used for this operation.

Example: INSERT INTO students (student_id, name, department, mark)

VALUES (4, 'Priya', 'CSE', 88);

Key Points:

- INSERT is used to add data.
- Column names identify where the values should be stored.
- The VALUES clause provides the new record.
- A primary key should uniquely identify the record.

5. Read Operation

Read means retrieving or viewing data stored in a database.

In SQL, the SELECT statement is used.

Example: SELECT * FROM students;

- To retrieve only selected columns:

  Example: SELECT name, mark FROM students;

- To retrieve records that satisfy a condition:

  Example: SELECT name, mark FROM students WHERE mark >= 80 ORDER BY mark DESC;

6. Update Operation

Update means changing existing data in a database.

The UPDATE statement is used.

Example: UPDATE students SET mark = 90 WHERE student_id=4;

The WHERE condition identifies the record that must be changed.

Without a suitable WHERE condition, an UPDATE statement may change multiple records.

7. Delete Operation

Delete means removing existing records from a database.

The DELETE statement is used.

Example: DELETE FROM students WHERE student_id=4;

The WHERE condition limits the deletion to the intended record.

Important: DELETE permanently removes the selected rows unless the database operation is rolled back or the data is otherwise recoverable.

8. CRUD Using SQL

SQL provides the commands required to perform the four CRUD Operations.

CRUD Operation	SQL Command	Purpose
Create	INSERT	Add a new record
Read	SELECT	Retrieve records
Update	UPDATE	Modify existing records
Delete	DELETE	Remove records

9. CRUD Using MySQL

MySQL is a relational database management system that supports SQL-based CRUD operations.

Example database setup:

```
CREATE DATABASE college_db;

USE college_db;

CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    department VARCHAR(30),
    mark INT
);
```

After creating the table, INSERT, SELECT, UPDATE, and DELETE can be used to manage its records.

10. Using WHERE Conditions Safely

The WHERE clause is important when updating or deleting specific records.

Example: UPDATE students SET department = 'CSE' WHERE student_id=2;

DELETE FROM students WHERE student_id=3;

Always check the condition before executing UPDATE or DELETE.

A missing or incorrect WHERE clause can affect unintended rows.

11. Primary Keys and CRUD

A primary key uniquely identifies each record in a table.

It is especially useful when performing UPDATE and DELETE Operations.

Example: UPDATE students SET mark = 95 WHERE student_id = 1;

Here, student_id is used to identify exactly which student record should be updated.

12. CRUD and Foreign Keys

A foreign key connects a record in one table with a related record in another table.

Example:
```
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    department_id INT,
    FOREIGN KEY (department_id)
    REFERENCES departments(department_id)
);
```

Foreign-key relationships help maintain consistency between related tables.

Depending on the database design and constraints, deleting or updating a parent record may be restricted or may affect related records.

13. CRUD in Application Development

In a real application, CRUD operations are usually connected to a user interface and a backend.

- Create – a form allows the user to enter new information.
- Read – a page or dashboard displays stored information.
- Update – an edit form changes an existing record.
- Delete – a delete action removes a selected record.

A typical flow is: User Interface → Application/Backend → SQL Query → Database → Result → User Interface.

14. Common CRUD Mistakes

- Using UPDATE without an appropriate WHERE condition.
- Using DELETE without checking which rows will be affected.
- Inserting duplicate values into columns that should be unique.
- Using incorrect data types for inserted or updated values.
- Not validating user input before sending it to the database.
- Not considering relationships between parent and child tables.

15. CRUD Best Practices

- Use meaningful table and column names.
- Define primary keys for important tables.
- Use foreign keys when relationships are required.
- Validate input before database operations.
- Use parameterized queries or prepared statements in applications.
- Check the result of every CRUD operation.
- Back up important data and use transactions where appropriate.
- Test CRUD operations with valid and invalid inputs.

16. Real World Usage

CRUD operations are used in many real-world systems.

- Student Management – add, view, edit, and remove student records.
- E-commerce – manage products, customers, and orders.
- Banking – manage accounts and transaction-related records.
- Hospital Systems – manage patient and appointment information.
- Employee Management – manage employee details and departments.
- Library Systems – manage books, members, and borrowing records.

17. Summary

CRUD is the foundation of database record management.

Create uses INSERT, Read uses SELECT, Update uses UPDATE, and Delete uses DELETE.

A strong understanding of CRUD helps developers connect database tables with practical application features.

Primary keys, Foreign keys, Validation, and Safe Conditions are important for reliable CRUD implementation.

18. Exercises

1. Create a students table with student_id, name, department, and mark.

2. Insert five student records into the table.

3. Display all student records.

4. Display only the name and department columns.

5. Find students whose marks are greater than or equal to 80.

6. Update the mark of one student using the primary key.

7. Delete one student using the primary key.

8. Create a departments table and connect it to students using a foreign key.

9. Write one example each for INSERT, SELECT, UPDATE, and DELETE.

10. Build a small student-management application that performs all four CRUD operations.