



Relational Database Design: ER Modelling, Normalisation and Physical Schema



Relational Database Design: ER Modelling, Normalisation and Physical Schema

Introduction:

Application databases built without a deliberate model accumulate duplicated facts, orphaned rows and inconsistent codes, so every new feature needs workarounds and reports disagree with the screens that fed them. Relational database design prevents this by turning business rules into entities, keys, dependencies and constraints before the first table is created. This Core Concept course takes developers, analysts and junior DBAs from requirements to a reviewed entity-relationship model, a normalised logical schema and a PostgreSQL physical design. Participants produce a Normalised Schema and Physical Design Pack for a case business application.

Course Objectives:

- Translate business requirements and rules into a conceptual entity-relationship model agreed with stakeholders
- Define entities, attributes, keys and relationships with correct cardinality and optionality in crow's foot notation
- Normalise a logical schema to Boyce-Codd Normal Form and justify any deliberate denormalisation
- Implement tables, data types, constraints and referential actions in PostgreSQL that enforce business rules inside the database
- Plan indexes, views, transaction boundaries and history tables to match application access patterns and audit needs
- Produce a Normalised Schema and Physical Design Pack with a data dictionary ready for design review

Target Audience:

- Application developers who create and change the tables behind business systems
 - Systems and business analysts who document data requirements and business rules for new applications
 - Junior database administrators who review and deploy schema changes
 - Software testers and integration specialists who depend on consistent keys and reference data
 - Technical leads responsible for data structures in in-house or extended packaged applications
-

Course Outline:

Day 1: Data Requirements, Business Rules and the Conceptual Model

- Conceptual, Logical and Physical Data Model Layers
- Requirements Elicitation: Business Rule Statements and Noun-Verb Analysis
- Entity Identification, Weak Entities and Entity Definition Cards
- Attribute Classification: Simple, Composite, Multivalued and Derived
- Current-State Schema Assessment: Redundancy and Anomaly Spotting Checklist

Day 2: Entity-Relationship Notation, Keys and Normal Forms

- Chen and Crow's Foot Notation for ER Diagrams
- Cardinality and Optionality: One-to-Many, Many-to-Many and Associative Entities
- Candidate, Primary, Alternate and Surrogate Key Selection Criteria
- Functional Dependency Analysis and Insertion, Update and Deletion Anomalies
- Normalisation Walk-Through: 1NF, 2NF, 3NF and Boyce-Codd Normal Form

Day 3: Logical-to-Physical Mapping in PostgreSQL

- ER-to-Table Mapping Rules for Supertypes, Subtypes and Recursive Relationships
- PostgreSQL Data Type Selection: Numeric, Text, Temporal, UUID and JSONB
- Declarative Constraints: NOT NULL, CHECK, Alternate-Key and EXCLUSION
- Foreign Keys and Referential Actions: RESTRICT, CASCADE, SET NULL and SET DEFAULT
- Naming Convention Standard and DDL Script Versioning with Migration Files

Day 4: Indexing, Denormalisation, Concurrency and History

- B-Tree, Partial, Composite and Covering Index Strategy from Access Paths
- Controlled Denormalisation: Summary Columns, Materialised Views and Trade-Off Log
- Views as an Interface Layer for Applications and Reporting
- ACID Transactions, Isolation Levels and Lock Contention Patterns
- Audit Trail and History Tables: Validity Periods and Trigger-Based Change Capture

Day 5: Lab: Case Application Schema and Design Pack

- Lab: Order-to-Invoice Case ER Model and Business Rule Register
 - Lab: Normalisation to BCNF and Dependency Proof for the Case Schema
 - Lab: PostgreSQL Build with Constraints, Indexes and Seed Test Data
 - Design Review Checklist and Peer Walkthrough of the Case Model
 - Normalised Schema and Physical Design Pack with Data Dictionary
-

Skills You Will Gain:

- Conceptual Data Modelling
- ER Diagramming
- Functional Dependency Analysis
- Schema Normalisation
- Referential Integrity Design
- Index Strategy Planning
- Temporal and Audit Table Design
- Data Dictionary Authoring

Why Attend This Course:

- Return with a Normalised Schema and Physical Design Pack for a case business application, built and tested in PostgreSQL
- Stop update anomalies and orphaned records at the source by letting the database enforce rules instead of application code
- Explain design choices, such as a surrogate key or a denormalised column, to developers, testers and reviewers with evidence
- Practise modelling on cases from retail, services and public administration alongside professionals from other organisations

Conclusion:

A well-designed relational schema lets applications grow without duplicated facts, broken links or conflicting codes. The week moves from requirements, business rules and conceptual modelling, through ER notation, keys and normalisation to Boyce-Codd Normal Form, to physical mapping, data types and constraints in PostgreSQL, then indexing, denormalisation, transactions and history tables. The final day applies every step in a lab on a case application, and each participant completes a Normalised Schema and Physical Design Pack with a data dictionary ready for team review.
