数 据 库 系 统 课 程 教 案

第 9 单元 **学时：** 4

教材内容	第 3 章　SQL 语言（3.5～3.7）
基本知识点	视图的概念及其与基本表的区别，掌握 SQL 的数据更新、SQL 的视图使用、SQL 的数据控制功能
教学重点	SQL 的数据更新、SQL 的视图使用、SQL 的数据控制功能
教学难点	SQL 的数据更新、SQL 的视图使用
要求掌握内容	SQL 的数据更新、SQL 的视图使用、SQL 的数据控制功能
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、什么是相关子查询与不相关子查询？ 2、简述 SELECT 语句的一般执行步骤。 **二、导入新课，明确本次授课的目的与要求** 1、掌握对空值的处理 2、理解视图的概念及其与基本表的区别。 3、掌握 SQL 的数据更新。 4、掌握 SQL 的视图使用。 5、掌握 SQL 的数据控制。 **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 幻灯片演示结合板书设计和举例加以阐述。
本章思考题和作业	P130 第 5⑻～⑾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

3.5 数据更新 ㈠插入数据：有插入单个元组、插入子查询结果两种方式 。

㈠插入数据：

1、插入单个元组

语句格式：INSERT INTO <表名> [(<属性列 1>[,<属性列 2 >…]) VALUES (<常量 1> [,<常量 2>] …)

功能：将新元组插入指定表中。其中：INTO 子句指定要插入数据的表名及属性列，属性列的顺序可与表定义中的顺序不一致。若没有指定属性列，则表示要插入的是一条完整的元组，且属性列属性与表定义中的顺序一致；若指定部分属性列，则插入的元组在其余属性列上取空值。VALUES 子句提供的值必须与 INTO 子句在值的个数和值的类型上匹配 。

[例 1]将一个新学生记录（95020，陈冬，男，IS，18 岁）插入到 Student 表中。

INSERT INTO Student VALUES ('95020'，'陈冬'，'男'，'IS'，18)；

[例 2]插入一条选课记录('95020'，'1')。

INSERT INTO SC(Sno，Cno) VALUES ('95020'，'1')；

新插入的记录在 Grade 列上取空值。

2、插入子查询结果

语句格式：INSERT INTO <表名> [(<属性列 1> [,<属性列 2>…]) 子查询；

功能：将子查询结果插入指定表中。其中：INTO、SELECT 子句的说明与插入单条元组类似。

[例 3]对每一个系，求学生的平均年龄，并把结果存入数据库。

建表：CREATE TABLE Deptage (Sdept CHAR(15) ，Avgage SMALLINT)；

插入数据：INSERT INTO Deptage(Sdept，Avgage)

SELECT Sdept，AVG(Sage) FROM Student GROUP BY Sdept；

注意点：无论是插入单个元组，还是插入子查询结果， DBMS 在执行插入语句时会自动检查所插元组是否破坏表上已定义的完整性规则(实体完整性、参照完整性、用户定义的完整性)：

对于有 NOT NULL 约束的属性列是否提供了非空值；对于有 UNIQUE 约束的属性列是否提供了非重复值；对于有值域约束的属性列所提供的属性值是否在值域范围内。

㈡修改数据：有修改某一个元组的值、修改多个元组的值、带子查询的修改三种方式，DBMS 也会自动检查三类完整性规则。

语句格式： UPDATE <表名> SET <列名>=<表达式>[,<列名>=<表达式>]… [WHERE <条件>]；

功能：修改指定表中满足 WHERE 子句条件的元组。其中：SET 子句指定修改方式、要修改的列和修改后的取值；WHERE 子句指定要修改的元组，缺省表示修改表中所有元组。

1、修改某一个元组的值[例 4]将学生 95001 的年龄改为 22 岁。 UPDATE Student SET Sage=22 WHERE Sno='95001'； 2、修改多个元组的值

[例 5]将所有学生的年龄增加 1 岁。 UPDATE Student SET Sage= Sage+1；

3、带子查询的修改语句

[例 6]将计算机系全体学生的成绩置零。

UPDATE SC SET Grade=0 WHERE 'CS'=(SELETE Sdept FROM Student WHERE Student.Sno=SC.Sno)；

㈢删除数据：有删除某一个元组的值、删除多个元组的值、带子查询的删除三种方式。DBMS 在执行删除语句时会检查所删元组是否破坏表上已定义的参照完整性规则(不允许删除或级联删除)。

本次授课具体内容（续）

语句格式：DELETE FROM <表名> [WHERE <条件>];

功能：删除指定表中满足 WHERE 子句条件的元组。其中：WHERE 子句指定要删除的元组，缺省表示要修改表中的所有元组。

1、删除某一个元组的值

[例 7]删除学号为 95019 的学生记录。 DELETE FROM Student WHERE Sno='95019';

2、删除多个元组的值

[例 8]删除所有的学生选课记录。 DELETE FROM SC;

3、带子查询的删除

[例 9]删除计算机系所有学生的选课记录。

DELETE FROM SC WHERE 'CS'=(SELETE Sdept FROM Student WHERE Student.Sno=SC.Sno);

3.6 空值的处理

1、空值的产生：空值是一个很特殊的值，含有不确定性。对关系运算带来特殊的问题，需要做特殊的处理。

2、空值的判断：判断一个属性的值是否为空值，用 IS NULL 或 IS NOT NULL 来表示

3、空值的约束条件：（1）属性定义（或者域定义）中；（2）有 NOT NULL 约束条件的不能取空值加了 UNIQUE 限制的属性不能取空值；（3）码属性不能取空值

4、空值的算术运算、比较运算和逻辑运算：（1）空值与另一个值（包括另一个空值）的算术运算的结果为空值；（2）空值与另一个值（包括另一个空值）的比较运算的结果为 UNKNOWN；有 UNKNOWN 后，传统二值（TRUE，FALSE）逻辑就扩展成了三值逻辑

3.7 视图

㈠视图的特点

1、视图是 RDBMS 提供给用户以多种角度观察数据库中数据的重要机制。

2、虚表，是从一个或几个基本表(或视图)导出的表。

3、只存放视图的定义，不会出现数据冗余。

4、基表中的数据发生变化，从视图中查询出的数据也随之改变。

㈡建立视图

语句格式：CREATE VIEW <视图名>[(<列名>[,<列名>]…)] AS <子查询> [WITH CHECK OPTION];

功能：DBMS 将视图的定义存入数据字典，并不执行其中的 SELECT 语句，查询视图时，按其定义从基本表中将数据查出。说明：组成视图的属性列名或全部省略或全部指定。若全部省略，则该视图由子查询中 SELECT 目标列中的诸字段组成；但下述三种情况必须明确指定组成视图的所有列名：某个目标列是集函数或列表达式、多表连接时选出了几个同名列作为视图的字段、需要在视图中为某个列启用新的更合适的名字。常见的视图形式：

1、行列子集视图：从单个基本表导出，视图只是去掉了基本表的某些行和某些列，但保留了码。

[例 1]建立信息系学生的视图。

CREATE VIEW IS_Student AS SELECT Sno，Sname，Sage FROM Student WHERE Sdept='IS';

2、WITH CHECK OPTION 视图：表示对视图进行增删改操作时不得破坏视图定义中的谓词条件(即子查询中的条件表达式)。

[例 2]建立信息系学生的视图，并要求透过该视图进行的更新操作只涉及信息系学生。

CREATE VIEW IS_Student AS SELECT Sno，Sname，Sage

FROM Student WHERE Sdept= 'IS' WITH CHECK OPTION;

对该 IS_Student 视图修改或删除操作时，DBMS 自动加上 Sdept= 'IS'的条件；插入操作时 DBMS 会自动检查 Sdept 属性值是否为'IS'，如果不是，则拒绝该插入操作；如果没有提供 Sdept 属性值，则自动定义 Sdept 为'IS'。

3、基于多个基表的视图

[例 3]建立信息系统选修了 1 号课程的学生视图。

CREATE VIEW IS_S1(Sno，Sname，Grade)

AS SELECT Student.Sno，Sname，Grade FROM Student，SC

WHERE Sdept= ¹IS¹ AND Student.Sno=SC.Sno AND SC.Cno= ¹1¹；

4、基于视图的视图：视图可建立在一个或多个已定义的视图上。

[例 4]建立信息系选修了 1 号课程且成绩在 90 分以上的学生视图。

CREATE VIEW IS_S2 AS SELECT Sno，Sname，Grade FROM IS_S1 WHERE Grade>=90；

5、带表达式的视图：必须明确定义组成视图的各个属性列名。

[例 5]定义一个反映学生出生年份的视图。

CREATE VIEW BT_S(Sno，Sname，Sbirth) AS SELECT Sno，Sname，2000-Sage FROM Student；

6、分组视图：带有集函数和 GROUP BY 子句查询的视图。

[例 6]将学生的学号及他的平均成绩定义为一个视图。

CREAT VIEW S_G(Sno，Gavg) AS SELECT Sno，AVG(Grade) FROM SC GROUP BY Sno；

7、一类不易扩充的视图：以 SELECT * 方式创建的视图，其可扩充性差，应尽可能避免。

[例 7]将 Student 表中所有女生记录定义为一个视图。

CREATE VIEW F_Student1(stdnum，name，sex，age，dept)

AS SELECT * FROM Student WHERE Ssex=¹女¹；

缺点：修改基表 Student 的结构后，基表与 F_Student1 视图的映象关系被破坏，导致该视图不能正确工作。解决方法：

CREATE VIEW F_Student2 (stdnum，name，sex，age，dept)

AS SELECT Sno，Sname，Ssex，Sage，Sdept FROM Student WHERE Ssex=¹女¹；

㈢删除视图

语句格式：DROP VIEW <视图名>；

功能：从数据字典中删除指定的视图定义。视图删除后，由该视图导出的其他视图定义仍在数据字典中，但这些视图已不能使用，必须使用 DROP VIEW 显式删除。基表删除后，由该基表导出的所有视图定义没有被删除，但却无法使用，必须使用 DROP VIEW 显式删除。

[例 8]删除视图 IS_S1。 DROP VIEW IS_S1；

㈣查询视图：视图定义后，用户可以象对基本表一样对视图进行查询。DBMS 实现视图查询时，首先进行有效性检查，检查所查询的表、视图是否存在。若存在，则从数据字典中取出视图的定义，把定义中的子查询和用户的查询结合起来，转换成等价的对基本表的查询，这一转换过程称为视图消解。

[例 1]在信息系学生的视图中找出年龄小于 20 岁的。

SELECT Sno，Sage FROM IS_Student WHERE Sage<20；

IS_Student 视图的定义 (视图定义例 1)为：

CREATE VIEW IS_Student AS SELECT Sno，Sname，Sage FROM Student WHERE Sdept=¹IS¹；

采用视图消解法转换后的查询语句为：

SELECT Sno，Sage FROM Student WHERE Sdept=¹IS¹ AND Sage<20；

[例 2]查询信息系统选修了 1 号课程的学生。

SELECT Sno，Sname FROM IS_Student，SC WHERE IS_Student.Sno =SC.Sno AND SC.Cno='1';

有些情况下，视图消解法不能生成正确查询，采用视图消解法的 DBMS 会限制这类查询。

[例 3]在 S_G 视图中查询平均成绩>=90 分的学生学号和平均成绩。

SELECT * FROM S_G WHERE Gavg>=90； S_G 视图定义为：CREATE VIEW S_G (Sno，Gavg)AS SELECT Sno，AVG(Grade) FROM SC GROUP BY Sno;

错误的转换： SELECT Sno，AVG(Grade) FROM SC WHERE AVG(Grade)>=90 GROUP BY Sno;

正确的转换：SELECT Sno，AVG(Grade) FROM SC GROUP BY Sno HAVING AVG(Grade)>=90;

㈤更新视图：通过视图实现插入、删除和修改数据。由于视图是不实际存储数据的虚表，因此对视图的更新最终要转换为对基本表的更新。 DBMS 实现对视图的更新也是采用视图消解法。用户指定 WITH CHECK OPTION 子句后，DBMS 在更新视图时会进行检查。

[例 1]将视图 IS_Student 中学号 95002 的学生姓名改为刘辰。

UPDATE IS_Student SET Sname= '刘辰' WHERE Sno= '95002';

转换后的语句：UPDATE Student SET Sname= '刘辰' WHERE Sno= '95002' AND Sdept= 'IS';

[例 2]向信息系学生视图 IS_S 中插入一个新的学生记录：95029，赵新，20 岁。

INSERT INTO IS_Student VALUES('95029'，'赵新'，20);

转换为对基本表的更新：

INSERT INTO Student(Sno，Sname，Sage，Sdept) VALUES('95029'，'赵新'，20，'IS');

[例 3]删除视图 CS_S 中学号为 95029 的记录。

DELETE FROM IS_Student WHERE Sno='95029';

转换为对基本表的更新：DELETE FROM Student WHERE Sno= '95029' AND Sdept='IS';

更新视图的限制：一些视图是不可更新的，因为对这些视图的更新不能唯一地有意义地转换成对相应基本表的更新。

1、允许对行列子集视图进行更新。

2、若视图是由两个以上基本表导出的，则此视图不允许更新。

3、若视图的字段来自字段表达式或常数，则不允许对此视图执行 INSERT 和 UPDATE 操作，但允许执行 DELETE 操作。

4、若视图的字段来自集函数，则此视图不允许更新。

5、若视图定义中含有 GROUP BY 子句，则此视图不允许更新。

6、若视图定义中含有 DISTINCT 短语，则此视图不允许更新。

7、一个不允许更新的视图上定义的视图也不允许更新。

8、若视图定义中有嵌套查询，并且内层查询的 FROM 子句中涉及的表也是导出该视图的基本表，则此视图不允许更新。

㈥视图的作用

1、能够简化用户的操作：当视图中数据不是直接来自基本表时，如基于多张表连接形成的视图、基于复杂嵌套查询的视图、含导出属性的视图等，定义视图能够简化用户的操作。

2、使用户能以多种角度看待同一数据：视图机制能使不同用户以不同方式看待同一数据，适应数据库共享的需要。

3、对重构数据库提供了一定程度的逻辑独立性：关系数据库中数据库的重构往往是不可避免的，重构数据库最常见的是将一个基本表垂直地分成多个基本表。

4、能够对机密数据提供安全保护：对不同用户定义不同视图，使每个用户只能看到他有权看到的数据。通过 WITH CHECK OPTION 可以对关键数据定义操作时间限制。

本次授课小结

本次授课讲述了视图的概念及其与基本表的区别、SQL 的数据更新、SQL 的视图使用、SQL 的数据控制。

学生课后复习时应着重于其中的第 2、3、4 点内容，为进一步学习后续章节打好基础。

实验	实验 3　SQL 数据查询