**Class diagram**

A **Class Diagram** shows the structure of a system by displaying **classes**, their **attributes**, **methods**, and the **relationships** between them.  
It helps in designing and understanding object-oriented systems.

**Relationships can be:**

* **Association** → general connection between classes
* **Inheritance** → one class inherits features from another
* **Aggregation/Composition** → a class is made of other classes
* **Dependency** → one class depends on another

**Example:**

* Book class: title, author → borrow(), return()
* Member class: name, ID → register(), payFees
* A Member borrows a Book → Association

**Cluster Index**

**🡺 Cluster Indexing** is a database indexing technique used to speed up data retrieval

* In a **Clustered Index**, the table’s rows are **physically stored** on disk in the same order as the index
* There’s **only one clustered index** per table because the data rows can only be sorted one way
* It makes searching for ranges of values faster
* Primary keys are often implemented as clustered indexes
* Clustered indexes are useful when:

1. You often search for **ranges of values** (e.g. dates, IDs).
2. You run queries that sort or order data by a specific column.

**Ex:**  
If you create a clustered index on a column like EmployeeID, the table’s rows are stored sorted by EmployeeID. So, searching for employees between ID 100 and 200 is fast

select \*

from Employees

where Employee\_ID between 100 and 200

-> is very fast because the database can directly read the matching block of rows

-> Without a clustered index, the database might have to scan the entire table to find those rowsA table of information on a black background

AI-generated content may be incorrect.

**Built in stored procedures**

**Built-in Stored Procedures** are **predefined procedures** provided by a database system (like SQL Server, MySQL) to help perform common tasks.

* They are already created for you by the database engine.
* You can run them to manage the database, get information, or perform administrative tasks.
* They usually start with special characters, like sp\_ in SQL Server or start with mysql. in MySQL.

**Ex in SQL Server:**

* sp\_help → shows info about tables, views, procedures.
* sp\_who → shows current users and sessions.
* sp\_rename → renames a table, column, or other object

exec sp\_help 'Employees'

**SQL Injection**

**SQL injection** (SQLi) :

🡪 is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

🡪This can allow an attacker to view data that they are not normally able to retrieve.

🡪 This might include data that belongs to other users, or any other data that the application can access.

🡪 In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.

**How to detect SQL injection vulnerabilities:**

You can detect SQL injection manually using a systematic set of tests against every entry point in the application. submit:

* The single quote character ' and look for errors or other anomalies.
* Some SQL-specific syntax that evaluates to the base (original) value of the entry point, and to a different value, and look for systematic differences in the application responses.
* Boolean conditions such as OR 1=1 and OR 1=2, and look for differences in the application's responses.
* Payloads designed to trigger time delays when executed within a SQL query, and look for differences in the time taken to respond.
* OAST payloads designed to trigger an out-of-band network interaction when executed within a SQL query, and monitor any resulting interactions.

**Some common SQL injection examples include:**

* [Retrieving hidden data](https://portswigger.net/web-security/sql-injection#retrieving-hidden-data), where you can modify a SQL query to return additional results.
* [Subverting application logic](https://portswigger.net/web-security/sql-injection#subverting-application-logic), where you can change a query to interfere with the application's logic.
* [UNION attacks](https://portswigger.net/web-security/sql-injection/union-attacks), where you can retrieve data from different database tables.
* [Blind SQL injection](https://portswigger.net/web-security/sql-injection/blind), where the results of a query you control are not returned in the application's responses.