SQL Command Reference Guide

1. ALTER TABLE

The ALTER TABLE statement is used to modify the structure of an existing table.

Commands with Examples:

- **Add a Column:**
- ALTER TABLE employees ADD age INT;

ALTER TABLE employees ADD age INT;

Adds a new column age of type INT to the employees table.

- **Modify a Column:**
- ALTER TABLE employees MODIFY age VARCHAR(3);

Changes the data type of the age column to VARCHAR(3).

- **Rename a Column:**
- ALTER TABLE employees RENAME COLUMN age TO years_old;

Renames the age column to years_old.

- **Drop a Column:**
- ALTER TABLE employees DROP COLUMN age;

Removes the age column from the employees table.

- **Add a Constraint:**
- ALTER TABLE employees ADD CONSTRAINT unique_email UNIQUE (email);

Adds a unique constraint to the email column.

- **Drop a Constraint:**
- ALTER TABLE employees DROP CONSTRAINT unique_email;

Removes the unique_email constraint.

- **Rename a Table:**
- ALTER TABLE employees RENAME TO staff;

Renames the employees table to staff.

- **Set or Drop Default Value:**
- ALTER TABLE employees MODIFY age INT DEFAULT 18;
- ALTER TABLE employees MODIFY age DROP DEFAULT;

Sets a default value of 18 for the age column and then removes it.

---

2. UPDATE

The UPDATE statement is used to modify existing data in a table.

Commands with Examples:

- **Update Specific Rows:**
- UPDATE employees SET salary = 50000 WHERE id = 1;

Updates the salary of the employee with id 1 to 50000.

- **Update Multiple Columns:**
- UPDATE employees SET salary = 60000, department = 'HR' WHERE id = 2;

Updates the salary and department for the employee with id 2.

- **Update All Rows:**
- UPDATE employees SET status = 'active';

Sets the status column to active for all employees.

---

3. DELETE

The DELETE statement is used to remove rows from a table.

Commands with Examples:

- **Delete Specific Rows:**
- DELETE FROM employees WHERE id = 3;

Deletes the employee with id 3.

- **Delete All Rows:**
- DELETE FROM employees;

Removes all rows from the employees table.

---

4. DROP

The DROP statement is used to delete entire database objects (tables, views, etc.).

Commands with Examples:

- **Drop a Table:**
- DROP TABLE employees;

Deletes the employees table.

- **Drop a Database:**
- DROP DATABASE company;

Deletes the company database.

- **Drop a Column:**
- ALTER TABLE employees DROP COLUMN department;

Removes the department column from the employees table.

- **Drop a Constraint:**
- ALTER TABLE employees DROP CONSTRAINT unique_email;

Removes the unique_email constraint.

---

5. SELECT

The SELECT statement is used to retrieve data from a table.

Commands with Examples:

- **Select All Columns:**
- SELECT * FROM employees;

Retrieves all columns from the employees table.

- **Select Specific Columns:**
- SELECT name, salary FROM employees;

Retrieves only the name and salary columns.

- **With Conditions (WHERE********):**
- SELECT name FROM employees WHERE salary > 50000;

Retrieves names of employees with a salary greater than 50000.

- **Order Results:**
- SELECT name FROM employees ORDER BY salary DESC;

Retrieves employee names ordered by salary in descending order.

- **Limit Results:**
- SELECT name FROM employees LIMIT 5;

Retrieves the first 5 employee names.

---

6. DESC

The DESC statement is used to describe the structure of a table.

Command with Example:

DESC employees;

Displays the column details of the employees table, including names, data types, and constraints.

---

7. WHERE

The WHERE clause is used to filter rows based on conditions.

Commands with Examples:

- **Basic Condition:**
- SELECT * FROM employees WHERE department = 'IT';

Retrieves all employees in the IT department.

- **Multiple Conditions:**
- SELECT * FROM employees WHERE salary > 40000 AND department = 'HR';

Retrieves employees with a salary above 40000 in the HR department.

- **Using Operators:**
- SELECT * FROM employees WHERE age BETWEEN 25 AND 35;

Retrieves employees aged between 25 and 35.

---

8. Operators

Comparison Operators:

- = : Equal to
- != or <> : Not equal to
- > : Greater than
- < : Less than
- >= : Greater than or equal to
- <= : Less than or equal to

Logical Operators:

- AND : All conditions must be true
- OR : At least one condition must be true
- NOT : Negates a condition

Other Operators:

- LIKE : Pattern matching
- SELECT * FROM employees WHERE name LIKE 'A%';

Retrieves employees whose names start with A.

- IN : Match any value in a list
- SELECT * FROM employees WHERE department IN ('IT', 'HR');

Retrieves employees in the IT or HR departments.

- BETWEEN : Range matching
- SELECT * FROM employees WHERE age BETWEEN 20 AND 30;

Retrieves employees aged between 20 and 30.

---

9. Data Types

Numeric Data Types:

- INT or INTEGER
- FLOAT
- DOUBLE
- DECIMAL(precision, scale)

String Data Types:

- CHAR(n)
- VARCHAR(n)
- TEXT

Date and Time Data Types:

- DATE
- DATETIME
- TIMESTAMP
- TIME

Boolean Data Type:

- BOOLEAN (or BIT in some databases)

---

10. Constraints

Common Constraints with Examples:

- **PRIMARY KEY:**
- ALTER TABLE employees ADD CONSTRAINT pk_id PRIMARY KEY (id);

Sets the id column as the primary key.

- **FOREIGN KEY:**

- ALTER TABLE orders ADD CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES customers(id);

Links the customer_id column in orders to the id column in customers.

- **UNIQUE:**
- ALTER TABLE employees ADD CONSTRAINT unique_email UNIQUE (email);

Ensures all values in the email column are unique.

- **NOT NULL:**
- ALTER TABLE employees MODIFY email VARCHAR(255) NOT NULL;

Ensures the email column cannot have NULL values.

- **CHECK:**
- ALTER TABLE employees ADD CONSTRAINT check_salary CHECK (salary > 0);

Ensures the salary column contains values greater than 0.

This guide now includes examples for each SQL command and feature. Let me know if further clarification is needed!