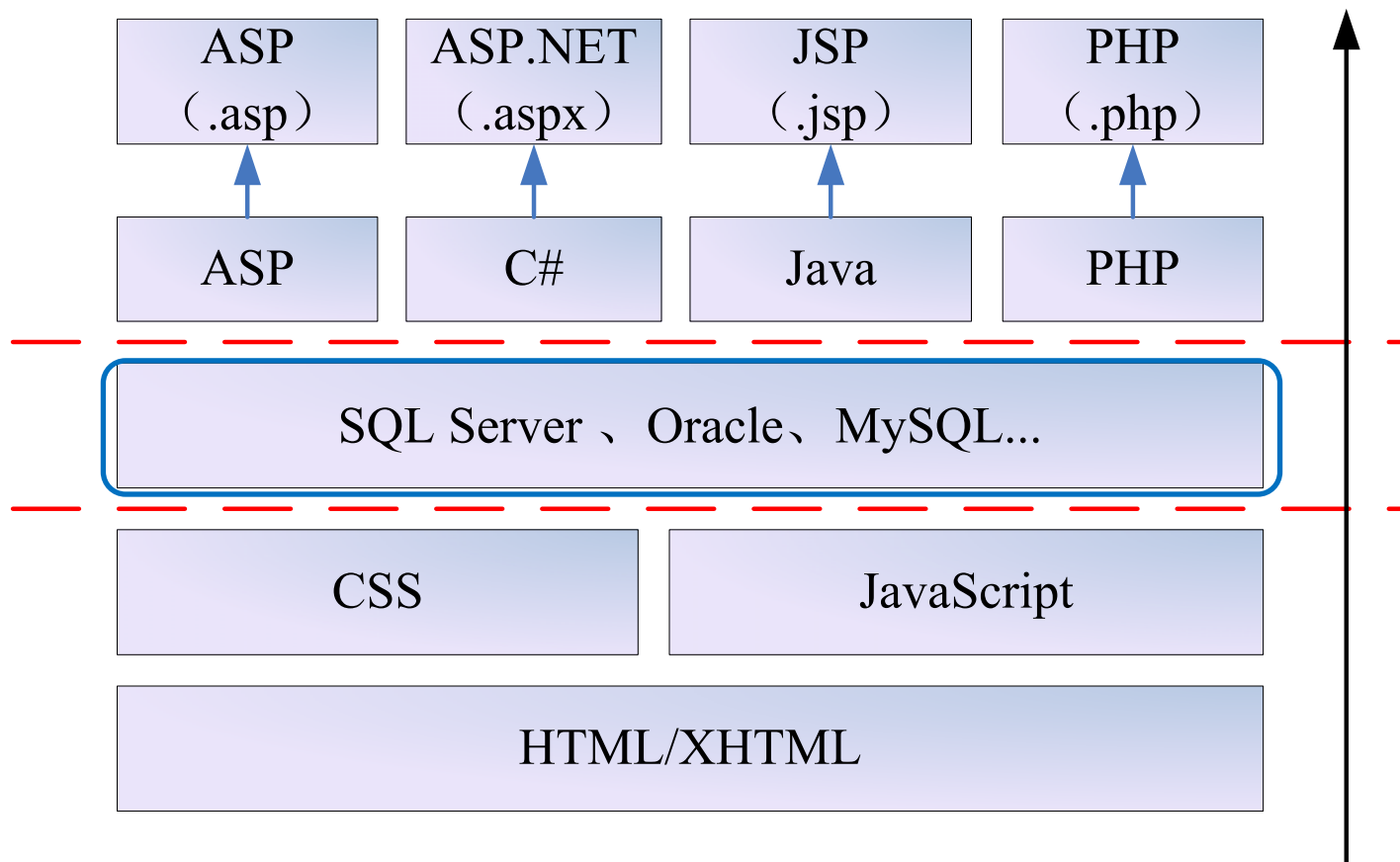


Power Designer

Email: 18442056@QQ.com

需要的掌握的相关知识

- 软件工程
- 系统分析与设计
- 数据库开发与设计
- 至少一门开发工具



为什么需要设计数据库



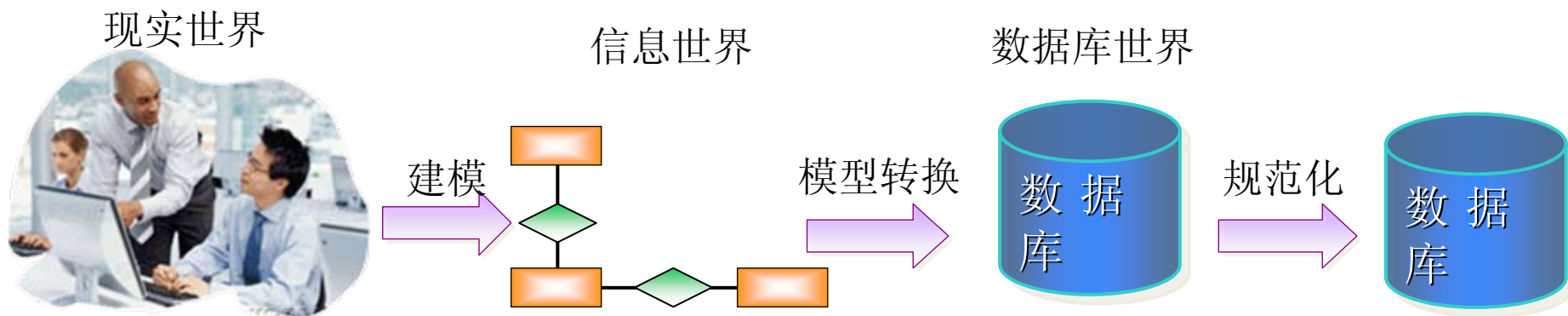
修建茅屋需要设计吗？



修建大厦需要设计吗？

结论：当数据库比较复杂时我们需要设计数据库

数据库开发周期



- 需求分析阶段：分析客户的业务和数据处理需求；
- 概要设计阶段：设计数据库的E-R模型图，确认需求信息的正确和完整；
- 详细设计阶段：将E-R图转换为多张表，进行逻辑设计，并应用数据库设计的三大范式进行审核；
- 代码编写阶段：选择具体数据库进行物理实现，并编写代码实现前端应用；
- 软件测试阶段：.....
- 安装部署：.....

字段名	名称	类型	长度	说明
F_ID	编号	NUMBER	4	自动增长编号- 主键
F_DOCUMENT_CODE	文档编码	VARCHAR	30	文档的编号
F_USER_CODE	用户编号	VARCHAR	30	用户的编号
F_IS_REVERT	是否回滚	BIT	1	是否回滚记录 0 为不同 1 为相同

字段名	名称	类型	长度	说明
F_ID	编号	NUMBER	4	自动增长编号- 主键
F_DOCUMENT_CODE	文档编码	VARCHAR	30	文档的编号

F_USER_CODE
F_IS_READ

字段名	名称	类型	长度	说明
F_DOCUMENT_ID	文档编号	NUMBER	4	自动增长编号- 主键
F_DOCUMENT_CODE	文档编码	VARCHAR	30	文档的编号
F_DOCUMENT_NAME	文档名称	VARCHAR	60	文档的标题
F_DOCUMENT_TYPE	文档类型	VARCHAR	30	文档类型，用于区分文档的类型
F_DOCUMENT_FILE_TYPE	文档文件类型	VARCHAR	8	文档文件类型，用于标识文档从数据库内获取出来后保存的文件类型
F_IS_PROMULGATE	是否发布	BIT	1	0 为未发布，1 为已发布
F_DOCUMENT_DEPARTMENT	文档所属部门	NUMBER	4	文档所属部门的编号
F_IS_DELETE	是否删除	BIT	1	0 为未删除，1 为已删除
F_CREATE_DATE	创建日期	DATE		文档添加进入的日期，当文档发布后将表示为文档发布日期
F_OWNER_USER_ID	创建者 ID	NUMBER	4	文档作者
F_DOCUMENT_DATE	文档内容	IMAGE	16	文档内容

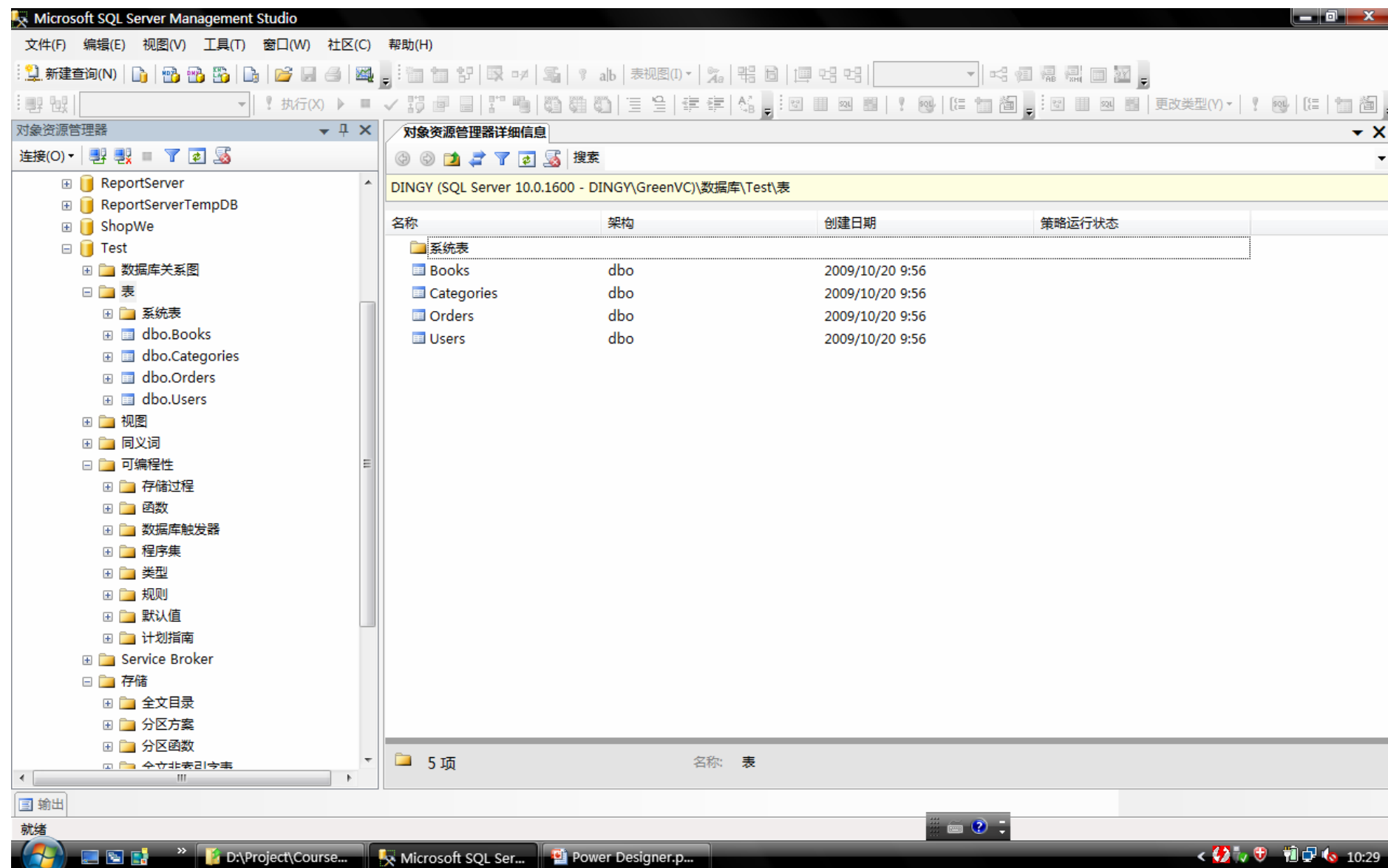
```

create table Users (
    UsersID          int          not null,
    UserCode         varchar(20)  not null,
    Pwd              varchar(100) not null,
    Name             nvarchar(10) not null,
    Address          ntext        null,
    Phone            varchar(20)   null,
    Email            varchar(100)  null,
    Role             int          not null,
    constraint PK_USERS primary key (UsersID)
)
go

create table Orders (
    OrdersID          int          not null,
    UsersID           int          not null,
    BooksID           int          not null,
    OrderDate         datetime     not null,
    Quantity          float        not null,
    UnitPrice         money        not null,
    Remark            ntext        null,
    constraint PK_ORDERS primary key (OrdersID)
)
go

create table Books (
    BooksID           int          not null,
    Title             nvarchar(100) not null,
    Author            nvarchar(50)  null,
    PublisherName     nvarchar(100) null,
    PublishDate       datetime     null,
    ISBN              varchar(20)   not null,
    Content           ntext        null,
    CategoriesID      int          not null,
    UnitPrice         money        not null,
    Stock            float        not null,
    Storage           float        not null,
    Catalog           ntext        not null,
    Discount          float        not null,
    constraint PK_BOOKS primary key (BooksID)
)
go

```



Power Designer简介

- Power Designer是Sybase公司的CASE工具集，使用它可以方便地对管理信息系统进行分析设计，它几乎包括了数据库模型设计的全过程。利用Power Designer可以制作数据流程图、概念数据模型、物理数据模型，可以生成多种客户端开发工具的应用程序，还可为数据仓库制作结构模型，也能对团队设计模型进行控制。它可与许多流行的数据库设计软件，例如：PowerBuilder，C#，Java等相配合使用来缩短开发时间和使系统设计更优化

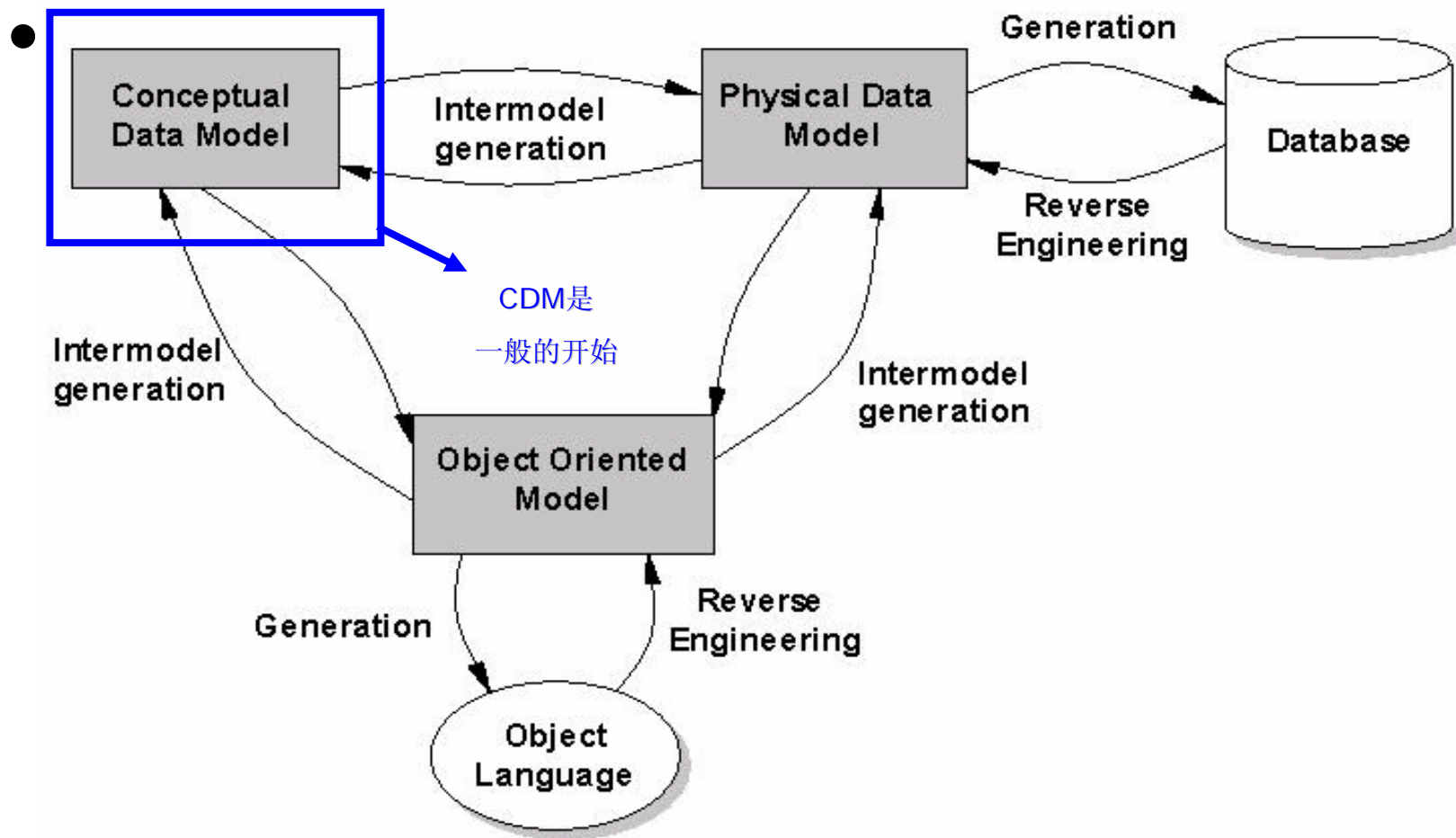
Power Designer的功能

- **Data Architect**：这是一个强大的数据库设计工具，使用Data Architect可利用实体-关系图为一个信息系统创建"概念数据模型"-CDM（Conceptual Data Model）。并且可根据CDM产生基于某一特定数据库管理系统的"物理数据模型"-PDM(Physical Data Model)。还可优化PDM，产生为特定DBMS创建数据库的SQL语句并可以文件形式存储以便在其他时刻运行这些SQL语句创建数据库。另外，Data Architect还可根据已存在的数据库反向生成PDM、CDM及创建数据库的SQL脚本。

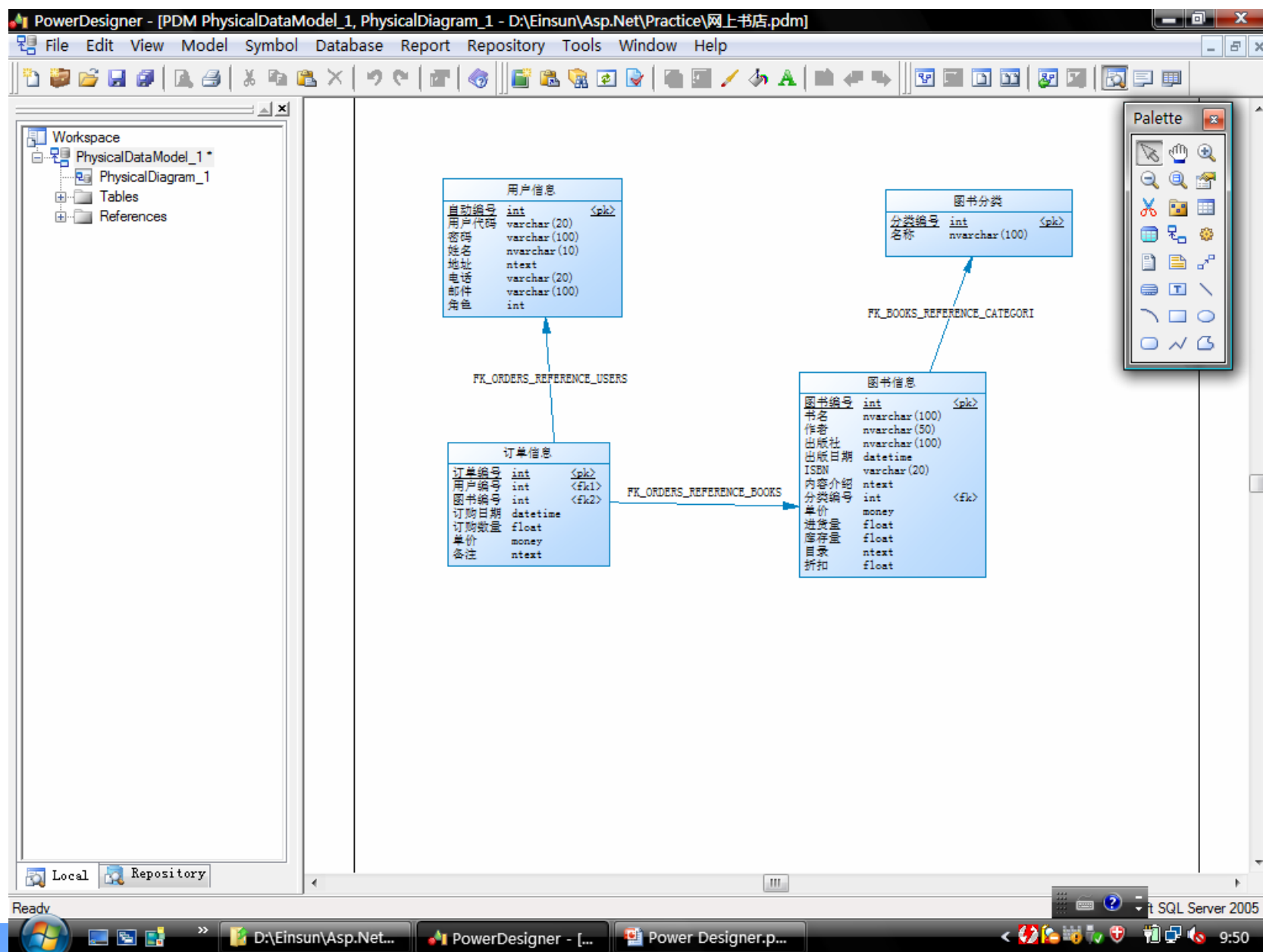
Power Designer的4种模型

- 概念数据模型 (CDM)
- 物理数据模型 (PDM)
- 面向对象模型 (OOM)
- 业务程序模型 (BPM)

CDM PDM OOM三者转换关系



Power Designer 软件介绍



关于业务规则

- 业务规则是业务活动中必须遵循的规则,是业务信息之间约束的表达式,它反映了业务信息数据之间的彝族完整性约束.每当信息实体中包含的信息发生变化的时候,系统都会检查这些信息是否违反特定的业务规则。当你使用业务规则约束 CDM 的时候,他们不被转变为可执行的代码。 业务规则可能被实现为 PDM 的约束。

业务规则类型	业务规则说明	业务规则举例
事实	信息系统中存在的事实	一个出版者可能出版一或多个的主题的图书
定义	信息系统中对象的特性	一位作家被一个名字和一个住址识别
公式	信息系统中的计算公式	总金额为所有订单金额的总和
确认	信息系统中需要的确认	支付所有作家一本书的版税百分比必须为版税的 100%
需求	信息系统中功能的详细说明	模型被设计以致版税的总数量不超过总售卖的 10%
约束	信息系统中数据之间的约束	销售开始日期必须迟于出版日期

如何创建业务规则

- 从创建数据库的作用看，似乎只有约束（**Constraint**）和确认（**Validation**）是有意义的（生成完整性约束）
- 建立规则后，再从实体中去选取相应的规则进行应用（也可手动键入约束，主键或外键亦可看成是一些特殊的约束）
- 可以通过预览（**Preview**）查看生成的SQL脚本检查内容

一些重要操作

- 关于数据项目：一个数据项目是一个基本信息。
- 创建实体：DB相关的主键属性就是Identifiers。
- 定义实体间的角色与联系
- 定义和使用域：一个域定义你能适用于多个数据项目的标准数据结构。当你修正一个域时,你将更新全部与域关联的数据项目。当你作任何变化的时候,这导致数据一致化特性比较容易。这样,就可以创建一些自己常用的域,来应用到数据对象上,就不会导致相同的用途,但是数据类型和长度不一致。
- 定义和使用继承

将CDM对象转换为PDM对象

- Tools—>Generate Physical Data Model
- 数据库类型选择
- 名称定义：设定组成物理对象名称规则
- 错误检查无误后生成相应的PDM
- 自动产生继承关系的父子实体（Table）

正向工程

- 正向工程：你能直接地从 PDM 产生一个数据库, 或产生一个能在你的数据库管理系统环境中运行的数据库脚本。
- 可以生成数据库脚本, 如果选择 ODBC 方式, 则可以直接连接到数据库, 从而直接产生数据库表以及其他数据库对象。
- Database —> Generate Database
- Database —> Modify Database

逆向工程

- 逆向工程：将已存在的数据库产生进新的 PDM 之内。数据来源可能是从脚本文件或一个开放数据库连接数据来源。
- 当你逆向工程使用脚本的时候,你能使用一个单一脚本文件或一些脚本文件。
- File->Reverse Engineer->Database

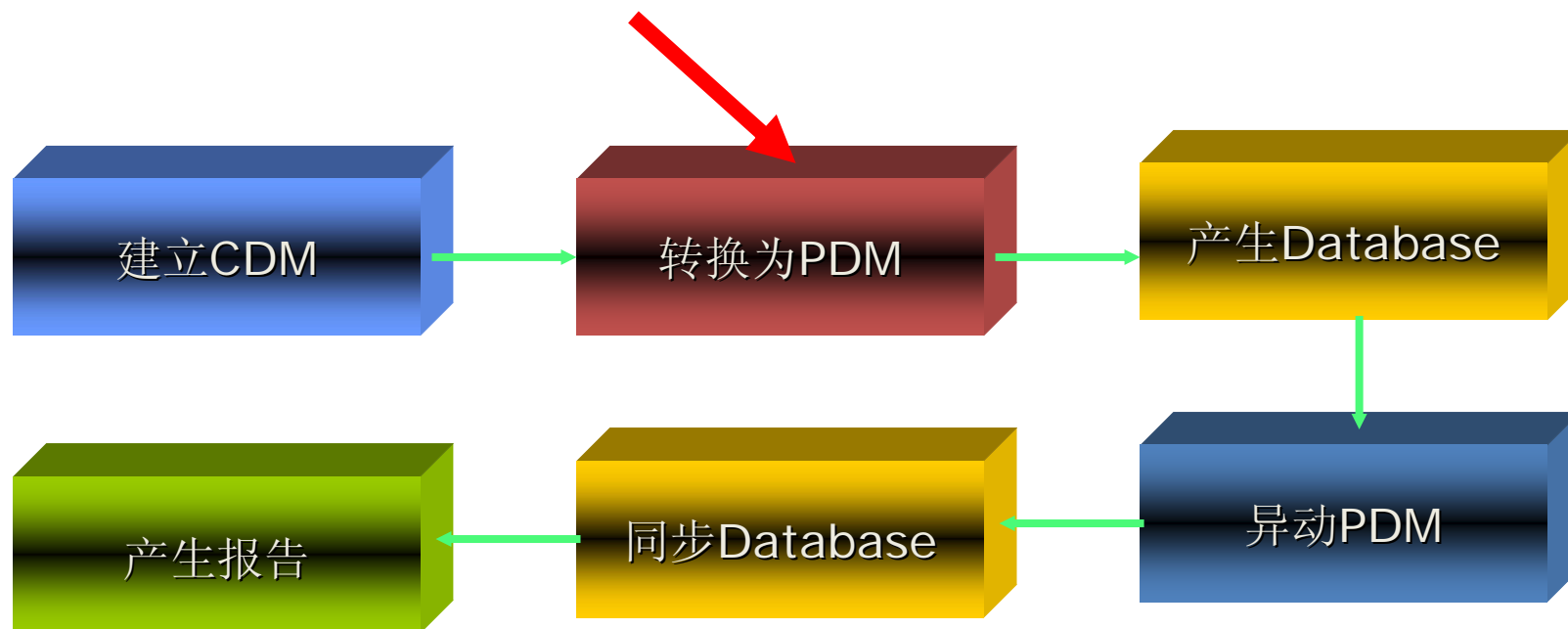
数据库相关操作

- Database菜单包含主要的数据库操作
- 连接数据库（Connect）
- 产生数据库（Generate Database）
- 同步数据库（Modify Database）
- 产生测试数据（Generate Test Data）
- 执行SQL脚本（Execute SQL）
- 创建Trigger和Procedure

生成报表

- 全部由简单鼠标操作完成
- 定义需要的栏位
- 调整输出格式（宽度、大小等）
- 可以产生为RTF或HTM文件

使用步骤

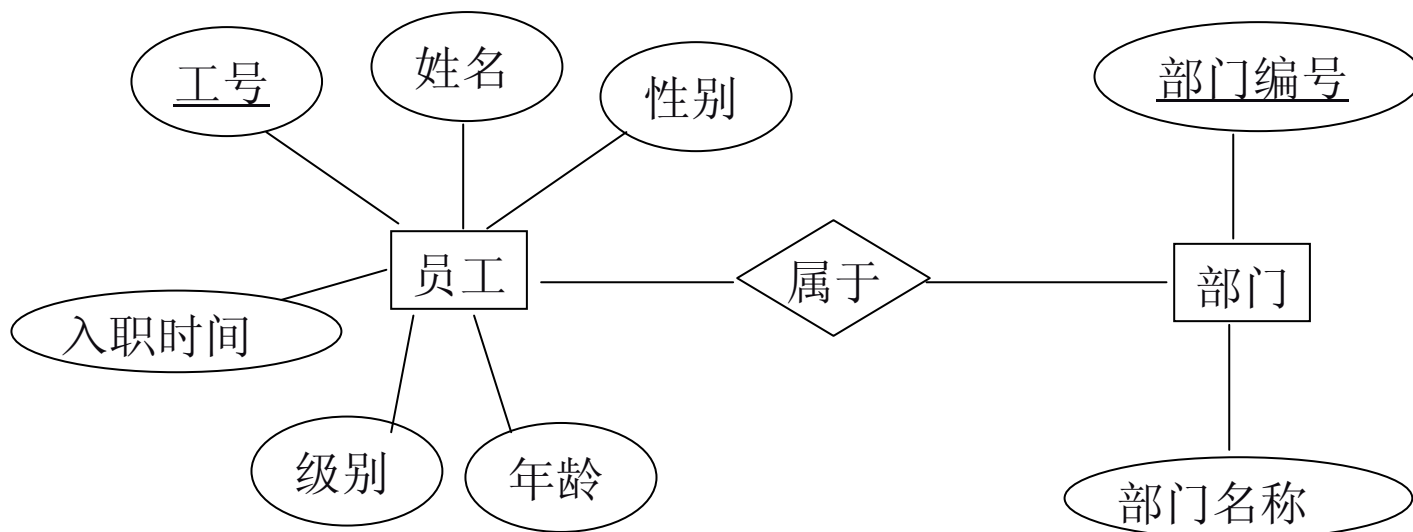


Power Designer小结

- Power Designer仅仅是实现的工具。
- Power Designer用途不局限与数据建模如业务流程建模、Web Services等。
- 并不是每个设计都需要用到Power Designer。

例如：小的系统，或Table数比较少的情況下就没有必要采用Power Designer了

实现如下关系的数据库



关系模式为一对一

- 员工（工号，姓名，性别，级别，年龄，入职时间，所属部门编号） 工号为主键。
- 部门（部门编号，部门名称） 部门编号为主键