**Database Workshop**

**Day 1:**

**DB Concept**

**Db Design**

**ERD**

**File based system**

**DB Life cycle**

**File based system & Disadvantage and limitations**

**Basic Definitions:**

**Database: collection of related data or tables**

**Database Management system (DBMS): tool 🡪 sql server to create db 🡪 mdf (doc)**

**Database System: sw + db (interface)**

**The ER Model: Entities & Attributes & relationships**

**Entity Relationship Modeling (ERD)**

**ER Diagram: Starting Example (Rectangles: entity set, Diamonds: relationship set , Ellipses: attributes)**

**Strong & weak entity**

**Types of attributes: (Composite, Multi-valued, derived, complex, simple)**

**Degree of relationship (Unary, Binary, Ternary)**

**Mapping cardinalities (1:1, 1:m, m:m)**

**Participation constraint (Total, Partial)**

**Day 2:**

**Recap of what we discussed in day 1**

**Relational DB definition (Table or entity, attribute or column or filed, Row or Record or tuble, DB)**

**Mapping -> DB schema**

**Mapping rule**

**Case study**

**ANSI SQL**

**Create DB (SQL server)**

**Day3:**

**Joins**

**DB constrains (DB integrity)**

**Day4:**

**IF**

**While**

**Variable**

**Function**

**1.Introduction to Databases**

* Briefly introduce the agenda for the day.
* Engage the trainees with a question:
  + **"Can you think of examples where databases are used in everyday life?"**

**Example:** Facebook, Instagram, Twitter.

**Example:** Amazon …etc

**Example:** ATM systems, online banking.

**Theme:** Understanding Databases, Their Design, and the Shift from File Systems

* **What is a Database?**
  + A systematic way to organize and store data for easy retrieval and management.
  + Example: A phonebook vs. a contact list on your smartphone.
* **Why Use Databases?**
  + Efficient handling of large data sets.
  + Reduced data redundancy.
  + Easy sharing and multi-user access.

**2. Understanding the Benefit of Good Database Design**

* **Why Good Design Matters:**
  + Ensures reliability and scalability.
  + Reduces redundancy and inconsistency.
  + Makes maintenance and updates easier.
* **Common Mistake:**
  + Jumping straight into SQL Server or database creation without a proper design.
* **Analogy:**
  + **Database = Building Architecture:**
    - You wouldn’t construct a building without a blueprint; similarly, you shouldn’t create a database without an ERD.

**3. Why Spend Time on ERD?**

* **What is an ERD (Entity-Relationship Diagram)?**
  + A visual representation of data and its relationships.
  + Components: Entities (tables), Attributes (columns), Relationships (links between tables).
* **Why Use ERDs?**
  + Prevents errors during database creation.
  + Saves time in debugging and redesigning later.
* **Discussion Prompt:**
  + **Ask:** **“Why do architects spend so much time on blueprints? How is this like database design?”**

**4. The Evolution of Data Storage: From File Systems to Databases**

**Do all applications have a database as their backend?**

* **File Systems:**
  + Data stored in plain text files with delimiters e.g., CSV files (**Comma-Separated Values**).
  + **Example:**

ID, Name, Age, Department

1, Alice, 28, HR

2, Bob, 34, IT

3, Charlie, 25, Marketing

**Why Were File Systems Used?**

* + Simple and suitable for small-scale systems.
* **Drawbacks:**
  + Data redundancy and inconsistency.
  + Lack of relationships between data.

**The Shift to Relational Databases:**

* Introduced by Dr. E. F. Codd in the 1970s.
* Concept of tables (relations) with rows (records) and columns (attributes).
* Addressed issues of redundancy and scalability.

**5. Are All Programs Built on Databases?**

* **Programs That Rely on Databases:**
  + E-commerce platforms (Amazon).
  + Banking systems (data for accounts and transactions).
  + Social media (profiles, posts, comments).
* **Programs Without Databases:**
  + Standalone calculators.
  + Word processors with local file storage.
* **Discussion Prompt:**
  + **Ask: “Why do some programs need databases while others don’t?”**

**Activity:**

* Present a flat-file dataset and show how it can be structured into a database table.
* **Example:** Convert a CSV file like this:

ID, Name, Age, Department

1, Alice, 28, HR

2, Bob, 34, IT

3, Charlie, 25, Marketing

Into a table with clearly defined rows and columns using a spreadsheet tool or a DBMS like SQLite Browser.

**6. Recap and Q&A**

* **Key Takeaways:**
  + Importance of database design and ERD.
  + Benefits of relational databases over file systems.
  + Why not all programs rely on databases.

**Database Design**

ERD, Database Mapping

**Is an ERD (Entity-Relationship Diagram) considering an actual database?**

**No, an ERD is not a database.**

* **What is an ERD?**An Entity-Relationship Diagram is a conceptual tool used to visualize the structure of a database. It represents entities (tables), attributes (fields), and the relationships between entities.
* **Why Isn’t it a Database?**An ERD does not store or manage data; it only serves as a blueprint for designing the database.

**Database Design Process Flowchart**

**Below is the flowchart representation of the database design process**

1. **Requirement Analysis**
   * Understand what data needs to be stored and how it will be used.
2. **Conceptual Design (ERD Creation)**
   * Draw an ERD (Entity-Relationship Diagram) to represent the data, entities, attributes, and relationships.
3. **Database Mapping (Logical Design)**
   * Apply DB Mapping Rules to convert the ERD into database tables.
4. **Normalization**
   * Refine the database structure to minimize redundancy and ensure consistency.
5. **Physical Design**
   * Define the actual storage structure and indexing mechanisms on a database management system (DBMS) like SQL Server or MySQL.
6. **Implementation**
   * Create the database using SQL commands and populate it with data.

**This ERD Needs to Go Through Database Mapping**

Database Mapping is the process of converting the conceptual design (ERD) into the logical structure of a database (tables, fields, and relationships).

**DB Mapping Rules**

Here are the steps for mapping:

1. **Entities → Tables**
   * Each entity in the ERD becomes a table in the database.
   * Example: A Student entity maps to a student table.
2. **Attributes → Columns**
   * Each attribute of an entity becomes a column in the corresponding table.
   * Example: The attributes StudentID, Name, and DOB in the Student entity become columns in the Student table.
3. **Primary Keys (PK)**
   * Identify a unique attribute or combination of attributes for each table.
   * Example: StudentID is the primary key of the Student table.
4. **Relationships → Foreign Keys (FK)**
   * Convert relationships between entities into foreign keys in the tables.
   * Example: A Course table has a CourseID, and a relationship between Student and Course creates a foreign key CourseID in the Student table.
5. **Multi-Valued Attributes**
   * Convert multi-valued attributes into separate tables with a relationship to the original table.
   * Example: If a student has multiple phone numbers, create a PhoneNumbers table linked to Student with a foreign key.

* **Activity:**
  1. Draw a sample ERD on the board and guide trainees through mapping it to tables.
* **Discussion:**
  1. "Why is mapping an essential step in database design?"

Mapping is essential because it acts as the bridge between design and implementation, ensuring that the database is structured correctly, maintains data integrity, and can efficiently meet current and future needs. Without mapping, the database might suffer from poor performance, inconsistency, and difficulty in maintenance or scaling