* **هل العلاقة ال Ternary كــ اداء افضل حاجة ؟ واي هو بديلها**

**Not always.  
it is better to break the ternary relationship into multiple binary relationships.**

* **Cross Join Uses:**

**- Generating all possible combinations.  
- Creating test data.**

* **Insert With Join**

**Very good to insert data from one or more existing tables into another table directly with one query.**

* **ازاي بنعمل العلاقة M : M في Self Join**

**Create a junction table to represent the relationship between entities within the same table. This allows you to query and retrieve related records from the same table based on the connection defined in the junction table.**

* **what is primary and secondary files in DB?**
* The Primary Data File is the data file that is the initial data file holding data in SQL Server and points to other files within your database.  Primary Data Files have a .mdf extension on them.
* Secondary Data Files are data files that hold all data that do not belong in the Primary Data File.  Data that doesn’t go into the Primary Data File is told to do so and set in a setting in SQL Server.  Secondary Files are optional.  These files should have a .ndf extension.
* **what is the extension of .mdf , .ndf and .ldf files and what they contain?**

**.mdf (Primary Data File):**

**This file is the main container for the database's data and objects. Every SQL Server database has one primary data file.**

**.ndf (Secondary Data File):**

**These are optional files that can be added to a database to store additional data, potentially on different storage locations. This can improve performance or manage large datasets.**

**.ldf (Transaction Log File):**

**This file is crucial for database recovery. It records all transactions (insert, update, delete) so that the database can be restored to a consistent state in case of a failure.**

* **What is Normalization and what’s the difference between the output of mapping and Normalization?**

**Normalization is a design process in relational databases where we organize data to:**

* **Reduce redundancy**
* **Improve data integrity**
* **Optimize storage**

**We do this by breaking large tables into smaller related tables and defining relationships using foreign keys.**

**Mapping: Converting an ER model to tables**

**Normalization: Cleaning and refining those tables**

**ER Diagram → Mapping → Raw Tables → Normalization → Optimized Tables**

* **What is Denormalization and when in business we need this?**
* **Denormalization is the process of intentionally introducing some redundancy into a database design by combining tables or adding duplicate data to improve read performance.**

**It’s basically the opposite of normalization.**

* **While normalization makes data consistent and reduces redundancy, it also creates multiple related tables, which means:**
* **More JOINs in queries**
* **More complex query logic**
* **Potentially slower read performance when fetching large reports or dashboards**

**So, to make read operations faster and simpler, we sometimes denormalize.**

* **What the meaning of clustered in hard disk by primary key?**

**A primary key clustered index is a type of database index that defines the primary key for a table and organizes the data in the table based on that key. Here are the key points to understand:**

1. **Primary Key: A primary key is a unique identifier for each record in a table. It ensures that no two rows have the same value for the primary key column(s) and that the primary key cannot be null.**
2. **Clustered Index: A clustered index determines the physical order of data in a table. When a table has a clustered index, the rows are stored on disk in the same order as the index. This means that the data itself is organized according to the primary key.**
3. **Single Clustered Index: A table can have only one clustered index because the data rows can be sorted in only one order. Typically, the primary key is defined as the clustered index, but this is not mandatory; you can have a different column as the clustered index.**
4. **Performance Benefits: Using a primary key clustered index can improve the performance of queries that search for records based on the primary key. It can also speed up range queries since the data is stored sequentially.**
5. **Implications of Changes: When you change the value of a primary key that is part of a clustered index, the database may need to move the entire row to maintain the order, which can impact performance.**

**Example**

**If you have a table called Employees with a primary key on the EmployeeID column, and this primary key is defined as a clustered index, the rows in the Employees table will be physically stored in the order of EmployeeID. This makes it efficient to retrieve records using the EmployeeID but can incur overhead when inserting or updating records that affect the order.**

**In summary, a primary key clustered index is a way to uniquely identify records while also determining how those records are stored physically in the database.**