# JavaWeb

## day03\_mysql约束

### 0.1学习目标

1.能够完成DQL操作

2.能够使用非空、唯一、主键、外键约束

3.能够理解多表关系

4.对数据库三大范式有一定了解

5.能够完成数据库的备份与还原

### 0.2 学习指南

DQL操作（重要）

使用约束（重要）

理解多表关系（重要）

数据库三大范式（了解）

数据库的备份与还原（重要）

### 第一堂课

本节知识点

1.1 今日内容

1.2 DQL排序查询

1.3 DQL聚合函数

1.4 DQL分组查询

本节目标

完成DQL排序查询、DQL聚合函数、DQL分组查询等操作

#### 今日内容介绍

##### 1.1.1知识概述

今天的内容介绍

##### 1.1.2视频详情



##### 1.1.3总结与补充

无

#### DQL排序查询

##### 1.2.1知识概述

语法：order by 子句  
            \* order by 排序字段1 排序方式1 ，  排序字段2 排序方式2...  
  
        \* 排序方式：  
            \* ASC：升序，默认的。  
            \* DESC：降序

##### 1.2.2视频详情



##### 1.2.3总结与补充

注意：  
            \* 如果有多个排序条件，则当前边的条件值一样时，才会判断第二条件。

##### 1.2.4课堂提问与练习

无

##### 1.2.5习题答案

无

##### 1.2.6视频缺陷

无

##### 1.2.7视频扩展

无

#### DQL聚合函数

##### 1.3.1知识概述

聚合函数对一组值执行计算并返回单一的值。除了 COUNT 以外，聚合函数忽略[空值](https://baike.baidu.com/item/%E7%A9%BA%E5%80%BC)。聚合函数经常与 SELECT 语句的 GROUP BY 子句一同使用。

聚合函数：将一列数据作为一个整体，进行纵向的计算。  
        1. count：计算个数  
            1. 一般选择非空的列：主键  
            2. count(\*)  
        2. max：计算最大值  
        3. min：计算最小值  
        4. sum：计算和  
        5. avg：计算平均值

##### 1.3.2视频详情



##### 1.3.3总结与补充

聚合函数的作用是为了方便程序员对某一列数据的操作，需要注意null值。

##### 1.3.4课堂提问与练习

无

##### 1.3.5习题答案

无

##### 1.3.6视频缺陷

无

##### 1.3.7视频扩展

无

#### DQL分组查询

##### 1.4.1知识概述

GROUP BY关键字可以将查询结果按照某个字段或多个字段进行分组。字段中值相等的为一组。基本的语法格式如下：

GROUP BY 属性名 [HAVING 条件表达式] [WITH ROLLUP]

* 属性名：是指按照该字段的值进行分组。
* HAVING 条件表达式：用来限制分组后的显示，符合条件表达式的结果将被显示。
* WITH ROLLUP：将会在所有记录的最后加上一条记录。加上的这一条记录是上面所有记录的总和。

1. 语法：group by 分组字段；  
        2. 注意：  
            1. 分组之后查询的字段：分组字段、聚合函数  
            2. where 和 having 的区别？  
                1. where 在分组之前进行限定，如果不满足条件，则不参与分组。having在分组之后进行限定，如果不满足结果，则不会被查询出来  
                2. where 后不可以跟聚合函数，having可以进行聚合函数的判断。

##### 1.4.2视频详情



##### 1.4.3总结与补充

ROUP BY关键字可以和GROUP\_CONCAT()函数一起使用。GROUP\_CONCAT()函数会把每个分组中指定的字段值都显示出来。

同时，GROUP BY关键字通常与集合函数一起使用。集合函数包括COUNT()函数、SUM()函数、AVG()函数、MAX()函数和MIN()函数等。

sql语句的执行过程是：from-->where-->group by -->having --- >order by --> select;

其执行顺序为：FROM-WHERE-GROUP BY-HAVING-SELECT-DISTINCT-UNION-ORDER BY

Mysql执行顺序：开始->FROM子句->WHERE子句->GROUP BY子句->HAVING子句->ORDER BY子句->SELECT子句->LIMIT子句->最终结果

聚合函数是针对结果集进行的，但是where条件并不是在查询出结果集之后运行，所以主函数放在where语句中，会出现错误，

而having不一样，having是针对结果集做筛选的，所以我门一般吧组函数放在having中，用having来代替where，having一般跟在group by后

##### 1.4.4课堂提问与练习

* 练习**DQL排序、聚合函数、分组等操作**

##### 1.4.5习题与答案

参考代码

##### 1.4.6视频缺陷

无

##### 1.4.7视频扩展

无

### 第二堂课

本节知识点：

2.1 DQL分页查询

2.2 约束的概述

2.3 约束\_非空约束

2.4 约束\_唯一约束

目标：

1、能够完成DQL分页查询

2、能够理解约束的相关概念

3、能够掌握非空约束与唯一约束

#### DQL\_分页查询

##### 2.1.1知识概述

Mysql分页关键字为limit。

为什么会用到分页呢？

因为列表内容太多了，所以使用分页进行显示。

##### 2.1.2视频详情



##### 2.1.3总结与补充

不同的数据库有不同的分页关键字和语法。

例如：mysql用的是limit，oracle用的是rownum。

##### 2.1.4课堂提问与练习

无

##### 2.1.5习题答案

无

##### 2.1.6视频缺陷

无

##### 2.1.7视频扩展

无

#### 2.2约束概述

##### 2.2.1知识概述

对表中的数据进行限定，保证数据的正确性、有效性和完整性。

##### 2.2.2视频详情



##### 2.2.3总结与补充

主键约束（Primary Key constraint）：要求主键列数据唯一，并且不允许为空。

[唯一约束](https://baike.baidu.com/item/%E5%94%AF%E4%B8%80%E7%BA%A6%E6%9D%9F/1976604)（Unique constraint）：要求该列唯一，允许为空，但只能出现一个[空值](https://baike.baidu.com/item/%E7%A9%BA%E5%80%BC/6506427)。

[检查约束](https://baike.baidu.com/item/%E6%A3%80%E6%9F%A5%E7%BA%A6%E6%9D%9F/10272981)（Check constraint）：某列取值范围限制，格式限制等，如有关年龄、邮箱（必须有@）的约束。

默认约束（Default constraint）：某列的默认值，如在数据库里有一项数据很多重复，可以设为默认值。

[外键](https://baike.baidu.com/item/%E5%A4%96%E9%94%AE/1232333)约束（Foreign Key constraint）：用于在两个表之间建立关系，需要指定引用主表的哪一列。

##### 2.2.4课堂提问与练习

无

##### 2.2.5习题答案

无

##### 2.2.6视频缺陷

无

##### 2.2.7视频扩展

无

#### 2.3 约束\_非空约束

##### 2.3.1知识概述

not null，某一列的值不能为null

1. 创建表时添加约束  
            CREATE TABLE stu(  
                id INT,  
                NAME VARCHAR(20) NOT NULL -- name为非空  
            );

 2. 创建表完后，添加非空约束  
            ALTER TABLE stu MODIFY NAME VARCHAR(20) NOT NULL;  
  
  3. 删除name的非空约束  
            ALTER TABLE stu MODIFY NAME VARCHAR(20);

##### 2.3.2视频详情



##### 2.3.3总结与补充

在实际应用中，根据实际情况来确定是否使用非空约束。

例如：注册用户时，用户名和密码必须不为null

##### 2.3.4课堂提问与练习

无

##### 2.3.5习题答案

无

##### 2.3.6视频缺陷

无

##### 2.3.7视频扩展

无

#### 2.4 约束\_唯一约束

##### 2.4.1知识概述

unique，某一列的值不能重复。注意：null值可以有多个。

唯一约束：unique，某一列的值不能重复  
        1. 注意：  
            \* 唯一约束可以有NULL值

        2. 在创建表时，添加唯一约束  
            CREATE TABLE stu(  
                id INT,  
                phone\_number VARCHAR(20) UNIQUE -- 手机号  
            );  
        3. 删除唯一约束  
            ALTER TABLE stu DROP INDEX phone\_number;  
        4. 在表创建完后，添加唯一约束  
            ALTER TABLE stu MODIFY phone\_number VARCHAR(20) UNIQUE;

##### 2.4.2视频详情



##### 2.4.3总结与补充

唯一约束在实际应用中，也要根据实际情况来进行使用。

例如：

注册账户时需要用到手机号，每个手机号只能注册一个账号，这时就要用到唯一约束。

##### 2.4.4课堂提问与练习

* 练习非空约束、唯一约束

##### 2.4.5习题答案

参考代码

##### 2.4.6视频缺陷

无

##### 2.4.7视频扩展

无

### 第三堂课

本节知识点

3.1 约束\_主键约束

3.2 约束\_主键约束\_自动增长

3.3 约束\_外键约束

本节目标

1. 掌握主键约束及自动增长
2. 掌握外键约束

#### 3.1约束\_主键约束

##### 3.1.1知识概述

primary key，非空并且唯一，一张表只能有一个主键，主键是表的唯一标识

主键约束：primary key。  
        1. 注意：  
            1. 含义：非空且唯一  
            2. 一张表只能有一个字段为主键  
            3. 主键就是表中记录的唯一标识  
  
        2. 在创建表时，添加主键约束  
            create table stu(  
                id int primary key,-- 给id添加主键约束  
                name varchar(20)  
            );  
  
        3. 删除主键  
            -- 错误 alter table stu modify id int ;  
            ALTER TABLE stu DROP PRIMARY KEY;  
  
        4. 创建完表后，添加主键  
            ALTER TABLE stu MODIFY id INT PRIMARY KEY;

##### 3.1.2视频详情



##### 3.1.3总结与补充

主键是一个表中能标识唯一行的标志（也有其他方法表示唯一行，如唯一列）。

主键主要用在查询单调数据，修改单调数据和删除单调数据上。

##### 3.1.4课堂提问与练习

主键约束的特点

##### 3.1.5习题答案

非空，唯一

##### 3.1.6视频缺陷

无

##### 3.1.7视频扩展

无

#### 3.2约束\_主键约束\_自动增长

##### 3.2.1知识概述

自动增长：  
            1.  概念：如果某一列是数值类型的，使用 auto\_increment 可以来完成值得自动增长  
  
            2. 在创建表时，添加主键约束，并且完成主键自增长  
            create table stu(  
                id int primary key auto\_increment,-- 给id添加主键约束  
                name varchar(20)  
            );  
  
              
            3. 删除自动增长  
            ALTER TABLE stu MODIFY id INT;  
            4. 添加自动增长  
            ALTER TABLE stu MODIFY id INT AUTO\_INCREMENT;

##### 3.2.2视频详情



##### 3.2.3总结与补充

自增主键

这种方式是使用数据库提供的自增数值型字段作为自增主键，它的优点是：

（1）数据库自动编号，速度快，而且是增量增长，按顺序存放，对于检索非常有利；

（2）数字型，占用空间小，易排序，在程序中传递也方便；

（3）如果通过非系统增加记录时，可以不用指定该字段，不用担心主键重复问题。

其实它的缺点也就是来自其优点，缺点如下：

（1）因为自动增长，在手动要插入指定ID的记录时会显得麻烦，尤其是当系统与其它系统集成时，需要数据导入时，很难保证原系统的ID不发生主键冲突（前提是老系统也是数字型的）。特别是在新系统上线时，新旧系统并行存在，并且是异库异构的数据库的情况下，需要双向同步时，自增主键将是你的噩梦；

（2）在系统集成或割接时，如果新旧系统主键不同是数字型就会导致修改主键数据类型，这也会导致其它有外键关联的表的修改，后果同样很严重；

（3）若系统也是数字型的，在导入时，为了区分新老数据，可能想在老数据主键前统一加一个字符标识（例如“o”，old）来表示这是老数据，那么自动增长的数字型又面临一个挑战。

##### 3.2.4课堂提问与练习

无

##### 3.2.5习题答案

无

##### 3.2.6视频缺陷

无

##### 3.2.7视频扩展

无

#### 3.3约束\_外键约束

##### 3.3.1知识概述

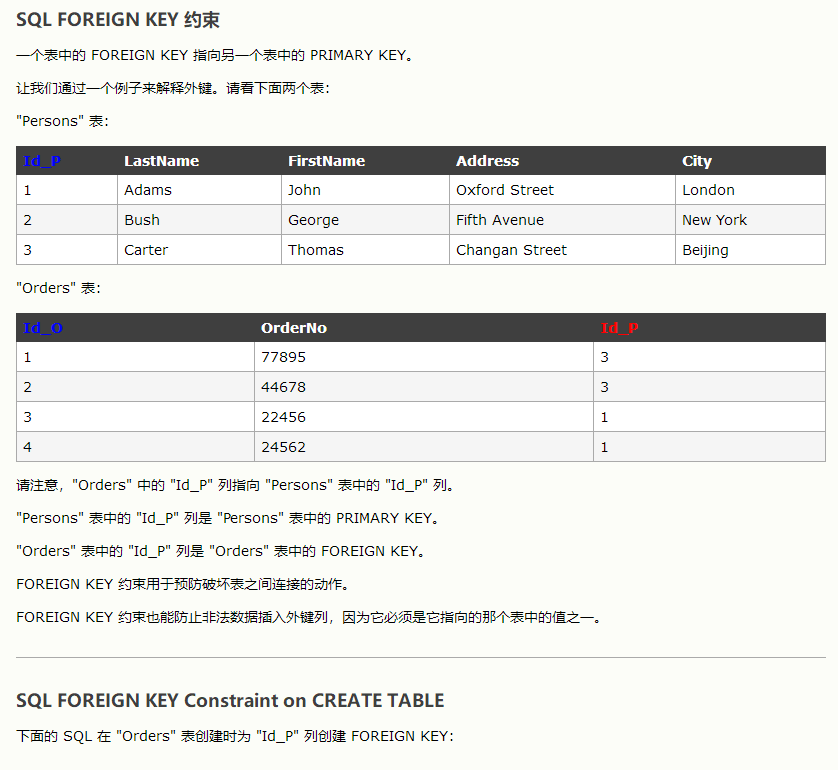
foreign key,让表于表产生关系，从而保证数据的正确性。

外键约束：foreign key,让表于表产生关系，从而保证数据的正确性。  
        1. 在创建表时，可以添加外键  
            \* 语法：  
                create table 表名(  
                    ....  
                    外键列  
                    constraint 外键名称 foreign key (外键列名称) references 主表名称(主表列名称)  
                );  
  
        2. 删除外键  
            ALTER TABLE 表名 DROP FOREIGN KEY 外键名称;  
  
        3. 创建表之后，添加外键  
            ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段名称) REFERENCES 主表名称(主表列名称);

##### 3.3.2视频详情



##### 3.3.3总结与补充



##### 3.3.4课堂提问与练习

* 练习主键约束及自动增长、外键约束

##### 3.3.5习题答案

无

##### 3.3.6视频缺陷

无

##### 3.3.7视频扩展

无

### 第四堂课

本节知识点：

4.1 约束\_外键约束\_级联操作

4.2多表关系的介绍及实现

本节目标 ：

能够完成约束\_外键约束\_级联操作。

能够理解并实现多表关系。

#### 约束\_外键约束\_级联操作

##### 4.1.1知识概述

A表与B表有外键约束，A中有外键，B中数据改变时，A中的外键字段对应做出改变。

1. 添加级联操作  
                语法：ALTER TABLE 表名 ADD CONSTRAINT 外键名称   
                        FOREIGN KEY (外键字段名称) REFERENCES 主表名称(主表列名称) ON UPDATE CASCADE ON DELETE CASCADE  ;  
            2. 分类：  
                1. 级联更新：ON UPDATE CASCADE   
                2. 级联删除：ON DELETE CASCADE

##### 4.1.2视频详情



##### 4.1.3总结与补充

级联操作有四种类型：级联更新（合并），级联保存，级联刷新，和级联删除。

详情：现在我们分别介绍以上四种级联类型的详细情况。

1.CascadeType.REFRESH 级联刷新（使用场景，当我们从数据库中请求了一条数据，在我们对这条数据进行业务处理的时候，另一个用户可能会修改数据库中的这条数据的记录，那么此时我们就需要进行级联刷新了，否则我们处理的数据就失去了意义。）

2.CascadeType.PERSIST 级联持久化（比如在发出保存订单操作时，会同时发出保存订单项的操作语句）   \*订单和订单项之间是一对多的关系。

3.CascadeType.MERGE 级联合并 也就是级联更新（当处于游离状态下的对象被修改了，那么与他相关联的实体中的对象也会执行修改，如果开启了级联更新，那么此实体对象的更新会波及到与其关联的实体对象）

4.CascadeType.REMOVE 级联删除 （比如：如果设置了级联删除，当删除订单时，会同时删除对应的订单项，但是需要注意此操作的先后顺序为先删除订单项，再删除订单）

注：即使我们定义了级联操作，也只有在我们使用EntityManage 对象的api方法时才会生效。比如说当我们调用了refresh()；方法时，级联刷新才会生效。当调用了remove()方法时，级联删除才会生效。而使用HQL语句进行删除，则不会触发级联删除操作。

##### 4.1.4课堂提问与练习

无

##### 4.1.5习题答案

无

##### 4.1.6视频缺陷

无

##### 4.1.7视频扩展

无

#### 多表关系介绍

##### 4.2.1知识概述

1. 一对一(了解)：  
                \* 如：人和身份证  
                \* 分析：一个人只有一个身份证，一个身份证只能对应一个人  
            2. 一对多(多对一)：  
                \* 如：部门和员工  
                \* 分析：一个部门有多个员工，一个员工只能对应一个部门  
            3. 多对多：  
                \* 如：学生和课程  
                \* 分析：一个学生可以选择很多门课程，一个课程也可以被很多学生选择

##### 4.2.2视频详情



##### 4.2.3总结与补充

确定表与表之间的关系，必须根据项目中的业务逻辑来设置。

##### 4.2.4课堂提问与练习

多表关系有哪几种？

。

##### 4.2.5习题答案

一对多，多对多，一对一

##### 4.2.6视频缺陷

无

##### 4.2.7视频扩展

无

#### 4.3多表关系\_一对多关系实现

##### 4.3.1知识概述

 一对多(多对一)：  
                \* 如：部门和员工  
                \* 实现方式：在多的一方建立外键，指向一的一方的主键。

##### 4.3.2视频详情



##### 4.3.3总结与补充

在多的一方设置外键，便于维护。一个部门可以有多个员工。

##### 4.3.4课堂提问与练习

无

。

##### 4.3.5习题答案

无

##### 4.3.6视频缺陷

无

##### 4.3.7视频扩展

无

#### 4.4多表关系\_多对多关系实现

##### 4.4.1知识概述

 多对多：  
                \* 如：学生和课程  
                \* 实现方式：多对多关系实现需要借助第三张中间表。中间表至少包含两个字段，这两个字段作为第三张表的外键，分别指向两张表的主键

##### 4.4.2视频详情



##### 4.4.3总结与补充

需要创建一张中间表来维护两张主表之间的关系，并且中间表最少要有两个字段。

##### 4.4.4课堂提问与练习

无

。

##### 4.4.5习题答案

无

##### 4.4.6视频缺陷

无

##### 4.4.7视频扩展

无

### 第五堂课

本节知识点：

5.1 多表关系\_一对一关系实现

5.2 多表关系案例

5.3 范式的相关概念

本节目标 ：

实现多表关系案例

了解范式的相关概念

#### 多表关系\_一对一关系实现

##### 5.1.1知识概述

一对一：  
                \* 如：人和身份证  
                \* 实现方式：一对一关系实现，可以在任意一方添加唯一外键指向另一方的主键。

##### 5.1.2视频详情



##### 5.1.3总结与补充

例如一个人只能有一个身份证号。在一张表可以实现。

##### 5.1.4课堂提问与练习

无

##### 5.1.5习题答

无

##### 5.1.6视频缺陷

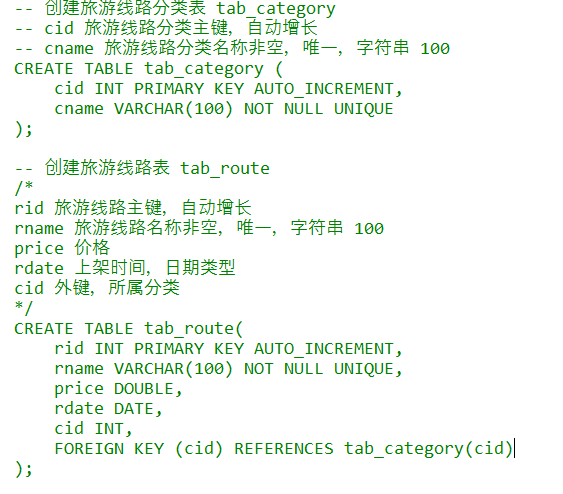
无

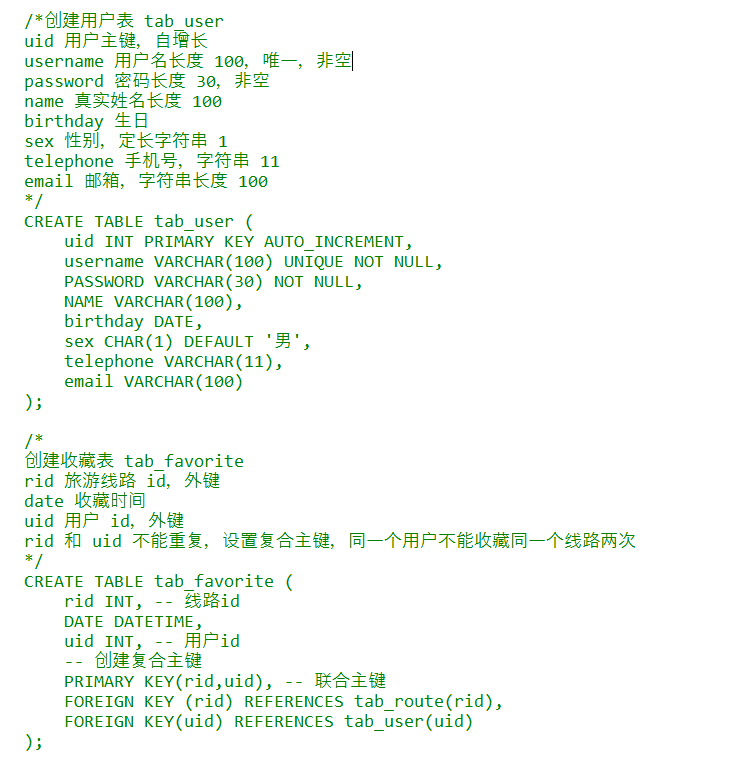
##### 5.1.7视频扩展

无

#### 多表关系\_案例

##### 5.2.1知识概述





##### 5.2.2视频详情



##### 5.2.3总结与补充

根据实际的业务需求来设计不同表之间的关系。首先要分析清楚功能的业务逻辑。

##### 5.2.4课堂提问与练习

练习多表案例

##### 5.2.5习题答案

参考代码

##### 5.2.6视频缺陷

无

##### 5.2.7视频扩展

无

#### 5.3范式概述

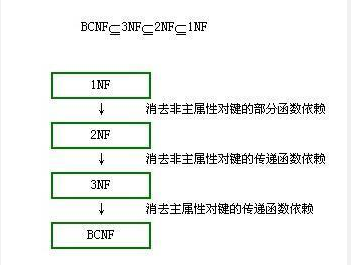
##### 5.3.1知识概述

##### 设计数据库时，需要遵循的一些规范。要遵循后边的范式要求，必须先遵循前边的所有范式要求             设计关系数据库时，遵从不同的规范要求，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。             目前关系数据库有六种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、第四范式(4NF）和第五范式（5NF，又称完美范式）。

##### 5.3.2视频详情



##### 5.3.3总结与补充



[第一范式](https://baike.sogou.com/v340108.htm)（[1NF](https://baike.sogou.com/v451979.htm)）、[第二范式](https://baike.sogou.com/v340341.htm)（[2NF](https://baike.sogou.com/v73031650.htm)）、[第三范式](https://baike.sogou.com/v319931.htm)（[3NF](https://baike.sogou.com/v8437713.htm)）、巴斯-科德范式（BCNF）、[第四范式](https://baike.sogou.com/v687755.htm)(4NF）和[第五范式](https://baike.sogou.com/v48527004.htm)（5NF，又称完美范式）。满足最低要求的范式是第一范式（1NF）。在第一范式的基础上进一步满足更多规范要求的称为第二范式（2NF），其余范式以次类推。一般说来，数据库只需满足第三范式(3NF）就行了。

##### 5.3.4课堂提问与练习

无

##### 5.3.5习题答案

##### 5.3.6视频缺陷

无

##### 5.3.7视频扩展

无

### 第六堂课

本节知识点：

6.1 三大范式详解

6.2 数据库的备份和还原

本节目标 ：

1. 知道数据库三大范式
2. 完成数据库的备份和还原

##### 6.1三大范式详解

##### 6.1.1知识概述

1. 第一范式（1NF）：每一列都是不可分割的原子数据项  
            2. 第二范式（2NF）：在1NF的基础上，非码属性必须完全依赖于码（在1NF基础上消除非主属性对主码的部分函数依赖）  
                \* 几个概念：  
                    1. 函数依赖：A-->B,如果通过A属性(属性组)的值，可以确定唯一B属性的值。则称B依赖于A  
                        例如：学号-->姓名。  （学号，课程名称） --> 分数  
                    2. 完全函数依赖：A-->B， 如果A是一个属性组，则B属性值得确定需要依赖于A属性组中所有的属性值。  
                        例如：（学号，课程名称） --> 分数  
                    3. 部分函数依赖：A-->B， 如果A是一个属性组，则B属性值得确定只需要依赖于A属性组中某一些值即可。  
                        例如：（学号，课程名称） -- > 姓名  
                    4. 传递函数依赖：A-->B, B -- >C . 如果通过A属性(属性组)的值，可以确定唯一B属性的值，在通过B属性（属性组）的值可以确定唯一C属性的值，则称 C 传递函数依赖于A  
                        例如：学号-->系名，系名-->系主任  
                    5. 码：如果在一张表中，一个属性或属性组，被其他所有属性所完全依赖，则称这个属性(属性组)为该表的码  
                        例如：该表中码为：（学号，课程名称）  
                        \* 主属性：码属性组中的所有属性  
                        \* 非主属性：除过码属性组的属性  
                          
            3. 第三范式（3NF）：在2NF基础上，任何非主属性不依赖于其它非主属性（在2NF基础上消除传递依赖）

##### 6.1.2视频详情



##### 6.1.3总结与补充

第二范式依赖于第一范式，第三范式依赖于第二范式，一般数据库只要符合第三范式就可以。

##### 6.1.4课堂提问与练习

数据库范式的作用？

##### 6.1.5习题答案

是为了在设计中更好的解决数据冗余，[数据有效性](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E6%9C%89%E6%95%88%E6%80%A7&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)检查，提高存储效率考虑。

##### 6.1.6视频缺陷

无

##### 6.1.7视频扩展

无

#### 6.2 数据库的备份和还原

##### 6.2.1知识概述

为了数据库中数据的安全，例如误删数据库或者表及表数据，造成数据丢失等问题，所以要进行数据库的备份和还原。

##### 6.2.2视频详情



##### 6.2.3总结与补充

在实际应用中，都会定时进行数据库的备份。

##### 6.2.4课堂提问与练习

练习数据库备份和还原

##### 6.2.5习题答案

参考笔记

##### 6.2.6视频缺陷

无

##### 6.2.7视频扩展

无