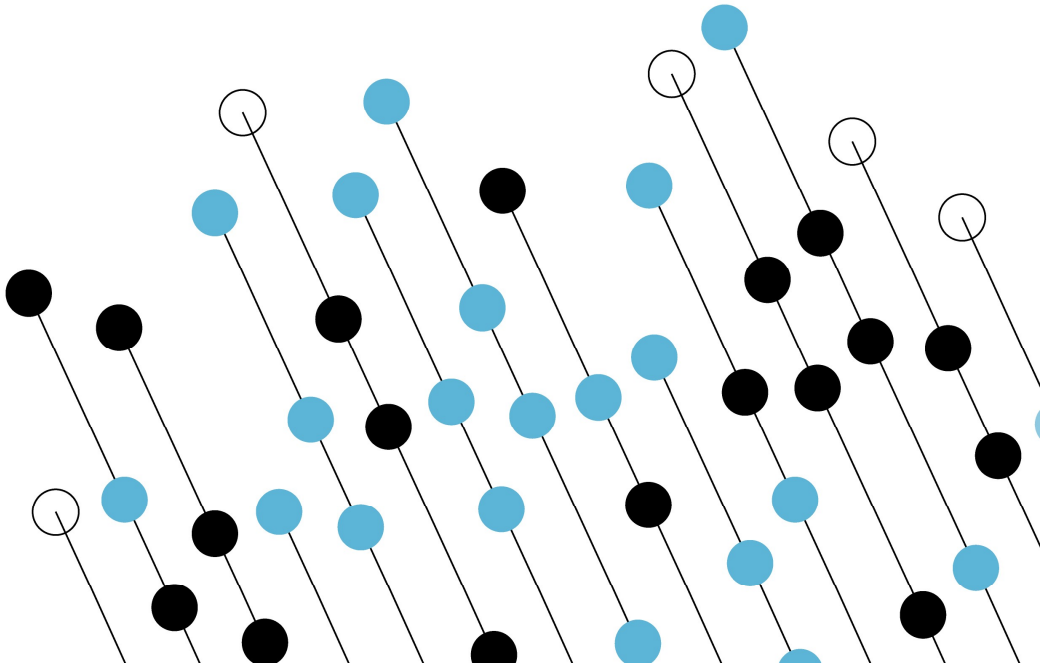
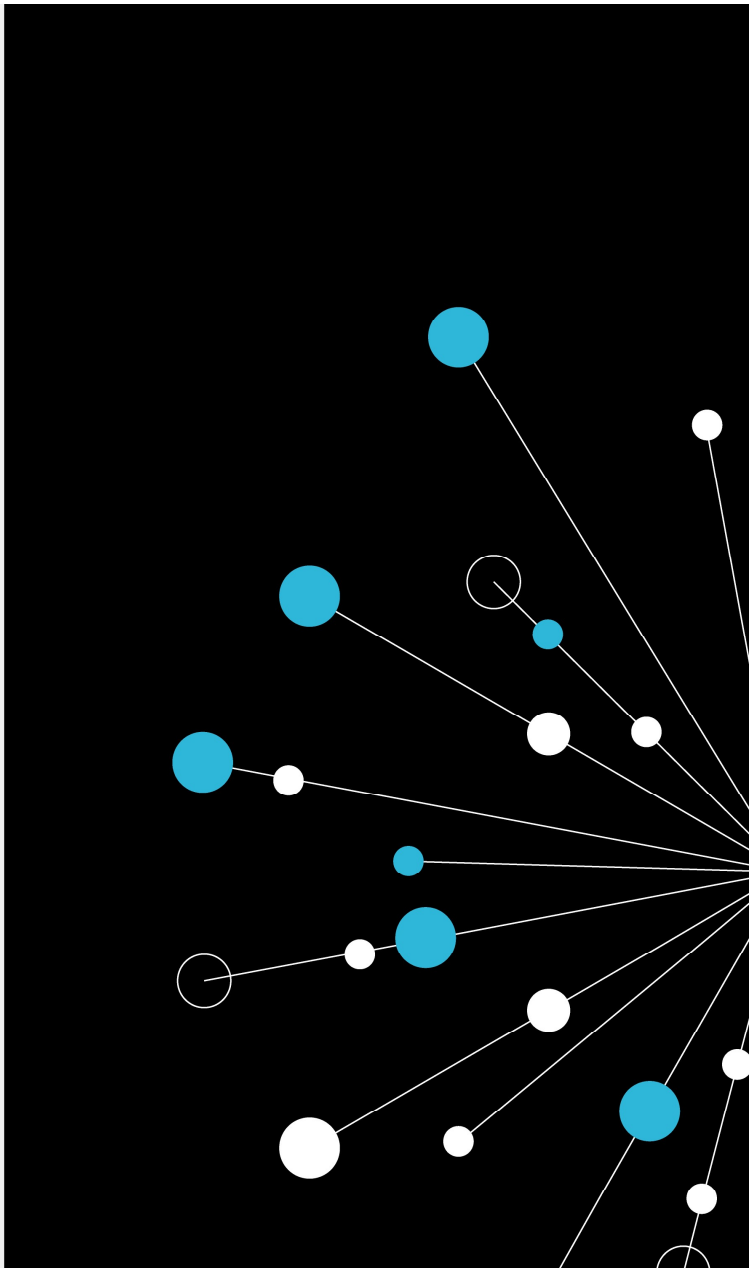
Introduction to Data Modelling



Data Modelling

Contents:

- What is data modelling and what is it used for?
- Types of relational data models
- Developing data models



QA

Poor database design

Poor database design means:

- Problems with creation
- Problems with modification
- Problems with deletion

ANOMALIES

From the business perspective, it is extremely difficult to function under such restrictions. Very often, this situation forces a database upgrade or replacement. That is why the design stage is so crucial.

Good database design

Good database design allows:

- the database to grow indefinitely
- to store new data
- to create new types of data

without changing the structure of the database.

Databases must be powerful but flexible at the same time, allowing business growth and supporting future needs.

In other words, they should be **future proof**. That is why it is extremely important to make sure the database has been designed properly from the very beginning, i.e. it contains appropriate tables and all the necessary fields in those tables.

Data modelling

- **A data model describes how data is represented within a system.** Any system that stores data will require a data model.
- Data models are built during the analysis and design phases of a project to ensure that the requirements for a new application or for re-engineering a legacy application are fully understood.
- A data model defines the logical structure of a database and determines how data is stored, accessed, and manipulated.
- A data model acts as a blueprint for designing a database and helps in understanding the relationships between different data elements.

Data models are represented graphically via **Entity Relationship Diagrams (ERD)** to make them easier to comprehend.

Types of relational data models

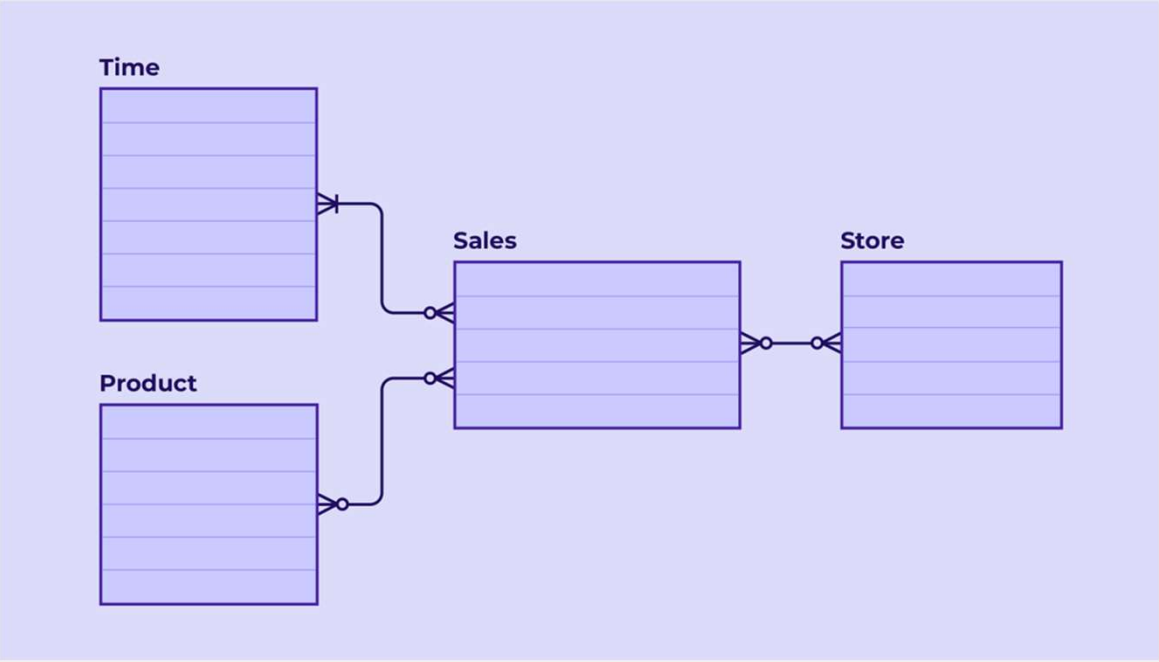
The process of data modelling is broken down into three stages that represent different depths of models:

- **Conceptual model:**
  - Identifies what concepts (entities) the data model is about.
  - Describes how the concepts relate to each other.
- **Logical data model:**
  - Identifies what information (attributes) will be stored about each concept (entity), including a unique identifier.
  - Describes the relationships identified in the conceptual data model in terms of these unique identifiers.
- **Physical data model:**
  - Precisely describes how the data will be stored and their types.
  - May introduce additional constraints that cannot be expressed in the previous two stages.
  - The physical data model is the only one that requires selecting a database. The conceptual and the logical data model are database agnostic.

Examples of relational data models: Conceptual data model

Examples of relational data models

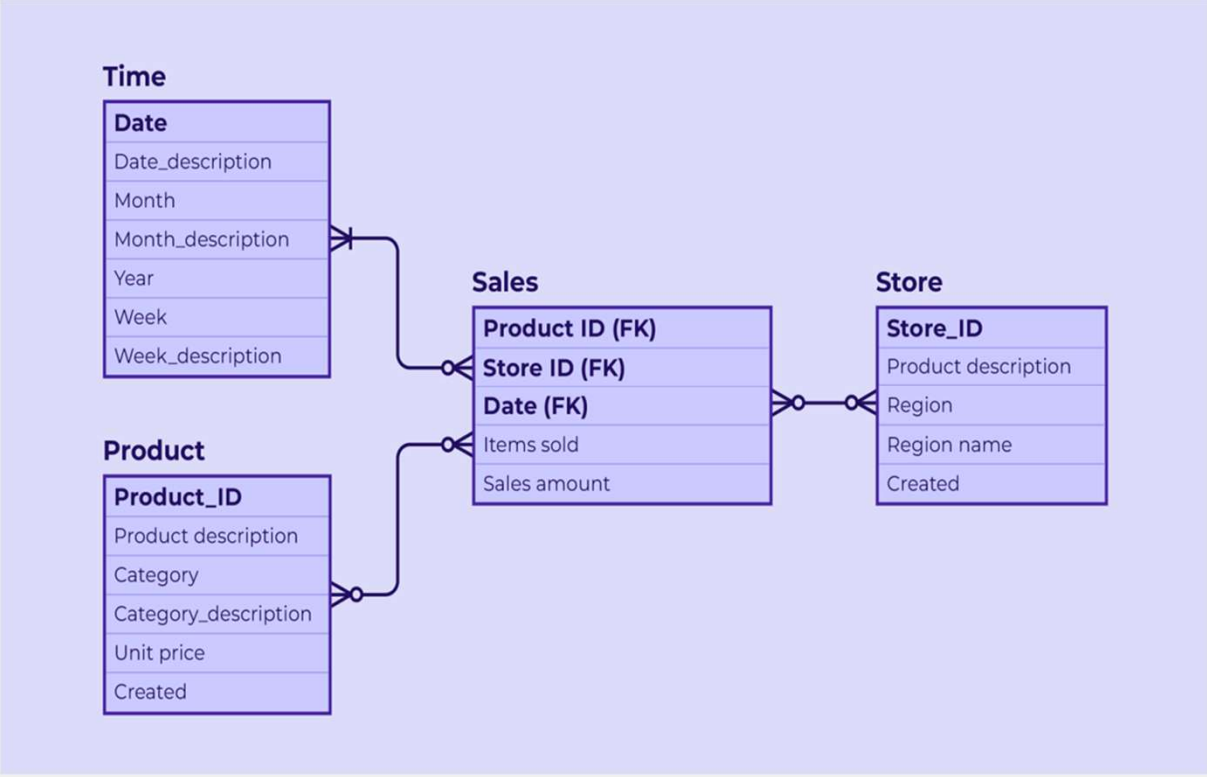
Conceptual data model of an online store. The entities are time, product information, sales information, and store information.



Examples of relational data models: Logical data model

Examples of relational data models

Logical data model of an online store. The entities are expanded upon and contain attributes, some of which are designated as key attributes – Primary Keys and Foreign Keys.



Examples of relational data models: Physical data model

Examples of relational data models

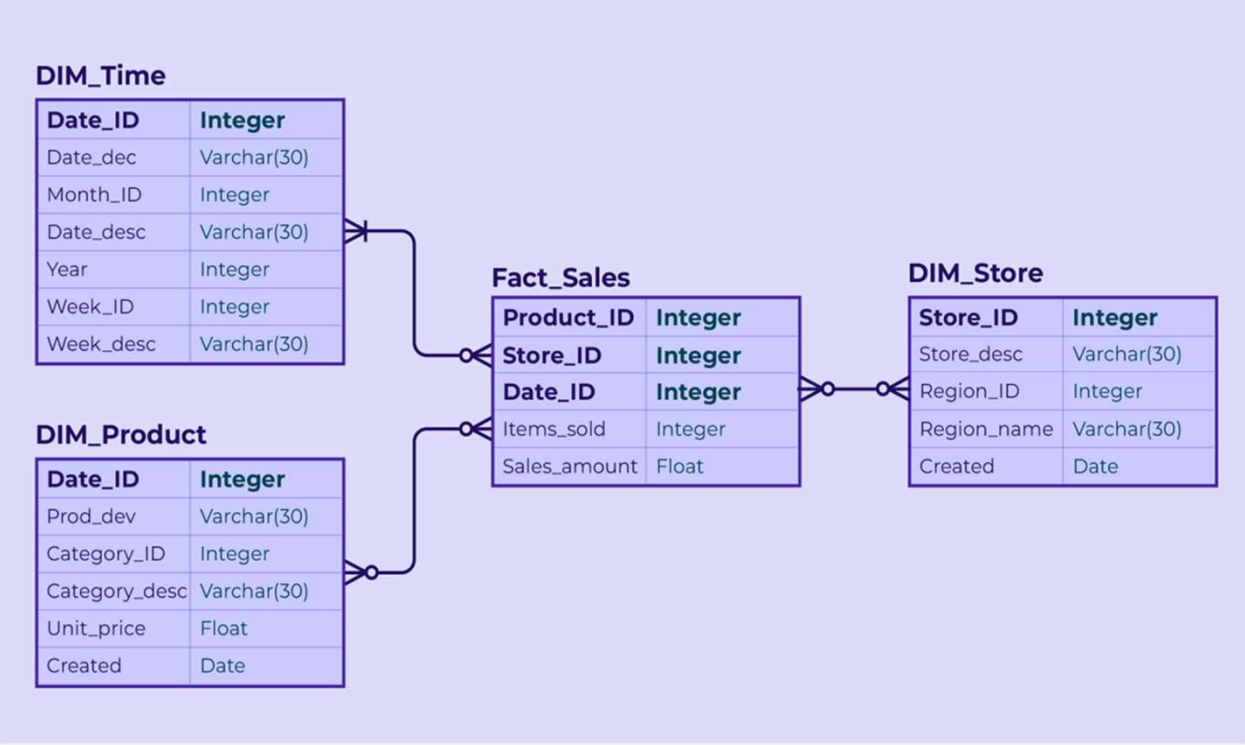
The physical data model represents how the model will be built in the database. When going from a logical to a physical model:

- entities become tables.
- attributes become columns.
- data types have to be specified for each column.
- primary and foreign keys are specified from the key attributes.

Examples of relational data models: Physical data model

Examples of relational data models

Physical data model for the online store.



Developing data models

The process for constructing a data model is often described in terms of how the task is approached:

- Starting from a complete list of business data requirements from which the models are built, or;
- Starting from a collection of examples of data that need to be stored and working from them.

These two approaches are known, respectively, as top-down and bottom-up. Their aims are the same – to produce a physical data model.

In the relational database model, there are two widely known approaches:

- Entity-Relationship modelling – a top-down approach
- Normalisation – a bottom-up approach