

■ Relational Modelling Principles – Summary

Relational Database Basics

- Database: structured collection of data (CRUD: Create, Read, Update, Delete).
- DBMS: software to manage databases (Desktop: Access, FileMaker / Server: SQL Server, Oracle, PostgreSQL, MySQL).
- Relational structures: Columns = attributes (single data type), Rows = records (unique, represent one entity).
- Rules of Relations: Cells hold single values, Columns unique, Rows unique, Order matters.

Structured Query Language (SQL)

- Used to create, insert, update, and retrieve data.
- Accessed via command-line (flexible, requires syntax) or GUI (easier).
- DBMS choice depends on company policy, features, and cost.

Preventing Data Anomalies

- Relational DB advantages: structured design prevents errors, inconsistencies, and inefficiency.
- Remove duplicates: avoids slow performance and maintenance issues.
- Eliminate inconsistent data: prevent typos/inconsistent entry styles (move to new table).
- Break data into components: use short, specific fields (e.g., split full name).
- Prevent conflicts: avoid storing calculated values, recalculate when needed.
- Require complete information: set mandatory fields to avoid missing data.
- Maintain consistent structure: avoid multiple values in one field.

Entity-Relationship Diagram (ERD)

- Entity = object (e.g., Student, Course).
- Attribute = property (e.g., Name, Age).
- Relationship = how entities connect (e.g., Student enrolls in Course).
- Tool: diagrams.net (can integrate with Confluence).

■ In short: Relational modelling ensures clean, consistent, and flexible data by following rules (unique rows, atomic values) and preventing anomalies (duplicates, inconsistencies, missing/derived data). SQL + ERD help design and manage these databases effectively.