数 据 库 系 统 课 程 教 案

第 7 单元 **学时：** 2

教材内容	第 3 章　SQL 语言（3.1～3.4.1）
基本知识点	SQL 语言发展的过程、SQL 语言的特点与两种使用方式、SQL 语言与非关系模型数据语言的不同、视图的概念和作用、SQL 语言对关系数据库模式的支持、SQL 的表定义功能、SQL 的索引定义功能、SQL 的单表查询
教学重点	视图的概念和作用、SQL 语言对关系数据库模式的支持、SQL 的表定义功能、SQL 的索引定义功能、SQL 的单表查询
教学难点	SQL 的单表查询
要求掌握内容	SQL 语言的特点与两种使用方式、SQL 语言与非关系模型数据语言的不同、视图的概念和作用、SQL 语言对关系数据库模式的支持、SQL 的表定义功能、SQL 的索引定义功能、SQL 的单表查询
教学思路，采用的教学方法和辅助手段，板书设计，重点如何突出，难点如何解决，师生互动等	**教学思路：** **一、复习旧课，巩固上次授课主要内容** 1、简述 SQL Server 的数据库结构。 2、简述存储数据库数据的操作系统文件的分类。 **二、导入新课，明确本次授课的目的与要求** 1、理解 SQL 语言的特点。 2、理解索引的概念。 3、理解唯一索引与聚簇索引的区别。 4、掌握 SQL 的表定义功能。 5、掌握 SQL 的索引定义功能。 6、掌握 SQL 的单表查询。 **三、讲解本次授课的具体内容** **教学方法：**整合教学内容，强化基础训练；努力营造生动活泼的课堂气氛，搭建师生间良好的沟通渠道；采用多媒体教学与传统的板书设计相结合的方式，教学手段灵活多变。 **辅助手段：**通过 PPT 幻灯片演示并加以阐述；对于关系代数中的连接和除运算，通过 PPT 幻灯片演示结合板书设计和举例加以阐述。
本章思考题和作业	P130 第 1、3 题
主要教材参考资料	1.《数据库系统概论》，萨师煊，王珊，高等教育出版社，2014.9 2.《数据库系统概论学习指导与习题解答》，王珊，张俊，高等教育出版社，2015.7
备 注	

本次授课具体内容

第 3 章　SQL 语言

3.1　SQL 概述：SQL 的特点：1、综合统一：SQL 语言集数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的功能于一体。

2、高度非过程化：非关系数据模型的数据操纵语言是面向过程的语言，操作必须指明存取路径；而用 SQL 语言进行数据操作，只要提出“做什么”，无须指明“怎么做”，因此无需了解存取路径。

3、面向集合的操作方式：非关系数据模型采用的是面向记录的操作方式，操作对象是一条记录；而 SQL 语言采用的集合操作方式，不仅操作对象、查询结果可以是元组的集合，而且一次插入、删除、更新操作的对象也可以是元组的集合。4、以同一种语法结构提供两种使用方法：SQL 语言既是自含式语言，能独立地用于联机交互；又是嵌入式语言，能嵌入到高级语言中进行混合编程。

5、语言简洁，易学易用，完成核心功能只用 9 个动词。

3.3　数据定义

㈠定义、修改与删除基本表

1、定义基本表

CREATE TABLE <表名> (<列名> <数据类型>[<列级完整性约束条件>]

[,<列名> <数据类型>[<列级完整性约束条件>]]…

[,<表级完整性约束条件>]);

其中：表名为所要定义的基本表的名字，列名为组成该表的各个属性(列)，列级完整性约束条件为涉及相应属性列的完整性约束条件,表级完整性约束条件为涉及一个或多个属性列的完整性约束条件。常用的完整性约束有：主码约束 PRIMARY　KEY、唯一性约束 UNIQUE、非空值约束 NOT NULL、参照完整性约束 FOREIGN KEY　REFERENCES。

[例 1]建立一个“学生”表 Student，它由学号 Sno、姓名 Sname、性别 Ssex、年龄 Sage、系别 Sdept 五个属性组成。其中学号不能为空，值是唯一的，并且姓名取值也唯一。

CREATE TABLE Student (Sno　CHAR(5)　NOT NULL　UNIQUE，　Sname　CHAR(20)　UNIQUE，

Ssex　CHAR(1) ，Sage　INT，Sdept　CHAR(15))；

2、修改基本表

ALTER TABLE <表名>[ADD <新列名> <数据类型> [完整性约束]]

[DROP <完整性约束名>] [DROP column <列名>][MODIFY <列名> <数据类型>];

其中：表名为要修改的基本表，ADD 子句为增加新列和新的完整性约束条件，DROP 子句为删除指定的完整性约束条件，MODIFY 子句为用于修改列名和数据类型。

[例 2]向 Student 表增加“入学时间”列，其数据类型为日期型。

ALTER TABLE Student ADD Scome DATE;

不论基本表中原来是否已有数据，新增加的列一律为空值。

本次授课具体内容（续）

[例 3]将年龄的数据类型改为半字长整数。 ALTER TABLE Student MODIFY Sage SMALLINT;

[例 4]删除学生姓名必须取唯一值的约束。 ALTER TABLE Student DROP UNIQUE(Sname);

3、删除基本表

DROP TABLE <表名>;

基本表定义一旦删除，表中的数据、表上建立的索引和视图都将自动删除。

[例 5]删除 Student 表。 DROP TABLE Student ;

㈡建立与删除索引

建立索引是加快查询速度的有效手段，建立与删除索引由 DBA 或表的属主负责完成，但有些 DBMS 自动建立 PRIMARY 或 KEY UNIQUE 列上的索引。

1、 建立索引

CREATE [UNIQUE] [CLUSTER] INDEX <索引名>ON <表名>(<列名>[<次序>][,<列名>[<次序>]]…);

说明：用<表名>指定要建索引的基本表；索引可以建立在该表的一列或多列上，各列名之间用逗号分隔；用<次序>指定索引值的排列次序，升序 ASC，降序 DESC，缺省 ASC； UNIQUE 表明此索引的每一个索引值只对应唯一的数据记录；CLUSTER 表示要建立的索引是聚簇索引。

[例 6]为学生-课程数据库中 Student,Course,SC 三个表建立索引。其中 Student 表按学号升序建唯一索引，Course 表按课程号升序建唯一索引，SC 表按学号升序和课程号降序建唯一索引。

CREATE UNIQUE INDEX Stusno ON Student(Sno);

CREATE UNIQUE INDEX Coucno ON Course(Cno);

CREATE UNIQUE INDEX SCno ON SC(Sno ASC，Cno DESC);

⑴唯一索引与聚簇索引：对于已含重复值的属性列不能建 UNIQUE 索引，对某个列建立 UNIQUE 索引后,插入新记录时 DBMS 会自动检查新记录在该列上是否取了重复值,这相当于增加了一个 UNIQUE 约束。

⑵聚簇索引：建立聚簇索引后，基表中数据也需要按指定的聚簇属性值的升序或降序存放，即聚簇索引的索引项顺序与表中记录的物理顺序一致。

例：CREATE CLUSTER INDEX Stusname ON Student(Sname);

在 Student 表的 Sname 列上建立一个聚簇索引,而且 Student 表中的记录将按照 Sname 值的升序存放。在一个基本表上最多只能建立一个聚簇索引，聚簇索引对于某些类型的查询，可以提高查询效率，聚簇索引在很少对基表进行增删操作或很少对其中的变长列进行修改操作的场合下非常适用。

2、删除索引

DROP INDEX <索引名>；删除索引时，系统会从数据字典中删去有关该索引的描述。

[例 7]删除 Student 表的 Stusname 索引。 DROP INDEX Stusname;

授课具体内容及实施过程

3.4 查询

㈠概述

查询语句格式：

SELECT [ALL|DISTINCT] <目标列表达式> [，<目标列表达式>]

FROM <表名或视图名>[，<表名或视图名>] …[WHERE <条件表达式>]

[GROUP BY <列名 1> [HAVING <条件表达式>]] [ORDER BY <列名 2> [ASC|DESC]]；

其中：SELECT 子句指定要显示的属性列；FROM 子句指定查询对象(基本表或视图)；WHERE 子句指定查询条件；GROUP BY 子句对查询结果按指定列的值分组，该属性列值相等的元组为一个组，通常会在每组中作用集函数；HAVING 短语筛选出只有满足指定条件的组；ORDER BY 子句对查询结果表按指定列值的升序或降序排序。

㈡单表查询：查询仅涉及一个表，是一种最简单的查询操作。

1、选择表中的若干列

⑴查询指定列或全部列

[例 1]查询全体学生的学号与姓名。SELECT Sno，Sname FROM Student；

[例 2]查询全体学生的姓名、学号、所在系。SELECT Sname，Sno，Sdept FROM Student；

[例 3]查询全体学生的详细记录。SELECT Sno，Sname，Ssex，Sage，Sdept FROM Student；

或 SELECT * FROM Student；

2、查询经过计算的值：SELECT 中的<目标列表达式>为算术表达式、字符串常量、函数、列别名等 。

[例 4]查全体学生的姓名及其出生年份。SELECT Sname，2000-Sage FROM Student；

[例 5]查询全体学生的姓名、出生年份和所有系，要求用小写字母表示所有系名。

SELECT Sname，'Year of Birth:'，2000-Sage，ISLOWER(Sdept) FROM Student；

可以使用列别名改变查询结果的列标题：

SELECT Sname NAME，'Year of Birth:' BIRTH，2000-Sage BIRTHDAY FROM Student；

3、选择表中的若干元组：在 WHERE 子句中设置查询条件，使用 DISTINCT 短语消除取值重复的行。

[例 6]查询选修了课程的学生学号。SELECT Sno FROM SC;

或 SELECT ALL Sno FROM SC; SELECT DISTINCT Sno FROM SC;

注意：DISTINCT 短语的作用范围是所有目标列。

⑴确定大小：在 WHERE 子句的<比较条件>中使用比较运算符。

[例 8]查询所有年龄在 20 岁以下的学生姓名及其年龄。

SELECT Sname，SageFROM Student WHERE Sage < 20；

授课具体内容及实施过程

或 SELECT Sname，Sage FROM Student WHERE NOT Sage >= 20；

⑵确定范围：在 WHERE 子句的<比较条件>中使用谓词 BETWEEN… AND 或 NOT BETWEEN … AND

[例 10]查询年龄在 20～23 岁(包括 20 岁和 23 岁)间的学生的姓名、系别和年龄。

SELECT Sname，Sdept，Sage FROM Student WHERE Sage BETWEEN 20 AND 23；

[例 11]查询年龄不在 20～23 岁之间的学生姓名、系别和年龄。

SELECT Sname，Sdept，Sage FROM Student WHERE Sage NOT BETWEEN 20 AND 23；

⑶确定集合：在 WHERE 子句的<比较条件>中使用用谓词 IN(值表)或 NOT IN (值表)，(值表)是用逗号分隔的一组取值。

[例 12]查询信息系、数学系和计 算机系学生的姓名和性别。

SELECT Sname，Ssex FROM Student WHERE Sdept IN ('IS'，'MA'，'CS')；

[例 13]查询既不是信息系、数学系，也不是计算机系的学生的姓名和性别。

SELECT Sname，Ssex FROM Student WHERE Sdept NOT IN ('IS'，'MA'，'CS')；

⑷字符串匹配：在 WHERE 子句的<比较条件>中使用谓词[NOT] LIKE'匹配串' [ESCAPE' 换码字符']。其中匹配串指定了匹配模板，匹配模板就是固定字符串或含通配符的字符串，当匹配模板为固定字符串时，可以用 =取代 LIKE ，用 != 或 < >取代 NOT LIKE 。通配符% (百分号)：代表任意长度(可以为 0)的字符串，例 a%b 表示以 a 开头、以 b 结尾的任意长度的字符串，如 acb、addgb、ab 等；通配符_ (下横线)代表任意单个字符，例 a_b 表示以 a 开头、以 b 结尾的长度为 3 的任意字符串，如 acb、afb 等。ESCAPE 短语：当用户要查询的字符串本身就含有 % 或 _ 时，要使用 ESCAPE '换码字符' 短语对通配符进行转义。

[例 14]查询学号为 95001 的学生详细情况。SELECT * FROM Student WHERE Sno LIKE '95001'；

等价于：SELECT * FROM Student WHERE Sno ='95001'；

[例 15]查询所有姓刘学生的姓名、学号和性别。

SELECT Sname，Sno，Ssex FROM Student WHERE Sname LIKE '刘%'；

[例 16]查询所有不姓刘的学生姓名。

SELECT Sname，Sno，Ssex FROM Student WHERE Sname NOT LIKE '刘%'；

使用换码字符将通配符转义为普通字符：

[例 17]查询 DB_Design 课程的课程号和学分。

SELECT Cno，Ccredit FROM Course WHERE Cname LIKE 'DB_Design' ESCAPE '\'；

[例 18]询以 DB_开头、且倒数第 3 个字符为 i 的课程的详细情况。

SELECT * FROM Course WHERE Cname LIKE 'DB_%i_ _' ESCAPE '\'；

⑸涉及空值的查询：在 WHERE 子句的<比较条件>中使用谓词 IS NULL 或 IS NOT NULL，注意 IS NULL 不能用= NULL 代替。

[例 21]某些学生选修课程后没有参加考试，所以有选课记录但没有考试成绩。查询缺少成绩的学生的学号和相应的课程号。SELECT Sno，Cno FROM SC WHERE Grade IS NULL;

[例 22]查所有有成绩的学生学号和课程号。SELECT Sno，Cno FROM SC WHERE Grade IS NOT NULL;

⑹多重条件查询：在 WHERE 子句的<比较条件>中使用逻辑运算符 AND 和 OR 来联结多个查询条件，AND 的优先级高于 OR， 可用括号改变优先级。

授课具体内容及实施过程

[例 23]查询计算机系年龄在 20 岁以下的学生姓名。

SELECT Sname FROM Student WHERE Sdept= 'CS' AND Sage<20;

⑺对查询结果排序：使用 ORDER BY 子句可以按一个或多个属性列排序，升序 ASC，降序 DESC，缺省为升序。当排序列含空值时，ASC 排序列为空值的元组最后显示，DESC 排序列为空值的元组最先显示。

[例 24]查询选修了 3 号课程的学生的学号及成绩，查询结果按分数降序排列。

SELECT Sno，Grade FROM SC WHERE Cno= '3' ORDER BY Grade DESC;

[例 25]查询全体学生情况，结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

SELECT * FROM Student ORDER BY Sdept，Sage DESC; ⑻使用集函数： 5 类主要集函数

计数COUNT([DISTINCT|ALL] *)或COUNT([DISTINCT|ALL] <列名>)

计算总和SUM([DISTINCT|ALL] <列名>)

计算平均值AVG([DISTINCT|ALL] <列名>)

求最大值MAX([DISTINCT|ALL] <列名>)

求最小值MIN([DISTINCT|ALL] <列名>)

其中：DISTINCT 短语在计算时要取消指定列中的重复值，ALL 短语不取消重复值，ALL 为缺省值。

[例 26]查询学生总人数。 SELECT COUNT(*) FROM Student;

[例 27]查询选修了课程的学生人数。 SELECT COUNT(DISTINCT Sno) FROM SC;

注：用 DISTINCT 以避免重复计算学生人数。

[例 28]计算 1 号课程的学生平均成绩。 SELECT AVG(Grade) FROM SC WHERE Cno= '1';

[例 29]查询选修 1 号课程的学生最高分数。SELECT MAX(Grade) FROM SC WHERE Cno= '1';

⑼对查询结果分组： GROUP BY 子句的作用对象是查询的中间结果表，分组方法是按指定一列或多列值分组，值相等的为一组。使用 GROUP BY 子句后，SELECT 子句的列名列表中只能出现分组属性和集函数。使用 GROUP BY 子句分组细化集函数的作用对象，未对查询结果分组，集函数将作用于整个查询结果；对查询结果分组后，集函数将分别作用于每个组。

[例 30]求各个课程号及相应的选课人数。SELECT Cno，COUNT(Sno) FROM SC GROUP BY Cno;

⑽使用 HAVING 短语筛选最终输出结果：只有满足 HAVING 短语指定条件的组才输出，HAVING 短语与 WHERE 子句的区别是 WHERE 子句作用于基表或视图，从中选择满足条件的元组；HAVING 短语作用于组，从中选择满足条件的组。

[例 31]查询选修了 3 门以上课程的学生学号。

SELECT Sno FROM SC GROUP BY Sno HAVING COUNT(*) >3;

查询有 3 门以上课程是 90 分以上的学生的学号及课程数。

SELECT Sno，COUNT(*) FROM SC WHERE Grade>=90

GROUP BY Sno HAVING COUNT(*)>=3;

授课具体内容及实施过程

本次授课小结

本次授课讲述了连接、除等专门的关系运算、抽象的元组关系演算、抽象的域关系演算。

学生课后复习时应着重于其中的第 1 点内容，为进一步学习后续章节打好基础。

实验	实验 1　数据库的创建与管理