

## **Data Base Life Cycle:**

The Database Life Cycle (DBLC) describes the step-by-step process of creating, using, and maintaining a database system. It ensures that databases are properly designed, implemented, and managed throughout their lifetime.

### **Phases of the Database Life Cycle (DBLC):**

#### **1. problem identification :**

What you do: Understand the business problem or need that the database must solve. Interview stakeholders and collect high-level goals.

Deliverables: Problem statement, scope, success criteria, stakeholder list, initial feasibility notes (budget/time/constraints).

Typical questions: What data must be stored? Who will use it? What reports/queries must the system support?

#### **2. Analysis (Requirements Collection & Analysis):**

What you do: Gather detailed functional and data requirements. Model business rules, identify entities, attributes, relationships, and constraints. Prioritize requirements.

Deliverables: Requirements specification document, use cases or user stories, preliminary ER (conceptual) model, data dictionary.

Typical activities: Workshops with users, review existing forms/reports, examine current systems and sample data.

#### **3. DB design ( logical design):**

What you do: Convert conceptual ER model into a logical relational schema: tables, keys (PK/FK), constraints, normalization to remove redundancy. Define views and transactions logically (no physical storage choices yet).

Deliverables: Logical schema diagrams, normalized table designs, data dictionary with column types and business rules.

Typical tools: ER diagram tools (Exceldraw).

#### **4. Mapping ( physical design):**

What you do: Map the logical schema to physical structures for the chosen DBMS. Choose datatypes, indexing strategy, partitioning, storage parameters, constraints, and backup/replication approach. Consider query patterns and expected data volumes.

Deliverables: Physical schema DDL (CREATE TABLE ...), index plan, storage/partitioning plan, backup/restore plan, capacity estimates.

#### **5. DB implementation:**

What you do: Create the database objects in the DBMS (tables, constraints, indexes, stored procedures, triggers). Load initial data (ETL), set up users/roles and security.

Deliverables: SQL DDL scripts, ETL scripts, seeded data, user/role definitions, configuration settings.

Tools: Microsoft SQL server , Orcal , mySQL.

#### **6. Testing:**

What you do: Test correctness, performance, security, concurrency, and recovery. Test queries, stored procedures, transactions, backup/restore, and user roles. Include unit tests for DB logic and integration tests with the application.

Deliverables: Test plans/cases, test results, performance baselines, bug reports.

#### **7. Create GUI or Application :**

What you do: Build the front-end forms, reports, APIs, and business logic that interact with the database. Implement input validation, transactions, and error handling.

Deliverables: Application code, API specifications, UI/UX mockups, integration tests.

## 8. Client “End User”:

What you do: Deploy database and application to production, provide training to end users, and set up monitoring/maintenance: backups, patches, performance tuning, capacity planning. Handle change requests and evolve schema as needs change.

Deliverables: Deployment scripts, runbook, monitoring dashboards, SLA docs, user manuals, training materials.

