数据设计

Chapter 7

北京航空航天大學
SCHOOL OF SOFTWARE BEIHANG UNIVERSITY 软件学院

内容安排

- 数据架构
- 数据库架构
- 数据库设计

数据存储：对象的持久化问题

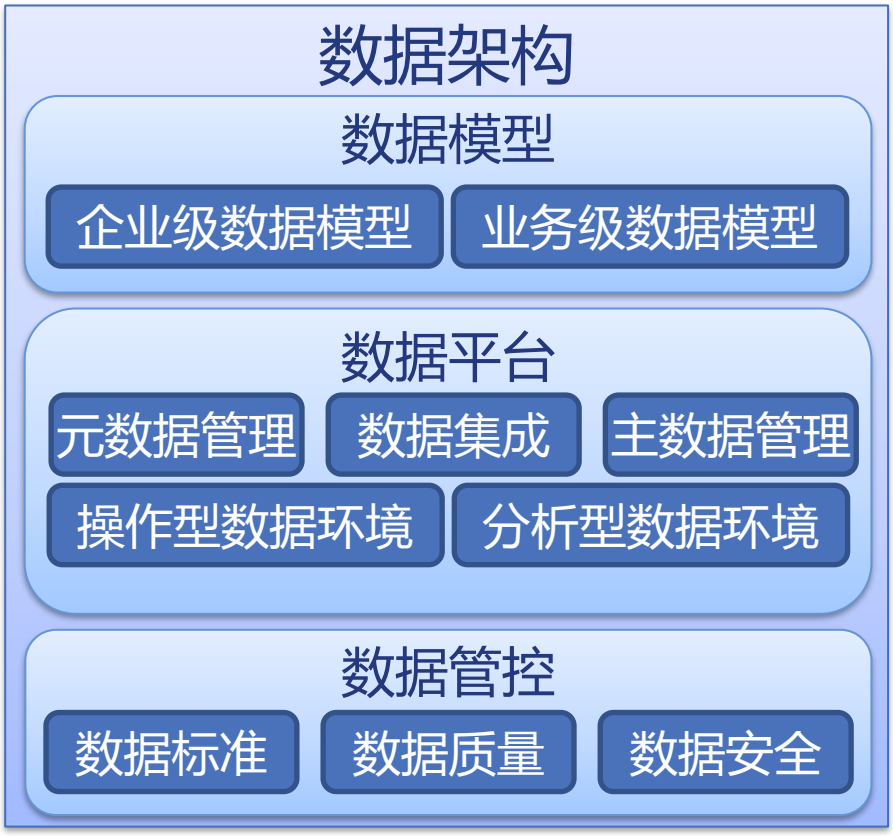
- 文件
  - 各种格式的文件（.txt, .ini…）
- 关系数据库（RDBMS）（最常用）

- 面向对象数据库（OODBMS）、对象存储设备

关系数据库正在向对象-关系数据发展

数据架构

- 数据架构是从跨企业应用系统的视角**统一对数据进行组织和规划**，提高跨系统间数据存储和共享的效率

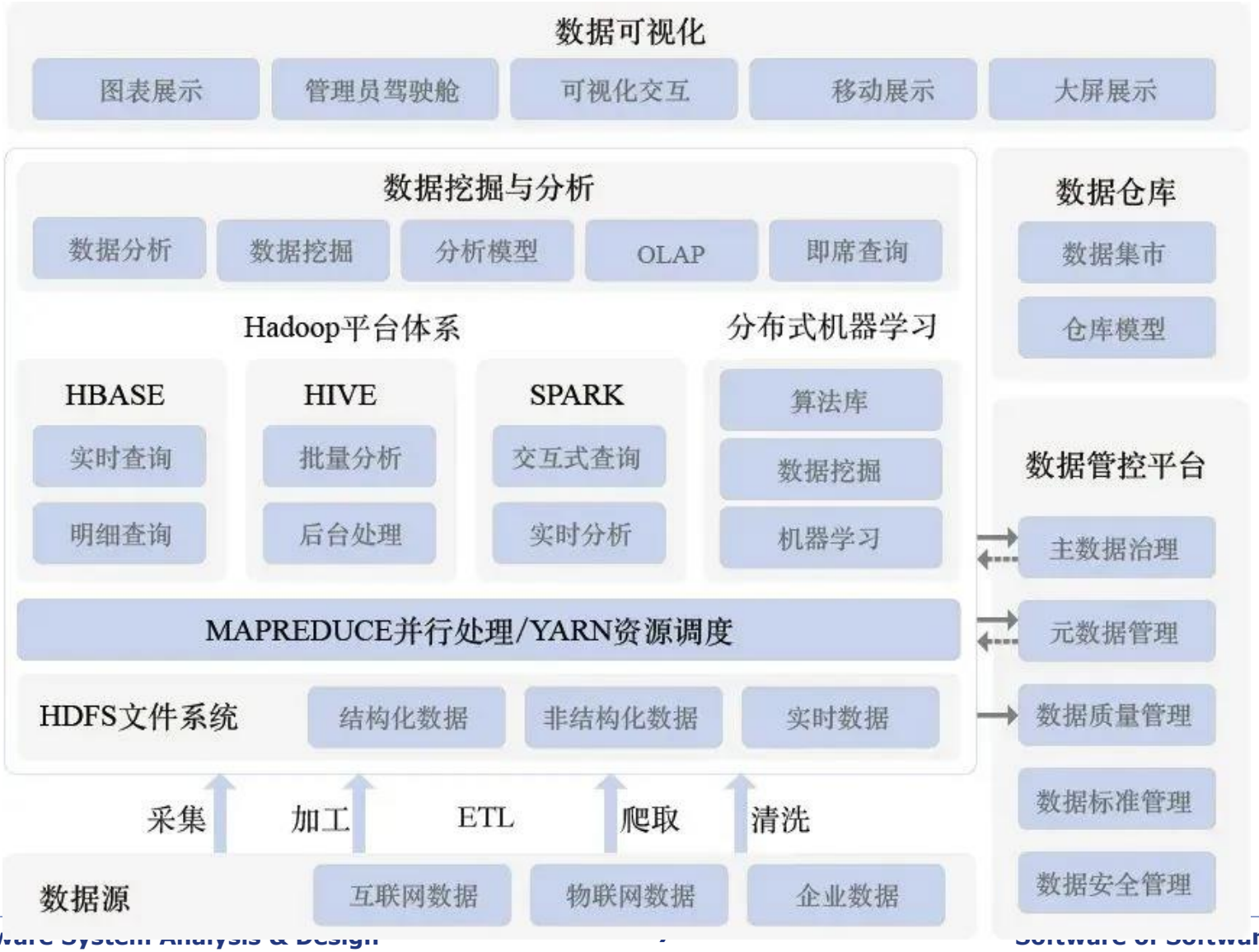


- **统一性**：各业务线统一数据视图，综合考虑实时/非实时、结构化/非结构数据管理策略
- **标准性**：借鉴成熟、科学的企业/行业数据标准
- **扩展性**：实现数据架构的模型化、层次化，支持在统一核心数据模型基础上的模型扩展

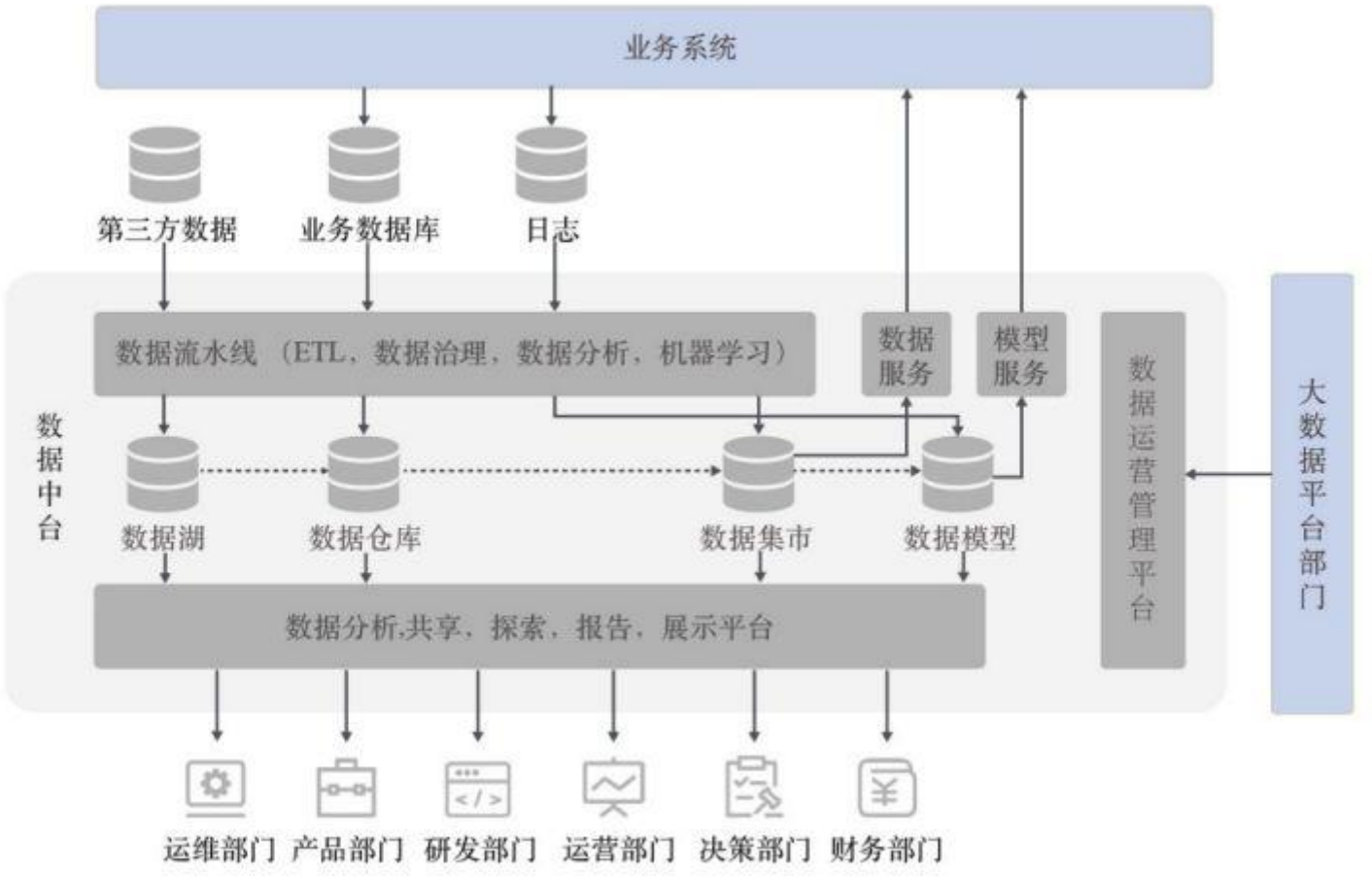
数据架构相关概念发展

- **数据库**：传统关系型数据库的主要应用，主要是基本的、日常的事务处理
- **数据仓库**：支持复杂的数据分析，侧重决策支持
- **数据集市**：“小型数据仓库”，仅包含单个主题，面向部门级业务或某一个特定的主题
- **数据湖**：存储企业原始数据的大型仓库，可供存取、处理、分析及传输
- **大数据平台**：以处理海量数据存储、计算及流数据实时计算等场景为主的一套基础设施
- **数据中台**：为了从数据中提取价值来支持更有效的数据运营，那么不能指导实际行动，创造实际价值的数据以及从数据中产生的知识是无用的，那花大价钱来做这个系统也没有必要。

经典的大数据平台架构



基础数据中台架构



数据中台示例（用友）

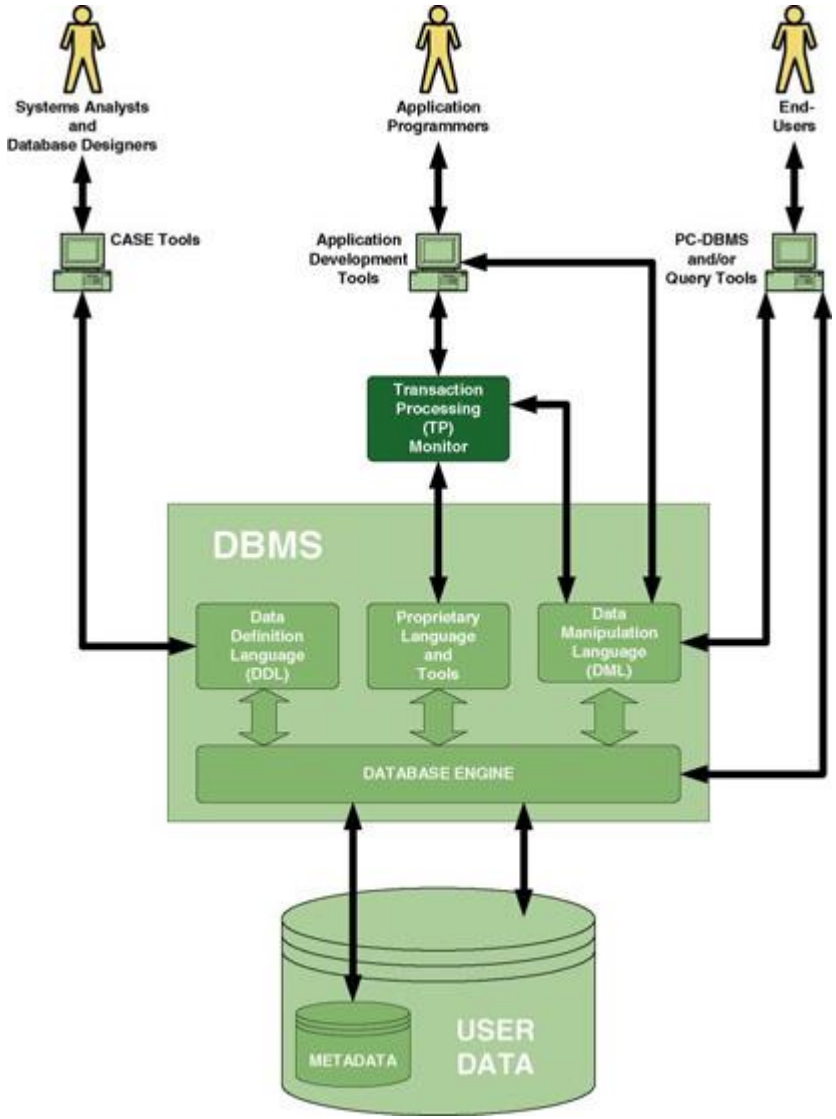
数据中台产品总体架构

用友 yonyou

数据库架构

- Database architecture – the database technology used to support data architecture
  - Including the database engine, database utilities, CASE tools, and database development tools.
- Database management system (DBMS) – special software used to create, access, control, and manage a database.
  - The core of the DBMS is its database engine.
  - A data definition language (DDL) is used to physically define tables, fields, and structural relationships.
  - A data manipulation language (DML) is used to create, read, update, and delete records in database and navigate between records.

Typical DBMS Architecture



Database Capacity Planning

- For each table sum the *field sizes*. This is the *record size*.
- For each table, multiply the *record size* times the number of entity instances to be included in the table (planning for growth). This is the *table size*.
- Sum the *table sizes*. This is the *database size*.
- Optionally, add a slack capacity buffer (e.g. 10percent) to account for unanticipated factors. This is the *anticipated database capacity*.

Database Distribution

- Centralization
  - Entire database on a single server in one physical location
- Horizontal distribution (also called partitioning)
  - Row assigned to different database servers/locations.
  - Efficient access and security
  - Cannot always be easily recombined for management analysis
- Vertical distribution (also called partitioning)
  - Specific table columns assigned to specific databases/servers
  - Similar advantages and disadvantages of Horizontal

Database Replication

- Replication
  - Data duplicated in multiple locations
  - DBMS coordinates updates and synchronization
  - Performance and accessibility advantages
  - Increases complexity

Relational Databases

- Relational database – a database that implements stored data in a series of two-dimensional tables that are “related” to one another via foreign keys.
  - The physical data model is called a schema.
  - The DDL and DML for a relational database is called SQL (Structured Query Language).
  - Triggers – programs embedded within a database that are automatically invoked by updates.
  - Stored procedures – programs embedded within a database that can be called from an application program.

数据库设计

- 数据库设计(Database Design)目标
  - 确定设计中的持久性类
  - 设计适当的数据库结构已存储持久化类
  - 为存储和取得持久性数据定义机制和策略，以满足系统的性能标准
- 输入
  - 对象模型
- 输出
  - 数据模型

用关系数据库来存储对象

你想把车停在一个面向对象的车库里。把车开进车库，下车，关上车门，然后回到你的房间。当你想出去的时候，只要走进车库，钻进汽车，启动，然后开走

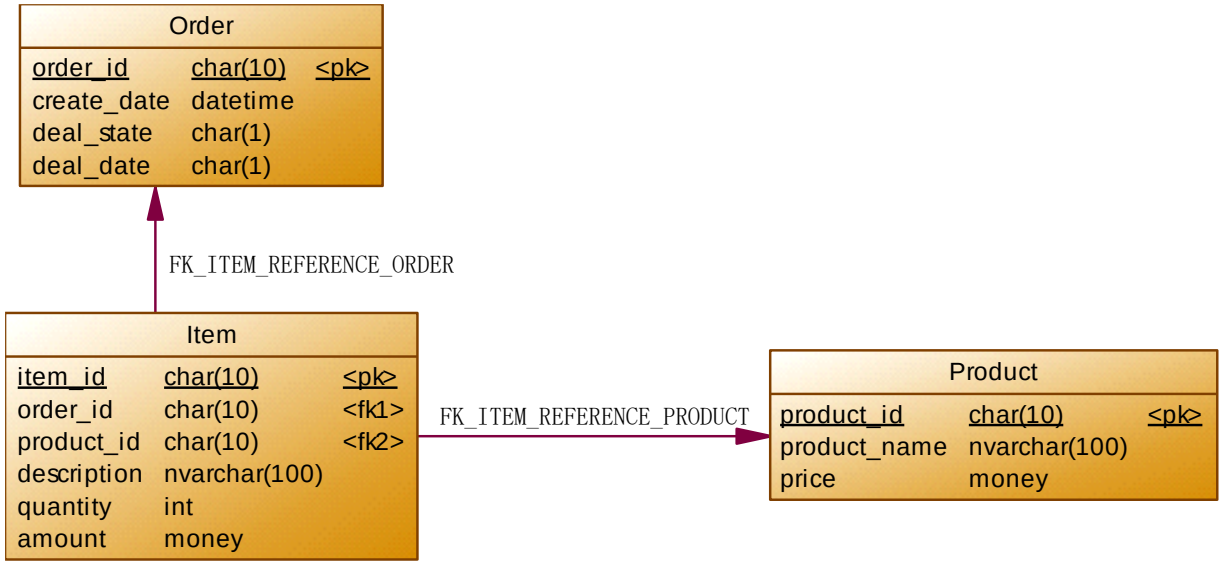
你想把车停在一个关系数据库的车库里把车开进车库，下车，卸下车门，将它们放在地上；卸下所有的车轮，将它们放到地上；卸下保险杠及其它的东西。然后回到你的房间。当你想出去的时候，走进车库，先安上车门，再安上保险杠，然后是车轮等等，都安完了，钻进汽车，点火，然后开走

关系数据库和面向对象

- RDBMS
  - 关注数据
  - 比较适合描述含有动态或临时关系的应用系统
  - 暴露数据(列值)
- 面向对象系统
  - 关注行为
  - 比较适合处理复杂的或特定于状态的行为，数据处于次要位置
  - 隐藏数据(封装)

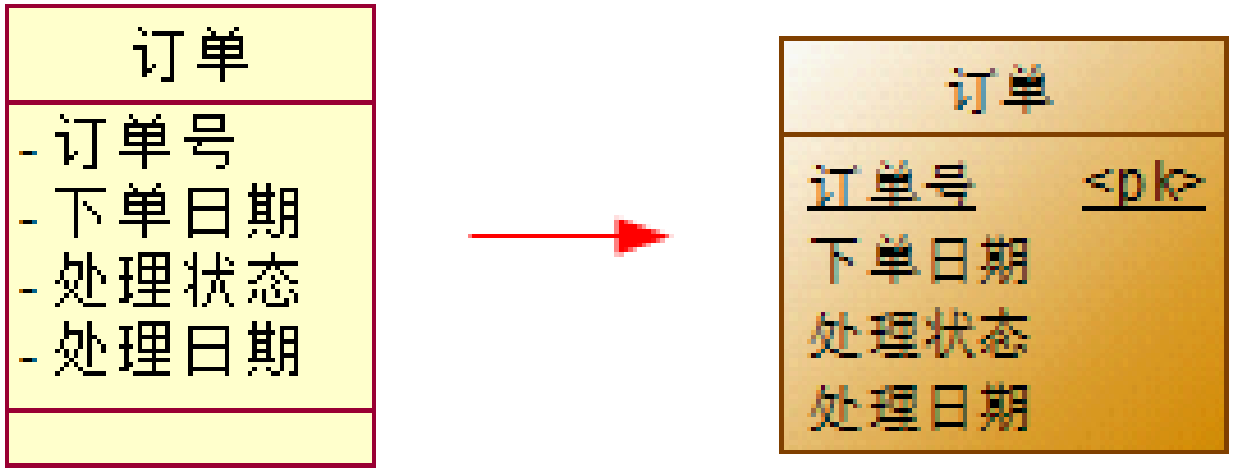
数据模型和对象模型

- 数据模型
  - 实体
  - 关系
- 对象模型
  - 类(属性)
  - 关联



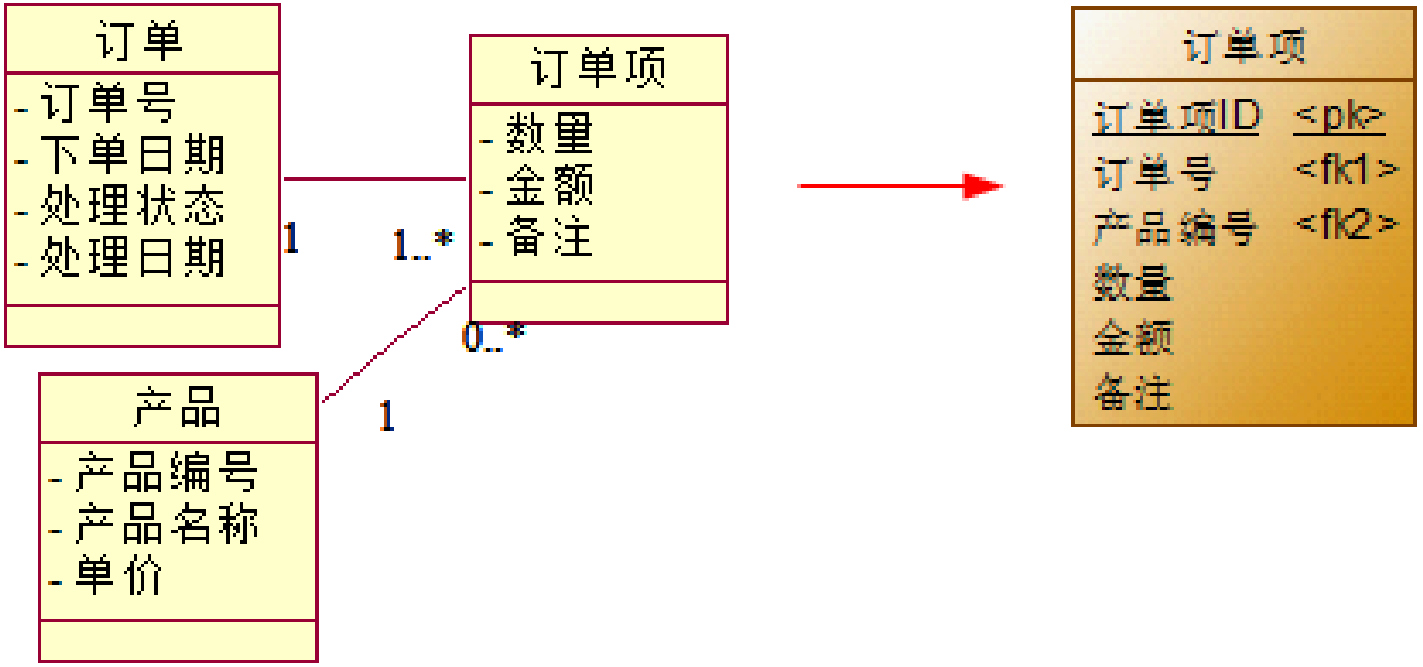
将永久性类映射为表

- 在一个关系数据库中
  - 表中的每一行都被认为是一个对象
  - 表中的列则对应于类的持久性属性



映射对象间的关联关系

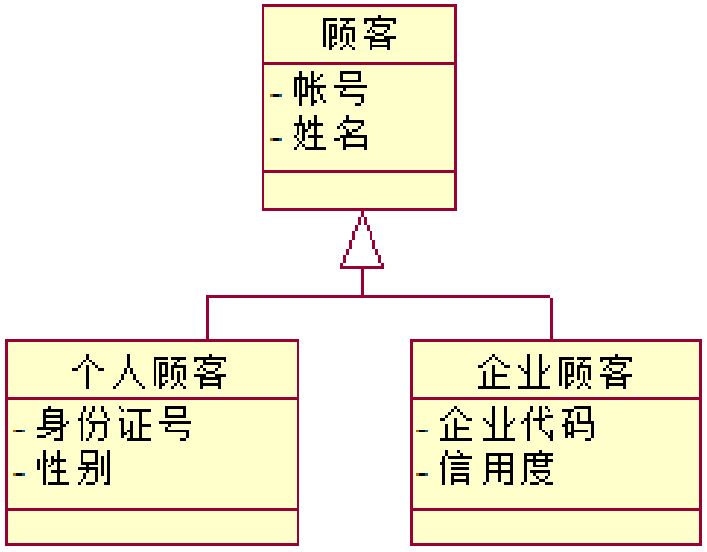
- 两个持久性对象间的关联关系表现为所关联对象的外键
  - 外键是一个表中的一列，其中含有所关联对象的主键值



映射对象间的泛化关系

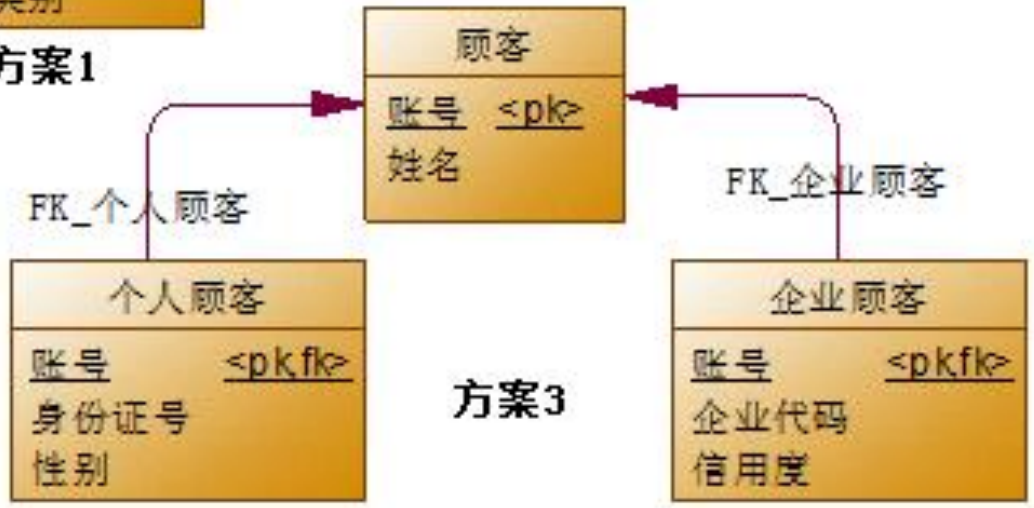
- 数据模型不支持直接方式的继承关系建模
- 两种解决方案:
  - 使用不同的表(规范化数据)
  - 复制所有继承的关联和属性(反规范化数据)

示例：映射泛化关系



方案1

方案2



方案3

将类行为映射到存储过程

- 可以利用存储过程和触发器来实现类的行为
- 确定是否有操作可以作为存储过程/触发器来实施
- 候选操作：
  - 处理持久性数据的操作
  - 在计算中所涉及的查询操作
  - 需要访问数据库以验证数据的操作

作业

- 作业5：数据库设计
  - 进度安排
    - 11月12日之前：基于需求分析和架构设计方案，结合之前有关数据库设计的方法，开展系统数据库设计方案，并着手编制数据库设计说明书
  - 提交
    - DDL：2022年11月12日晚上10:00
    - 交付物：数据库设计说明书，提交一份Word文档，以及可能的数据库设计模型文件
      - 如果采用分布式数据库、或分库等数据架构策略，应给出数据架构的设计方案
      - 数据库设计说明书应包含详细的数据库设计方案，表结构的详细说明；如果采用NoSQL数据库，也应给出对应的结构