

# 《数据库系统》教案

授课学期： 2025—2026 学年第 1 学期

---

课程名称： 数据库系统

---

专 业： 信息工程系

---

班 级： 电子 242、电子 241 (3+)

---

任课教师： 李童彬

---

科目名称	数据库系统	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元 1 认知数据库	班级	电子 242、241 (三加)	人数	51
学生学习条件分析	<p>1.起点分析: (1)已学习 Python 程序设计等代码编写调试类课程。 (2)个别同学接触过数据库。</p> <p>2.重点分析: (1)理解数据库、数据库管理系统、数据模型和数据库应用系统等基本概念。 (2)掌握 MySQL 数据库的安装和配置。 (3)熟悉 MySQL 的环境。</p> <p>3.难点分析: (1)理解数据库、数据库管理系统、数据模型和数据库应用系统等基本概念。</p>				
教学方法手段	<p>1. 教学方法: (学习方法) 讲授法、演示法、自主学习。</p> <p>2. 教学手段: 多媒体、网络平台</p>				
教学重难点	<p><b>教学重点:</b> 1、安装和配置 MySQL 数据库 2、heidSQL 管理工具的使用</p> <p><b>教学难点:</b> 1、处理安装中遇到的一般问题 2、命令行模式下命令的使用</p>				
教学资源	<p>1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a>、<a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a>, 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导</p>				
能力指标	<p>1. 能正确完成 MySQL 数据库的安装与配置 C1、D1 2. 能完成 HeidiSql 软件包的安装 C1、D1 3. 能使用命令方式和图形管理工具连接和断开服务器的操作。C1、D1</p>				
教学目标	<p>知识</p> <p>1. 理解数据库、数据库管理系统、数据库应用系统等基本概念。 2. 掌握数据模型, 特别是关系模型的特点。 3. 掌握 MySQL 数据库的安装与配置。 4. 掌握 HeidiSql 软件包的安装。</p> <p>能力</p> <p>1. 能完成 MySQL 数据库的安装与配置。 2. 能完成 HeidiSql 软件包的安装。 3. 能使用命令方式和图形管理工具连接和断开服务器的操作。</p> <p>素质</p>				

	<p>1. 提升解决数据库安装与配置过程中出现问题的能力。</p> <p>思政</p> <p>1. 通过数据库类型介绍，提升国产数据使用的民族自豪感。</p> <p>2. 通过数据库的广泛应用，增强学生科技报国、不断奋斗的理想和信念。</p>
教师课前准备	<p>1. 多媒体机房安装配置好上课所需的开发环境。</p> <p>2. 设计好课堂教学环节。</p> <p>3. 准备好教学案例和教学课件等材料。</p>
学生课前准备	<p>查阅资料，观看学习通技能点视频</p>
<p>活动历程 (强调成果导向、主要教学法的历程，含辅助手段、时间分配)</p>	<div data-bbox="416 629 1252 1108" data-label="Diagram"> </div> <p>一、准备活动</p> <p>(一) 课程定位、课程主要学习内容、学习方法介绍，激发学生的学习兴趣。(5分钟)</p> <p>(二) 新课导入(5分钟)</p> <p>1. 学生讨论(3min)</p> <p>(1) 谈谈你对数据库的感性认识。</p> <p>2. 教师总结(2min)</p> <p>(1) 总结学生讨论内容。</p> <p>(2) 课程引入：数据库的基本概念。</p> <p>二、发展活动</p> <p>(一) 任务 1：数据库概述（70 分钟）</p> <p>(1) 讲解数据与数据库（思政融入）。(5min)</p> <p>数据库应用广泛，各行各业都在应用，帮助我们提升管理效率。通过数据库的广泛应用，增强学生科技报国、不断奋斗的理想和信念。</p> <p>(2) 介绍数据库技术的发展（思政融入）。(20min)</p> <p>引入国产数据库的各种结构化数据和非结构化数据的厂商和产品介绍。国产替代拉动了国产数据库需求。国产自主可控信创最初是 2016 年提出，2020 年随着电子公文的推进经历两年，在 2022 年上半年整体收官，下一步进入政府下沉部门和行业国产化替代的</p>

阶段。数据库作为国产替代核心产品，华为高斯、达梦、金仓、星环等产品已经从起步的“能用”阶段发展到逐步“好用”的阶段。国产数据库产品在信创推动的政府和其他行业得到广泛地应用，是自身迭代升级的机会，拉进和国外厂商之间的差距。目前党政、金融、电信、石油、电力、交通等行业出台的相关政策，将全面带动其他行业国产替代。

学生讨论我国信创产业的发展，提升国产数据使用的民族自豪感。

- (3) 讲解数据库管理系统 (DBMS)。(10min)
- (4) 讲解数据库系统的组成。(5min)
- (5) 分析数据库系统的特点。(5min)
- (6) 讲解结构化查询语言 SQL。(10min)
- (7) 分析常见的关系型数据库管理系统。(5min)
- (8) 分析总结学生容易出现问题的知识点。(10min)

思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。

——第 1-2 节结束——

(二) 任务 2: MySQL 的安装与配置(80 分钟)

(1) 演示 MySQL 服务器安装与配置。(10min)

配置的一般步骤

步骤一：选择配置 MySQL 服务器

步骤二：服务器类型和网络设置

步骤三：身份验证方法设置

步骤四：账户和角色设置

步骤五：Windows 服务配置

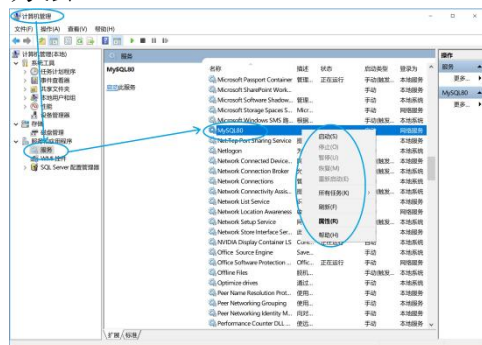
步骤六：应用配置

步骤七：样本和实例配置


(2) 演示 HeidiSql 软件的安装。(10 min)

(3) 演示连接与断开服务器。(20min)

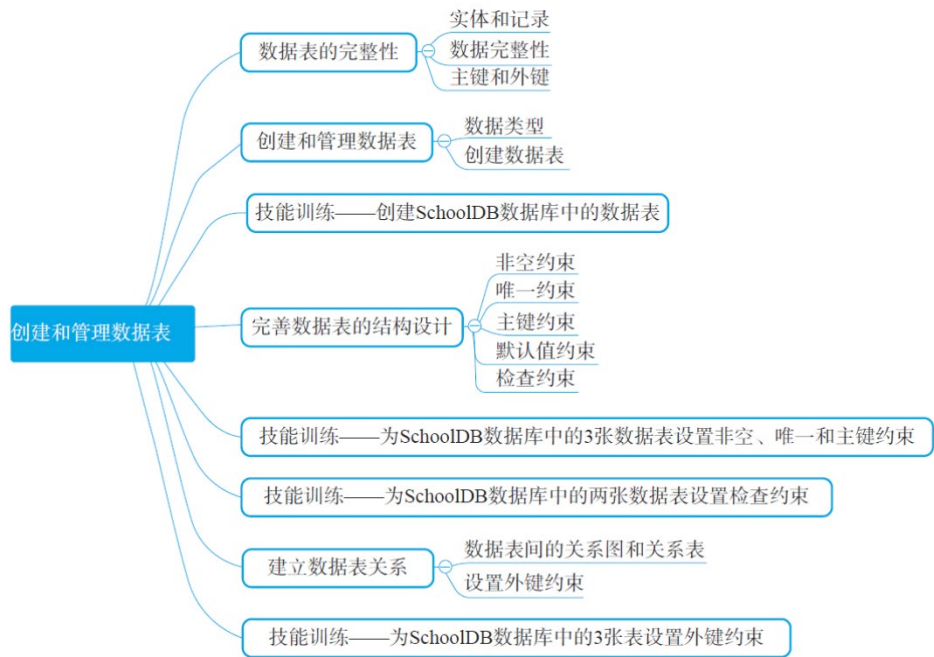
方法一：



方法二：

	 <p>(4) 每位学生熟悉 MySQL 的环境，独立完成实训任务（15min）。</p> <p>(5) 协助学生解决安装、配置中遇到的问题。（10min）</p> <p>(6) 分析总结学生出现的典型问题。（5min）</p> <p>(7) 学生在学习通上，完成阶段测试 1，见网络教学平台。（10min）</p> <p>思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。</p> <p>——第 3-4 节结束——</p>
学习评量	<p>1.阶段测试一，共计 100 分。</p> <p>2.评量方法：超星网络教学平台阶段测试。</p>
课后作业	学习通上完成视频学习
教学反思	<p>目标或能力指针达成度、学习表现、内容、方法、资源、业师协同等向度(至少写三项)</p> <p>1. 目标或能力指标达成度 全班 80%学生达到了教学目标和能力指标。</p> <p>2.学习表现 通过课前任务的布置，学生按要求完成任务单，课上教师提出问题，学生通过讨论，解决问题，个别学生在软件安装过程中出现的问题，经过指导和同学讨论完成任务。</p> <p>3.资源 (1) 学生总结本单元学习情况，教师点评，指出存在的问题及解决方法，做好资料积累工作，保存好各位同学操作的文件。 (2) 对学生操作中出现的记录，以便对下一年度的学生操作指出注意事项。</p>

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元 2 创建和管理数据库表	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)对数据库、表的创建没有概念。 (2)个别同学接触过数据库。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学重难点	<b>教学重点:</b> 1、数据库、表创建和修改命令的使用方法和步骤 <b>教学难点:</b> 1、在“MySQL 8.0 Command Line Client”模式下操作数据库、表				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 能合理有效地对数据库进行定义。 2. 掌握使用命令行方式创建、管理数据库表的能力。			C1 D1	
教学目标	知识 1. 熟悉使用命令行进行 MySQL 数据库操作方式。 2. 掌握数据库和表的创建、修改和删除。 能力 1. 能将关系模型转化为相应的库表。 2. 能使用 MySQL 命令创建、修改相应的库表。 3. 能对数据库表进行数据表约束的创建、修改。 素质 1. 提升解决数据库设计过程中出现问题的能力。 2. 培养良好的学习责任意识。 思政 1. 通过不同级别字符集的设置, 培养学生计算机系统的整体意识。 2. 通过字符集和字符规则的匹配, 强调树立数据库程序开发规范重要性。 3. 树立从多角度去看待事物与思考问题的意识。				
教师课前准备	1. 多媒体机房安装配置好上课所需的开发环境。 2. 设计好课堂教学环节。 3. 准备好教学案例和教学课件等材料。				
学生课前准备	查阅资料, 观看学习通技能点视频				



活动历程  
(强调成果  
导向、主  
要教学法的  
历程，  
含辅助手  
段、时间  
分配)

### 一、准备活动

(一) 上次课复习、本次课主要学习内容介绍，指出重点难点，引起学生重视。(2分钟)

(二) 新课导入(3分钟)

1. 学生讨论(2min)

如果你要建立一个学生成绩管理系统，你会创建哪些表？

2. 教师总结(1min)

(1) 总结学生讨论内容。

(2) 课程引入：数据库和表。

### 二、发展活动

(一) 任务 1：图形化创建数据库表(75分钟)

(1) heidsql 中图形化创建数据库的基本操作(思政融入)。(10min)

**强调字符集、校对规则的设置等支持中文。**

◇ 字符集和校验规则

#### ■ 字符的编码

◆ 简称：Encoding

◆ 字符 (Character) 是指人类语言中最小的表义符号，如A、B等

◆ 对给定每个字符赋予一个数值，用数值来代表对应的字符，这一数值就是字符的编码

◆ 例如：

✓ 给字符A赋予数值0，给字符B赋予数值1

✓ 则0就是字符A的编码，1就是字符B的编码

## ■ 字符集

◆ 简称: Character Set

◆ 给定一系列字符并赋予对应的编码后, 所有这些字符和编码对组成的集合

◆ 例如:

✓ 给定字符列表为{'A', 'B'}时, {'A'=>0, 'B'=>1}就是一个字符集

校验规则

简称: Collation

指在同一字符集内字符之间的比较规则

确定字符集的校对规则后, 才能在一个字符集上定义什么是等价的字符, 以及字符之间的大小关系

每个字符校对规则将唯一对应一种字符集, 但一个字符集可以对应多种字符校对规则, 其中有一个是默认字符集的校对规则 (Default Collation)

**字符集和校对规则层次**

**MySQL 对字符集的支持细化到 4 个层次**

**服务器、数据库、数据表和连接层 4 个层次**

**为避免乱码问题的出现, 从连接层级、客户端和结果返回级、数据库级、表级、服务器级等各个层级使用一致的字符集和校对规则, 此处强调一致性, 培养学生计算机系统的整体意识。**

MySQL 支持中文的常用字符集主要有 3 种

UTF-8 字符集。互联网广泛支持的 Unicode 字符集, 长度为 3 字节

GBK 字符集。主要用于显示汉字, 长度为 2 字节

GB2312

(2) 演示创建数据库示例。(5min)

(3) 技能训练创建数据库。(5min)

(4) 管理数据 (查看、打开、修改、删除)。(5min)

(5) 学生操作完成。(15min)

(6) 分析完成技能训练-管理数据库。(5min)

(7) 学生操作完成技能训练-数据库操作综合训练。(20min)

(8) 协助学生解决实训中遇到的问题。(5min)

(9) 分析总结学生出现的典型问题。(5min)

——第 1-2 节结束——

(一) 任务 2: 命令行方式创建数据库表 (80 分钟)

(1) 创建数据库的 SQL 语句 (思政融入)。(5min)

高校学生成绩管理系统 “的开发团队设计出了 SchoolDB 数据库的关系模型, 现在需要使用关系数据库软件 MySQL 来创建高校学生成绩管理系统的数据库, 并对该数据库进行管理。

强调字符集的设置 为 GB2312 等支持中文的字符集。强调树立数据库程序开发规范重要性。

(2) 演示创建数据库示例。(5min)

**CREATE {DATABASE|SCHEMA} [IF NOT EXISTS]数据库名  
[DEFAULT CHARACTER SET [=] 字符集名  
[DEFAULT] COLLATE [=] 校对规则名];**

- (3) 技能训练创建数据库。(10min)
- (4) 管理数据（查看、打开、修改、删除）。(5min)
- (5) 学生操作完成。(15min)
- (6) 分析教材 P41 完成技能训练-管理数据库要点。(5min)
- (7) 学生操作完成技能训练-数据库操作综合训练(思政融入)。(25min)

学生操作过程中会发现数据库是由不同维度数据聚集而成的数据集；通过多维数据来体现事物的全貌；引导学生理解事物之间存在的联系，逐步树立从多角度去看待事物与思考问题的意识。

- (8) 协助学生解决实训中遇到的问题。(5min)
- (9) 分析总结学生出现的典型问题。(5min)

——第 3-4 节结束——

(二) 任务 3：数据表的完整性和数据类型(80 分钟)

一、准备活动(5min)

- 1、活跃课堂气氛，自我介绍，学习通签到、点名。(1min)
- 2、回顾上节课知识点。(4min)

二、发展活动(70min)

1. 教师讲解数据表

- 数据库是存放数据的容器
  - ◆ 数据库本身是无法存储数据的，要存储数据必须创建数据表
  - ◆ 表是数据库存放数据的对象实体
  - ◆ 没有表，数据库中其他的对象就都没有意义
- 关系数据库中的数据表是二维表格
  - ◆ 由行和列组成
  - ◆ 每一行称为一条记录
  - ◆ 每一列称为一个字段，描述记录的某一特征
- 根据单元需求设计数据表
  - ◆ 一个数据库中要包含多少张数据表
  - ◆ 一个表应该包含几列
  - ◆ 各个列要存放什么类型的数据
  - ◆ 列值是否允许为空

2. 教师讲解实体和记录

- 实体
  - ◆ 是所有客观存在、可以被描述的事物
  - ◆ 例如：学生、课程、教室、假期
  - ◆ 描述实体
  - ◆ 针对实体的特征进行描述
  - ◆ 例如：
    - ✓ 学生，可以从学号、姓名、性别、出生日期、班级及家庭住址等方面进行描述

✓ 课程，可以从课程编号、课程名称、学时及学分等几个方面进行描述

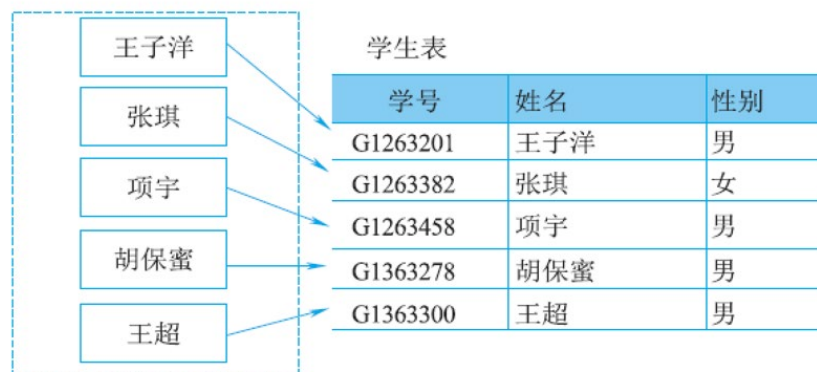
■ 学生实体

- ◆ 不同的学生：学号、姓名、性别、出生日期、班级及家庭住址等也不同
- ◆ 但只要是对学生的描述，描述的格式是一样的
- ◆ 在相同的格式下，不同的数据体现了不同的实体

■ 数据库中用数据表来存储这种相同类型和格式的实体

- ◆ 记录：每一行对应一个实体，通常也叫做一条记录
- ◆ 字段：表中的每一列，如学号、姓名等，通常也称为“字段”

■ 分析：“学生表”



■ 数据库由很多表组成

- ◆ 存储实体的数据表
- ◆ 表达实体关系的表
- ◆ 例如：
  - ✓ 学生和课程之间是存在联系
  - ✓ 某个学生会学习某门课程
  - ✓ 需要建立学生与课程的“关系”
  - ✓ 在数据库中这种关系也是通过表来存储

3. 教师讲解数据完整性

■ 数据完整性是指数据的准确性

- ◆ 准确性是通过数据库表的约束来实现的
- ◆ 例如：在存储学生信息的表中
  - ✓ 如果允许任意输入学生信息，则同一个学生的信息在同一张表中可能会重复出现
  - ✓ 如果不对表中存储的性别加以限制，那么学生的性别可能出现除男或女以外的值

■ 数据完整性一般包含四种类型

- ◆ 实体完整性
- ◆ 域完整性
- ◆ 参照完整性
- ◆ 用户自定义的完整性

■ 实体完整性

- ◆ 实体完整性要求表中的每一条记录反映不同的实体
- ◆ 不能存在相同的记录
- ◆ 实现途径：
  - ✓ 通过主键约束
  - ✓ 标识列属性
  - ✓ 唯一约束或索引

■ 域完整性

- ◆ 指表中字段输入值的有效性
- ◆ 实现途径：
  - ✓ 设置字段的类型
  - ✓ 设置取值范围（CHECK 约束）
  - ✓ 设置默认值（DEFAULT）
  - ✓ 设置非空约束等
- ◆ 例如：性别只能为取值为男或女，确保不合格的数据不进入数据库表中

■ 参照完整性

- ◆ 在输入或删除记录时，保证了两张表中相关联字段的值的一致性
- ◆ 例如：管理学生信息时
  - ✓ 学生表中存储学生的信息
  - ✓ 成绩表中存储考试成绩的信息
  - ✓ 并且成绩表中有一列数值为学号，通过这个学号的值在学生表能查找到学生的详细信息
  - ✓ 在成绩表中，被存入的学号必须是在学生表中已经存在的，否则不能存到成绩表中

学号	姓名	性别	...
G1263201	王子洋	男	
G1263382	张琪	女	
G1263458	项宇	男	
G1363278	胡保蜜	男	
G1363300	王超	男	

学号	课程	成绩	...
G1263201	大学英语	76	
G1263201	高等数学	88	
G1263382	大学英语	99	
G1263382	高等数学	80	
G1263458	大学英语	91	
G1263458	高等数学	62	
G1263999	大学英语	75	学号不存在，出现异常
...	...	...	

学号	课程	成绩	...
G1263201	大学英语	76	
G1263201	高等数学	88	
G1263382	大学英语	99	
G1263382	高等数学	80	
G1263458	大学英语	91	
G1263458	高等数学	62	

■ 用户自定义完整性

- ◆ 用户自定义完整性用来定义特定的规则
- ◆ 通过数据库的规则、触发器及存储过程等方法进行约束
- ◆ 涉及到数据库编程相关功能

## 主键

- ◆ 简写：Primary Key, PK
- ◆ 主键约束可以实现数据的实体完整性
- ◆ 规范化的数据库中的每张表都必须设置主键约束
- ◆ 主键的字段值必须是唯一的，不允许重复，也不能为空
- ◆ 一张表只能定义一个主键，主键可以是单一字段，也可以是多个字段组合

## 例如：

- ◆ 在学生表中，设置“学号”为主键，因为在一所学校内部学号是唯一的
- ◆ 在学生表中，不能设置“姓名”为主键，因为同名情况比较普遍

## 外键

- ◆ 简写：Foreign Key, FK
- ◆ 外键约束可以使一个数据库的多张表之间建立关联
- ◆ 外键约束可以保证数据的参照完整性
- ◆ 例如：在成绩表的学号字段上建立外键约束，关联到学生表的学号字段
  - ✓ 学生表称为“主表”
  - ✓ 成绩表称为“从表”（或称“相关表”）

## 4. 数据类型

■ 数据类型是数据的一种特征，决定数据的存储格式

■ 每个列、变量、表达式和参数都有各自的数据类型

### ■ MySQL中常用的数据类型

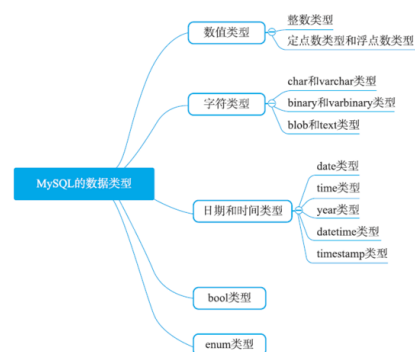
- ◆ 数值类型
- ◆ 字符类型
- ◆ 日期和时间类型
- ◆ bool类型
- ◆ enum类型

■ 数据类型是数据的一种特征，决定数据的存储格式

■ 每个列、变量、表达式和参数都有各自的数据类型

### ■ MySQL中常用的数据类型

- ◆ 数值类型
- ◆ 字符类型
- ◆ 日期和时间类型
- ◆ bool类型
- ◆ enum类型



## ■ 数值类型的范围

- ◆ 在标准SQL数值数据类型基础上，有一定的扩展

## ■ 数值类型分类

- ◆ 整型
- ◆ 定点数类型
- ◆ 浮点数类型

## ■ 数值类型数据显示宽度设置

- ◆ 通过可选的宽度显示指示器选项

## ■ 例如：

- ◆ 数据类型定义成：bigint(20)
- ◆ 表示该类型定义的数据在显示时，需要占20位的宽度
- ◆ 如果不足20位时，则在左侧自动补空格

## ■ 整数类型

- ◆ 精确整数数字的值
- ◆ 类型名称及特征

类型名称	字节数	范围(有符号)	范围(无符号)	解 释
TINYINT	1	(-128,127)	(0,255)	小整数值,如年龄等
SMALLINT	2	(-2 <sup>15</sup> ,2 <sup>15</sup> -1)	(0,2 <sup>16</sup> -1)	较大整数值
MEDIUMINT	3	(-2 <sup>23</sup> ,2 <sup>23</sup> -1)	(0,2 <sup>24</sup> -1)	大整数值
INT 或 INTEGER	4	(-2 <sup>31</sup> ,2 <sup>31</sup> -1)	(0,2 <sup>32</sup> -1)	中等范围的大整数值,如距离等
BIGINT	8	(-2 <sup>63</sup> ,2 <sup>63</sup> -1)	(0,2 <sup>64</sup> -1)	极大整数值

## ■ 数值类型数据显示宽度设置

- ◆ 通过可选的宽度显示指示器选项

## ■ 例如：

- ◆ 数据类型定义成：bigint(20)
- ◆ 表示该类型定义的数据在显示时，需要占20位的宽度
- ◆ 如果不足20位时，则在左侧自动补空格

## ■ 数值类型数据显示宽度设置

- ◆ 通过可选的宽度显示指示器选项

## ■ 注意：

- ◆ 宽度显示指示器不会影响数据类型对数据大小的规定
- ◆ 例如：TINYINT (5)
- ◆ 不表示该数能有5位数字，其大小仍然是 (-128, 127)
- ◆ 只是该数字在显示的时候占5位的宽度，在左边自动补齐空

## ■ 定点数和浮点数

- ◆ 定点数类型和浮点数类型都表示小数，表示为：DECIMAL(M,D)
- ◆ 定点数类型在数据库中存放精确的值
- ◆ 浮点数类型在数据库中存放的是近似值
  - ✓ 包括单精度浮点数 (FLOAT)
  - ✓ 双精度浮点数 (DOUBLE)

类型名称	字节数	范围(有符号)	范围(无符号)	解 释
DECIMAL(M,D)	M+2	依赖于 M 和 D 的值	依赖于 M 和 D 的值	精确的小数值, 数额、考试成绩等
FLOAT(M,D)	4	(-3.402 823 466E+38, 1.175 494 351E-38)	0, (1.175 494 351E-38, 3.402 823 466E+38)	单精度浮点数值, 成绩、温度等较小的
DOUBLE(M,D)	8	(-1.7976931348623157E+308, 2.2250738585072014E-308)	0, (2.225 073 858 507 201 4E-308, 1.797 693 134 862 315 7E+308)	双精度浮点数值, 科学数据

## ■ 字符类型最常用的数据类型之一

## ■ 字符类型的表示

- ◆ 字符类型的数据通常被放在一对单引号中

## ■ 字符类型包括

- ◆ 以字符个数来限定数据长度的CHAR和VARCHAR
- ◆ 以文本方式存放数据的TEXT
- ◆ 以二进制方式存放数据的BLOB
- ◆ 以字节为单位来存储二进制数据的BINARY和VARBINARY
- ◆ 以枚举方式列出可能取值的数据类型ENUM和SET

## ■ CHAR和VARCHAR字符类型

- ◆ 都以字符个数来限定数据长度，常用来存储字符串数据，如名字
- ◆ 区别
  - ✓ 保存和检索的方式不同
  - ✓ 最大长度不同
  - ✓ 是否保留尾部空格不同

类型名称	字节数	解 释
CHAR (n)	0~255	定长字符串,当数据的长度达不到最大长度时,在它们的右边填充空格以达到最大长度,如身份证号(固定18位)、邮编(6位)、手机号(11位); <i>n</i> 表示用户要保存的最大字符数
VARCHAR (n)	0~65535	变长字符串,只保存需要的字符数,不进行填充。若列值的长度超过声明的长度,将对值进行裁剪以使其合适,如姓名、地址、商品名称

## ■ CHAR和VARCHAR字符类型

- ◆ CHAR常用来存储长度固定的字符串变量
- ◆ 如：身份证号（固定18位）、邮编（6位）、手机号（11位）
- ◆ VARCHAR类型存储长度无法固定的数据
- ◆ 如：姓名、地址

## ■ BINARY和VARBINARY类型

- ◆ BINARY和VARBINARY类似于CHAR和VARCHAR
- ◆ 不同：存储以字节为单位的二进制数据

类型名称	字节数	解 释
BINARY (n)	<i>n</i> +4	定长二进制数据, <i>n</i> 的范围为1~255,当数据的长度达不到最大长度时,不足部分以0填充
VARBINARY (n)	实际长度+4	变长二进制数据, <i>n</i> 的范围为1~65535,只保存需要的长度,不进行填充

## ■ BLOB和TEXT类型

- ◆ TEXT是字符型长对象类型，BLOB是二进制长对象类型
- ◆ TEXT是以文本方式存储数据，常用于存储长型的文本数据
- ◆ 如：新闻事件、博客、产品描述等
- ◆ 按文本的长短，有4种TEXT类型

类型名称	字节数	解 释
TINYTEXT	0~255 个字符	短文本字符串
TEXT	0~65 535 个字符	长文本数据
MEDIUMTEXT	0~16 777 215 (即 $2^{24}-1$ ) 个字符	中等长度文本数据
LONGTEXT	0~4 294 967 295 (即 $2^{32}-1$ ) 个字符	极大长度文本数据

## ■ 字符类型最常用的数据类型之一

### ■ 字符类型的表示

- ◆ 字符类型的数据通常被放在一对单引号中

### ■ 字符类型包括

- ◆ 以字符个数来限定数据长度的CHAR和VARCHAR
- ◆ 以文本方式存放数据的TEXT
- ◆ 以二进制方式存放数据的BLOB
- ◆ 以字节为单位来存储二进制数据的BINARY和VARBINARY
- ◆ 以枚举方式列出可能取值的数据类型ENUM和SET

## ■ BLOB和TEXT类型

- ◆ BLOB类型常用来存储图片、视频、音频、附件等二进制数据
- ◆ 按数据长度，有4种类型
  - ✓ TINYBLOB
  - ✓ BLOB
  - ✓ MEDIUMBLOB
  - ✓ LONGBLOB

类型名称	字节数	解 释
TINYBLOB	0~255 字节	不超过 255 字节的二进制数据
BLOB	0~65 535 字节 (64 KB)	二进制形式的长数据
MEDIUMBLOB	0~16 777 215 字节 (16 MB)	二进制形式的中等长度数据
LONGBLOB	0~4 294 967 295 字节 (4 GB)	二进制形式的极长数据

■ 时间和日期类型最常用的数据类型之一

■ 例如：出生日期、借阅时间

■ 类型名称及特征

类型名称	字节数	格式	解 释
DATE	3	YYYY-MM-DD	日期值
TIME	3	HH:MM:SS	时间值或持续时间
YEAR	1	YYYY	年份值
DATETIME	8	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	8	YYYYMMDD HHMMSS	混合日期和时间值,时间戳

■ DATE类型

◆ 日期类型

■ 一般使用“年-月-日”的格式表示

■ 允许使用不严格的语法，如以下表示是等价的

◆ “2021-12-31”

◆ “2021.12.31”

◆ “2021/12/31”

◆ “2016@12@31”

■ TIME类型

◆ 时间类型

■ 一般使用“时:分:秒”的格式表示

■ 允许使用不严格的语法，且时、分、秒的值小于10时，无需加0

◆ “10:6:3”

◆ “10:06:03”

■ 还可以表示为“HH:MM:SS. fraction”格式

◆ fraction为分秒或毫秒

◆ 如：“10:6:3.120”

■ YEAR类型

◆ 年类型

■ 其值可以是字符串，也可以是数值

◆ 如：2021、“2021”

## ■ DATETIME类型

### ◆ 日期时间类型

## ■ 使用“年-月-日 时:分:秒”的格式表示，也可采用不严格格式

## ■ 日期和时间直接用空格分隔

### ◆ “2021-12-31 10:6:3”

### ◆ “2021.12.31 10:06:03”

### ◆ “2021/12/31 10:6:3.120”

## ■ MySQL中本身没有bool类型

## ■ 为了与其他关系数据库相兼容，MySQL提供了bool类型的映射

## ■ bool类型会被转换成tinyint数据类型

## ■ 枚举类型

### ◆ 必须定义枚举值

### ◆ 该种数据类型的值只能是定义时的枚举值

### ◆ 如果输入了枚举值之外的数值，则插入命令将报错

## ■ 枚举值一般为字符串，但可以为NULL

### ◆ 字符串按照枚举顺序，枚举值的索引依次被定义为0, 1, 2, 3,

### ◆ NULL的枚举值仍然为NULL

学生认真听讲并完成实训

三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)

思政点：理解事物间的联系是普遍存在的，引申出处理好人际关系的重要性，数据存储关系（内存越大，数据存储量越大，处理问题速度效率越高）同时融进思政元素（大格局）。

——第 5-6 节结束——

一、教学导入（5 分钟）

1. 学生讨论(3min)

(1) 如何在数据库中创建一个表格？

2. 教师总结(2min)

(1) 总结学生讨论内容。

(2) 课程引入：数据表的创建，可以有效的存储数据。

(四) 任务 4：命令行方式创建管理数据表（75 分钟）

(4) 每个学生跟练习相应的案例。（10min）

一、准备活动(5min)

1、活跃课堂气氛，自我介绍，学习通签到、点名。（1min）

2、回顾上节课知识点。（4min）

二、发展活动(70min)

1. 教师讲解数据表（5min）
  - 数据表
    - ◆ 是数据库中最重要对象
    - ◆ 整个数据库中的数据都是物理存储在各个数据表中
  - 数据表包含系统表和用户表
    - ◆ 系统表是创建数据库的时候自动生成的，用来保存数据库自身的信息
    - ◆ 用户表存储用户数据
    - ◆ 数据表的创建方式
    - ◆ 使用命令行模式创建
    - ◆ 使用 navicat 客户端创建
2. 使用命令行模式创建数据表（15min）
  - 语法
 

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]表名
(
    列名 1 数据类型 [约束][索引][注释],
    列名 2 数据类型 [约束][索引][注释],
    列名 3 数据类型 [约束][索引][注释],
    ...
    列名 n 数据类型 [约束][索引][注释]
)ENGINE=存储引擎;
```

    - TEMPORARY
      - ◆ 加该关键字表示创建临时表
      - ◆ 否则表示持久表
    - IF NOT EXISTS
      - ◆ 在创建表前判断，只有当该表不存在时才执行 CREATE TABLE 操作
      - ◆ 使用此选项可避免出现表已经存在无法再新建的错误
    - 表名
      - ◆ 要创建的表名
      - ◆ 表名必须符合标识符的命名规则
    - 列名
      - ◆ 表中列的名字
      - ◆ 列名必须符合标识符的命名规则，长度不能超过 64 个字符
      - ◆ 在表中要唯一
      - ◆ 如果有 MySQL 保留字则必须用英文单引号括起来
    - 数据类型
      - ◆ 列的数据类型
      - ◆ 有的数据类型需要指明长度 n，并用括号括起来
    - 约束
      - ◆ 包括：非空约束、默认值约束、主键约束、唯一性约

束、外键约束、检查约束等

◆ 可以在创建表后在添加约束

■ 存储引擎

◆ MySQL 8 中默认的存储引擎为 InnoDB

◆ 通常可以省略

【演示示例 3-1】使用命令行创建数据表（10min）

■ 需求

◆ 为数据库SchoolDB创建数据表：年级表Grade

◆ 不带任何约束

◆ 表结构

列名	数据类型	长度	字段说明	属性	备注
gradeId	int	4	年级编号	非空	主键
gradeName	varchar(50)	50	年级名称	非空	

```
CREATE TABLE Grade
(
    gradeId INT(4) COMMENT '年级编号',
    gradeName VARCHAR(50) COMMENT '年级名称'
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci
```

! 补充说明：如果“ENGINE”后都采用默认的方式，可以省略。

```
CREATE TABLE grade
(
    gradeId INT(4) COMMENT '年级编号',
    gradeName VARCHAR(50) COMMENT '年级名称'
);
```

学生动手操作，完成实训（35min）。

思政点：遇到问题，要学会看报错，或者利用已有的工具解决问题。发扬工匠精神，将执着专注、精益求精、一丝不苟、追求卓越的理念融入到单元开发中。

三、课堂总结(5min)

思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。

——第 7-8 节结束——

（四）任务 4：管理数据表(80 分钟)

一、准备活动(5min)

1、活跃课堂气氛，自我介绍，学习通签到、点名。（1min）

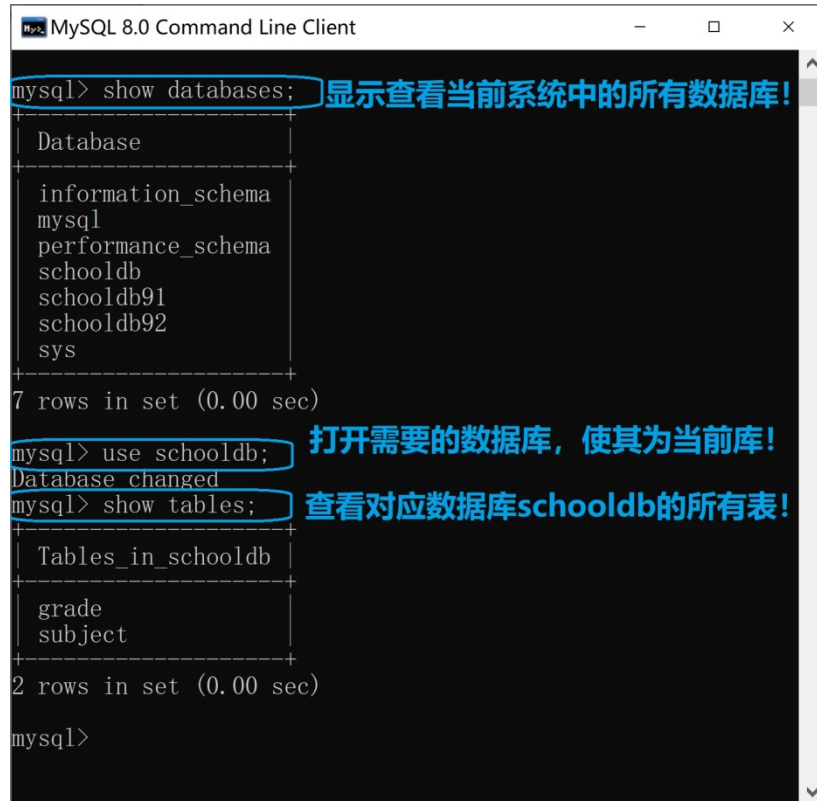
2、回顾上节课知识点。（4min）

二、发展活动(70min)

1. 教师讲解管理数据表的常用方法

- 查看数据表
- 复制数据表
- 修改数据表名
- 修改数据表结构
- 删除数据表

2. 操作演示：查看数据表名称 SHOW TABLES;



```
MySQL 8.0 Command Line Client
mysql> show databases; 显示查看当前系统中的所有数据库!
+-----+
| Database |
+-----+
| information_schema |
| mysql         |
| performance_schema |
| schooldb      |
| schooldb91    |
| schooldb92    |
| sys          |
+-----+
7 rows in set (0.00 sec)

mysql> use schooldb; 打开需要的数据库，使其为当前库!
Database changed
mysql> show tables; 查看对应数据库schooldb的所有表!
+-----+
| Tables_in_schooldb |
+-----+
| grade              |
| subject            |
+-----+
2 rows in set (0.00 sec)

mysql>
```

学生操作演示

3. 操作演示：查看数据表结构 DESCRIBE 表名;

```

MySQL 8.0 Command Line Client
syntax to use near 'describe student' at line 1
mysql> show tables;
+-----+
| Tables_in_schooldb |
+-----+
| grade               |
| result              |
| student             |
| subject             |
+-----+
4 rows in set (0.00 sec)

mysql> describe student;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| studentno  | varchar(20)   | YES  |     | NULL    |       |
| loginPwd   | varchar(20)   | YES  |     | NULL    |       |
| studentName | varchar(50)   | YES  |     | NULL    |       |
| sex        | char(4)       | YES  |     | NULL    |       |
| gradeId    | int           | YES  |     | NULL    |       |
| phone      | varchar(20)   | YES  |     | NULL    |       |
| address    | varchar(255)  | YES  |     | NULL    |       |
| bornDate   | date          | YES  |     | NULL    |       |
| email      | varchar(50)   | YES  |     | NULL    |       |
| identityCard | varchar(18)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>

```

学生操作演示

#### 4. 复制数据表

##### 语法

**CREATE TABLE [IF NOT EXISTS] 新表名  
[LIKE 参照表名]  
[AS (SELECT 语句)]**

##### 分析

- LIKE关键字创建一个新的空表，与参照表名结构完全相同
- AS关键字可以复制表的内容，但索引和完整性约束不会
- 一般只能在同一个数据库内部复制表，不能跨数据库复制

##### 【演示示例 3-3】

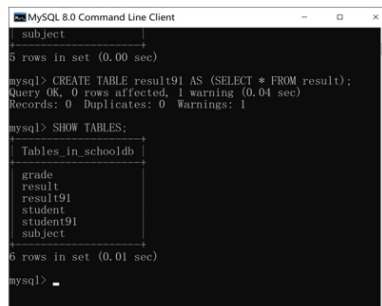
#### 复制数据库 SchoolDB 中的数据表

- 需求
  - ◆ 数据库 SchoolDB 中复制表
  - ◆ 创建表 student91，表结构直接取自 student 表
  - ◆ 创建表 result91，其结构和数据都取自 result 表

## 🔍分析代码

```
CREATE TABLE student91 LIKE student;  
CREATE TABLE result91 AS (SELECT * FROM result);  
SHOW TABLES;
```

## 🔍分析结果



```
MySQL 8.0 Command Line Client  
subject  
5 rows in set (0.00 sec)  
mysql> CREATE TABLE result91 AS (SELECT * FROM result);  
Query OK, 0 rows affected, 1 warning (0.04 sec)  
Records: 0 Duplicates: 0 Warnings: 1  
mysql> SHOW TABLES;  
Tables_in_schooldb  
grade  
result  
result91  
student  
student91  
subject  
6 rows in set (0.01 sec)  
mysql>
```

 操作演示：复制数据表

### 5. 修改数据表名称

■ 命令：RENAME TABLE

■ 语法格式：

**RENAME TABLE 旧表名 1 TO 新表名 1  
[旧表名 2 TO 新表名 2]…;**

■ 命令：ALTER TABLE

■ 增加数据列

**ALTER [IGNORE] TABLE 表名**

**ADD 新增列的列名 数据类型[FIRST|AFTER 参照列的列名]**

■ 删除数据列

**ALTER [IGNORE] TABLE 表名 DROP [column] 列名**

■ 命令：ALTER TABLE

■ 为表中列重命名

**ALTER [IGNORE] TABLE 表名**

**CHANGE [column] 旧列名 列定义 [FIRST|AFTER 列名]**

■ 修改表中列的数据类型

**ALTER [IGNORE] TABLE 表名**

**MODIFY [column] 列名 列定义 [FIRST|AFTER 列名]**

■ 命令: DROP TABLE

■ 语法格式: |

### **DROP TABLE [IF EXISTS] 表名 1[,表名 2]...**

思政点: 面对不同的单元要求, 需要使用不同的命令, 所以, 培养学生坚韧执着、刻苦钻研、迎难而上的品质和精益求精的工匠精神。

学生操作演示并完成实训

三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)

思政点: 下课, 关闭电脑、整理桌面, 养成良好的行为习惯。

——第 9-10 节结束——

(五) 任务 5: 完善数据表的结构设计(80 分钟)

一、准备活动(5min)

1、活跃课堂气氛, 自我介绍, 学习通签到、点名。(1min)

2、回顾上节课知识点。(4min)

二、发展活动(70min)

1. 教师讲解完善数据表的结构设计

■ 数据完整性

◆ 实体完整性

◆ 域完整性

◆ 参照完整性

◆ 用户自定义完整性

■ 途径: 对表添加必要的约束

■ 约束: 存入到数据表中的数据列的取值所必须遵守的规则

■ 效果: 当录入或者插入数据时, 只有符合规则的值才能被接受

2. 任务一: 非空约束的设置

■ 可以对数据表的列设置: “是否允许为空”

◆ 列是否允许为空和具体的要求相关

■ 设置了非空约束的列, 表明该列的取值不允许为空

■ 例如:

◆ 学生的地址不是很重要, 可以为空

◆ 而姓名是重要的、不可或缺的信息, 就不应该允许为空

■ 设置途径

◆ 创建表时设置非空约束

◆ 对已经存在的表设置非空约束

### 语法

- 列名的后面加上 “NOT NULL”

### 举例

- 需要为表Grade中gradeId列和gradeName列添加非空约束

```
CREATE TABLE grade
(
    gradeID INT(4) NOT NULL COMMENT '年级编号',
    gradeName VARCHAR(50) NOT NULL COMMENT '年级名称'
);
```

对已经存在的表设置非空约束

### 语法

```
ALTER [IGNORE] TABLE 表名
    MODIFY 列名 数据类型 NOT NULL;
```

### 举例

- 在数据库SchoolDB中，对已经创建的表Grade设置gradeId列和gradeName列为非空约束

```
ALTER TABLE Grade
    MODIFY gradeId INT(4) NOT NULL;
```

【演示示例 3-4】设置非空约束

### 需求

- 在数据库SchoolDB中
  - ◆ 对已经创建的表Grade设置gradeId列和gradeName列为非空约束

### 分析代码

```
ALTER TABLE Grade
    MODIFY gradeId INT(4) NOT NULL;
ALTER TABLE Grade
    MODIFY gradeName VARCHAR(50) NOT NULL;
DESCRIBE Grade;
```

```

MySQL 8.0 Command Line Client
mysql>
mysql> ALTER TABLE Grade
-> MODIFY gradeId INT(4) NOT NULL;
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> ALTER TABLE Grade
-> MODIFY gradeName VARCHAR(50) NOT NULL;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE Grade;
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| gradeId    | int       | NO   |     | NULL    |       |
| gradeName  | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

修改后的非空约束!

Field	Type	Null	Key	Default	Extra
gradeId	int	NO		NULL	
gradeName	varchar(50)	NO		NULL	

### 3. 任务二：唯一约束

- 若某些列的值需要唯一，将该列设置“唯一约束”
- 结果：该列的任何两行都不能有相同的列值
- 允许 NULL 值的列设置唯一约束后，该列中 NULL 值只能出现一次
- 一个表中可以为多个列设置唯一约束
- 设置途径
  - ◆ 创建表时设置唯一约束
  - ◆ 对已经存在的表设置唯一约束

#### 创建数据表时设置唯一约束

##### 语法

- 列名的后面加上“UNIQUE”

##### 举例

- 将学生信息表Student中身份证号设置为唯一约束

```
identityCard VARCHAR(18) UNIQUE COMMENT '身份证号'
```

#### 对已经存在的表设置唯一约束

## 语法

**ALTER TABLE 表名**

**ADD [CONSTRAINT]约束名 UNIQUE KEY(列名);**

## 举例

### ■ 在数据库SchoolDB中

- ◆ 对已经创建的表student设置身份证号列identityCard为唯一约束
- ◆ 约束名称为UN\_ic

**ALTER TABLE Student**

**ADD CONSTRAINT UN\_ic UNIQUE KEY(identityCard);**

【演示示例 3-5】设置唯一约束

## 需求

### ■ 在数据库SchoolDB中

- ◆ 对已经创建的表student设置身份证号列identityCard为唯一约束
- ◆ 约束名称为UN\_ic

## 分析代码

**ALTER TABLE Student**

**ADD CONSTRAINT UN\_ic UNIQUE KEY(identityCard);**

**DESCRIBE Student;**

```
MySQL 8.0 Command Line Client
10 rows in set (0.00 sec)

mysql> ALTER TABLE Student
-> ADD CONSTRAINT UN_ic UNIQUE KEY(identityCard);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE student;
```

Field	Type	Null	Key	Default	Extra
studentno	varchar(20)	YES		NULL	
loginPwd	varchar(20)	YES		NULL	
studentName	varchar(50)	YES		NULL	
sex	char(4)	YES		NULL	
gradeId	int	YES		NULL	
phone	varchar(20)	YES		NULL	
address	varchar(255)	YES		NULL	
bornDate	date	YES		NULL	
email	varchar(50)	YES		NULL	
identityCard	varchar(18)	YES	UNI	NULL	

10 rows in set (0.00 sec)

```
mysql>
```


设置唯一约束成功!

## 学生操作演示

4. 任务三：使用可视化的方式创建非空约束和唯一约束

学生自主探究并展示。

思政点：设置约束时引导学生学生规范做事、严谨做人，做人做事都要讲规矩、守底线。

	 <p>学生操作演示并完成实训 三、课堂总结(5min) 全班共同对学习内容进行整合总结。(5min) <b>思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。</b> ——第 11-12 节结束——</p>
学习评量	<p>1.阶段测试一，共计 100 分。 2.评量方法：超星网络教学平台阶段测试。</p>
课后作业	<p>学习通完成视频学习</p>
教学反思	<p>目标或能力指针达成度、学习表现、内容、方法、资源、业师协同等向度(至少写三项)</p> <p>1.目标或能力指标达成度 全班 80%学生达到了教学目标和能力指标。</p> <p>2.学习表现 通过课前任务的布置，学生按要求完成任务单，课上教师提出问题，学生通过讨论，解决问题，个别学生在数据库定义过程中出现的问题，经过指导和同学讨论完成任务。</p> <p>3.资源 (1) 学生总结本单元学习情况，教师点评，指出存在的问题及解决方法，做好资料积累工作，保存好各位同学操作的文件。 (2) 对学生操作中出现的记录，以便对下一年度的学生操作指出注意事项。</p>

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元三 SQL 语句操作数据表	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	<p>1.起点分析:</p> <p>(1)掌握了数据库和表的定义。</p> <p>(2)能正确的进行数据库、表的创建、修改、删除。</p>				
教学方法手段	<p>1.教学方法: (学习方法) 讲授法、演示法、自主学习。</p> <p>2.教学手段: 多媒体、网络平台</p>				
教学资源	<p>1.个人资源: 教材、教学 PPT、实训指导书</p> <p>2.学校资源: 超星平台中的公共资源</p> <p>3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a>、<a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a>, 出版社资源: 配套微课</p> <p>4.企业资源: 企业工程案例、优秀校友的线上辅导</p>				
能力指标	<p>1. 掌握使用 MySQL 命令向表中插入、修改、删除数据的方法。 C1</p> <p>2. 提升解决数据库设计过程中出现问题的能力。 D1</p> <p>3. 培养良好的规范意识。 A1</p>				
教学目标	<p>知识</p> <p>1. 掌握使用 MySQL 命令向表中插入数据的方法。</p> <p>2. 掌握使用 MySQL 命令向表中更新、删除数据的方法。</p> <p>3. 理解 INSERT 语句、UPDATE 语句、DELETE 语句语法;</p> <p>能力</p> <p>1. 能根据要求实现数据的插入。</p> <p>2. 能根据要求实现数据的更新、删除等操作。</p> <p>素质</p> <p>1. 提升解决数据库操作中出现问题的能力。</p> <p>2. 培养良好的学习责任意识。</p> <p>思政</p> <p>1. 培养规则意识, 遵纪守法意识。</p> <p>2. 树立程序编写过程中细致入微的工作态度。</p> <p>3. 引导学生建立规范编写程序代码的意识, 培养学生的职业素养与道德规范。</p>				
教学重难点	<p>1. MySQL 系列命令的规则和应用</p> <p>2. 理解数据表的约束和数据表之间的关系, 并通过数据验证是否有效</p>				
教师课前准备	<p>1. 多媒体机房安装配置好上课所需的开发环境。</p> <p>2. 设计好课堂教学环节。</p> <p>3. 准备好教学案例和教学课件等材料。</p>				

学生课前准备	查阅资料，观看学习通技能点视频
活动历程 (强调成果导向、主要教学法的历程，含辅助手段、时间分配)	<p>一、准备活动(5min)</p> <p>1、活跃课堂气氛，自我介绍，学习通签到、点名。(1min)</p> <p>2、回顾上节课知识点。(4min)</p> <p>二、发展活动(70min)</p> <p>1.教师讲解 SQL 语言简介</p> <ul style="list-style-type: none"> <li>■ 一门针对数据库而言的语言</li> <li>■ 可以创建数据库、创建数据表、添加约束等</li> <li>■ 可以针对数据库的数据进行增、删、改、查等操作</li> <li>■ 可以创建视图、存储过程等</li> <li>■ 可以赋予用户权限等</li> <li>■ SQL 的全称：结构化查询语言 (Structured Query Language)</li> <li>■ SQL 语言是一种数据库查询和程序设计语言 <ul style="list-style-type: none"> <li>◆ 用于存取、查询、更新数据</li> <li>◆ 管理关系数据库系统</li> </ul> </li> <li>■ SQL 结构简洁、功能强大、应用广泛 <ul style="list-style-type: none"> <li>◆ Oracle、DB2、Microsoft SQL Server、MySQL、Access 等数据库支持</li> </ul> </li> <li>■ SQL 语言不同于 C#和 Java 等程序设计语言 <ul style="list-style-type: none"> <li>◆ 只能被数据库识别的指令</li> <li>◆ 在程序中，可以组织 SQL 语句发送给数据库，数据库再执行相应的操作</li> </ul> </li> <li>■ 例如：在 Java 程序中要得到 MySQL 数据库表中的记录 <ul style="list-style-type: none"> <li>◆ 在 Java 程序中编写 SQL 语句</li> <li>◆ 然后发送到数据库</li> <li>◆ 数据库接收到的 SQL 语句并执行</li> <li>◆ 再把执行结果返回给 Java 程序</li> </ul> </li> </ul> <p>2.教师讲解 SQL 组成</p> <ul style="list-style-type: none"> <li>■ MySQL 数据库中，SQL 由 4 部分组成 <ul style="list-style-type: none"> <li>◆ 数据定义语言</li> <li>◆ 数据操纵语言</li> <li>◆ 数据控制语言</li> <li>◆ MySQL 增加的语言元素</li> </ul> </li> <li>■ 数据定义语言 (DDL, Data Definition Language) <ul style="list-style-type: none"> <li>◆ 用于对数据库及各种对象进行创建 (CREATE)、删除 (DROP)、修改 (ALTER) 等操作</li> <li>◆ 数据库对象包括：表、默认约束、规则、视图、触发器、存储过程等</li> <li>◆ 数据操纵语言 (DML, Data Manipulation Language) <ul style="list-style-type: none"> <li>◆ 用于对数据库中的数据进行操作</li> <li>◆ 包括：查询 (SEIECT)、插入 (INSERT)、修改 (UPDATE)、删除 (DEIETE) 等语句</li> </ul> </li> </ul> </li> <li>■ 数据控制语言 (DCL, Data Control Language)</li> </ul>

- ◆ 用于安全管理，授予或回收用户操作数据库对象的权限
- ◆ 包括：授权（GRANT）、收回（REVOKE）等语句
- ◆ MySQL 增加的语言元素
- ◆ 为了方便用户编程而增加的语言元素
- ◆ 包括：常量、变量、运算符、函数、流程控制语句和注释等

### 3.教师讲解 MySQL 运算符

- 运算符：是一种符号
  - ◆ 是用来进行列间或者变量之间的运算的符号
  - ◆ MySQL 中运算符分类
  - ◆ 算术运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符

#### 算术运算符

- 算术运算符：加、减、乘、除、求余数
- 运算符的符号和功能列表

运算符	说 明
+	加运算,求两个数或表达式相加的和
-	减运算,求两个数或表达式相减的差
*	乘运算,求两个数或表达式相乘的积
/	除运算,求两个数或表达式相除的商
DIV	除运算,求两个数或表达式相除的商
%	取余数运算,求两个数或表达式相除的
MOD	取余数运算,求两个数或表达式相除的

- 运算符两边的数据类型不一致时，要进行数据类型的转换
  - ◆ 字符型与整型运算时，字符型将被转换成整型
  - ◆ 整型与浮点型运算时，由整型向浮点型转换
  - ◆ 在做除法时，如果除数为 0，则商为空
  - ◆ 求余数时，如果被除数为带小数点的小数时，小数点后面的数字不参与运算，直接作为余数
- 字符型与整型运算时，字符型将被转换成整型
- 例如：
  - ◆ 表达式 “3.15+'300'” 的值为 303.15，“3.15+'300.1'” 的值为 303.25

```
MySQL 8.0 Co... - □ ×
mysql> select 3.15+'300';
+-----+
| 3.15+'300' |
+-----+
|          303.15 |
+-----+
1 row in set (0.00 sec)

mysql> select 3.15+'300.1';
+-----+
| 3.15+'300.1' |
+-----+
|           303.25 |
+-----+
1 row in set (0.00 sec)

mysql> █
```

- 整型与浮点型运算时，由整型向浮点型转换
- 例如：
  - ◆ 表达式“3+5.5”的值为 8.5

```
MySQL 8.0 Comma... - □ ×
mysql> select 3+5.5;
+-----+
| 3+5.5 |
+-----+
|      8.5 |
+-----+
1 row in set (0.00 sec)

mysql>
```

- 在做除法时，如果除数为 0，则商为空
- 例如：
  - ◆ 表达式“3/0”的值为“NULL”，表示空

```
MySQL 8.0 Co... - □ ×
mysql> select 3/0;
+-----+
| 3/0   |
+-----+
| NULL  |
+-----+
1 row in set (0.00 sec)

mysql>
```

- 求余数时，如果被除数为带小数点的小数时，小数点后面的数字不参与运算，直接作为余数
- 例如：
  - ◆ 表达式“3.5%2”的值为 1.5

```
MySQL 8.0 Co... - □ ×
mysql> select 3.5%2;
+-----+
| 3.5%2 |
+-----+
| 1.5   |
+-----+
1 row in set (0.00 sec)

mysql> █
```

### 关系运算符

- 关系运算符
  - ◆ 判断两个表达式之间的大小关系
  - ◆ BLOB 和 TEXT 等字符数据类型的表达式不可以使用关系运算符
- 关系运算符包括
  - ◆ 等于、大于、小于、大于或等于、小于或等于、不等于
  - ◆ 是否为空、在某个区间

运算符	说 明
=	等于,例如,age=23
>	大于,例如,price>100
<	小于
>=	大于或等于
<=	小于或等于
<>	不等于
!=	不等于(非 SQL 92 标准)
is null	字段是否为空
is not null	字段是否不为空
between...and	在某个区间

- 关系运算符表达式的结果
  - ◆ 运算结果为真时用 1 表示
  - ◆ 运算结果为假时用 0 表示
- 例如
  - ◆ 35>30 的值为 1
  - ◆ 35<=30 的值为 1
- 关于空的表达
  - ◆ 判断一条记录的某个字段是否为空，不能使用逻辑运算符等于“=”
  - ◆ 应该使用“is null”或者“is not null”
  - ◆ 例如：判断学生信息表中某条记录的字段 email 的值是否为空
  - ◆ 正确表达式：email is null
  - ◆ 错误表达式：email = null
- between...and 的应用
  - ◆ between...and 表示在某个范围内。如果在范围内结果为 1，否则结果为 0
  - ◆ 在计算时含等于，如 between 50 and 100，包含 50 和 100
- 例如：
  - ◆ 表达式“25 between 50 and 100”的值为 0
  - ◆ 表达式“50 between 50 and 100”的值为 1

```

MySQL 8.0 Command Line Client
mysql> select 25 between 50 and 100,50 between 50 and 100,100 between 50 and 100;
+-----+-----+-----+
| 25 between 50 and 100 | 50 between 50 and 100 | 100 between 50 and 100 |
+-----+-----+-----+
| 0 | 1 | 1 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

### 逻辑运算符

- 与运算规则：“全真为真，有假即假”
  - ◆ 真—非 0 的值
  - ◆ 假—0
- “全真为真”
  - ◆ 两边的两个操作数都为真时，与运算的结果是真
  - ◆ 例如：表达式“1 and 5”的值为 1
- “有假即假”
  - ◆ 两边的两个运算符只要有一个运算数是假，与运算的结果就假
  - ◆ 例如：表达式“0 and 5”的值为 0
- 或运算规则：“有真为真，全假为假”
- “全假为假”
  - ◆ 两边的两个操作数都为假时，运算的结果是假
  - ◆ 例如：表达式“0 or 0”的值为 0
- “有真即真”
  - ◆ 两边的两个运算符只要有一个运算数是真，与运算的结果就真
  - ◆ 例如：表达式“0 and 5”的值为 1
- 非运算规则：“颠倒黑白”
  - ◆ 将真变成假，将假变成真
- 例如：
  - ◆ 表达式：“not 5”的值为 0
  - ◆ 表达式：“not 0”的值为 1

```

MySQL 8.0 Command Lin...
mysql> select not 5,not 0;
+-----+-----+
| not 5 | not 0 |
+-----+-----+
| 0 | 1 |
+-----+-----+
1 row in set (0.00 sec)

mysql>

```

### 赋值运算符

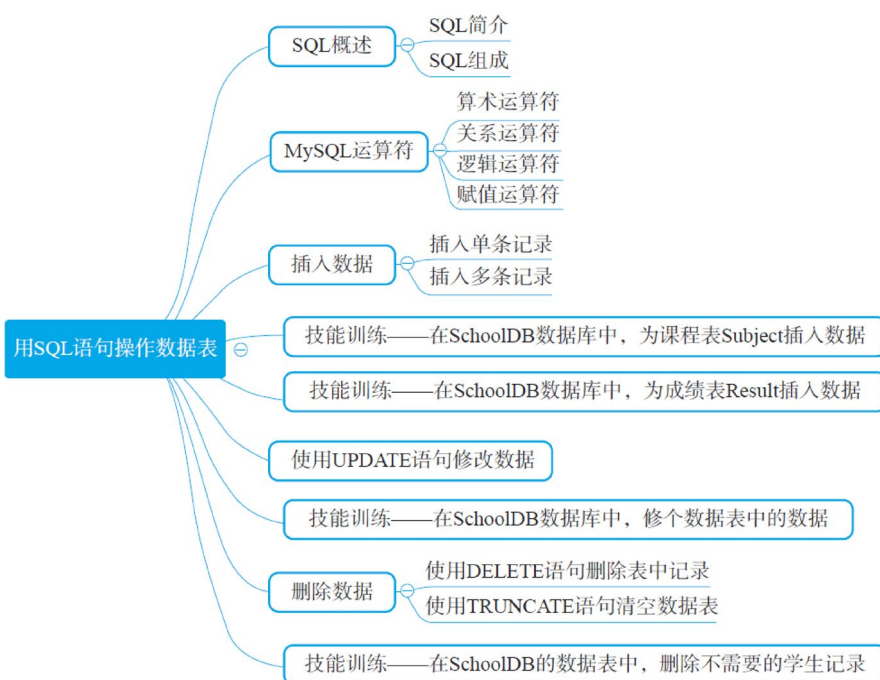
- 赋值运算符：“=”（等号）
  - ◆ 用于将一个数或变量或表达式赋值给另一个变量或者字段
- 例如：需要将姓名“张成叔”赋值给变量 name
  - ◆ 使用表示式：name='张成叔'
- 赋值运算符和比较运算符相等是同一个符号“=”
  - ◆ 需要根据语句的功能来确定是赋值，还是相等

思政点：通过讲解运算符的运算规则，告诫学生每人要懂得不以规则不能成方圆。每个人必须按照科学规律做事，怀着实实在在的看法做事，探究任何事情的客观规律，按部就班，引导学生有规则意识，要养成遵纪守法的好习惯，做到有井有条生活、工作。

### 学生完成实训

#### 三、课堂总结(5min)

全班共同对学习内容进行整合总结。（5min）



思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。  
——第 1-2 节结束——

#### 一、准备活动(5min)

- 1、活跃课堂气氛，自我介绍，学习通签到、点名。（1min）
- 2、回顾上节课知识点。（4min）

#### 二、发展活动(70min)

##### 1.教师讲解插入数据

- 插入数据是对数据表进行后续操作的基础
- 插入数据以行为单位
- 插入数据的方式
  - ◆ 可以一次插入一行记录

- ◆ 可以一次插入多行记录
- ◆ 可以将 SELECT 语句的查询结果批量插入到数据表中

## 2.插入单条数据

### 语法

```
INSERT [INTO] 表名 [(列名 1,列名 2,...)]
VALUES({表达式|DEFAULT}, ...),(... ),...;
```

### 分析

#### ■ 表名

- ◆ 即插入数据表的名称
- ◆ 表名前的“INTO”可以省略不写

### 语法

```
INSERT [INTO] 表名 [(列名 1,列名 2,...)]
VALUES({表达式|DEFAULT}, ...),(... ),...;
```

### 分析

#### ■ 列名：如果要给所有列都插入数据，列名可以省略

- ◆ 如果只给表的部分列插入数据，需要指定具体的列名
- ◆ 对于没有指出的列，将按下面的原则来处理
  - ✓ 具有 auto\_increment 属性的列，即标识列，系统自动生成序号值来唯一标识
  - ✓ 具有默认值的列，其值为默认值
  - ✓ 没有默认值的列，若允许为空值，则其值为空值；若不允许为空值，则出错
  - ✓ 类型为 timestamp 的列，系统自动赋值

#### ■ VALUES 子句：包含各列需要插入的数据清单

- ◆ 数据的顺序要与列的顺序相对应
- ◆ 值的类型要与列名的类型一致
- ◆ 若表名后不给出列名，则在 VALUES 子句中要给表中的每一列赋值，如果列值为空，则其值必须置为 NULL，否则会出错

#### ■ VALUES 子句：包含各列需要插入的数据清单

- ◆ 字符类型、日期类型的列，其值用单引号(')引起来，加以区别
- ◆ 如果只给表的部分列插入数据，需要指定具体的列名
- ◆ 不能为标识列指定值，因为它是自动增长的
- ◆ 不支持中文的符号，包括空格都必须在英文状态下输入

任务一：

### 语法

```
INSERT [INTO] 表名 [(列名 1,列名 2,...)]  
VALUES({表达式|DEFAULT}, ...),(... ),...;
```

### 讲解

- 为成绩表Result插入数据是不能给定“id”列的值
  - ◆ Id列为自动标识列，其值是自动生成的

```
INSERT Result(studentNo,subjectId,studentResult,examID)  
VALUES('G1263201',13,85,'2020-1-5')
```

### 【演示示例 4-1】

为年级表 Grade 插入数据

- 在数据库SchoolDB中
- 为年级表Grade添加6条数据

gradeId	gradeName
1	S1
2	S2
3	S3
4	S4
5	S5
6	S6

```
USE SchoolDB;  
INSERT INTO Grade (gradeId,gradeName) VALUES(1,'S1');  
INSERT INTO Grade (gradeId,gradeName) VALUES(2,'S2');  
INSERT INTO Grade (gradeId,gradeName) VALUES(3,'S3');  
INSERT INTO Grade (gradeId,gradeName) VALUES(4,'S4');  
INSERT INTO Grade (gradeId,gradeName) VALUES(5,'S5');  
INSERT INTO Grade (gradeId,gradeName) VALUES(6,'S6');  
SELECT * FROM Grade;
```

```

MySQL 8.0 Command Line Client
mysql> USE SchoolDB;
Database changed
mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(1,'S1');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(2,'S2');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(3,'S3');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(4,'S4');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(5,'S5');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Grade (gradeId,gradeName) VALUES(6,'S6');
Query OK, 1 row affected (0.00 sec)

mysql> select * from grade;
+----+-----+
| gradeId | gradeName |
+----+-----+
| 1       | S1       |
| 2       | S2       |
| 3       | S3       |
| 4       | S4       |
| 5       | S5       |
| 6       | S6       |
+----+-----+
6 rows in set (0.00 sec)

mysql>

```



**! 注意**

- 插入数据前要使用 **“USE SchoolDB;”** 数据库
- 年级名称为字符型，需要加**单引号**来区别，如'S1'
- 为了查看数据插入成功的效果，使用查询命令 **“SELECT”**
- 每条INSERT语句后面都要加 **“;”**，代表一条语句结束

**【技能训练 4-1】**

为课程表 Subject 插入数据

- 在 **“演示示例4-1”** 的基础上
- ◆ 打开SchoolDB数据库，为课程表Subject插入测试数据

subjectId	subjectName	classHour	gradeId
1	C 语言程序设计	64	1
2	大学英语	96	1
3	图形图像处理	64	1
4	网页设计	64	2
5	C# 面向对象设计	64	2
6	数据库设计与应用	96	2
7	Android 应用开发	64	3
8	Java 面向对象设计	64	3
9	Web 客户端编程	64	3
10	数据结构与算法	64	4
11	JavaWeb 应用开发	64	4
12	计算机网络基础	64	4

- 必须确保年级表Grade中数据完整
  - ◆ 涉及主外键约束问题
- 在插入数据之前
  - ◆ 使用 “SELECT \* FROM Subject;” 语句来查看并确认该表中无数据
  - ◆ 否则需要先使用 “DELETE” 命令来删除所有数据，确保数据的清洁
- 字符型的课程名称字段值都需要使用 “' ’” 括起来
  - ◆ 如：'C语言程序设计'

```

mysql> select * from subject;
+-----+-----+-----+-----+
| SubjectID | SubjectName          | ClassHour | GradeId |
+-----+-----+-----+-----+
| 1         | C语言程序设计        | 64        | 1       |
| 2         | 大学英语            | 96        | 1       |
| 3         | 图形图像处理        | 64        | 1       |
| 4         | 网页设计            | 64        | 2       |
| 5         | C#面向对象设计      | 64        | 2       |
| 6         | 数据库设计与应用    | 96        | 2       |
| 7         | Android应用开发     | 64        | 3       |
| 8         | Java面向对象设计    | 64        | 3       |
| 9         | Web客户端编程       | 64        | 3       |
| 10        | 数据结构与算法      | 64        | 4       |
| 11        | JavaWeb应用开发     | 64        | 4       |
| 12        | 计算机网络基础      | 64        | 4       |
| 13        | 软件测试技术        | 32        | 5       |
| 14        | Linux操作系统        | 32        | 5       |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>
  
```

### 插入多条数据

#### 语法

```
INSERT [INTO] 表名 [(列名清单)]
VALUES(列值清单 1),(列值清单 2),..., (列值清单 n);
```

#### 分析

##### ■ 列值清单

- ◆ 是指与列名对应的值
- ◆ 每一组值使用一对 “ ( ) ” 括起来，代表一条记录的值
- ◆ 多条记录值之间用 “ , ” 分隔

#### 【演示示例 4-2】

为年级表 Grade 插入数据

学号	密码	姓名	性别	年级	电话	地址	出生日期	邮箱	身份证号
G1263201	1	王子洋	男	5	18655290000	安徽省蚌埠市	1993/8/7	wzy@163.com	340423199308070000
G1263382	1	张琪	女	5	15678090000	安徽省合肥市	1993/5/7	zhangqi@126.com	340104199305070000
G1263458	1	项宇	男	5	18298000000		1992/12/10	xiangyu@163.com	340881199212100000
G1363278	1	胡保蜜	男	3	18965000000	北京市通州区	1993/6/29		346542199306290000
G1363300	1	王超	男	3	18123560000	上海市闵行区	1993/4/30	wangchao@126.com	340409199304300000
G1363301	1	党志鹏	男	3	15876550000	安徽省芜湖市	1994/12/20	dzp@suho.com	456765199412200000
G1363302	1	胡仲友	男	3	15032450000	安徽省淮南市	1994/6/13	12454344@qq.com	340043199406130000
G1363303	1	朱晓燕	女	3	15155670000	安徽省安庆市	1994/4/18	yanyan@163.com	
G1463337	1	高伟	男	1	18390870000	山东省济南市	1995/6/7		450504199506070000
G1463342	1	胡俊文	男	1	13976870000	河南省郑州市	1995/4/20		340408199504200000
G1463354	1	陈大伟	男	1	15067340000	四川省成都市	1995/8/23	wangkuan@163.com	340422199508230000
G1463358	1	温海南	男	1	18028760000		1995/1/30		450560199501300000
G1463383	1	钱嫣然	女	1	18656430000	安徽省潜山市	1994/1/14	yanran@126.com	340408199401140000
G1463388	1	卫丹丹	女	1	15134870000	广东省深圳市	1995/4/17		340411199504170000

**注意**

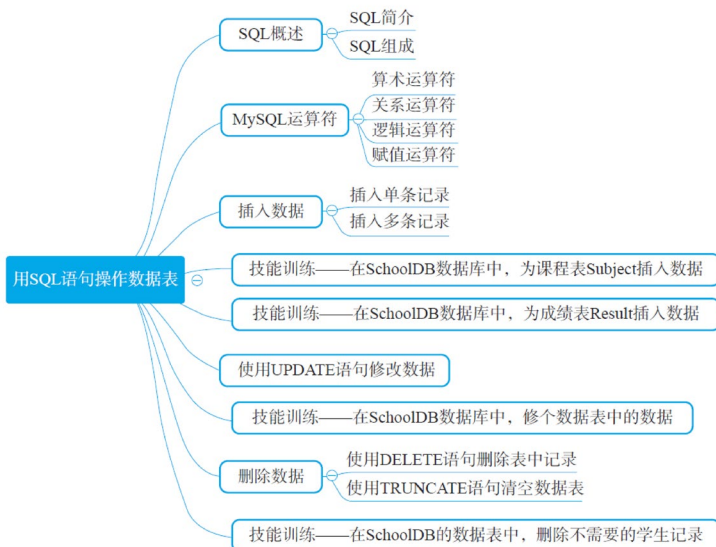
- 在插入数据之前使用 **"SELECT \* FROM Student;"** 语句来查看并确认该表中无数据
- 如果已经存在数据需要先使用 **"DELETE"** 命令来删除所有数据，确保数据的清洁
- 对于未指定值的列，需要使用 **"DEFAULT"** 或者 **"NULL"** 表示，确保值清单中值的个数一定要与列清单中列的数量相同
- 学号列为**主键列**，不能出现相同的学号
- 主外键约束问题
  - ◆ 学生信息表Student中的年级编号gradeId列要**参照**年级表Grade中的年级编号gradeId列
  - ◆ 在学生信息表Student中插入数据前，必须确保年级表Grade中已经有了可参照的数据，否则插入数据失败
- 唯一约束问题
  - ◆ identityCard列设置了 **"唯一约束"**，表中允许出现一个空值 "NULL"
  - ◆ 如果给出的列值清单中出现第2个空值 "NULL"，则会提示语法错误

思政点：使用不同的方法创建记录，培养利用多种思路解决问题的能力，插入数据时要遵照规则和数据类型要求，引导学生理解做任何事情都要知深浅，心中有‘度’。这里的度是做人的标准，意味着人应当遵循的规则。

学生完成实训

三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)



思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。  
——第 3-4 节结束——

### 一、准备活动(5min)

- 1、活跃课堂气氛，自我介绍，学习通签到、点名。(1min)
- 2、回顾上节课知识点。(4min)

### 二、发展活动(70min)

#### 1.教师讲解插入查询的结果

##### 语法

**INSERT [INTO] 表名 [(列名 1,列名 2,...)]  
SELECT 查询语句;**

##### 分析

##### ■ 表名

- ◆ 必须是已经创建的数据表
- ◆ 并且和查询使用的是同一个数据库中表

##### ■ SELECT查询语句

- ◆ 查询语句将在下一章中学习
- ◆ 一般格式

**SELECT \* FROM 表名 WHERE 条件;**

三种插入数据形式比较

**INSERT [INTO] 表名 [(列名 1,列名 2,...)]  
VALUES({表达式|DEFAULT}, ..., (... ),...);**

**INSERT [INTO] 表名 [(列名清单)]  
VALUES(列值清单 1),(列值清单 2),..., (列值清单 n);**

**INSERT [INTO] 表名 [(列名 1,列名 2,...)]  
SELECT 查询语句;**

思政点：使用不同的方法创建记录，培养利用多种思路解决问题的能力。

**【演示示例 4-3】**

将查询结果插入到数据表中

- 数据库 SchoolDB 中
- 新建一张新表 Result92，其表结构与 Result 相同
- 用来存储 Result 表中成绩不及格的信息

USE SchoolDB;

CREATE TABLE Result92 LIKE Result;

INSERT INTO Result92

SELECT \* FROM Result WHERE studentResult<60;

SELECT \* FROM Result92;

MySQL 8.0 Command Line Client  
Records: 8 Duplicates: 0 Warnings: 0

```
mysql> select * from result92;
```

id	studentno	subjectId	studentResult	examDate
88	G1263382	14	56.00	2019-11-16 00:00:00
90	G1263458	14	0.00	2019-11-16 00:00:00
91	G1363278	7	55.00	2020-01-05 00:00:00
95	G1363300	8	49.00	2020-01-07 00:00:00
99	G1363301	9	55.00	2019-11-20 00:00:00
102	G1363302	8	56.00	2020-01-07 00:00:00
110	G1463337	3	56.00	2020-01-09 00:00:00
113	G1463354	1	52.00	2019-11-20 00:00:00

8 rows in set (0.00 sec)

```
mysql>
```

**! 注意**

- 创建数据表Result92前要确保该表不存在
  - ◆ 如果存在，再查看表中是否有数据
  - ◆ 如果不是需要的数据，可以先删除已经存在的表Result92
- 通过复制创建的新数据表Result92没有建立外键约束
  - ◆ 与学生信息表Student和课程表Subject表之间没有建立起参照关系

**【技能训练 4-3】**

为成绩表 Result 插入数据

- 在“演示示例 4-1”、“技能训练 4-1”和“演示示例 4-2”基础上
- 打开 SchoolDB 数据库，为成绩表 Result 插入测试数据
- 数据见指定教 P87 页，表 4-7

studentNo	subjectId	studentResult	examDate
G1263201	13	76	2019/11/15
G1263201	14	88	2019/11/16
G1263382	13	79	2019/11/15
G1263382	14	56	2019/11/16
G1263458	13	92	2019/11/15
G1263458	14	0	2019/11/16
G1363278	7	55	2020/1/5
G1363278	8	78	2020/1/7
G1363278	9	76	2020/1/6
G1363300	7	83	2020/1/5
G1363300	8	49	2020/1/7
G1363300	9	64	2020/1/6
G1363301	7	65	2020/1/5
G1363301	8	87	2020/1/7
G1363301	9	55	2019/11/20
G1363301	9	90	2020/1/6
G1363302	7	80	2020/1/5
G1363302	8	56	2020/1/7
G1363302	9	87	2020/1/6
G1363303	7	61	2020/1/5
G1363303	8	87	2020/1/7
G1363303	9	81	2020/1/6
G1463337	1	82	2019/11/20
G1463337	1	90	2020/1/5
G1463337	2	92	2019/1/8
G1463337	3	56	2020/1/9
G1463342	1	86	2020/1/5
G1463342	2	68	2019/1/8
G1463354	1	52	2019/11/20
G1463354	1	67	2020/1/5
G1463354	2	68	2020/1/7
G1463354	3	75	2020/1/9
G1463358	1	65	2020/1/5
G1463358	2	92	2020/1/7
G1463383	1	88	2019/11/20
G1463383	1	87	2020/1/5
G1463383	2	92	2020/1/7
G1463383	3	89	2020/1/9
G1463388	1	80	2019/11/20
G1463388	1	78	2020/1/5
G1463388	2	92	2020/1/7
G1463388	3	83	2020/1/9

## ■ 在插入数据之前

- ◆ 使用 “SELECT \* FROM Result;” 确认该表中无数据
- ◆ 如有数据，先使用 “DELETE” 命令来删除所有数据，确保数据清洁

## ■ Result表中，id列为标识列

- ◆ 该列的值由系统自动生成，用户不能为该列赋值，也无法控制该列
- ◆ 如：出现清单 “(100,'G1263201',14,88,'2019/11/16')” 时，执行

MySQL 8.0 Command Line Client  
Records: 42 Duplicates: 0 Warnings: 0

```
mysql> select * from result;
```

id	studentno	subjectId	studentResult	examDate
85	G1263201	13	76.00	2019-11-15 00:00:00
86	G1263201	14	88.00	2019-11-16 00:00:00
87	G1263382	13	79.00	2019-11-15 00:00:00
88	G1263382	14	56.00	2019-11-16 00:00:00
89	G1263458	13	92.00	2019-11-15 00:00:00
90	G1263458	14	0.00	2019-11-16 00:00:00
91	G1363278	7	55.00	2020-01-05 00:00:00
92	G1363278	8	78.00	2020-01-07 00:00:00
93	G1363278	9	76.00	2020-01-06 00:00:00
94	G1363300	7	83.00	2020-01-05 00:00:00
95	G1363300	8	49.00	2020-01-07 00:00:00
96	G1363300	9	64.00	2020-01-06 00:00:00
97	G1363301	7	65.00	2020-01-05 00:00:00
98	G1363301	8	87.00	2020-01-07 00:00:00
99	G1363301	9	55.00	2019-11-20 00:00:00
100	G1363301	9	90.00	2020-01-06 00:00:00
101	G1363302	7	80.00	2020-01-05 00:00:00
102	G1363302	8	56.00	2020-01-07 00:00:00
103	G1363302	9	87.00	2020-01-06 00:00:00
104	G1363303	7	61.00	2020-01-05 00:00:00
105	G1363303	8	87.00	2020-01-07 00:00:00
106	G1363303	9	81.00	2020-01-06 00:00:00
107	G1463337	1	82.00	2019-11-20 00:00:00
108	G1463337	1	90.00	2020-01-05 00:00:00
109	G1463337	2	92.00	2019-01-08 00:00:00
110	G1463337	3	56.00	2020-01-09 00:00:00
111	G1463342	1	86.00	2020-01-05 00:00:00
112	G1463342	2	68.00	2019-01-08 00:00:00
113	G1463354	1	52.00	2019-11-20 00:00:00
114	G1463354	1	67.00	2020-01-05 00:00:00
115	G1463354	2	68.00	2020-01-07 00:00:00
116	G1463354	3	75.00	2020-01-09 00:00:00
117	G1463358	1	65.00	2020-01-05 00:00:00
118	G1463358	2	92.00	2020-01-07 00:00:00
119	G1463383	1	88.00	2019-11-20 00:00:00
120	G1463383	1	87.00	2020-01-05 00:00:00
121	G1463383	2	92.00	2020-01-07 00:00:00
122	G1463383	3	89.00	2020-01-09 00:00:00
123	G1463388	1	80.00	2019-11-20 00:00:00
124	G1463388	1	78.00	2020-01-09 00:00:00
125	G1463388	2	92.00	2020-01-07 00:00:00
126	G1463388	3	83.00	2020-01-09 00:00:00

42 rows in set (0.00 sec)

```
mysql>
```

## ■ 关于主外键约束问题

- ◆ 成绩表Result中的studentNo列参照Student表中的studentNo
- ◆ 成绩表Result中subjectId列参照Subject表中的subjectId列

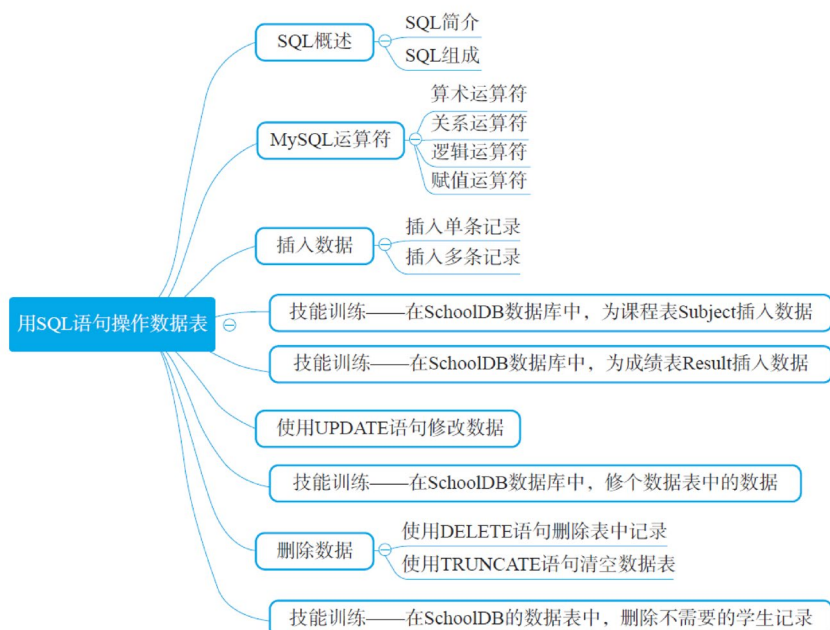
## ■ studentResult列设置了 “>=0” 的检查约束

- ◆ 该列不能出现空值
- ◆ “G1263458” 同学的成绩为0分，不能用NULL代替

学生完成实训

### 三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)



**思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。**

5-6 节结束——

### 一、准备活动(5min)

1、活跃课堂气氛，自我介绍，学习通签到、点名。(1min)

2、回顾上节课知识点。(4min)

### 二、发展活动(70min)

#### 1.教师讲解 使用 UPDATE 语句修改数据

##### ■ 应用场合

- ◆ 学生信息、课程信息和成绩信息等数据表中需要维护的数据发生变化
- ◆ 对已经存在的数据进行修改，添加和删除记录

##### ■ 使用语句：UPDATE

- ◆ 可以用来修改单个表中的数据
- ◆ 可以修改多个表中的数据

##### ≡ 语法

#### UPDATE 表名

SET 列名 1=表达式 1,列名 2=表达式 2,...

[WHERE 更新条件]

##### ■ 表名

- ◆ 即需要修改数据的数据表名称

##### ■ SET 子句

- ◆ 根据 WHERE 子句中指定的条件对符合条件的数据进行修改
- ◆ 若语句中不设定 WHERE 子句，则更新表中所有行
- ◆ 指定数据修改的内容

- ◆ 可以多个数据列的更新值，多个数据列之间以逗号“,”分隔
  - 列名和表达式列表
    - ◆ 列名 1、列名 2...为要修改列值的列名
    - ◆ 表达式 1、表达式 2...可以是常量、变量或表达式
    - ◆ 可以同时修改所在数据行的多个列值，中间用逗号隔开
- 【演示示例 4-4】更新数据表中的数据

🔍需求分析

- 数据库SchoolDB中
- 复制学生信息Student的结构和数据得到新表Student93
- 将Student93中所有邮箱地址为空的学生的邮箱都改为“未注册邮箱@163.com”

🔍代码分析

```
USE SchoolDB;
CREATE TABLE Student93 LIKE Student;
INSERT INTO Student93 SELECT * FROM Student;
SELECT * FROM Student93 WHERE email is null;
UPDATE Student93 SET email='未注册邮箱@163.com'
WHERE email is null;
SELECT * FROM Student93 WHERE email='未注册邮箱@163.com';
```

MySQL 8.0 Command Line Client

```
mysql> SELECT * FROM Student93 WHERE email is null;
```

studentno	loginPwd	studentName	sex	gradeId	phone	address	bornDate	email	identityCard
G1363278	1	胡保蜜	男	3	18965000000	北京市通州区	1993-06-29	NULL	346542199306290000
G1463337	1	高伟	男	1	18390870000	安徽省灵璧县	1995-06-07	NULL	450504199506070000
G1463342	1	胡俊文	男	1	13976870000	安徽省定远县	1995-04-20	NULL	340408199504200000
G1463358	1	温海南	男	1	18028760000	地址不详	1995-01-30	NULL	450560199501300000
G1463388	1	卫丹丹	女	1	15134870000	安徽省凤阳县	1995-04-17	NULL	340411199504170000

5 rows in set (0.00 sec)

```
mysql> UPDATE Student93 SET email='未注册邮箱@163.com'
WHERE email is null;
```

Query OK, 5 rows affected (0.01 sec)  
Rows matched: 5 Changed: 5 Warnings: 0

修改部分同学邮箱账号前后的对比结果!

```
mysql> SELECT * FROM Student93 WHERE email='未注册邮箱@163.com';
```

studentno	loginPwd	studentName	sex	gradeId	phone	address	bornDate	email	identityCard
G1363278	1	胡保蜜	男	3	18965000000	北京市通州区	1993-06-29	未注册邮箱@163.com	346542199306290000
G1463337	1	高伟	男	1	18390870000	安徽省灵璧县	1995-06-07	未注册邮箱@163.com	450504199506070000
G1463342	1	胡俊文	男	1	13976870000	安徽省定远县	1995-04-20	未注册邮箱@163.com	340408199504200000
G1463358	1	温海南	男	1	18028760000	地址不详	1995-01-30	未注册邮箱@163.com	450560199501300000
G1463388	1	卫丹丹	女	1	15134870000	安徽省凤阳县	1995-04-17	未注册邮箱@163.com	340411199504170000

5 rows in set (0.00 sec)

```
mysql>
```

- 为了保护数据库中的原始数据，将原始数据表复制一份
  - 在复制后的数据表中进行操作，不影响原表中的数据
  - 在修改数据的前和后都使用“SELECT”命令详细数据
  - 对照检查修改前后的数据是否符合要求
- 思政点：更新数据会破坏原有的数据，所以，要养成备份数据的好习惯。为客户提供容错机制，引导学生养成严于律己，宽以待人的处事原则。

【技能训练 4-3】修改数据中数据

- 在 SchoolDB 数据库中
- 修改学号为 G1363300 的学生地址为“山东省济南市文化路 2

号院”

- 修改学号为 G1363301 的学生的所属年级为 2
- 修改“大学英语”课程的学时数为 55
- 将 2020 年 1 月 7 日考试的“Java 面向对象设计”课程分数低于 60 分的学生全部提高 5 分
- 将学号为 G1363300 的学生在 2020 年 1 月 5 日考试的“Android 应用开发”课程的分数修改为 80
- 将邮件账号为空的学生的邮件账号统一修改为“未知@”
  - 在修改数据之前，先打开数据库
  - 修改学生的住址，参考语句

```
UPDATE Student
```

```
SET Address='山东省济南市文化路 2 号院'
```

```
WHERE StudentNo='G1363300';
```

### ■ 修改“Java面向对象设计”课程的分数

- ◆ 通过“SELECT”语句查询课程表Subject
- ◆ 查询得到课程“Java面向对象设计”的课程编号为8
- ◆ 再使用“UPDATE”语句对Result表进行成绩的更新
- ◆ 修改前后使用SELECT语句查询结果，对比分析

### ■ 修改“Java面向对象设计”课程的分数

```
SELECT * FROM Subject;
```

```
SELECT * FROM Result
```

```
WHERE examDate='2020-1-7' AND subjectId=8 AND studentResult<60;
```

```
UPDATE Result SET studentResult=studentResult+5
```

```
WHERE examDate='2020-1-7' AND subjectId=8 AND studentResult<60;
```

```
SELECT * FROM Result
```

```
WHERE examDate='2020-1-7' AND subjectId=8 AND studentResult<65;
```

### ■ 邮箱为空的表达方式

```
UPDATE Student SET email='未知@' WHERE email IS NULL
```

### ■ 多个条件的组合

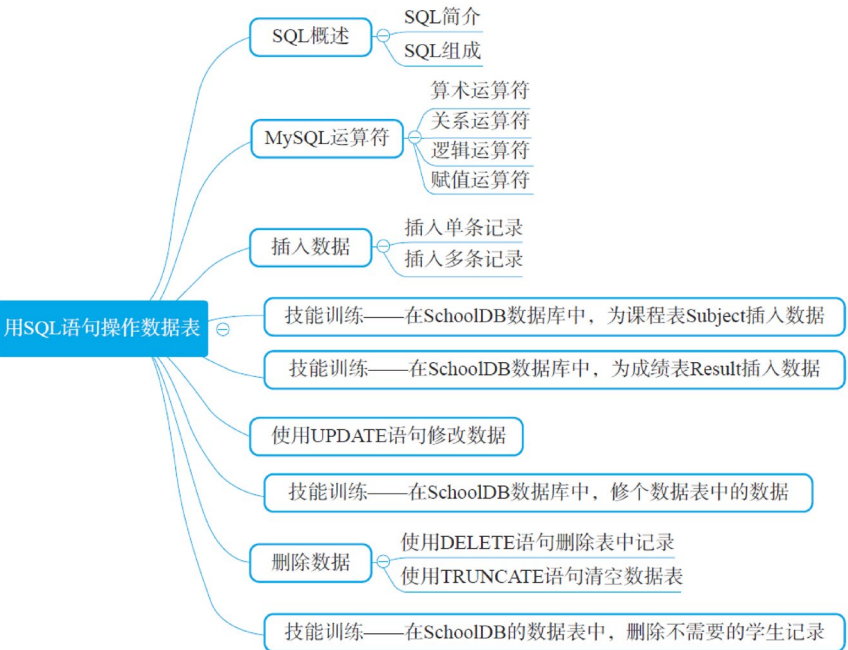
- ◆ 有学号、日期、课程的限制，又有分数的限制
- ◆ 需要使用逻辑与运算符“AND”来连接多个条件

```
WHERE examDate='2020-1-7' AND subjectId=8 AND studentResult<60
```

学生完成实训

三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)

	 <p>思政点：下课，关闭电脑、整理桌面，养成良好的行为习惯。 5-6 节结束——</p>
学习评量	<p>1.阶段测试三，共计 100 分。 2.评量方法：超星网络教学平台阶段测试。</p>
课后作业	<p>学习通中视频学习</p>
教学反思	<p>目标或能力指针达成度、学习表现、内容、方法、资源、业师协同等向度(至少写三项)</p> <p>1.目标或能力指标达成度 全班 90%学生达到了教学目标和能力指标。</p> <p>2.学习表现 通过课前任务的布置，学生按要求完成任务单，课上教师提出问题，学生通过讨论，解决问题，个别学生在数据库更新过程中出现的问题，经过指导和同学讨论完成任务。</p> <p>3.资源 (1) 学生总结本单元学习情况，教师点评，指出存在的问题及解决方法，做好资料积累工作，保存好各位同学操作的文件。 (2) 对学生操作中出现的记录，以便对下一年度的学生操作指出注意事项。</p>

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元四 单表查询和模糊查询	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)掌握数据库和表的创建、修改和删除。 (2)能对数据表记录进行增、删、改。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 理解使用 sql 语句进行数据的查询的语法和机制。 2. 能根据查询要求编写相应的 sql 语句,完成查询。			C1	D1
教学目标	<p>知识</p> <ol style="list-style-type: none"> <li>理解查询处理的机制。</li> <li>掌握常用的系统函数。</li> <li>掌握 select 语句查询的语法。</li> <li>掌握等值连接的原理。</li> <li>理解和掌握分组查询和连接查询的机制。</li> </ol> <p>能力</p> <ol style="list-style-type: none"> <li>熟练使用 select 语句单表、多表的查询。</li> <li>使用 LIKE\BETWEEN\IN 等进行模糊查询的能力。</li> <li>使用聚合函数统计和汇总查询的能力。</li> <li>对查询结果进行排序的能力。</li> <li>具备使用 GROUP BY 和 HAVING 子句实现分组和筛选查询。</li> </ol> <p>素质</p> <ol style="list-style-type: none"> <li>提升精益求精的工匠精神。</li> <li>培养良好的编程习惯和规范。</li> </ol> <p>思政</p> <ol style="list-style-type: none"> <li>培养将所学知识应用于服务社会的责任与奉献精神。</li> <li>引导学生树立合法查询的意识,明确哪些数据的查询属于违法行为。</li> <li>发扬工匠精神,将执着专注、精益求精、一丝不苟、追求卓越的理念融入到单元开发中。</li> </ol>				
教学重难点	<p>教学重点:</p> <ol style="list-style-type: none"> <li>使用查询条件的构造</li> <li>灵活运用聚合函数对成绩信息进行汇总统计</li> </ol>				

	<b>教学难点：</b> 1、使用表达式、运算符和函数解决实际问题
<b>教师课前准备</b>	1. 多媒体机房安装配置好上课所需的开发环境。 2. 设计好课堂教学环节。 3. 准备好教学案例和教学课件等材料。
<b>学生课前准备</b>	查阅资料，观看学习通技能点视频
<b>活动历程</b> (强调成果导向、主要教学法的历程，含辅助手段、时间分配)	<div style="text-align: center;"> </div> <p>(一) 任务一 查询基础</p> <p>一、准备活动(5min)</p> <p>1、活跃课堂气氛，自我介绍，学习通签到、点名。(1min)</p> <p>2、回顾上节课知识点。(4min)</p> <p>二、发展活动(70min)</p> <p>1.教师讲解查询</p> <ul style="list-style-type: none"> <li>■ 查询是针对表中已经存在的数据行而言的，可理解为“筛选”       <ul style="list-style-type: none"> <li>◆ 从数据表中将符合条件的数据<b>抽取</b>出来</li> </ul> </li> <li>■ 数据表在接受查询请求时，它将逐行判断是否符合查询条件       <ul style="list-style-type: none"> <li>◆ 如果符合查询条件就提取出来，然后把所有选中的行组织在一起，构成查询的结果，称作：<b>记录集</b></li> <li>◆ 记录集的结构类似于表结构</li> </ul> </li> <li>■ 在查询记录集上可以再次进行查询</li> </ul>

## 语法

```
SELECT *|字段列表  
FROM 表名  
[WHERE 查询条件]  
[ORDER BY 排序的列名 [ASC|DESC];
```

## 分析

### ■ ELECT子句

- ◆ 指定要查看的列，即字段，即列出查询结果中要显示的字段名
- ◆ 字段列表可以是多个列名或表达式，之间用逗号间隔
- ◆ 表达式可以是列名、函数或常数的列表
- ◆ 如果要查看所有列，使用 "\*" 表示

### ■ FROM 子句

- ◆ 指定要查询的表，可以是表名或视图名
- ◆ FROM 中可以是多个表名，它们之间用逗号间隔

### ■ WHERE 子句

- ◆ 用于给出限制查询的条件或多个表的连接条件
- ◆ 根据具体的查询要求进行选择使用
- ◆ 可以为空，即查询表中所有记录

### ■ ORDER BY 子句

- ◆ 用于对查询的结果进行排序
- ◆ 指明排序的列名，可以是多列，使用逗号分隔
- ◆ 排序时可以指定升序或者降序，默认为升序

## 使用 SELECT 语句进行查询

### 讲解

#### ■ 查询所有的数据行和列

- ◆ SELECT后面加上 "\*" ，代表所有的列
- ◆ 不需要加任何条件，即：WHERE子句为空

### 演示

#### ■ 查询课程表Subject中所有的课程信息

```
USE SchoolDB;  
SELECT * FROM Subject;
```

```
MySQL 8.0 Command Line Client
mysql> SELECT * FROM Subject;
+-----+-----+-----+-----+
| subjectId | subjectName | classHour | gradeId |
+-----+-----+-----+-----+
| 1 | C语言程序设计 | 64 | 1 |
| 2 | 大学英语 | 96 | 1 |
| 3 | 图形图像处理 | 64 | 1 |
| 4 | 网页设计 | 64 | 2 |
| 5 | C#面向对象设计 | 64 | 2 |
| 6 | 数据库设计与应用 | 96 | 2 |
| 7 | Android应用开发 | 64 | 3 |
| 8 | Java面向对象设计 | 64 | 3 |
| 9 | Web客户端编程 | 64 | 3 |
| 10 | 数据结构与算法 | 64 | 4 |
| 11 | JavaWeb应用开发 | 64 | 4 |
| 12 | 计算机网络基础 | 64 | 4 |
| 13 | 软件测试技术 | 32 | 5 |
| 14 | Linux操作系统 | 32 | 5 |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)
mysql>
```

 讲解

■ 查询指定列

- ◆ SELECT后面加上列出需要列名，即显示的列名
- ◆ 列名之间用逗号分隔

 演示

■ 查询Student表中所有学生的学号、姓名和邮箱账号信

```
USE SchoolDB;
```

```
SELECT studentNo,studentName,email FROM Student;
```

```
MySQL 8.0 Command Line Client
mysql> SELECT studentNo,studentName,email FROM Student;
+-----+-----+-----+
| studentNo | studentName | email |
+-----+-----+-----+
| G1263201 | 王子洋 | wzy@163.com |
| G1263382 | 张琪 | zhangqi@126.com |
| G1263458 | 项宇 | xiangyu@163.com |
| G1363278 | 胡保蜜 | NULL |
| G1363300 | 王超 | wangchao@126.com |
| G1363301 | 党志鹏 | dzp@sohu.com |
| G1363302 | 胡仲友 | 12454344@qq.com |
| G1363303 | 朱晓燕 | yanyan@163.com |
| G1463337 | 高伟 | NULL |
| G1463342 | 胡俊文 | NULL |
| G1463354 | 陈大伟 | wangkuan@163.com |
| G1463358 | 温海南 | NULL |
| G1463383 | 钱嫣然 | yanran@126.com |
| G1463388 | 卫丹丹 | NULL |
+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>
```

**讲解**

■ 改变查询结果的列标题

- ◆ 列名一般要求使用英文名称
- ◆ 但是在中国，用户更加希望查询的结果标题用汉字表示，一目了然

**语法**

字段名称 AS 别名

■ 查询Student表中所有学生的学号、姓名和邮箱账号信息

- ◆ 要求使用中文汉字为列标题

```
USE SchoolDB;
SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'
FROM Student;
```

```

MySQL 8.0 Command Line Client
mysql> SELECT studentNo AS '学号', studentName AS '姓名', email AS '邮箱账号'
-> FROM Student;

```

学号	姓名	邮箱账号
G1263201	王子洋	wzy@163.com
G1263382	张琪	zhangqi@126.com
G1263458	项宇	xiangyu@163.com
G1363278	胡保蜜	NULL
G1363300	王超	wangchao@126.com
G1363301	党志鹏	dzp@sohu.com
G1363302	胡仲友	12454344@qq.com
G1363303	朱晓燕	yanyan@163.com
G1463337	高伟	NULL
G1463342	胡俊文	NULL
G1463354	陈大伟	wangkuan@163.com
G1463358	温海南	NULL
G1463383	钱嫣然	yanran@126.com
G1463388	卫丹丹	NULL

14 rows in set (0.00 sec)

```

mysql>

```

自定义汉字列名!

 讲解

■ 限制查询结果返回记录的行数

- ◆ 只需要显示返回结果的部分记录行
- ◆ 使用LIMIT子句来限定

 语法

LIMIT 行数

或

LIMIT 起始行的偏移量,返回的记录行数

■ 查询Student表中的前4行以及第6行开始的4行记录

- ◆ 显示学号、姓名和邮箱账号信息
- ◆ 使用中文汉字为列标题

```

USE SchoolDB;
SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'
FROM Student
LIMIT 4;
SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'
FROM Student
LIMIT 5,4;

```

```

MySQL 8.0 Command Line Client

mysql> SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'
-> FROM Student
-> LIMIT 4;
+----+-----+-----+
| 学号 | 姓名 | 邮箱账号 |
+----+-----+-----+
| G1263201 | 王子洋 | wzy@163.com |
| G1263382 | 张琪 | zhangqi@126.com |
| G1263458 | 项宇 | xiangyu@163.com |
| G1363278 | 胡保蜜 | NULL |
+----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'
-> FROM Student
-> LIMIT 5,4;
+----+-----+-----+
| 学号 | 姓名 | 邮箱账号 |
+----+-----+-----+
| G1363301 | 党志鹏 | dzp@sohu.com |
| G1363302 | 胡仲友 | 12454344@qq.com |
| G1363303 | 朱晓燕 | yanyan@163.com |
| G1463337 | 高伟 | NULL |
+----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

查询前4条记录的结果!

查询第6条记录开始的4条记录的结果!

**讲解**

■ 消除查询结果的重复行

- ◆ 重复行：查询结果中可能存在内容相同的行
- ◆ 将DISTINCT关键字写在SELECT字段列表所有列名的前面

**演示**

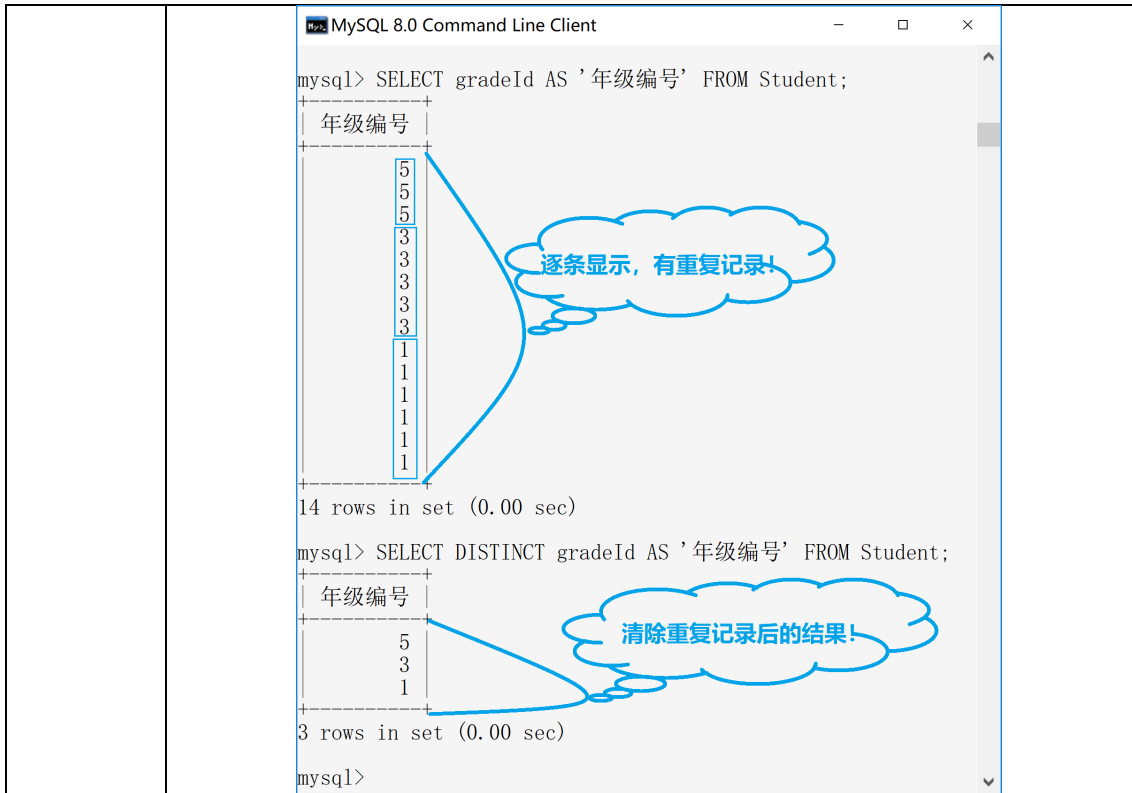
■ 查询Student表中的年级编号

- ◆ 要求删除结果集中的重复记录

```

USE SchoolDB;
SELECT gradeId AS '年级编号' FROM Student;
SELECT DISTINCT gradeId AS '年级编号' FROM Student;

```



**讲解**

■ 查询中使用计算列

- ◆ 需要对查询结果进行简单计算
- ◆ 在SELECT子句中可以使用算术运算符 (+、-、\*、/ 和%等)

**演示**

■ 查询表Subject表中所有的课程信息

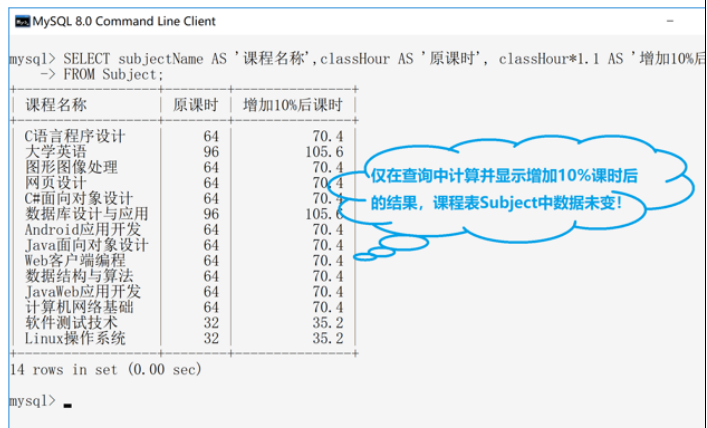
- ◆ 学校考虑为所有课程增加课时，增加比例的原课时的10%
- ◆ 只需要显示课程名称、原课时和增加后课时信息

USE SchoolDB;

SELECT subjectName AS '课程名称',classHour AS '原课时',

classHour\*1.1 AS '增加 10%后课时'

FROM Subject;



## ■ 使用WHERE子句查询部分行

- ◆ 部分行：查询某门课及格的学生信息
- ◆ WHERE子句用来限制查询结果的数据行
- ◆ WHERE后面是条件表达式，表示满足条件表达式的记录行才显示

## ■ 条件表达式

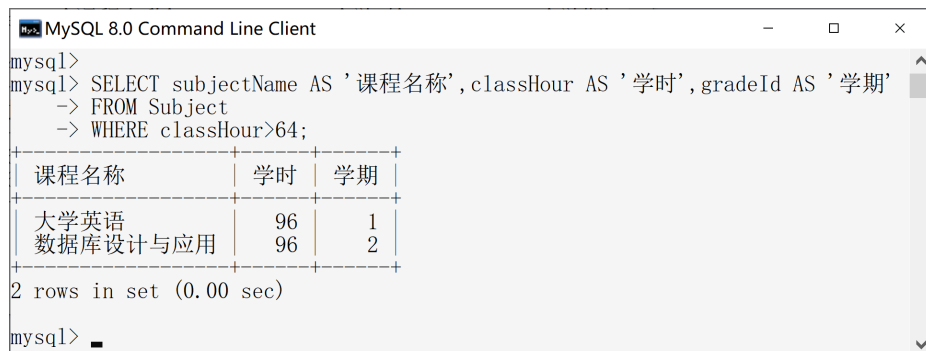
- ◆ 通常由一个或多个逻辑表达式组成
- ◆ 而逻辑表达式通常会涉及比较运算符、逻辑运算符、模式匹配等
- ◆ 要求表达式的值逻辑真或者逻辑假

## ■ 查询Subject表中学时超过64的课程信息

- ◆ 显示课程名称、学时和学期
- ◆ 使用中文汉字标题

USE SchoolDB;

```
SELECT subjectName AS '课程名称',classHour AS '学时',gradeId AS '学期'  
FROM Subject  
WHERE classHour>64;
```



```
MySQL 8.0 Command Line Client  
mysql>  
mysql> SELECT subjectName AS '课程名称',classHour AS '学时',gradeId AS '学期'  
-> FROM Subject  
-> WHERE classHour>64;
```

课程名称	学时	学期
大学英语	96	1
数据库设计与应用	96	2

```
2 rows in set (0.00 sec)  
mysql> _
```

## 🔊 讲解

### ■ 查询空值

- ◆ 空值表示未知的不确定的值，不是空格也不是空字符串
- ◆ 当需要判定一个表达式的值是否为空值时，使用IS NULL关键字

## ☰ 语法

表达式 IS [NOT] NULL

### ■ 功能分析

- ◆ 当不使用NOT时，若表达式为空值，则返回TRUE，否则返回FALSE
- ◆ 当使用NOT时，结果刚好相反查询空值

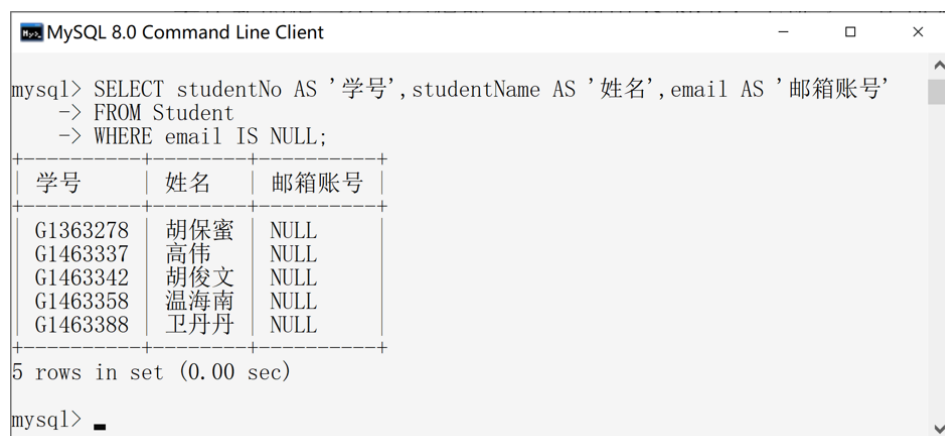
## 演示

### ■ 查询Student表中邮箱账号为空的学生信息

- ◆ 显示学号、姓名和邮箱账号信息
- ◆ 要求使用中文汉字为列标题

```
USE SchoolDB;
```

```
SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱'  
FROM Student  
WHERE email IS NULL;
```



```
MySQL 8.0 Command Line Client  
mysql> SELECT studentNo AS '学号',studentName AS '姓名',email AS '邮箱账号'  
-> FROM Student  
-> WHERE email IS NULL;
```

学号	姓名	邮箱账号
G1363278	胡保蜜	NULL
G1463337	高伟	NULL
G1463342	胡俊文	NULL
G1463358	温海南	NULL
G1463388	卫丹丹	NULL

```
5 rows in set (0.00 sec)  
mysql>
```

思政点：发扬工匠精神，将执着专注、精益求精、一丝不苟、追求卓越的理念融入到单元开发中

### 学生完成实训任务

#### 三、课堂总结(5min)

全班共同对学习内容进行整合总结。(5min)

——第 1-2 节结束——

#### (二) 任务二：简单查询和排序(80 分钟)

##### 一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。(1min)
2. 学习通预习技能点的学习情况检查。(1min)
3. 复习。(1min)

##### 二、发展活动(35min)

1. 新课导入(5min)

## 查询

- 查询是针对表中已经存在的数据行而言的，可理解为“筛选”
  - ◆ 从数据表中将符合条件的数据**抽取**出来
- 数据表在接受查询请求时，它将逐行判断是否符合查询条件
  - ◆ 如果符合查询条件就提取出来，然后把所有选中的行组织在一起，构成查询的结果，称作：**记录集**
  - ◆ 记录集的结构类似于表结构
- 在查询记录集上可以再次进行查询

### 3. 课程新授（10 分钟）

## 使用SELECT语句进行查询

### 语法

```
SELECT *|字段列表  
FROM 表名  
[WHERE 查询条件]  
[ORDER BY 排序的列名 [ASC|DESC];
```

### 分析

#### ■ FROM子句

- ◆ 指定要查询的表，可以是表名或视图名
- ◆ FROM中可以是多个表名，它们之间用逗号间隔

### 分析

#### ■ WHERE子句

- ◆ 用于给出限制查询的条件或多个表的连接条件
- ◆ 根据具体的查询要求进行选择使用
- ◆ 可以为空，即查询表中所有记录

### 分析

#### ■ ORDER BY子句

- ◆ 用于对查询的结果进行排序
- ◆ 指明排序的列名，可以是多列，使用逗号分隔
- ◆ 排序时可以指定升序或者降序，默认为升序

### 4. 操作演示（10 分钟）

## 使用SELECT语句进行查询

### 讲解

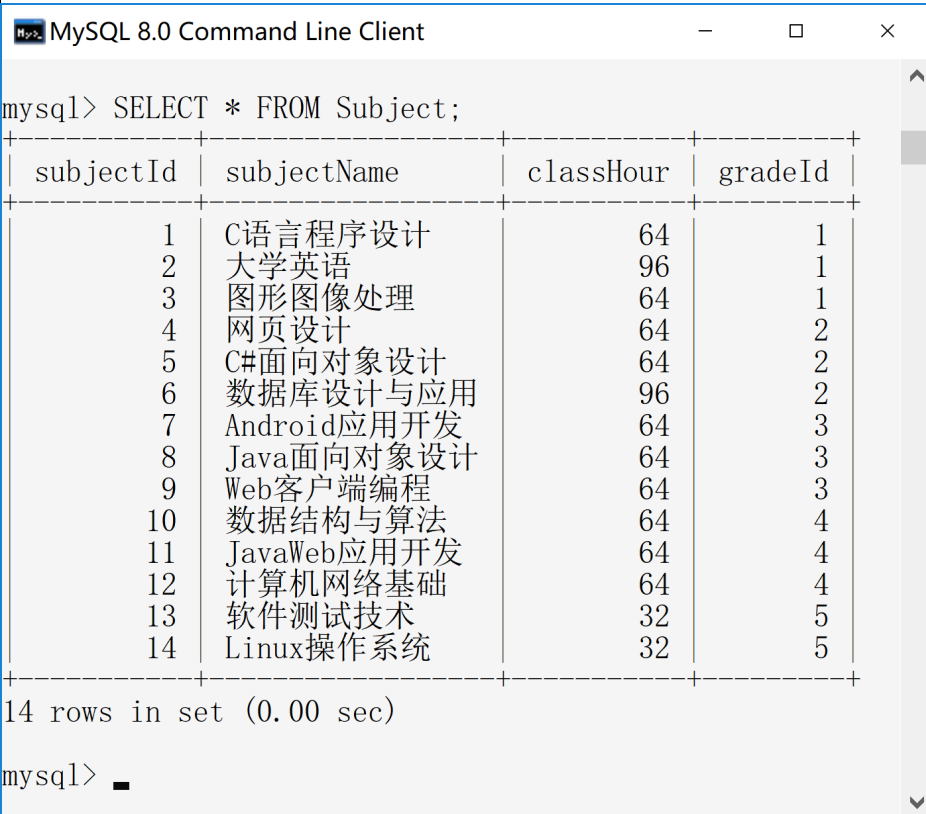
- 查询所有的数据行和列
  - ◆ SELECT后面加上 “\*”，代表所有的列
  - ◆ 不需要加任何条件，即：WHERE子句为空

### 演示

- 查询课程表Subject中所有的课程信息

```
USE SchoolDB;
```

```
SELECT * FROM Subject;
```



```
mysql> SELECT * FROM Subject;
```

subjectId	subjectName	classHour	gradeId
1	C语言程序设计	64	1
2	大学英语	96	1
3	图形图像处理	64	1
4	网页设计	64	2
5	C#面向对象设计	64	2
6	数据库设计与应用	96	2
7	Android应用开发	64	3
8	Java面向对象设计	64	3
9	Web客户端编程	64	3
10	数据结构与算法	64	4
11	JavaWeb应用开发	64	4
12	计算机网络基础	64	4
13	软件测试技术	32	5
14	Linux操作系统	32	5

```
14 rows in set (0.00 sec)
```

```
mysql> █
```

### 三 小结（2分钟）

MySQL 中最主要、最核心的部分是查询功能。查询语言用来对已经存在于数据库中的数按照特定的组合、条件或者一定次序进行检索。

**课间活动，大家站立走动，活动腰部、颈椎，进行健康教育**

### 四 深入活动（40min）

## 1. 课程新授演示（10 分钟）

### 使用ORDER BY子句进行查询排序

#### 语法

```
SELECT *|字段列表  
FROM 表名  
[WHERE 查询条件]  
[ORDER BY 排序的列名 [ASC|DESC];
```

```
ORDER BY 次序表达式 1 [ASC|DESC][,次序表达式 2 [ASC|DESC]]..
```

#### 分析

##### ELECT子句

- ◆ 指定要查看的列，即字段，即列出查询结果中要显示的字段名

#### 语法

```
ORDER BY 次序表达式 1 [ASC|DESC][,次序表达式 2 [ASC|DESC]]...
```

#### 分析

##### 次序表达式

- ◆ 为排序的表达式，可以是列名或者列的别名，也可以是一个表达式
- ◆ 关键字ASC表示升序排列，DESC表示降序排列，默认值为ASC
- ◆ 排序时，空值（NULL）被认为是最小值，参与排序
- ◆ 可以为多个表达式，且每个可以单独标明升序或降序，用逗号分隔
- ◆ 当有多个次序表达式时，按照从前往后的顺序执行，当表达式1的值相等的情况下，再按照表达式2的排序，依次类推

## 2. 学生操作跟练习，完成实训作业，并提交

——第 3-4 节结束——

### （三）任务三：函数查询（字符串函数）（80 分钟）

#### 一、教学导入（5 分钟）

##### 1. 学生讨论(3min)

- （1）什么是函数，函数有什么作用？

##### 2. 教师总结(2min)

- （1）总结学生讨论内容。
- （2）课程引入：为实现一些复杂的运算，MySQL 提供了丰富的内置函数。

### 函数

- 函数是完成特定功能的一组SQL语句的集合
- 应用：在数据查询中经常会使用函数来实现一些复杂运算
- MySQL提供了丰富的内置函数
  - ◆ 字符串函数
  - ◆ 日期和时间函数
  - ◆ 聚合函数

#### 二、发展活动（75 分钟）

- （1）讲解字符串函数。（5min）

## 字符串函数

### ■ 功能

- ◆ 字符串函数主要针对字符型数据进行操作和运算

### ■ 应用

- ◆ 为了实现某些复杂的查询功能，使用字符串函数对字符型数据进行处理

### ■ 字符串函数中包含的字符串必须要用单引号括起来

```
SELECT ASCII('email');
```

(2) 分别对不同的字符串函数讲解，演示，学生通过讨论、跟练等方式进行学习。(30min)

函数	功能	示例和说明
ASCII(字符串表达式)	返回字符串表达式最左端字符的 ASCII 值。返回值为整数	SELECT ASCII('email'); 说明:返回字母 e 的 ASCH 码值 101
CHAR(整型表达式)	返回整型 ASCII 码转换的字符	SELECT CHAR(65); 说明:返回 ASCII 码值为 65 的字符 A
LENGTH(字符串表达式)	返回字符串表达式的长度	SELECT LENGTH('email'); 说明:返回 email 的长度值 5
LEFT(字符串表达式,长度)	返回从字符串表达式左边开始指定长度个字符	SELECT LEFT('telephone',3); 说明:返回 telephone 左边开始 3 个字符,返回 tcl
RIGHT(字符串表达式,长度)	返回从字符串表达式右边开始指定长度个字符	SELECT RIGHT('telephone',3); 说明:返回 telephone 右边开始 3 个字符,返回 one
TRIM(字符串表达式)	返回删除字符串表达式首部和尾部的所有空格,返回值为字符串	SELECT TRIM(' I like MySQL! '); 说明:删除 I like MySQL! 前后的所有空格,返回 I like MySQL! ,□代表空格,下同
LTRIM(字符串表达式)	删除字符串中前面的空格,返回值为字符串	SELECT LTRIM(' I like MySQL! '); 说明:删除 I like MySQL! 前面的所有空格,返回 I like MySQL! □□□
RTRIM(字符串表达式)	删除字符串中尾部的空格,返回值为字符串	SELECT RTRIM(' I like MySQL! '); 说明:删除 I like MySQL! 后面的所有空格,返回 I like MySQL!

函数	功能	示例和说明
REPLACE(字符串1,字符串2,字符串3)	用字符串3替换字符串1中出现的字符串2,最后返回替换后的字符串	SELECT REPLACE('Welcome tK BEIJING!','K','o'); 说明:将 Welcome tK BEIJING! 中出现的 K 替换成 o,返回 Welcome to BEIJING!
SUBSTRING(字符串表达式,指定位置,长度)	返回字符串表达式从指定位置开始指定长度的子串	SELECT SUBSTRING('telephone',5,5); 说明:取字符串 telephone 中从第 5 个字符 p 开始连续 5 个字符构成的子串,返回 phone
CONCAT(字符串1,字符串2, ..., 字符串n)	返回字符串1,字符串2, ..., 字符串n连接起来构成的字符串	SELECT CONCAT('中国','北京'); 说明:字符串中国和北京连接起来,返回中国北京
LOCATE(字符串1,字符串2)	返回字符串1的第1个字符在字符串2中的序号,从1开始计数	SELECT LOCATE('Happy','I am happy!'); 返回 6

(3) 每位学生完成 ftp 上的实训任务单。(40min)

思政点:通过函数的主要功能讲解,让学生理解函数可以模块化的实现一定的功能,讲究合作共赢,善于利用资源,各取所长。

——第 5-6 节结束——

一、教学导入 (5 分钟)

1. 学生讨论(3min)
  - (1) 怎样获取系统当前日期?
  - (2) 中奖系统的中奖数据是如何获得的?
2. 教师总结(2min)
  - (1) 总结学生讨论内容。
  - (2) 课程引入：日期函数和数学函数。

## 二、发展活动（75 分钟）

- (1) 讲解日期函数。(5min)

### 日期函数

#### ■ 功能

- ◆ 日期和时间函数主要用来处理日期和时间值
- ◆ 日期函数帮助提取日期值中的年月和日，以便分别操作它们

#### ■ 通常不能对日期直接运用数学运算

- ◆ 例如：如果执行一个诸如“当前日期+1”的语句，MySQL无法理解要增加的是一日、一月还是一年

- (4) 分别对不同的日期函数讲解，演示，学生通过讨论、跟练等方式进行学习。（30min）

函数	功能	示例和说明
NOW()	获得当前的日期和时间,它以 YYYY-MM-DD HH:MM:SS 的格式返回当前的日期和时间	SELECT NOW(); 返回 2020-08-08 10:26:49
CURTIME()	返回当前时间	SELECT CURTIME(); 返回 10:28:40
CURDATE()	返回当前日期	SELECT CURDATE(); 返回 2018-08-25
YEAR()	分析一个日期值并返回其中关于年的部分	SELECT YEAR(20200808142800), YEAR('2021-11-11'); 返回 2020 和 2021
MONTH()	以数值格式返回参数中月的部分	SELECT MONTH(20200808142800); 返回 8
MONTHNAME()	以字符串的格式返回参数中月的部分	SELECT MONTHNAME('2021-11-22'); 返回 November
DAYNAME()	以字符串形式返回星期名	SELECT DAYNAME('2021-11-22'); 返回 Monday
函数	功能	示例和说明
WEEK()	返回指定的日期是一年的第几个星期	SELECT WEEK(20200808142800); 返回 31
YEARWEEK()	返回指定的日期是哪一年的哪一个星期	SELECT YEARWEEK('2021-11-22'); 返回 202147,表示 2021 年第 47 周
HOUR()	返回时间值的小时部分	SELECT HOUR(155300); 返回 15
MINUTE()	返回时间值的分钟部分	SELECT MINUTE('13:55:00'); 返回 55
SECOND()	返回时间值的秒部分	SELECT SECOND(132445); 返回 45
DATE_FORMAT (日期时间,格式符)	将日期时间按照指定的格式符输出,格式主要有 %Y,%y,%m,%d 等	SELECT DATE_FORMAT(now(), '%Y,%y,%m,%d'); 返回 2020,20,08,12

- (5) 讲解演示数学函数，学生跟练。(30min)

## 数学函数

### ■ 功能

- ◆ 数学函数用于对数值型数据进行处理

### ■ 常用的数学函数分析

函数	功能	示例和说明
RAND()	返回从 0 到 1 之间的随机 float 值	SELECT RAND(); 返回 0.767553509644696 (该值随机产生)
ABS(数值表达式)	取数值表达式的绝对值	SELECT ABS(-26); 返回 26
CEILING(数值表达式)	向上取整,取大于或等于指定数值、表达式的最小整数	SELECT CEILING(28.5); 返回 29
FLOOR(数值表达式)	向下取整,取小于或等于指定表达式的最大整数	SELECT FLOOR(28.5); 返回 28
POWER(数值表达式 1, 数值表达式 2)	取数值表达式的幂值	SELECT POWER(3, 2); 返回 9
ROUND(数值表达式)	将数值表达式四舍五入为指定精度	SELECT ROUND(28.543, 1); 返回 28.5

函数	功能	示例和说明
SIGN(数值表达式)	对于正数返回 +1,对于负数返回 -1,对于 0 则返回 0	SELECT SIGN(-28); 返回 -1
SQRT(数值表达式)	取浮点表达式的平方根	SELECT SQRT(4); 返回 2

(5)每位学生独立完成实训任务 (10min)

——第 7-8 节结束——

(五) 任务五：模糊查询 (80 分钟)

一、准备活动 (3min)

1. 衢职有礼师生互动, 清理桌面、键盘鼠标以及显示器—劳动教育。学习通签到。(1min)

2. 学习通预习技能点的学习情况检查。(1min)

3. 复习。(1min)

二、发展活动 (75min)

1. 新课导入 (5min)

不需要精确查询的时候, 怎么办?

2. 课程新授 (10min)

## 通配符和模糊查询

### ■ 通配符

- ◆ 是一类字符
- ◆ 它可以代替一个或多个真正的字符, 查找信息时作为替代字符出现

### ■ 模糊查询

- ◆ 主要通过模式匹配来实现
- ◆ 不需要精确的查询条件, 而是某些列值的一部分, 不要求与列值完全相等
- ◆ 例如: 要查找用户表中姓李的客户的的相关信息

## ■ 模糊查询会用到LIKE运算符

- ◆ 用于指出一个字符串与指定字符串是否相匹配
- ◆ 需要与通配符一起使用

## ■ 常用通配符：“\_”和“%”

- ◆ “%”代表0个或多个字符
- ◆ “\_”代表单个字符

3. 操作演示，学生跟练习（30min）

## 使用LIKE进行模糊查询

### 语法

表达式 1 [NOT] LIKE 表达式 2

### 分析

- 表达式1和表达式2一般要求是字符型表达式
- 表达式的值
  - ◆ 为逻辑真或者逻辑假
  - ◆ 一般用在WHERE子句中，作为查询、更新和删除的条件出现

### 分析

- 由于MySQL默认不区分大小写，要区分大小写时需要更换字符集的校对规则
- 转义字符
  - ◆ 如果给出的查询条件本身包含特殊符号中的一个或全部（\_或%）
  - ◆ 必须使用转义字符来实现查询
  - ◆ 转义字符为单个字符，没有默认值

## 使用BETWEEN在某个范围内进行查询

- 关键字BETWEEN可以查找那些介于两个已知值之间的一组值
  - ◆ 要实现这种查找，必须知道查找的初值和终值
  - ◆ 并且初值要小于等于终值，初值和终值用AND关键字分开
  - ◆ 查询结果包含初值和终值
- 例如：查询分数在70（含）85（含）之间的信息

```
SELECT * FROM result  
WHERE studentResult BETWEEN 70 AND 85;
```

## 使用IN在列举值内进行查询

- 应用场合
  - ◆ 查询的值是指定的某些值之一
  - ◆ 使用带列举值的IN关键字来进行查询
  - ◆ 将列举值放在圆括号里，用逗号分开
  - ◆ 列举值类型必须与匹配的列具有相同的数据类型
- 例如：查询第2、第3和第4学年开设课程的详细信息

```
SELECT * FROM Subject  
WHERE gradeId IN(2,3,4) ORDER BY gradeId;
```

4.每位学生独立完成实训任务（30min）

——第 9-10 节结束——

（六）任务六：聚合函数查询（80分钟）

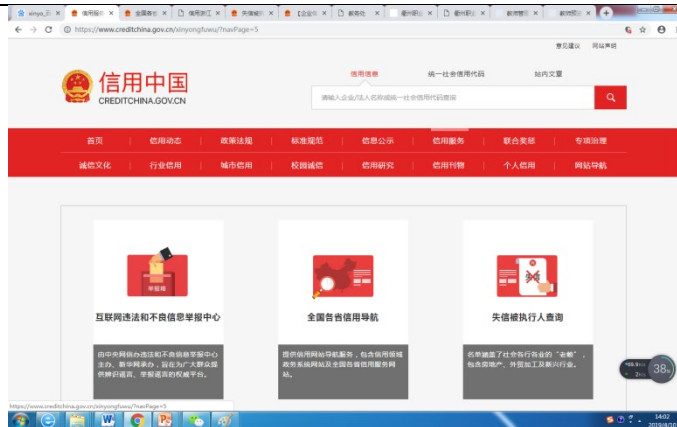
一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。（1min）
2. 学习通预习技能点的学习情况检查。（1min）
3. 复习。（1min）

二、发展活动(35min)

1. 新课导入（2min）

学习通技能点的学习检查情况-存在的刷视频情况，引入诚信的重要性，通过诚信中国的建设，导入本次课的学习内容。（2min）



2. 课程新授 (15min)

引入中国梦的实现，需要科技的支撑，学好知识，科技报国。



(1) 重难点分析 (1min)。

聚合函数可以对一组(列)值进行计算，并返回一个值，常用的聚合函数包括：SUM(和)，AVG(平均值)，MAX(最大值)，MIN(最小值)，COUNT(统计记录条数)等。

(2) 讲解、演示使用聚合函数查询的语法结构 (2min)

SELECT 聚合函数(列名)

FROM 表名

[WHERE 条件]

其中“聚合函数”指的是SUM, AVG, MAX, MIN, COUNT中的一个，“列名”是被计算列名称。

(2) 讲解、演示 SUM 函数案例 (2min)

例 1: 计算 xs\_kc 表中学号为 2016110101 的学生所选课程的成绩总和，并将查询结果中的列名设为“学号 2016110101 总成绩”。

满足 (1) where 子句

(2) 成绩总和 sum

(3) 针对谁来求和？成绩列

(4) 计算机结果列名设置 AS

```
mysql> select sum<成绩> as 学号2016110101成绩 from xs_kc
-> where 学号='2016110101';
+-----+
| 学号2016110101成绩 |
+-----+
|                    205 |
+-----+
1 row in set (0.00 sec)
```

(4) 讲解、学生操作演示 avg、max、min、count 聚合函数案例 (10min)

强调返回值设置别名，注意命名规范

AVG 函数用于计算指定字段的平均值或符合特定条件的指定字段平均值，在计算时将忽略值为 NULL 的记录，但不忽略值为 0 的记录。

例 2 计算 xs\_kc 表中学号为 2016110101 的学生的平均成绩。

```
mysql> select avg<成绩> 平均成绩 from xskc where 学号='2016110101';
+-----+
| 平均成绩 |
+-----+
|        56.5 |
+-----+
1 row in set (0.00 sec)
```

例 3 计算 xs\_kc 表中学号为 2016110101 的学生的非 0 成绩的平均分。

```
mysql> select avg<成绩> 平均成绩 from xskc
-> where 学号='2016110101' and not 成绩=0;
+-----+
| 平均成绩 |
+-----+
| 75.33333333333333 |
+-----+
1 row in set (0.00 sec)
```

形成良好的编程习惯，强化 sql 语句的编写规范

①使用比较贴切的英文单词组合，做到文词达意；

②字段别名尽量使用下划线分割多个单词；

③字段别名时不要省略 as 关键字

3. 分组完成课堂单元，并在学习通中进行组间竞赛 (17min)，团队协作，互相帮助。

(1) 上机辅导，问题解答。

(2) 每组同学参考教材、学习通技能点视频，完成课堂单元的任务。

	<p>4. 学习通中查看各组完成情况 1min (诚信背书)。</p> <p>三、小结、布置作业(2min)</p> <p>1. 总结使用聚合函数查询的语法和步骤，强调重难点。(1min)</p> <p>2. 布置作业。(1min)</p> <p><b>课间活动，大家站立走动，活动腰部、颈椎，进行健康教育</b></p> <p><b>完成实训《使用聚合函数查询》，巡回指导</b></p>
学习评量	<p>1.阶段测试四，共计 100 分。</p> <p>2.评量方法：超星网络教学平台阶段测试。</p>
课后作业	学习通中视频学习
教学反思	<p>目标或能力指针达成度、学习表现、内容、方法、资源、业师协同等向度(至少写三项)</p> <p>1.目标或能力指标达成度</p> <p>全班 80%学生达到了教学目标和能力指标。</p> <p>2.学习表现</p> <p>通过课前任务的布置，学生按要求完成任务单，课上教师提出问题，学生通过讨论，解决问题，个别学生在数据库定义过程中出现的问题，经过指导和同学讨论完成任务。</p> <p>3.资源</p> <p>(1) 学生总结本单元学习情况，教师点评，指出存在的问题及解决方法，做好资料积累工作，保存好各位同学操作的文件。</p> <p>(2) 对学生操作中出现的记录，以便对下一年度的学生操作指出注意事项。</p>

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	期中测试	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)掌握了数据库和表的定义。 (2)能正确的进行数据库、表的创建、修改、删除。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 掌握使用 MySQL 命令向表中插入、修改、删除数据的方法。C1 2. 培养良好的规范意识。A1				
教学目标	知识 1. 掌握使用 MySQL 命令向表中插入数据的方法。 2. 掌握使用 MySQL 命令向表中更新、删除数据的方法。 3. 理解 INSERT 语句、UPDATE 语句、DELETE 语句语法; 能力 1. 能根据要求实现数据的插入。 2. 能根据要求实现数据的更新、删除等操作。 素质 1. 提升解决数据库操作中出现问题的能力。 2. 培养良好的学习责任意识。 思政 1. 设置两次考试机会, 取最高分, 树立学生不断优化的意识以及积极向上的精神面貌。				
教学重难点	5. MySQL 系列命令的规则和应用 6. 理解数据表的约束和数据表之间的关系, 并通过数据验证是否有效				
教师课前准备	1. 学习通设置好试卷。				
学生课前准备	查阅资料, 观看学习通技能点视频				
活动历程(强调成果导向、主	2022-2023 第二学期期中测验				

要教学法的历程，含辅助手段、时间分配)

### 一、 程序填空题 （共 6 题， 55 分）

1、 现在有成绩表 `xs_kc`， 包含学号、 课程号、 成绩等字段信息 ， 逻辑表达式请写在同一个括号内

查询所有不及格学生的学号、 课程号和成绩信息 `select ( ) , ( ) , ( ) from xs_kc where ( ) ;`

(5.0)

2、 现在有学生表 `xsqk`， 包含姓名、 学号、 班级、 身份证号码、 地址等字段信息 ，

查询所有姓李和姓张的学生信息（一个表达式， 请写在同一空格里面）

`( ) * ( ) ( ) where 姓名 like '李%' ( ) ( ) ;`

(10.0)

3、 现在有成绩表 `xs_kc`， 包含学号、 课程号、 成绩等字段信息 ， 查询成绩在 60 分至 70 分的学生成绩信息,并按照分数从高到低排序(逻辑表达式请写在同一个括号内)

`select 成绩 from xs_kc where ( ) ( ) ( ) ( ) ( ) ;`

(10.0)

4、 现在有课程表 `kc`， 包含课程号、 课程名称、 授课教师等信息

查询课程号为“102”的授课教师姓名(表达式写在同一空格里)

`select( )from ( ) where ( ) ;`

(10.0)

5、 现在有学生表 `xsqk`， 包含姓名、 学号、 班级、 性别、 出生日期、 身份证号码、 地址等字段信息 ，

统计出生日期在 1998-01-01 年以后的女生人数 `select( )from( )( )and( ) ;`（表达式写在同一空格）

(10.0)

6、 现在有成绩表 `xs_kc`， 包含学号、 课程号、 成绩等字段信息 ， 查询选修课程号 101 的最高成绩， 最低成绩和平均成绩；

`select ( ) , ( ) , ( ) from xs_kc where ( )`

(10.0)

### 二、 单选题 （共 9 题， 45 分）

1、

`Employee` 表中 `LastName` 列保存顾客的姓， `FirstName` 列保存顾客的名。 现在， 需要查询顾客姓名的组合， 例如， `LastName` 列中的“张”， 同一行 `FirstName` 列中的“国华”， 查询结果应该返回“张国华”， 则正确的查询语句应该是（ ）。

(5.0)

A、 `SELECT CONCAT(LastName,FirstName) FROM Employee;`

B、 `SELECT * FROM Employee ORDER BY LastName,FirstName;`

C、 SELECT LastName+FirstName FROM Employee;

D、 SELECT LastName AND FirstName FROM Employee;

2、现在 Students 表中已经存储了数据，Nation 列的数据存储了学生的民族信息，默认值应该为“汉族”。可是在设计表的时候这个默认特征没有考虑，现在已经输入了大量的数据。对于少数民族的学生，民族的信息已经输入，对于“汉族”的学生，数据都为空值。此时，要解决这个问题比较好的办法是（ ）。

(5.0)

A、 在表中为该列添加 NOT NULL 约束

B、 使用“UPDATE Students SET Nation='汉族' WHERE Nation IS NULL”进行数据更新

C、 使用“UPDATE Students SET Default='汉族'”进行数据更新

D、 手动输入所有的汉族

3、一个小组正在开发一个大型的银行存款系统，系统中包含上百万行顾客的信息。现在正在调试 SQL 语句，以进行查询的优化。可是，他们每次执行查询时，都返回好几百万行数据，显示查询结果非常费时。此时，比较好的解决办法是（ ）。

(5.0)

A、 删除这些数据

B、 把这些数据转换到文本文件中，再在文本文件中查找

C、 在查询语句中使用 LIMIT 子句限制返回行数

D、 在查询语句中使用 ORDER BY 子句进行排序

4、执行以下 SQL 语句“SELECT SName,SAddress FROM Students limit 40;”结果返回了 20 行数据，则（ ）。

(5.0)

A、 表 Students 中只有 40 行数据

B、 表 Students 中只有 20 行数据

C、 表 Students 中大约有 50 行数据

D、 表 Students 中大约有 100 行数据

5、以下（ ）能够得到今天属于哪个月份。(5.0)

A、 SELECT MONTH();

B、 SELECT DATE();

C、 SELECT MONTH(DATE());

D、 SELECT MONTH(NOW());

6、SQL 查询中使用 ORDER BY 子句指出的是（ ）。(5.0)

A、 查询目标

B、 查询结果排序

C、 查询视图

D、 查询条件

7、查询 Student 表中的所有非空 email 信息，以下语句正确的是（ ）。

	<p>(5.0)</p> <p>A、 SELECT email FROM Student Where email!=null;  B、 SELECT email FROM Student Where email not is null;  C、 SELECT email FROM Student Where email&lt;&gt;null;  D、 SELECT email FROM Student Where email is not null;</p> <p>8、 用于求系统日期函数的是 ( )。(5.0)</p> <p>A、 YEAR()  B、 NOW()  C、 COUNT()  D、 SUM()</p> <p>9、 在查询结果集中将 NAME 字段显示为联系人, 应该使用 ( ) 语句。</p> <p>(5.0)</p> <p>A、 SELECT name FROM Customers AS '联系人';  B、 SELECT name='联系人' FROM Customers;  C、 SELECT * FROM Customers WHERE name='联系人';  D、 SELECT name as '联系人' FROM Customers;</p>
学习评量	<p>1.期中测试, 共计 100 分。  2.评量方法: 超星网络教学平台阶段测试。</p>
课后作业	综合实训
教学反思	

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元五 分组和多表查询	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)掌握了数据表单表查询。 (2)能根据需求进行在一张表中进行数据查询。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 掌握使用 MySQL 命令在分类统计的方法。 2. 能根据需求选择多张表进行连接查询 2. 培养良好的规范意识。			C1 D1 A1	
教学目标	知识 1. 理解和掌握分组查询和连接查询的机制。 2. 使用 GROUP BY 和 HAVING 子句实现分组和筛选查询。 能力 1. 能根据查询需求多表连接查询。 2. 能根据根据查询需求, 使用子查询实现。 素质 1. 提升解决数据库操作中出现问题的能力。 2. 培养良好的学习责任意识。 思政 1. 提升技术创新、与时俱进的开拓精神。 2. 提升履行公民责任的意识, 提升服务社会, 科技报国的意识 3. 提升学生规范编写程序代码的意识				
教学重难点	1. 内连接和外连接查询的区别及应用场景 2. 灵活使用多表连接查询或者子查询解决实际问题				
教师课前准备	1. 学习通设置好试卷。				
学生课前准备	查阅资料, 观看学习通技能点视频				



(一) 任务一 分组查询 Group by

一、准备活动(3min)

活动历程 (强调成果) 1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。(1min)

2. 学习通预习技能点的学习情况检查。(1min)

3. 复习。(1min)

二、发展活动(35min)

1. 新课导入

第七次全国人口普查，获得大量的基础数据，各行各业要根据自己的需求，对数据进行一个分类汇总。导入本次课的学习内容。(2min)



3. 课程新授(25min)

人口普查获得基础数据后，要对数据进行分类汇总，整理后，才是有用的数据，助力国家发展。



引导学生了解人口普查，履行公民责任

(1) 重难点分析。

分类汇总的查询方法，可以按照指定的列将数据分成多个类别，然后按照类别进行汇总。按照查询功能的不同，可分为简单分类查询、统计功能分类查询、多字段分类查询和采用 Having 子句的分类查询。

(2) 讲解、演示、跟练简单分类查询

语法规则：

SELECT 字段列表 FROM 表名 [WHERE 条件]

GROUP BY 列名 1[, ...n]

分类的依据是 GROUP BY 子句中指定的列名来对数据记录进行分类。

例 在 xs\_kc 表中，按选修的课程号对所有选修情况进行分类。

```
mysql> select * from xs_kc group by 课程号;
```

学号	课程号	成绩	学分
2016110101	101	83	2
2016110101	102	64	5
2016110101	103	58	0
2016110105	105	67	4
2016110201	106	78	4
2016110202	107	85	4
2016110203	108	61	2
2016110204	109	18	0
2016110301	110	63	4

```
9 rows in set (0.00 sec)
```

(3) 实现统计功能的分类查询

如果只进行简单分类查询，是没有太大意义的。可以将分类查询

与统计函数一起使用，以实现统计功能的分类查询。

语法规则：

```
SELECT GROUP_CONCAT(列名) FROM 表名 WHERE 条件  
GROUP BY 列名 1[, ...n];
```

其中，GROUP\_CONCAT() 函数可以显示出每个分组中指定的字段值。

例 查询 xs\_kc 表，按课程号进行分组显示，并显示出选修了该课程的学号。

```
mysql> select 课程号, group_concat(学号) 学号  
-> from xs_kc  
-> group by 课程号;
```

课程号	学号
101	2016110101,2016110103,2016110105
102	2016110101,2016110102,2016110106
103	2016110101,2016110102,2016110104
105	2016110105
106	2016110201,2016110202
107	2016110202
108	2016110203
109	2016110204
110	2016110301

```
9 rows in set (0.06 sec)
```

#### (4) 多字段分类查询

在 MySQL 中进行分类查询时，除了可以对一个字段进行分类查询外，还可以对多个字段进行分类。

语法规则：

```
SELECT 字段列表 FROM 表名  
WHERE 条件  
GROUP BY 列名 1, 列名 2, ...;
```

在 GROUP BY 子句中，按照列出的列名先后次序进行分类。

例 查询 xsqk 表，先按性别进行分类，然后再查年不同性别的学生学了哪些不同的专业

```
mysql> select 性别,专业名 from xsqk
-> group by 性别,专业名;
+-----+-----+
| 性别 | 专业名 |
+-----+-----+
| 女   | 云计算 |
| 女   | 信息安全 |
| 女   | 机器人设计 |
| 女   | 网络工程 |
| 男   | 云计算 |
| 男   | 信息安全 |
| 男   | 机器人设计 |
| 男   | 网络工程 |
+-----+-----+
8 rows in set (0.00 sec)
```

三、小结、布置作业(2min)

小结:

分类汇总查询主要解决实际应用中, 用户需要得到不同类别的汇总数据的需求。 通过将指定的列分成多个类别, 按照按照类别进行汇总的数据查询方法。

关键词: GROUP BY

课间活动, 大家站立走动, 活动腰部、颈椎, 进行健康教育  
完成实训, 巡回指导 (40 分钟)

——第 1、2 节结束——

(二) 任务二 Having 子句分组筛选

一、准备活动(3min)

1. 衢职有礼师生互动, 清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。(1min)
2. 学习通预习技能点的学习情况检查。(1min)
3. 复习。(1min)

二、发展活动(35min)

1. **新课导入**-采用 HAVING 子句的分类查询 (10min)

在实际应用中, 有的需求还需要在分组统计的基础上进行筛选对分组统计的结果进行过滤

比如: 统计总分达到 200 分的学生信息

HAVING 子句

实现分组后的筛选功能

HAVING 子句要用在 GROUP BY 子句后, 用于过滤分组后的结果与 WHERE 子句比较

WHERE 子句是用来在 FROM 子句中选择行


HAVING 子句是在 GROUP BY 子句后选择行

## 使用HAVING子句进行分组筛选

■ HAVING和WHERE子句可以一起使用

■ 使用顺序



 示例

```
SELECT subjectId AS '课程编号',COUNT(*) AS '及格人数',
       AVG(studentResult) AS '及格学生的平均分'
FROM Result
WHERE studentResult>=60
GROUP BY subjectId
HAVING AVG(studentResult)>=80
ORDER BY AVG(studentResult) DESC;
```

3. 操作演示-（10 分钟）

### 【演示示例6-4】 分组筛选的应用

 需求

- 在数据库SchoolDB的学生信息表Student中
  - ◆ 查询年级总人数超过4的年级编号

 分析

- 通过分组查询获取每个年级的总人数
  - ◆ 使用GROUP BY子句，按照年级编号分组
  - ◆ 使用COUNT()函数统计
- 在GROUP BY后面使用HAVING子句，条件：“COUNT(\*)>4”

MySQL 8.0 Command Line Client

```
mysql> SELECT gradeId AS '年级', COUNT(*) AS '人数'
-> FROM Student
-> GROUP BY gradeId;
```

年级	人数
5	2
3	5
1	6

3 rows in set (0.00 sec)

```
mysql> SELECT gradeId AS '年级', COUNT(*) AS '人数'
-> FROM Student
-> GROUP BY gradeId
-> HAVING COUNT(*) > 4;
```

年级	人数
3	5
1	6

2 rows in set (0.00 sec)

**统计每个年级的人数!**

**在统计的基础上, 筛选大于4人的年级信息!**

4. 学生跟练 (10 分钟)

5. 小结

### 【演示示例6-4】分组筛选的应用

#### 注意

- 使用WHERE子句是不能满足查询要求的
  - ◆ WHERE子句只能对没有分组统计前的数据进行筛选
- 用HAVING子句来指定筛选的条件
  - ◆ 该子句中的条件通常是统计函数
  - ◆ 如: 条件为 "COUNT(\*)>4"

课间活动, 大家站立走动, 活动腰部、颈椎, 进行健康教育  
根据实训要求, 完成实训提交作业 (40min)

——第 3、4 节结束——

### (三) 任务三 内连接查询

#### 一、准备活动 (3min)

1. 衢职有礼师生互动, 清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。 (1min)
2. 学习通预习技能点的学习情况检查。 (1min)

3. 复习。(1min)

二、发展活动(35min)

1. 新课导入-大国战略，技术强国（5分钟）

党的十八届五中全会通过的“十三五”规划建议，明确提出实施网络强国战略以及与之密切相关的“互联网+”行动计划。国家着力实现关键技术自主可控，为维护国家安全提供技术保障。关键技术还有待提高自主控制的技术能力。

引导学生明确，建设网络强国不仅要有各类技术的支撑，作为未来从事软件技术领域的学生，更应该明确专业领域内工作岗位和工作内容的社会价值，自觉树立远大职业理想，将职业生涯、职业发展脉络和国家发展的历史进程融合起来。

2. 课程新授（18min）

在实际应用中，很多查询需要的数据要来源于多张表，比如：在查询成绩的结果中显示学生姓名和课程名称涉及到三张表，成绩表 Result、学生信息表 Student、课程表 Subject、查询命令 SELECT 支持多表连接查询。

### ①内连接查询

(1) 重难点分析。

内连接查询是最典型、最常用的连接查询，它根据表中共同的列来进行匹配。特别是两张表存在主外键关系时通常会使用内连接查询。关键要分析两张表的连接条件，根据应用需求完成。

(2) 讲解、演示、跟练内连接查询

语法规则：

**SELECT 显示列表**  
**FROM 表 1,表 2[,...]**  
**WHERE 连接条件;**

分析：

将查询需要数据所在的数据表分别列在 FROM 子句后面  
表名之间用逗号分隔

将表与表之间连接的条件作为 WHERE 子句的条件表达式

【演示示例 6-6】2 张表内连接查询的应用

### 需求

- 数据库SchoolDB中，查询学生的成绩
  - ◆ 显示学号、姓名、课程编号和成绩，按照学号排列

### 分析

- 需要查询的结果来源于2张表，需要2张表连接查询
- 成绩表中的学号引用了学生信息表的学号，连接条件：学号相等
- Student和Result中学号列的名称都是“studentNo”
  - ◆ 需要在列名的前面加上表的名称，并使用点“.”作为连接符
  - ◆ 如：“Student.studentNo”和“Result.studentNo”

### MySQL 8.0 Command Line Client

```
mysql> SELECT Student.studentNo AS '学号', Student.studentName AS  
-> Result.subjectId AS '课程编号', Result.studentResult AS '成绩'  
-> FROM Student, Result  
-> WHERE Student.studentNo=Result.studentNo  
-> ORDER BY Student.studentNo;
```

学号	姓名	课程编号	成绩
G1263201	王子洋	13	76.00
G1263201	王子洋	14	88.00
G1263382	张琪	13	79.00
G1263382	张琪	14	56.00
G1263458	项宇	13	92.00
G1263458	项宇	14	0.00
G1363278	胡保蜜	7	55.00
G1363278	胡保蜜	8	78.00
G1363278	胡保蜜	9	76.00
G1363300	王超	7	83.00
G1363300	王超	8	49.00
G1363300	王超	9	64.00
G1363301	党志鹏	7	65.00
G1363301	党志鹏	8	87.00
G1363301	党志鹏	9	55.00
G1363301	党志鹏	9	90.00
G1363302	胡仲友	7	80.00
G1363302	胡仲友	8	56.00
G1363302	胡仲友	9	87.00

只显示了前面的

(3) inner join on 实现 内连接查询

## 在FROM子句中使用INNER JOIN...ON

### 语法

```
SELECT 显示列表  
FROM 表 1 INNER JOIN 表 2  
ON (连接条件);
```

### 分析

- 通过FROM子句中使用INNER JOIN...ON关键字
  - ◆ 不需要在WHERE子句中设计表连接的条件表达式
  - ◆ 可以连接2张表，也可以连接3张表等

## 【演示示例6-7】 查询某门课及格学生的信息

### 需求

- 数据库SchoolDB中
- 查询课程编号为3的及格学生的姓名和分数

### 分析

- 需要查询的结果来源于2张表，需要2张表连接查询
- 采用组合条件
  - ◆ 统计及格的学生使用WHERE子句
  - ◆ 课程编号为“3”也是使用WHERE子句
  - ◆ 使用逻辑运算符“AND”连接2个条件

### (4) 外连接查询

- 外连接查询与内连接查询最大区别
  - ◆ 外连接查询中参与连接的表有主从之分
  - ◆ 以主表的每行数据匹配从表的数据，将符合条件的数据直接返回到结果集中
  - ◆ 对于不符合连接条件的列，在结果集中被填上NULL值（空值）
- 外连接分类
  - ◆ 左外连接（LEFT OUTER JOIN）
  - ◆ 右外连接（RIGHT OUTER JOIN）
  - ◆ 其中OUTER关键字可以省略

3. 学生自主练习完成 演示示例 6-6、6-7（10min）

三、小结、布置作业(2min)

小结：

多表查询主要解决实际应用中，用户需要得到不同的数据表中的

数据。 通过利用相同字段进行连接的方法，达成查询目标。

关键词：

Inner join on

课间活动，大家站立走动，活动腰部、颈椎，进行健康教育  
根据实训要求，完成实训，提交作业，教师巡回指导

—第 5、6 节结束—

一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。(1min)

2. 学习通预习技能点的学习情况检查。(1min)

3. 复习。(1min)

二、发展活动(35min)

1. 新课导入-数据缺失(5分钟)

## 外连接查询

### ■ 应用场合

- ◆ 使用内连接查询会出现查询信息不完整情况
- ◆ 比如：查询所有课程的考试成绩，如果某门课没有被学生选修
- ◆ 则该门课程不会出现在查询结果结果中
- ◆ 如果需要查询未被选修的课程信息，可以通过外连接查询实现

2. 新课讲授(10分钟)

### ■ 外连接查询与内连接查询最大区别

- ◆ 外连接查询中参与连接的表有主从之分
- ◆ 以主表的每行数据匹配从表的数据，将符合条件的数据直接返回到结果集中
- ◆ 对于不符合连接条件的列，在结果集中被填上NULL值(空值)

### ■ 外连接分类

- ◆ 左外连接 (LEFT OUTER JOIN)
- ◆ 右外连接 (RIGHT OUTER JOIN)
- ◆ 其中OUTER关键字可以省略

## 左外连接查询

### ■ 左外连接 (LEFT OUTER JOIN)

- ◆ 结果表中除了匹配行外(即与内连接查询结果相同的部分)
- ◆ 还包括左表中有的但右表中不匹配的行，从右表选择列的列值为NULL
- ◆ 查询结果包括
  - ✓ 第一个命名表(“左”表，出现在JOIN子句的最左边)中的所有行
  - ✓ 不包括右表中的不匹配行
- ◆ 连接条件和方式与内连接相同，主要体现在查询的结果集不同上

### 3. 操作演示（10 分钟）

#### 【演示示例6-9】统计所有课程选修情况

##### 需求

- 数据库SchoolDB中，查询统计所有课程的选修及成绩信息
- 显示学号、姓名、课程名称和考试成绩，按照课程编号升序排序
  - ◆ 如果在Result表中有成绩，则表示该同学选修了该门课
  - ◆ 如果某门课Result表中没有一条成绩记录，则表示无人选修该课程，对应课程的学号和成绩显示为NULL

##### 分析

- 要查询包括没有被选修的课程信息，使用外链接
- 使用左外连接，左表为课程表Subject，右表为成绩表Result

USE SchoolDB;

```
SELECT J.subjectId AS '课程编号',SubjectName AS 课程名称,  
StudentNo AS '学号',StudentResult AS '分数'
```

```
FROM Subject AS J
```

```
LEFT JOIN Result AS R ON J.subjectId=R.subjectId;
```

### 4. 学生跟练习（10 分钟）

#### 三、小结、布置作业(2min)

##### 小结：

多表查询主要解决实际应用中，用户需要得到不同的数据表中的数据。通过利用相同字段进行连接的方法，达成查询目标。

##### 关键词：

Left outer join on

Right outer join on

课间活动，大家站立走动，活动腰部、颈椎，进行健康教育  
根据实训要求，完成实训，提交作业，教师巡回指导

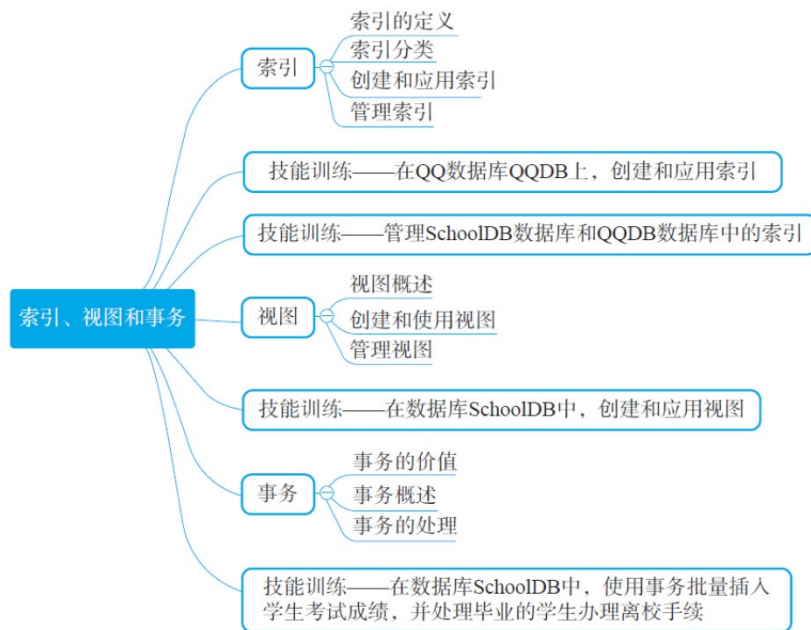
——第 7、8 节结束——

学习评量

课后作业

教学反思

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元六 索引和视图	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)掌握了数据表的连接查询。 (2)能根据需求进行在多张表中进行数据统计查询。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 掌握创建索引和视图的方法。 1. 能根据需求正确添加索引, 创建视图 2. 培养良好的规范意识。			C1 D1 A1	
教学目标	知识 1. 理解索引、视图和事务的概念和价值。 2. 掌握索引和视图的创建和管理方法。 能力 1. 能根据设计需求创建、管理应用索引。 2. 能根据设计需求创建、管理应用视图。 3. 能创建事务, 并应用事务解决实际问题。 素质 1. 提升解决数据库操作中出现问题的能力。 2. 培养不断优化提升系统效率的意识。 思政 1. 树立大局意识。 2. 提升学生规范编写程序代码的意识				
教学重难点	1. 灵活应用视图和索引解决实际问题。 2. 事务的创建、提交和回滚机制的理解和应用				
教师课前准备	1. 学习通设置好试卷。				
学生课前准备	查阅资料, 观看学习通技能点视频				



活动历程  
(强调成果  
导向、主要  
教学法的历  
程, 含辅助  
手段、时间  
分配)

任务一 索引  
一、准备活动(3min)  
1. 衢职有礼师生互动, 清理桌面、键盘鼠标以及显示器-劳动教育。学  
习通签到。(1min)  
2. 学习通预习技能点的学习情况检查。(1min)  
3. 复习。(1min)  
二、发展活动(35min)  
1. 新课导入(5min)  
汉语字典中的汉字按页存放, 一般都有汉语拼音目录(索引)、  
偏旁部首目录等, 我们可以根据拼音或偏旁部首, 快速查找某个字词



思政点: 引出中国传统文化的博大精深, 汉字的精妙, 文化的传承, 树立学生文化自信, 以及身为中国人的自豪感。

2. 课程新授(20min)  
语法讲解、操作演示

## 创建和应用索引

### ■ 创建索引主要的3种方式

- ◆ 在已经存在的表上使用CREATE INDEX创建索引
- ◆ 使用ALTER TABLE语句创建索引
- ◆ 在创建表时创建索引

### ■ 同一个表中可以创建多个索引

## 使用CREATE INDEX创建索引

### ☰ 语法

```
CREATE [索引类型] INDEX 索引名  
ON 表名(列名|(长度)|[ASC|DESC],...)
```

### 🔍 分析

#### ■ 索引名

- ◆ 索引的名称
- ◆ 在一个表中可以创建多个索引，但是索引名必须唯一

思政点命名规范：娴熟驾驭和运用标识符的命名规则，告诫学生在上课学习、日常生活和将来的工作岗位中要遵守相应的制度和规定，并用以约束和指导自己的行为，以后才能成为合格的社会公民。

学生跟练

## 使用ALTER TABLE语句创建索引

### 语法

**ALTER TABLE** 表名

**ADD** 索引类型 **INDEX** [索引名](列名,...)

|**ADD PRIMARY KEY**(列名,...)

### 分析

#### 索引类型

◆ 普通索引

◆ 唯一索引

◆ 全文索引

学生跟练

## 在创建表时创建索引

### 语法

**CREATE TABLE** 表名(列名,...|[索引项])

其中，索引项基本语法格式如下：

**PRIMARY KEY**(列名,...)

|**INDEX** [索引名](列名,...)

|**UNIQUE [INDEX]**[索引名](列名,...)

|**FULLTEXT [INDEX]**[索引名](列名,...)

学生跟练，完成实训作业，巡回指导（10min）

第一节结束

课间活动，大家站立走动，活动腰部、颈椎，进行健康教育

#### 1. 新课导入（5min）

为什么需要视图？保证信息的安全性，针对不同的用户设计不同的控制，拒绝未授权的访问请求。

思政点：通过视图机制把需要保密的数据对无权存取这些数据的用户藏起来，从而对数据提供一定程度的安全保护。进而延申数据泄露给安全造成的危害，培养了同学们在数据安全方面的法律意识，激发了他们维护国家数据安全、反对数据制裁的家国情怀。

#### 2. 内容新授（10min）

讲解，操作演示

## 什么是视图

- 视图是一张**虚拟表**
  - ◆ 表示一张表的部分数据或多张表的综合数据
  - ◆ 其结构和数据是建立在对表的查询基础上
- 视图中不存放数据
  - ◆ 数据依然存放在视图所引用的原始表中
  - ◆ 保存的是一种查询机制

## 视图的价值

- 对最终用户
  - ◆ 结果更容易理解
  - ◆ 获得数据更容易
- 对开发人员
  - ◆ 限制数据检索更容易
  - ◆ 维护应用程序更方便

## 创建视图

### 语法

```
CREATE VIEW 视图名[(列名列表)]
AS SELECT 语句
[WITH CHECK OPTION]
```

### 分析

#### ■ 列名列表

- ◆ 为可选项，为视图中的列定义明确的名称
- ◆ 列名之间用逗号隔开
- ◆ 列名列表中的名称数目必须等于SELECT语句检索的列数
- ◆ 若使用与源表或视图中相同的列名时可以省略列名列表

## 应用视图

### 语法

```
SELECT * FROM 视图名;
```

### 分析

- 先创建视图
- 视图以一张虚拟表的形式存在与数据库中
- 可以像基本表一样使用视图
- 视图主要用于与查询

3. 学生完成对应实训，巡回指导（25min）。

——第 1、2 节结束——

任务二 事务

一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。（1min）

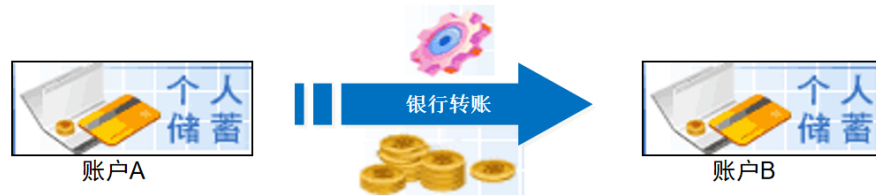
2. 学习通预习技能点的学习情况检查。（1min）

3. 复习。（1min）

二、发展活动(35min)

1. 新课导入-为什么需要事务，什么是事务（10min）

## 为什么需要事务



### ■ 银行转账问题：

- ◆ 假定资金从账户A转到账户B，至少需要两步
- ◆ 步骤1：账户A的资金减少
- ◆ 步骤2：然后账户B的资金相应增加

思政点：数据库中的数据是高度共享的，如果没有采取正确的并发控制技术，必然会引起数据不一致的问题。事务就是一种重要的保障机制。同理，人类社会也需要保障机制，从而引申学生尊重和遵守规则的品格。

## 什么是需要事务

### ■ 事务是单个的工作单元

- ◆ 如果某一事务执行成功，则在该事务中进行的所有数据更改均会提交
- ◆ 如果事务遇到错误且必须取消或回滚，则所有数据更改均被清除

### ■ 事务是一种机制、一个操作序列

- ◆ 它包含了一组数据库操作命令
- ◆ 把所有的命令作为一个整体一起向系统提交或撤销操作请求
- ◆ 即：这一组数据库命令要么都执行，要么都不执行

2. 操作演示-（10分钟）

## 【演示示例8-3】模拟银行转账业务

### 代码分析

```
USE SchoolDB;
DROP TABLE IF EXISTS bank;
CREATE TABLE bank
(
    customerName CHAR(10),
    currentMoney DECIMAL(10,2);
ALTER TABLE bank
    ADD CONSTRAINT CK_currentMoney CHECK(currentMoney>=1);
INSERT INTO bank(customerName,currentMoney)
    VALUES('张三',1000);
INSERT INTO bank(customerName,currentMoney)
    VALUES('李四',1);
SELECT * FROM bank;
```

3. 学生跟练，教师巡回指导（10 分钟）

4. 小结（5min）

## 执行事务的语法

### ■ 提交事务

- ◆ 标志一个事务成功提交
- ◆ 自事务开始至提交语句之间执行的所有数据更新将永久地保
- ◆ 并释放连接时占用的资源

### ■ 提交事务的语句

**COMMIT;**

课间活动，大家站立走动，活动腰部、颈椎，进行健康教育  
根据实训要求，完成实训提交作业（40min）

—第 3、4 节结束—

学习评量

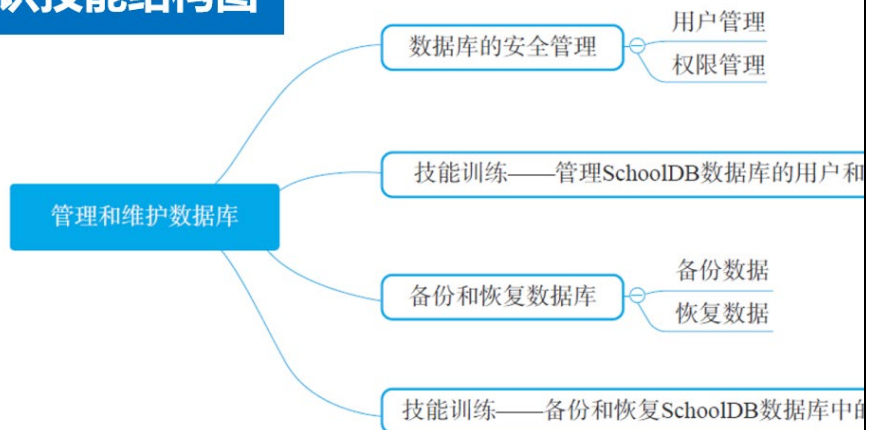
课后作业

教学反思

科目名称	数据库技术	学年	2	学期	1
教材来源	MySQL 数据库设计与应用	授课学时	54		
授课类型	<input type="checkbox"/> 理论型, <input checked="" type="checkbox"/> 理实一体型, <input type="checkbox"/> 实践型				
单元名称	单元七 存储过程	班级	电子 231 (自主、三加)	人数	45
学生学习条件分析	1.起点分析: (1)能使用 MySQL 进行简单的编程。				
教学方法手段	1.教学方法: (学习方法) 讲授法、演示法、自主学习。 2.教学手段: 多媒体、网络平台				
教学资源	1.个人资源: 教材、教学 PPT、实训指导书 2.学校资源: 超星平台中的公共资源 3.网络、社会或出版社资源: 论坛: <a href="https://bbs.csdn.net/forums/MySQL/">https://bbs.csdn.net/forums/MySQL/</a> 、 <a href="https://www.mysql.com/cn/">https://www.mysql.com/cn/</a> , 出版社资源: 配套微课 4.企业资源: 企业工程案例、优秀校友的线上辅导				
能力指标	1. 能进行数据的日常管理和维护 2. 培养较强的数据安全意识			D1 A2	
教学目标	知识 1. 掌握用户管理和权限管理的相关知识。 2. 掌握数据备份和回复的方法。 能力 1. 能使用 SQL 语句创建和删除用户账户。 2. 能使用 SQL 语句授予和回收权限。 3. 能备份和恢复数据库数据。 素质 1. 具备较强的数据安全意识。 思政 1. 培养学生在工作生活中要有居安思危的忧患意识。				
教学重难点	1. 权限的授予与回收。 2. 不同计算机之前迁移数据库和数据。				
教师课前准备	1. 学习通设置好试卷。				
学生课前准备	查阅资料, 观看学习通技能点视频				
活动历程 (强调成果导向、主要教学法的历程, 含辅助					

手段、时间  
分配)

## 知识技能结构图



### 任务一 数据库用户管理

#### 一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。(1min)
2. 学习通预习技能点的学习情况检查。(1min)
3. 复习。(1min)

#### 二、发展活动(35min)

##### 1. 新课导入(5min)

## 用户管理

### ■ 用户要访问数据库

- ◆ 首先必须能连接数据库所在的MySQL服务器
- ◆ 必须拥有登录MySQL服务器的用户名和密码

### ■ MySQL的访问控制分为两个阶段

- ◆ 第一个阶段：服务器验证是否允许连接
- ◆ 第二个阶段：连接成功后，验证每个请求是否具有实施的权限

### ■ 例如：要查看表中的数据

- ◆ MySQL会检查是否具有对这个表的SELECT权限
- ◆ 要执行某个存储过程，MySQL会检查是否具有该存储过程的执行

思政点：对于数据库而言，没有什么比安全性更重要了，它直接影响个人、企事业、部门以及国家的安全，所谓安全无小事。用户分级管理可以提升数据库安全性，进而引导学生作为IT行业的从业者，必须具备良好的职业道

德和职业素养，牢固树立数据安全意识。

2. 课程新授（25min）

语法讲解、操作演示（10min）

## 创建用户

### ■ 由系统管理员创建登录账户

- ◆ 给定用户名和登录密码
- ◆ 登录的位置和默认连接的数据库

### ■ 系统管理员是root用户

- ◆ 拥有最高的权限
- ◆ 可以完成所有的操作
- ◆ 它的密码在安装MySQL服务器时设置

## 创建用户

### 语法

```
CREATE USER 用户名  
@主机名[IDENTIFIED BY [PASSWORD] '密码']
```

### 分析

- 用户名：创建用户的名字
- 主机名
  - ◆ 指定创建用户所连接主机
  - ◆ 可以：某个IP地址主机名（如localhost）、某个
  - ◆ 也可包含：通配符%（任意个字符）、\_（任意单

思政点命名规范：用户名命名规则，告诫学生在上课学习、日常生活和将来的工作岗位中要遵守相应的制度和规定，并用以约束和指导自己的行为，以后才能成为合格的社会公民。

3. 学生跟练，完成实训作业，巡回指导（15min）

第1节结束

课间活动，大家站立走动，活动腰部、颈椎，进行健康教育

1. 新课导入 (5min)

为创建的用户创建不同的权限，每个用户允许做权利范围以内的，不允不可越界。

## ■ 权限管理

### ■ 用户执行SQL语句

- ◆ MySQL将逐级进行权限检查
- ◆ 看用户是否具有操作对象的SQL语句的执行权限

### ■ MySQL的权限管理

- ◆ 允许做权利范围以内的事情，不允许越界
- ◆ 例如：只允许执行SELECT操作，那么就不能执行INSERT

思政点：通过对不同的用户赋予不同的权限，进行数据库管理，让深刻体会到在社会中，每个人要做好自己该做的事情，进而逐渐养成的责任意识。。

2. 内容新授 (15min)

讲解，操作演示

## ■ 权限管理

### ■ MySQL中用户权限的五个层级

- ◆ 全局层级
- ◆ 数据库层级
- ◆ 表层级
- ◆ 列层级
- ◆ 子程序层级

权限	权限级别	权限说明
CREATE	数据库、表或索引	创建数据库、表或索引权限
DROP	数据库或表	删除数据库或表权限
GRANT OPTION	数据库、表或保存的程序	赋予权限选项
REFERENCES	数据库或表	
ALTER	表	更改表,如添加字段、索引、约束
DELETE	表	删除数据权限
INDEX	表	索引权限
INSERT	表	插入权限
SELECT	表	查询权限
UPDATE	表	更新权限
CREATE VIEW	视图	创建视图权限
SHOW VIEW	视图	查看视图权限
CREATE ROUTINE	存储过程	创建存储过程权限
权限	权限级别	权限说明
ALTER ROUTINE	存储过程	修改存储过程权限
EXECUTE	存储过程	执行存储过程
FILE	服务器主机上的文件访问	文件访问权限
CREATE TEMPORARY	服务器管理	创建临时表权限
LOCK TABLES	服务器管理	锁表管理
CREATE USER	服务器管理	创建用户权限
PROCESS	服务器管理	查看进程权限
RELOAD	服务器管理	执行 FLUSH.REFRESH.RELOAD 等命令
REPLICATION CLIENT	服务器管理	复制权限
REPLICATION SLAVE	服务器管理	复制权限
SHOW DATABASES	服务器管理	查看数据库权限
SHUTDOWN	服务器管理	关闭数据库权限
SUPER	服务器管理	执行 KILL 线程权限

## 权限授予

### 语法

```
GRANT 权限 1(列名列表 1)[, 权限 2(列名列表 2)]...  
ON|目标|{表名|*|*.|库名.*}  
TO 用户 1|[IDENTIFIED BY|PASSWORD] '密码 1'  
[,用户 2|[IDENTIFIED BY|PASSWORD] '密码 2']...  
[WITH 权限限制 1|权限限制 2]...
```

### 分析

#### ■ 权限

- ◆ 权限的名称，如SELECT、UPDATE等
- ◆ 给不同的对象授予权限的值也不相同

## 权限收回

### 语法

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 用户 1[,用
```

### 分析

#### ■ 回收该用户的所有权限

3. 学生完成对应实训，巡回指导（25min）。

——第1、2节结束——

### 任务二 数据库备份与恢复

#### 一、准备活动(3min)

1. 衢职有礼师生互动，清理桌面、键盘鼠标以及显示器-劳动教育。学习通签到。（1min）

2. 学习通预习技能点的学习情况检查。（1min）

3. 复习。（1min）

#### 二、发展活动(35min)

1. 新课导入-为什么需数据备份（10min）

如果系统管理员在对学生信息表进行管理，误删了重要的学生数据，为了挽回误操作造成的损失，数据库管理人员需要对数据库进行数据备份，在出现操作事故后可以将之前的数据还原。

## 备份数据库

- 目的：当数据库发生故障时可通过备份数据文件恢复数据
- 原因：
  - ◆ 存储介质故障：保存数据文件的磁盘设备损坏
  - ◆ 用户的错误操作：用户有意或无意删除了重要数据，甚至整个数据库
  - ◆ 服务器的瘫痪：数据库服务器因为软件漏洞彻底瘫痪
- 备份方式
  - ◆ 使用mysqldump备份
  - ◆ 使用SQL命令备份数据表
  - ◆ 使用mysql命令备份数据

思政点：数据库会因为各种原因造成破坏，因此需要做好数据库的备份工作。进而引申到学生在工作生活中，也要有居安思危的忧患意识，不能停滞不前，需要不断的创新来提升自己，提升学生的创新意识。

## 使用mysqldump备份数据库或者指定表

### 语法

```
mysqldump -u user -h host -p password db [tb1,[tb2,...]]>filename
```

### 分析

- -u后的user表示用户名
- -h后的host表示主机名
- -p后的password表示用户密码
  - ◆ -p选项和密码之间不能有空格
  - ◆ 如果是本地MySQL服务器，则-h选项可以省略

## ■ 使用mysqldump备份多个数据库

### ☰ 语法

```
mysqldump -u user -h host -p password  
--databases db1 [db2 [db3...]]>filename
```

### 🔍 分析

- --databases: 要备份多个数据库
  - ◆ 后面至少要指定一个数据库名称
  - ◆ 多个数据库用空格隔开
- db1、db2、db3表示要备份的多个数据库的名称

## ■ 使用SQL命令备份数据表

### ☰ 语法

```
SELECT columns FROM tablename [WHERE condition]  
INTO OUTFILE 'filename'[OPTION]
```

### 🔍 分析

- OPTION表示设置相应的选项
  - ◆ 决定数据行在文件中存放的格式
    - ✓ fields terminated by'string': 字段的分隔符为字符串对象, 默认为 “\t”
    - ✓ lines starting by'string': 每行开始的字符串符号, 默认不使用任何字符
    - ✓ lines terminated by'string': 每行结束的字符串符号, 默认为 “\n”
    - ✓ fields escaped by 'char': 设置转义字符的字符符号, 默认使用 “\”

## ■ 使用mysql命令恢复数据

### ☰ 语法

```
mysql -u root -p password [db]<filename
```

### 🔍 分析

- Db: 表示要还原的数据名称
  - ◆ 为可选项, 可以指定数据库名, 也可以不指定
  - ◆ 如果使用--all-databases备份所有数据库, 则还原时无需指定数据库
  - ◆ 如果指定了数据库, 则数据库要先创建

2、操作演示- (学生跟练, 教师巡回指导 (10 分钟))  
小结 (5min)

	<p><b>关键步骤</b></p> <ul style="list-style-type: none"> <li>■ 执行数据备份或恢复操作时 <ul style="list-style-type: none"> <li>◆ 当不使用默认目录，需要修改配置文件</li> <li>◆ 保存配置文件后，重新启动MySQL数据服务器</li> </ul> </li> <li>■ 不同操作系统下的目录分隔符存在差异，注意区分</li> <li>■ 数据库备份在项目中极其重要 <ul style="list-style-type: none"> <li>◆ 要做到定时备份与即时故障恢复</li> </ul> </li> </ul> <p>课间活动，大家站立走动，活动腰部、颈椎，进行健康教育 根据实训要求，完成实训提交作业（40min） ——第 3、4 节结束——</p>
学习评量	
课后作业	
教学反思	