

《数据库设计》实验报告

学 号： 1005183121姓 名： 周子杰日 期： 2021. 7. 2

得 分： _____

一、 实验内容

- ◇ 数据库设计
- ◇ 数据仓库

二、 知识整理

数据库三个世界

- 现实世界 / 概念设计
- 信息世界
- 数据世界

数据库设计

- 概念设计
 - 概念
 - 概念结构设计是整个数据库设计的关键,它通过对用户需求进行综合,归纳与抽象,形成了一个独立于具体 DBMS 的概念模型。
 - 三要素
 - 实体
 - 属性
 - 关系
 - E-R 图

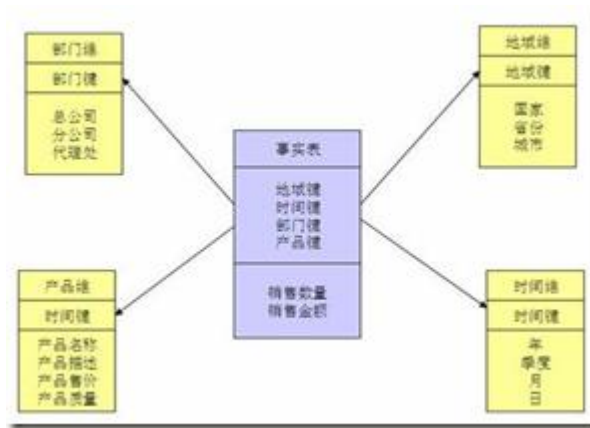
- 工具
 - starUML
 - processon.com（一个在线网站）
 - word
 - ERWin
 - Power Designer
 - ER/Studio
- 逻辑设计
 - 概念
 - 逻辑结构设计是将概念结构转换为某个 DBMS 所支持的数据模型，并将进行优化。
 - 三大模型
 - 层次模型：树形结构
 - 网状模型：允许多个/无双亲结点+复合联系
 - 关系模型：二维表（表：关系、行：元组、列：属性、主码、域、分量、关系模式）
- 物理设计
 - 概念
 - 物理设计是为逻辑数据结构模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）。
 - 五大约束
 - 主键约束（Primay Key Coustraint）唯一性，非空性；
 - 唯一约束（Unique Counstraint）唯一性，可以空，但只能有一个；
 - 缺省约束（Default Counstraint）该数据的默认值；
 - 外键约束（Foreign Key Counstraint）需要建立两表间的关系；
 - 空值约束（Not Null Counstraint）设置非空约束，该字段不能为空。

- 范式
 - 1NF
 - 确保每列保持原子性（属性不可分）
 - 2NF
 - ，且每一个非主属性完全依赖于任何一个候选码
 - 消除非主属性对码的部分函数依赖
 - 3NF
 - R 中不存在这样的码 X, 属性组 Y 以及非属性组 Z, 使得 $X \rightarrow Y$, $Y \rightarrow Z$ 成立
 - 消除非主属性对码的传递函数依赖
 - BCNF
 - $X \rightarrow Y$ （非平凡的）时 X 必含有码
 - 每个决定因素（ $X \rightarrow Y$ 中的 X）必包含码
 - 消除主属性对码的部分和传递函数依赖
 - 4NF
 - R 的每个非平凡多值依赖 $X \twoheadrightarrow Y$ 都含有候选码
 - 消除非平凡且非函数的多值依赖
 - 5NF
- 操作对象
 - 触发器对象等
- 评价
 - 存储空间、存取时间、维护代价

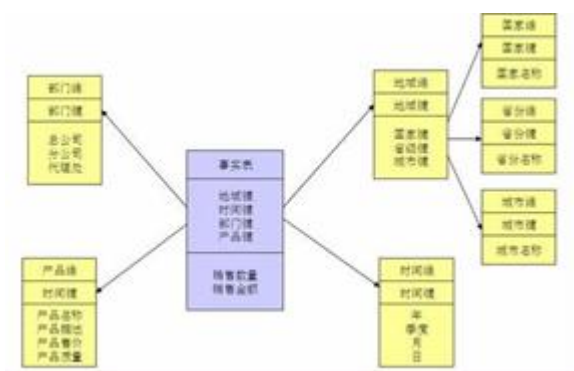
数据仓库

- 三个模式（结构大小、规模角度）
 - 企业仓库
 - 最复杂
 - 数据集市
 - 相对较小

- 虚拟仓库
 - 范畴更小
- 三种模式（系统角度）
 - 星型结构 / 星型模式
 - 一个庞大的事实表+若干个维度表
 - 描述：星型模型是数据集市维度建模中推荐的建模方法。星型模型是以事实表为中心，所有的维度表直接连接在事实表上，像星星一样。星型模型的特点是数据组织直观，执行效率高。因为在数据集市的建设过程中，数据经过了预处理，比如按照维度进行了汇总，排序等等，数据量减少，执行的效率就比较高。

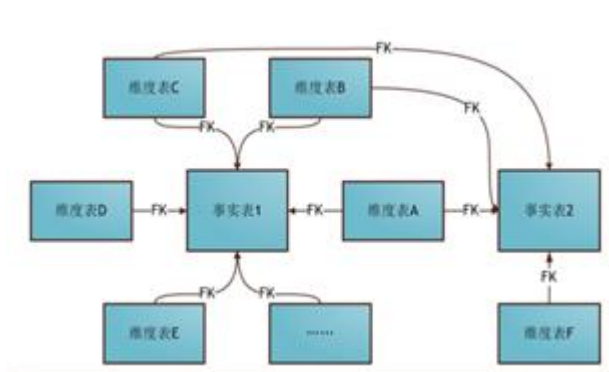


- 雪花模式
 - 是星型模式的扩展
 - 对维度表进行细分、降阶
 - 描述：雪花模型也是维度建模中的一种选择。雪花模型的维度表可以拥有其他维度表的，虽然这种模型相比星型模型更规范一些，但是由于这种模型不太容易理解，维护成本比较高，而且性能方面需要关联多层维表，性能也比星型模型要低。所以一般不是很常用。



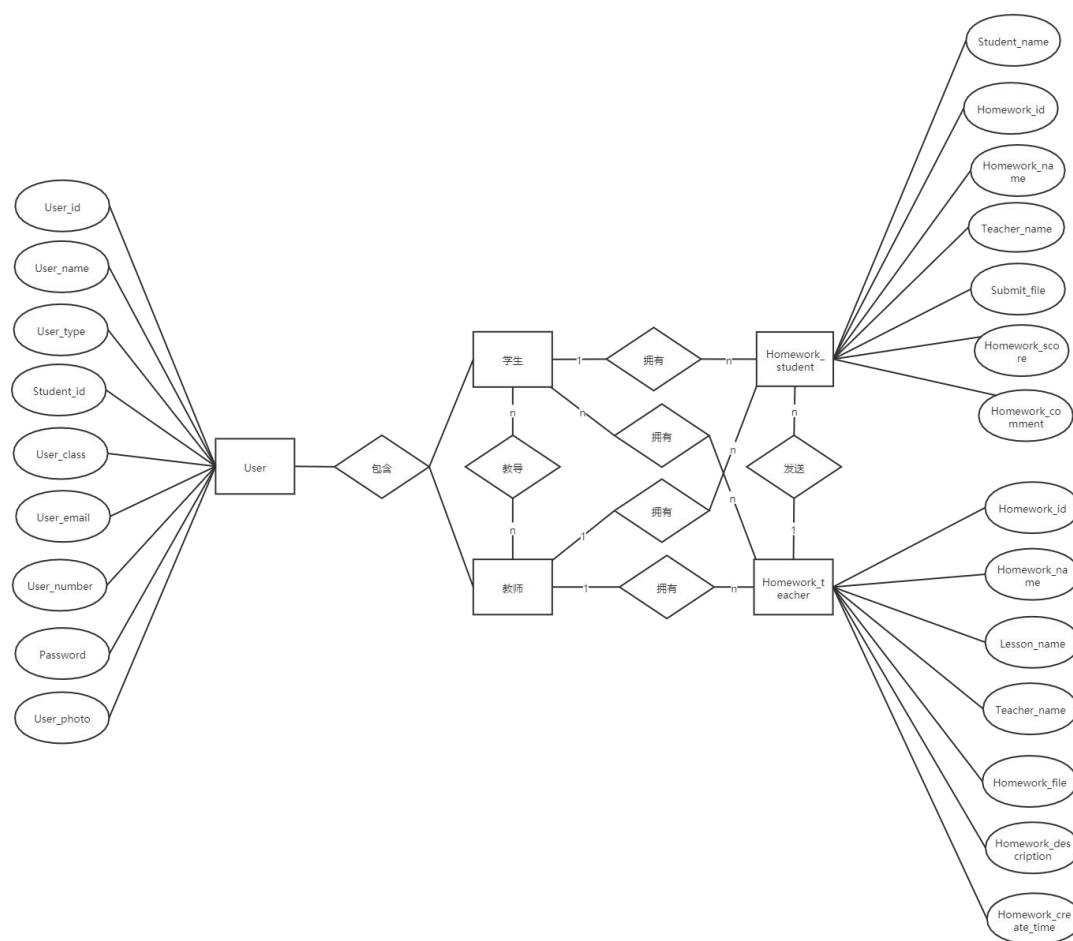
事实星际模式

- 多个事实表对应多个维度表
- 对事实表和维度表都进行细化
- 描述：星座模型是星型模型延伸而来，星型模型是基于一张事实表的，而星座模型是基于多张事实表的，而且共享维度信息。通过构建一致性维度，来建设星座模型，也是很好的选择。比如同一主题的细节表和汇总表共享维度，不同主题的事实表，可以通过在维度上互相补充来生成可以共享的维度。



三、 实践操作

概念设计（E-R 图设计）：



E-R 图

逻辑设计（E-R 图转换为关系表）：**实体：**

- 1) 用户：包括教师和学生，在数据库中以 User_type 区分，其包含字段可参照数据字典。
- 2) 教师作业：教师发布的作业，其包含字段参照数据字典。
- 3) 学生作业：学生接受的作业，其包含字段参照数据字典。

关系：

- 1) 学生和教师的关系为多对多 $n:n$ 。一个学生可以拥有多个教师，一个教师也可以拥有多个学生。
- 2) 学生和学生作业的关系为一对多 $1:n$ 。一个学生可以拥有多个学生作业，但一个学生作业智能属于一个学生。

- 3) 学生和教师作业属于多对多 $n:n$ 。一个学生可以拥有多个教师作业，一个教师作业也可以属于多个学生。
- 4) 教师和学生作业属于一对多 $1:n$ 。一个教师可以拥有多个学生作业，但一个学生作业只能属于一个教师。
- 5) 教师 and 教师作业属于一对多 $1:n$ 。一个教师可以拥有多个教师作业，但一个教师作业只能属于一个教师。
- 6) 教师作业和学生作业属于一对多 $1:n$ 。一个教师作业可以发送给多个学生产生多个学生作业，但一个学生作业只能来源于一个教师作业。

属性:

User 表

序号	字段名	字符数据类型	字段描述
1	User_id	string	系统分配的 id, 主键
2	User_name	String	名字
3	User_type	Int	用户类型
4	User_class	Int	学术需要填写班级
5	Student_id	String	学号
6	User_email	String	电子邮件
7	User_number	string	登录用的用户名
8	Password	string	密码
9	User_photo	file	头像

对于 User 表而言，其主键为 User_id，需要非空和唯一。同时，User_name 需要唯一。

Homework_student 表

序号	字段名	字符数据类型	字段描述
1	Student_name	String	学生名字
2	Homework_id	Int	作业 id, 主键

3	Homework_name	String	作业名字
4	Teacher_name	String	批阅老师名字
5	Submit_file	file	文件
6	Homework_score	Double	得分
7	Homework_comment	string	评语

对于 Homework_student 表而言，其主键为 Homework_id，需要非空和唯一。
Student_name 和 Teacher_name 需要唯一。

Homework_teacher 表

序号	字段名	字符数据类型	字段描述
1	Homework_id	Int	分配的 id，主键
2	Homework_name	String	作业名字
3	Lesson_name	String	所属课程
4	Teacher_name	String	布置作业的老师名字
5	Homework_file	File	文件
6	Homework_description	String	描述
7	Homework_create_time	Data	创建时间

对于 Homework_teacher 表而言，其主键为 Homework_id，需要非空和唯一。
Teacher_name 唯一。

物理设计：

环境：

系统：Windows 10

数据库：MySQL 5.5.57

建表：

```
1. CREATE DATABASE xiaoxueqi_test;
2. USE `xiaoxueqi_test`;
3.
```



```
4. CREATE TABLE `homework_teacher` (  
5.   `Homework_id` INT(11) NOT NULL AUTO_INCREMENT,  
6.   `Homework_name` VARCHAR(30) NOT NULL,  
7.   `Lesson_name` VARCHAR(30) NOT NULL,  
8.   `Teacher_name` VARCHAR(30) NOT NULL UNIQUE,  
9.   `Homework_file` VARCHAR(100) DEFAULT NULL,  
10.  `Homework_description` VARCHAR(100) DEFAULT NULL,  
11.  `Homework_create_time` DATE DEFAULT NULL,  
12.  PRIMARY KEY (`Homework_id`)  
13. ) ENGINE=INNODB DEFAULT CHARSET=utf8;  
14.  
15. CREATE TABLE `homework_student` (  
16.   `Student_name` VARCHAR(30) DEFAULT NULL UNIQUE,  
17.   `Homework_id` INT(11) DEFAULT NULL UNIQUE,  
18.   `Homework_name` VARCHAR(30) DEFAULT NULL,  
19.   `Teacher_name` VARCHAR(30) DEFAULT NULL,  
20.   `Submit_file` VARCHAR(50) DEFAULT NULL,  
21.   `Homework_score` DOUBLE DEFAULT NULL,  
22.   `Homework_comment` VARCHAR(50) DEFAULT NULL  
23. ) ENGINE=INNODB DEFAULT CHARSET=utf8;  
24.  
25. CREATE TABLE `user` (  
26.   `User_id` INT(11) NOT NULL AUTO_INCREMENT UNIQUE,  
27.   `User_name` VARCHAR(30) DEFAULT NULL UNIQUE,  
28.   `User_type` INT(11) DEFAULT NULL,  
29.   `User_class` INT(11) DEFAULT NULL,  
30.   `Student_id` INT(11) DEFAULT NULL UNIQUE,  
31.   `User_email` VARCHAR(30) DEFAULT NULL,  
32.   `Password` VARCHAR(30) DEFAULT NULL,  
33.   `User_photo` VARCHAR(100) DEFAULT NULL,  
34.   PRIMARY KEY (`User_id`)  
35. ) ENGINE=INNODB DEFAULT CHARSET=utf8;
```

视图对象:

```
1. CREATE VIEW StudentWithHomeworkA  
2. AS  
3. SELECT Student_id, User_name  
4. FROM `homework_student`, `user`  
5. WHERE `homework_student`.`Student_name` = `user`.`User_name`  
6. AND `homework_student`.`Homework_name` = 'A';
```

为有作业 A 的学生建立一个视图。

触发器对象：

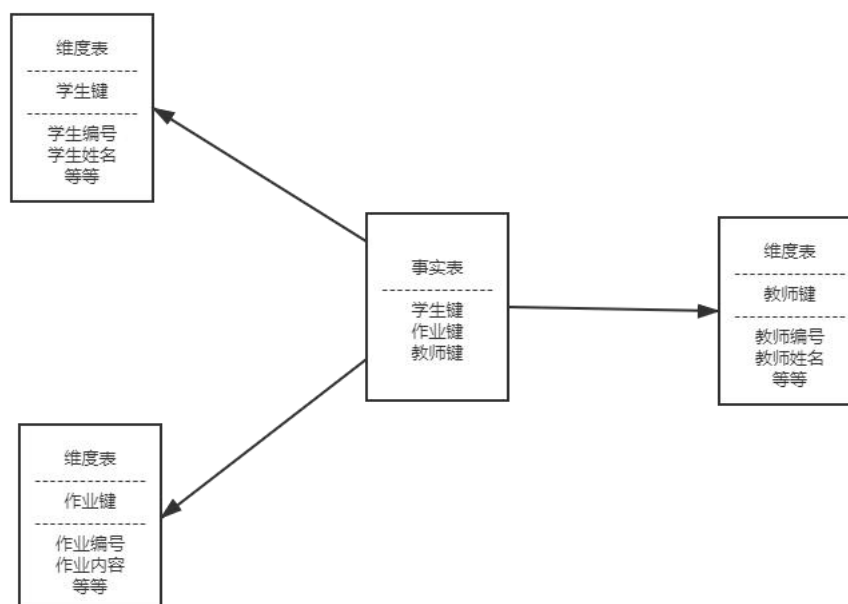
```

1.  -- 每次在 User 表中增加类型为学生的成员时统计学生个数到 Stu_num 中
2.  CREATE TABLE Stu_num(
3.    num INT
4.  );
5.  CREATE TRIGGER Student_number
6.  AFTER INSERT ON `user`
7.  REFERENCING
8.    NEW TABLE AS Stu
9.  FOR EACH statement
10.   INSERT INTO Stu_num(num)
11.   SELECT COUNT(*) FROM Stu
12.   WHERE Stu.type = 1;

```

每次加入学生时统计学生人数，并将结果加入到 Stu_num 表中去

数据仓库：



星型模式

这是一个简单的数据仓库设计，采用的是含有一个事实表和三个维度表的星型模式。

四、 心得体会

本次授课主要分为两部分，理论知识学习和实践操作环节。

通过对关于数据库设计的资料的查找，我发现自己在基础知识上的不足，很感谢老师提供这次机会让我能够对之前所学加深印象与理解。同时，也对数据仓库有了一定的了解。

在实践上，我完成了数据库设计中的概念设计、逻辑设计和物理设计，也简单设计了一个星型模式的数据仓库。总得来说，还是收获颇多的。