Database and SQL Commands

Mastering Data Management and Querying for Modern Systems

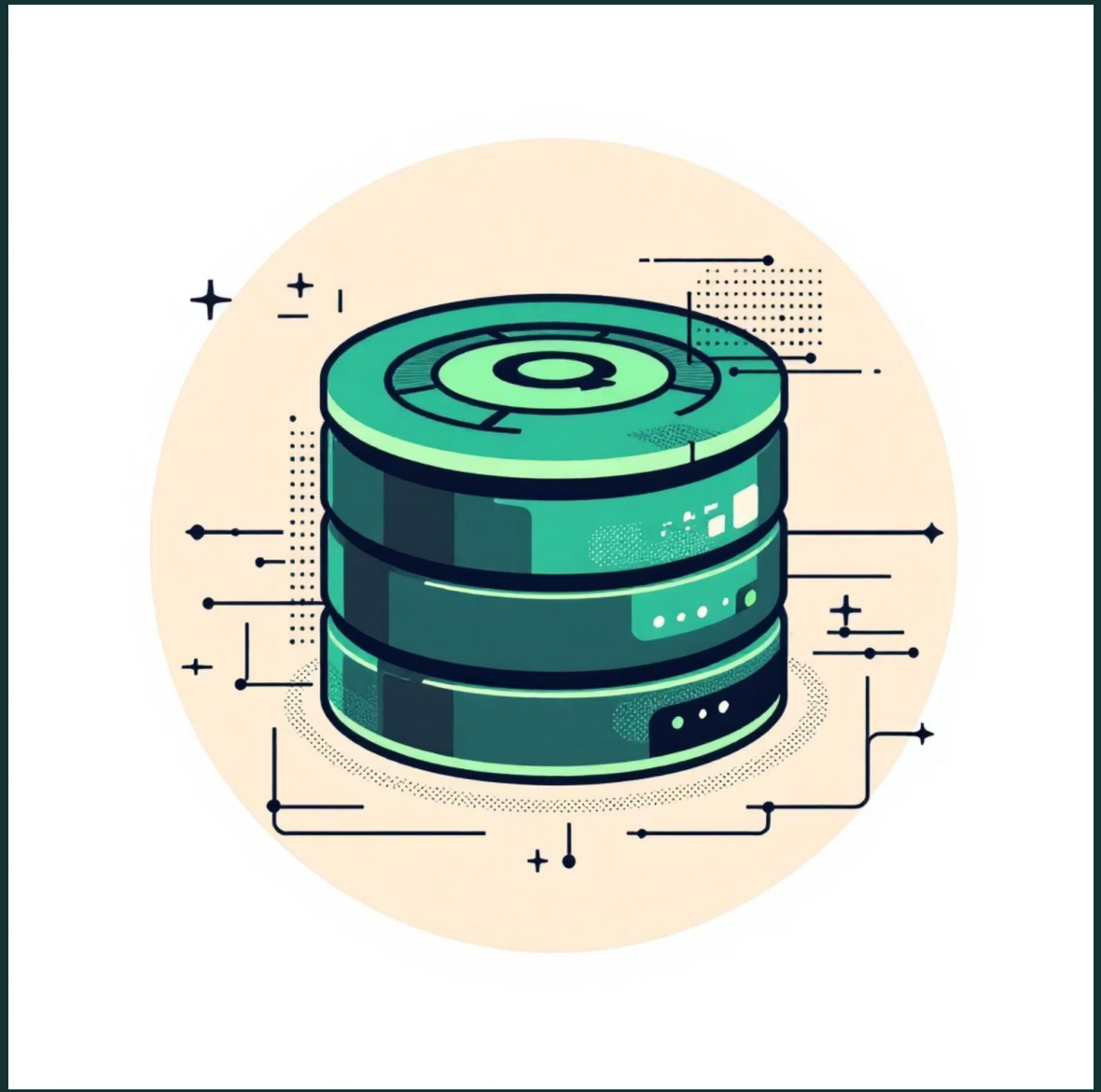
What is a Database?

A database is an **organized collection of data** that can be easily accessed, managed, and updated. Think of it as a digital filing cabinet, structured to store information efficiently and reliably.

Example: Students Table

Data is typically stored in tables, with rows representing individual records and columns representing specific attributes.

ID	Name	Age	Grade
1	John	20	A
2	Priya	19	B
3	Ahmed	21	A
4	Neha	18	C



DDL: Data Definition Language (DDL)

Data Definition Language (DDL) commands are used to define, modify, and manage the structure of database objects, such as tables. They essentially build the schema or blueprint of your database.

1

CREATE

Used to create new database objects, like tables, views, or indexes.

```
CREATE TABLE Students(ID INT, Name VARCHAR(50), Age INT, Grade CHAR(1));
```

2

ALTER

Used to modify the structure of an existing database object, such as adding a new column to a table.

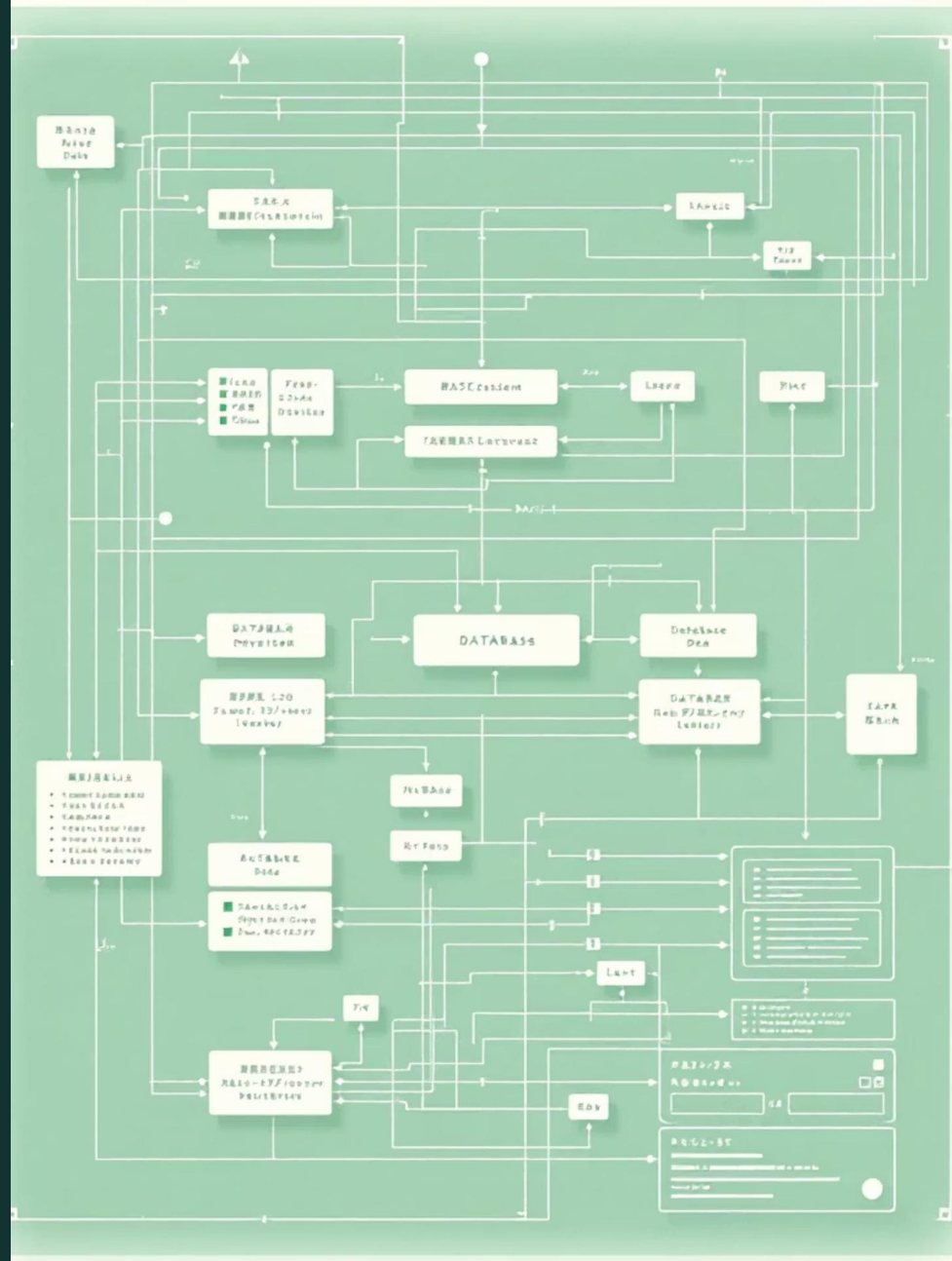
```
ALTER TABLE Students ADD Address VARCHAR(50);
```

3

DROP

Used to delete an entire database object from the database, permanently removing its structure and data.

```
DROP TABLE Students;
```



DML: Data Manipulation Language

Data Manipulation Language (DML) commands are used for managing data within database objects. These commands allow you to insert, update, and delete records, interacting directly with the information stored.

INSERT

Adds new rows or records into a table.

```
INSERT INTO Students
VALUES(5,'Sara',20,'B');
```

UPDATE

Modifies existing data within one or more rows of a table.

```
UPDATE Students SET
Grade='A+' WHERE ID=2;
```

DELETE

Removes one or more rows from a table based on specified conditions.

```
DELETE FROM Students WHERE ID=4;
```

These DML commands are essential for keeping your database content current and relevant.

DQL: Data Query Language (DQL),

Data Query Language (DQL), primarily represented by the **SELECT** statement, is used to retrieve data from a database. This is where you extract the insights from your stored information.

- Select All Records

```
SELECT * FROM Students;
```

Retrieves every column and row from the table.

- Select Specific Columns

```
SELECT Name, Grade FROM Students;
```

Only fetches the specified columns.

- Select with Condition (WHERE)

```
SELECT * FROM Students WHERE Age > 18;
```

Filters rows based on a specific condition.

- Select with ORDER BY

```
SELECT * FROM Students ORDER BY Name ASC;
```

Sorts the result set by one or more columns.

- Aggregate Functions (COUNT, AVG, MAX)

```
SELECT COUNT(*) AS TotalStudents FROM Students;
```

Performs calculations on a set of rows.

- Select with GROUP BY

```
SELECT Grade, COUNT(*) FROM Students GROUP BY Grade;
```

Groups rows that have the same values in specified columns.

- Select with LIKE (Pattern Matching)

```
SELECT * FROM Students WHERE Name LIKE 'A%';
```

Searches for patterns in text columns.

Connecting Data: Understanding SQL JOINs

When data is spread across multiple related tables, **JOIN** clauses are used to combine rows from two or more tables based on a related column between them. This is crucial for retrieving comprehensive information.

Tables for JOIN Examples:

Students Table:

ID	Name
1	John
2	Priya
3	Ahmed
4	Neha

Courses Table:

CourseID	CourseName	StudentID
101	Math	1
102	Science	2
103	English	1
104	History	3

INNER JOIN

Returns rows when there is a match in both tables. Only students with courses, and courses with students.

```
SELECT S.Name, C.CourseName FROM Students S INNER JOIN Courses C ON S.ID = C.StudentID;
```

LEFT JOIN (LEFT OUTER JOIN)

Returns all rows from the left table, and matching rows from the right table. If no match, NULL for right columns.

```
SELECT S.Name, C.CourseName FROM Students S LEFT JOIN Courses C ON S.ID = C.StudentID;
```

RIGHT JOIN (RIGHT OUTER JOIN)

Returns all rows from the right table, and matching rows from the left table. If no match, NULL for left columns.

```
SELECT S.Name, C.CourseName FROM Students S RIGHT JOIN Courses C ON S.ID = C.StudentID;
```

SQL Commands: A Comprehensive Overview

SQL commands are categorized based on their function, allowing for structured interaction with databases.

DDL: Data Definition Language

Commands like **CREATE, ALTER, DROP** are used to define and modify the database structure.

```
CREATE TABLE Students(...);
```

DML: Data Manipulation Language

Commands such as **INSERT, UPDATE, DELETE** are used to manage the data within tables.

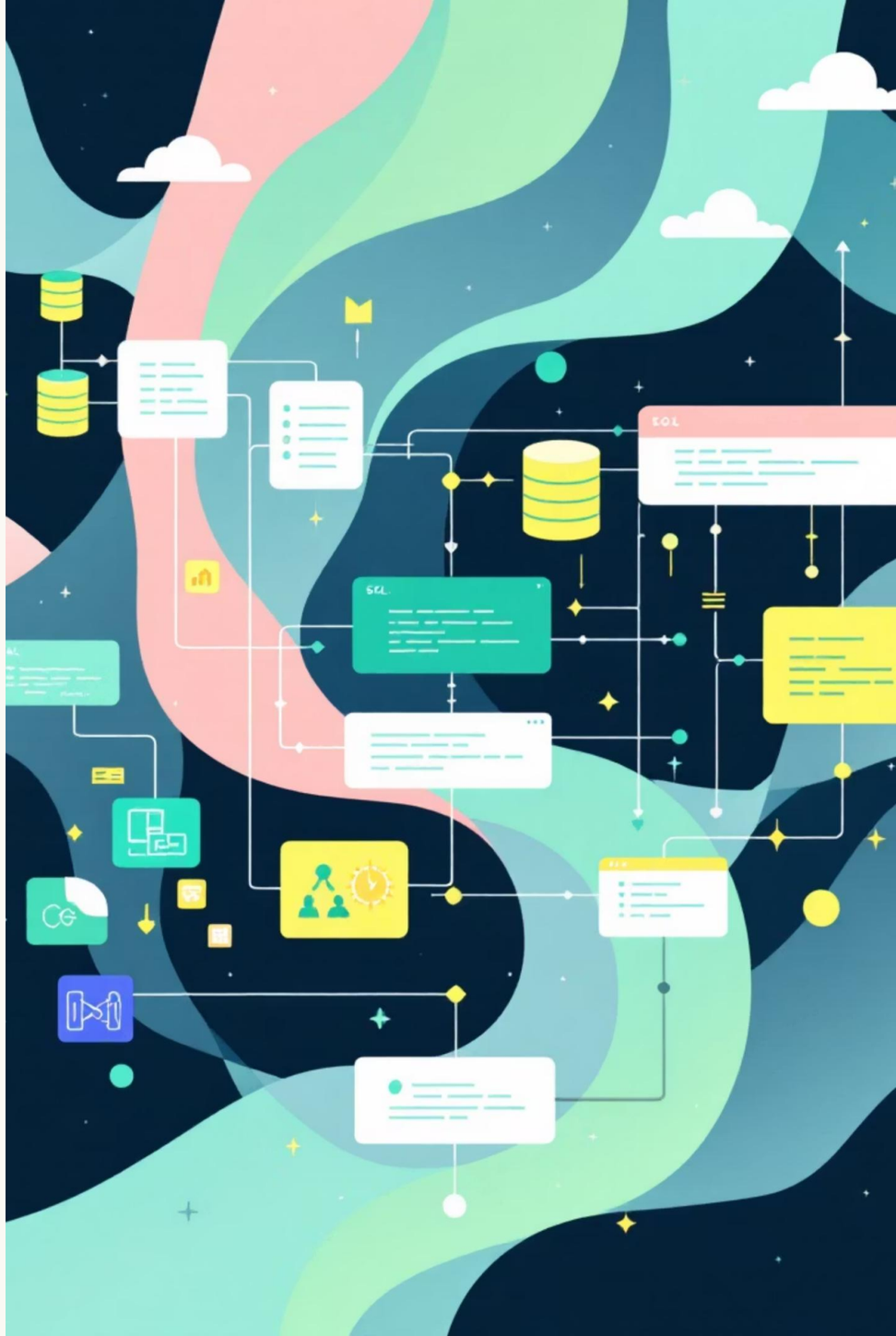
```
INSERT INTO Students VALUES(...);
```

DQL: Data Query Language

Primarily the **SELECT** statement, used for retrieving specific data from tables.

```
SELECT * FROM Students WHERE Age > 18;
```

Understanding these categories is key to effective database management and powerful data querying.



Thank You!

We appreciate your attention and hope this overview helps you embark on your database journey.

Any Questions?