个人项目报告：逻辑设计

引言

逻辑设计的任务和目标

概念设计阶段是独立于任何一种数据模型的，但是逻辑设计阶段就与选用的 DBMS 产品发生关系了。系统逻辑设计的任务就是将概念设计阶段设计好的基本 E-R 图转化为选用 DBMS 产品所支持的数据模型相符合的逻辑结构。数据库的逻辑结构是一种抽象的概念，主要用来描述各类数据类型的关系与结构。

规范化

一个低一级范式的关系模式，通过模式分解可以转换为若干个高级范式的关系模式的集合，这样的过程称作规范化。

范式

如果一个关系满足某个指定的约束集，则称它属于某种特定的范式（Normal Form）；目前关系数据库有六种范式，各范式呈递次规范，越高的范式数据冗余越小。最常用的是第一到第三范式。

第一范式

满足最低要求约束的称为第一范式，简称 1NF。当一个关系只包含原子值这一约束时，称为 1NF。原子值即为二维表的每一行和列的交叉位置上总是精确地存在的一个值，而不是值集，即不能“表中有表”。

第二范式

如果关系模式 R 满足第一范式，并且 R 的所有非主属性都完全依赖于 R 的每一个候选关键属性，称 R 满足第二范式，简记为 2NF。

需要注意的是：

1. 如果关系 R 的全体属性都是 R 的主属性，那么 R 满足 2NF
2. 从 1NF 中消除非主属性对码的部分函数依赖，则可获得 2NF 关系
3. 在 2NF 中，允许主属性部分函数依赖于码

■ 第三范式

如果关系模式 R 满足第二范式，且每个非主属性都不传递依赖于 R 的任何码，称 R 满足第三范式，简记为 3NF。

● 关系模型设计

数据库设计是对数据进行组织化和结构化的过程，即关系模型的设计。

对于项目规模小、用户数量少的情况，处理数据库中的表结构相对轻松；但随着项目的发展，相应的数据库架构、关系模型表结构越来越复杂，之前写的 SQL 语句是很笨拙并且效率低下的。更可怕的是，由于表结构定义不合理，会导致对数据的增删改查不方便不高效；最致命的是，扩展性极差，不能应对业务的变化。此时对我们计算机专业学生关系模式设计能力要求提高了，我们要学习和掌握数据库的规范化流程，以指导我们更好的设计数据库的表结构，减少冗余的数据，借此可以提高数据库的存储效率，数据完整性和可扩展性。

简洁、结构明晰的表结构对数据库的设计是相当重要的。规范化的表结构设计，在以后的数据维护中，不会发生插入（insert）、删除（delete）和更新（update）时的异常。反之，数据库表结构设计不合理，不仅会给数据库的使用和维护带来各种各样的问题，而且可能存储了大量不需要的冗余信息，浪费系统资源。

关于数据库关系模型设计的问题，就是在实际项目中，我们应该构造几个关系模型，每个关系（表）由哪些属性（列）组成，不同关系之间有什么关联。

■ E-R 图向关系模型的转换规则

联系转换成为关系模型时，要根据联系方式的不同采用不同的转换方式。

1. 1:1 联系的转换方法
   a) 将 1:1 联系转换为一个独立的关系：与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，且每个实体的码均是该关系的候选码
   b) 将 1:1 联系与某一端实体集所对应的关系合并，则需要在被合并关系中增加属性，其新增的属性为联系本身的属性与联系相关的另一个实体集的码
2. 1:n 联系的转换方法
   a) 一种方法是将联系转换为一个独立的关系，其关系的属性由与该联系相连的各实体集的码以及联系本身的属性组成，而该关系的码为 n 端实体集的码
   b) 另一种方法是在 n 端实体集中增加新的属性，新属性由联系对应的 1 端实体集的码和联系自身的属性构成，新增属性后原关系的码不变
3. m:n 联系的转换方法
   在向关系模型转换时，一个 m:n 联系转换为一个关系。转换方法为：与该联系相连的各实体集的码以及联系本身的属性均转换为关系的属性，新关系的码为两个相连实体码的组合（该码为多属性构成的组合码）。

■ E-R 图向关系模型的转换

根据概念设计中的 E-R 图，可知有如下几种联系：

1. 1:1 联系：学生与退宿、学生与离校
2. 1:n 联系：宿舍与学生、宿舍管理员与宿舍、宿舍管理员与卫生检查、学生与来访人员、学生与宿舍财产

故据此可以得到具体的 E-R 图向关系模型的转换结果。

1. student 表：（Sno，Sname，Sgender，Sdept，Scheckin，Dno）
2. dorm 表（Dno，Dphone，Pno）
3. repair 表（Rsubmit，Rsolve，Rreason，Dno，Pno）
4. check 表（ChTime，ChResult，Pno，Dno）
5. fee 表（Fcost，Fleftfee，Fctfee，Dno）
6. visitor 表（Vname，Vgender，Vin，Vout，Sno）
7. leave 表（Lout，Lin，Sno）
8. quit 表（Qout，Qreason，Sno）
9. manager 表（Mno，Mpassword）
10. possession 表（Pno，Pname，Sno）

● 表结构设计

■ 表结构设计原则

1. 应该根据系统结构中的组件划分，针对每个组件所处理的业务进行组件单元的数据库设计，而不是针对整个系统进行数据库设计
2. 不同组件间所对应的数据库表之间的关联应该尽可能减少，为系统或表结构的重构提供可能性
3. 采用领域模型驱动的方式和自顶向下的思路进行数据库设计，首先分析系业务，根据职责定义对象
4. 一个表中的所有非关键字属性都依赖于整个关键字。关键字可以是一个属性，也可以是多个属性的集合，总之应该确保关键字能够保证唯一性
5. 应针对所以表的主键和外键建立索引，有针对性得建立组合属性的索引，提高检索效率

■ 表结构设计实现

表 2.1 住宿学生表

student

属性名	字段	类型	长度	约束
学号	Sno	integer		主键，not null

姓名	Sname	varchar2	20	not null
性别	Sgender	varchar2	4	‘female’或‘male’，not null
专业	Sdept	varchar2	40	not null
宿舍号	Dno	varchar2	10	外键，not null
入住时间	Scheckin	date	8	not null

表 2.2 学生宿舍表

dorm

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	主键，not null
宿舍电话	Dphone	varchar2	15	not null
物品编号	Pno	integer		外键，not null

表 2.3 报修信息表

repair

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
提交日期	Rsubmit	date	8	not null
解决日期	Rsolve	date	8	not null
报修原因	Rreason	varchar2	50	not null
物品编号	Pno	integer		外键，not null

表 2.4 卫生检查信息表

check

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
检查时间	ChTime	date	8	not null
检查结果	ChResult	varchar2	10	not null
物品编号	Pno	integer		外键，not null

表 2.5 水电信息表

fee

属性名	字段	类型	长度	约束
宿舍号	Dno	varchar2	10	外键，not null
已用费用	Fcost	integer		not null
剩余费用	Fleftfee	integer		not null
续交费用	Fctfee	integer		not null

表 2.6 来访人员表

visitor

属性名	字段	类型	长度	约束
姓名	Vname	varchar2	20	not null
性别	Vgender	varchar2	4	‘female’或‘male’，not null
进入时间	Vin	timestamp	20	not null
离开时间	Vout	timestamp	20	not null
学号	Sno	integer		外键，not null

表 2.7 离校表

leave

属性名	字段	类型	长度	约束
学号	Sno	integer		外键，not null
离校时间	Lout	timestamp	20	not null
返校时间	Lin	timestamp	20	not null

表 2.8 退宿表

quit

属性名	字段	类型	长度	约束
学号	Sno	integer		外键，not null
退宿时间	Qout	time	8	not null
退宿原因	Qreason	varchar2	40	not null

表 2.9 宿舍管理员表

manager

属性名	字段	类型	长度	约束
管理员 ID	Mno	integer		主键，not null
管理员密码	Mpassword	varchar2	20	not null

表 2.10 宿舍财产表

possession

属性名	字段	类型	长度	约束
物品编号	Pno	integer		主键，not null
物品名	Pname	varchar2	20	not null
学号	Sno	integer		外键，not null