



2024年春季学期

数据库系统概论

An Introduction to Database Systems

第三章 关系数据库标准语言SQL

中国科学技术大学 大数据学院

黄振亚, huangzhy@ustc.edu.cn



第三章 关系数据库标准语言SQL

298

3.1 SQL概述

3.2 学生-课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 空值的处理

3.7 视图

3.8 小结



3.5 数据更新

299

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据



3.5.1 插入数据

300

- 两种插入数据方式
 1. 插入元组
 2. 插入子查询结果（学习过select后再介绍）
 - 可以一次插入多个元组



二、插入子查询结果 (select学过以后再介绍)

311

□ 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
- INTO子句(与插入元组类似)
- 子查询
 - **SELECT子句目标列必须与INTO子句匹配**
 - 值的个数
 - 值的类型



插入子查询结果（续）

312

[例3.72] 对每一个系，求学生的平均年龄，并把结果存入数据库。

第一步：建表

```
CREATE TABLE Dept_age  
  (Sdept CHAR(15)          /* 系名*/  
   Avg_age SMALLINT);    /* 学生平均年龄*/
```

第二步：插入数据

```
INSERT  
INTO Dept_age(Sdept, Avg_age)  
  SELECT Sdept, AVG(Sage)  
  FROM Student  
  GROUP BY Sdept
```



插入数据

313

- **RDBMS**在执行插入语句时会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
 - **NOT NULL**约束
 - **UNIQUE**约束
 - 值域约束

PS: RDBMS检查功能有限，应用程序中做更复杂的检查
数据完整性非常重要



3.5 数据更新

314

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据



3.4.2 修改数据

315

□ 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
- **SET**子句给出<表达式>的值用于取代相应的属性列
- 如果省略**WHERE**子句，表示要修改表中的所有元组



修改数据（续）

317

- 三种修改方式
 1. 修改某一个元组的值
 2. 修改多个元组的值
 3. 带子查询的修改语句



3. 带子查询的修改语句（之后学习）

320

[例3.75] 将计算机科学系全体学生的成绩置0。

```
UPDATE SC
```

```
SET Grade=0
```

```
WHERE Sno IN
```

```
(SELETE Sno
```

```
FROM Student
```

```
WHERE Sdept = 'CS');
```



3. 带子查询的修改语句（之后学习）

321

[例3.75] 将计算机科学系全体学生的成绩置0。

```
UPDATE SC
```

```
SET Grade=0
```

```
WHERE 'CS' =
```

```
(SELETE Sdept
```

```
FROM Student
```

```
WHERE Student.Sno = SC.Sno);
```



修改数据（续）

322

- ✓ RDBMS在执行修改语句时会检查修改操作是否破坏表上已定义的完整性规则
 - 实体完整性
 - 主码不允许修改
 - 用户定义的完整性
 - NOT NULL约束
 - UNIQUE约束
 - 值域约束



3.5 数据更新

323

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据



3.5.3 删除数据

324

□ 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

□ WHERE子句

- 指定要删除的元组
- 缺省表示要删除表中的所有元组，表的定义仍在字典中



删除数据（续）

325

- 三种删除方式
 1. 删除某一个元组的值
 2. 删除多个元组的值
 3. 带子查询的删除语句



3. 带子查询的删除语句（之后学习）

329

[例3.78] 删除计算机科学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. 带子查询的删除语句

330

[例3.78] 删除计算机科学系所有学生的选课记录(**Mysql 例子**)。

```
DELETE
FROM SC
WHERE Sno IN (
  SELECT Sno
  FROM Student
  WHERE Sdept = 'CS'
);
```

执行后可能会报错！主要是因为**Mysql**运行在**safe-updates**模式下，该模式会导致非主键条件下无法执行**update**或者**delete**命令！

✘ 83 19:54:00 D... Error Code: 1175. You are using safe update mode and you trie... 0.00091 sec



3. 带子查询的删除语句

331

[例3.78] 删除计算机科学系所有学生的选课记录(**Mysql 例子**)。

```
SET SQL_SAFE_UPDATES = 0;
```

解除安全模式后再运行上述删除指令，即可完成删除

删除前

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

SC 31

删除后

Sno	Cno	Grade
HULL	HULL	HULL



第三章 关系数据库标准语言SQL

335

3.1 SQL概述

3.2 学生-课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 空值的处理

3.7 视图

3.8 小结



3.6 空值的处理

336

- 空值就是“不知道”或“不存在”或“无意义”的值。
- 一般有以下几种情况：
 - 该属性应该有一个值，但目前不知道它的具体值
 - 该属性不应该有值
 - 由于某种原因不便于填写



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。

□ 空值的产生

[例 3.79]向SC表中插入一个元组，学生号是”200215126”，课程号是”1”，**成绩为空**。

```
INSERT INTO SC(Sno,Cno,Grade)
```

```
VALUES('200215126 ','1',NULL); /*该学生还没有考试成绩，取空值*/
```

或

```
INSERT INTO SC(Sno,Cno)
```

```
VALUES(' 200215126 ','1'); /*没有赋值的属性，其值为空值*/
```



空值的产生（续）

[例3.80] 将Student表中学生号为”200215200”的学生所属的系改为空值。

```
UPDATE Student  
SET Sdept = NULL  
WHERE Sno='200215200';
```



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。

[例 3.81] 从 Student 表中找出漏填了数据的学生信息

```
SELECT *  
FROM Student  
WHERE Sname IS NULL OR Ssex IS NULL OR Sage IS  
NULL OR Sdept IS NULL;
```




3. 空值的约束条件

- 属性定义（或者域定义）中
 - 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
 - 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

- 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
- 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



4. 空值的算术运算、比较运算和逻辑运算

表3.8 逻辑运算符真值表

x	y	x AND y	x OR y	NOT x
T	T	T	T	F
T	U	U	T	F
T	F	F	T	F
U	T	U	T	U
U	U	U	U	U
U	F	F	U	U
F	T	F	T	T
F	U	F	U	T
F	F	F	F	T

T表示TRUE, F表示FALSE, U表示UNKNOWN



空值的算术运算、比较运算和逻辑运算（续）

[例3.82] 找出选修1号课程的不及格的学生。

```
SELECT Sno  
FROM SC  
WHERE Grade < 60 AND Cno='1';
```

查询结果不包括缺考的学生，因为他们的Grade值为
null。



空值的算术运算、比较运算和逻辑运算（续）

[例 3.83] 选出选修1号课程的不及格的学生以及缺考的学生。

```
SELECT Sno
FROM SC
WHERE Grade < 60 AND Cno='1'
UNION
SELECT Sno
FROM SC
WHERE Grade IS NULL AND Cno='1'
```

或者

```
SELECT Sno
FROM SC
WHERE Cno='1' AND (Grade<60 OR Grade IS NULL);
```

课后尝试



第三章 关系数据库标准语言SQL

345

3.1 SQL概述

3.2 学生-课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 空值的处理

3.7 视图

3.8 小结



3.7 视图

346

视图的特点

- **虚表**，是从一个或几个基本表（或视图）导出的表
- 只存放视图的定义，不存放视图对应的数据
- 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



3.7 视图

347

基于视图的操作

- 查询
- 删除
- 受限更新
- 定义基于该视图的新视图



3.7 视图

348

3.7.1 定义视图

3.7.2 查询视图

3.7.3 更新视图

3.7.4 视图的作用



3.6.1 定义视图

349

- 建立视图
- 删除视图



一、建立视图

350

□ 语句格式

CREATE VIEW

<视图名> [(<列名> [, <列名>]...)]

AS <子查询>

[WITH CHECK OPTION];

- 组成视图的属性列名：全部省略或全部指定
- 子查询可以是任意的SELECT语句，是否可以含有ORDER BY子句和DISTINCT短语，则决定具体系统的实现



建立视图（续）

351

- 组成视图的属性列名：全部省略或全部指定
 - 全部省略：
 - 由子查询中SELECT目标列中的诸字段组成
 - 明确指定视图的所有列名：
 - 某个目标列是聚集函数或列表表达式
 - 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



建立视图（续）

352

- **WITH CHECK OPTION**
 - 对视图进行UPDATE，INSERT和DELETE操作时要保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）



建立视图（续）

353

[例3.84] 建立信息系学生的视图。

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



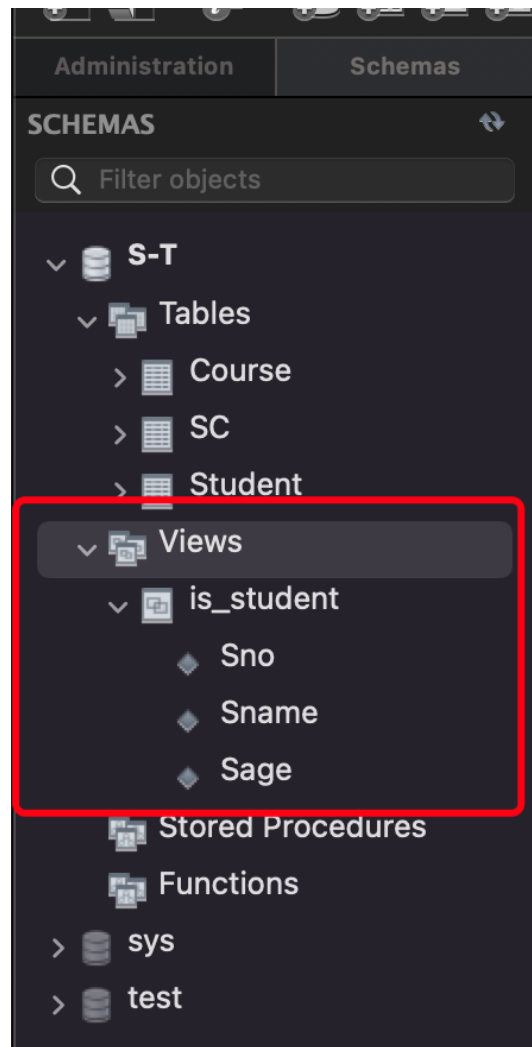
建立视图（续）

[例3.84] 建立信息系学生的视图

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

可以通过下方输出信息或者左侧的 **schemas** 查看视图创建情况！

```
✓ 96 20:06:42 CREATE V... 0 row(s) affected
```





建立视图（续）

355

- RDBMS执行CREATE VIEW语句时只是把视图定义存入数据字典，并不执行其中的SELECT语句。
- 在对视图查询时，按视图的定义从基本表中将数据查出。



建立视图（续）

356

[例3.85]建立信息系学生的视图，并要求进行修改和插入操作时仍需保证该视图只有信息系IS的学生。

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

定义IS_Student视图时加上了WITH CHECK OPTION子句，对该视图进行插入、修改和删除操作时，RDBMS会自动加上Sdept='IS'的条件。



建立视图（续）

357

对IS_Student视图的更新操作：

- 修改操作：自动加上Sdept= 'IS'的条件
- 删除操作：自动加上Sdept= 'IS'的条件
- 插入操作：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'



建立视图（续）

358

- 若一个视图是从单个基本表导出的，并且只是去掉了基本表的某些行和某些列，但保留了主码，我们称这类视图为行列子集视图。
 - IS_Student视图就是一个行列子集视图
 - 行列子集视图，是可以更新的



建立视图（续）

359

□ 基于多个基本表的视图

[例3.86] 建立信息系选修了1号课程的学生视图。（成绩单）

```
CREATE VIEW IS_S1(Sno, Sname, Grade)
```

```
AS
```

```
SELECT Student.Sno, Sname, Grade
```

```
FROM Student, SC
```

```
WHERE Sdept= 'IS' AND
```

```
Student.Sno=SC.Sno AND
```

```
SC.Cno= '1';
```



建立视图（续）

360

□ 基于视图的视图

[例3.87] 建立信息系选修了1号课程且成绩在90分以上的学生的视图。

```
CREATE VIEW IS_S2
```

```
AS
```

```
SELECT Sno, Sname, Grade
```

```
FROM IS_S1
```

```
WHERE Grade>=90;
```



建立视图（续）

361

□ 带表达式的视图

[例3.88] 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, 2023-Sage
FROM Student;
```



建立视图（续）

362

□ 分组视图

[例3.89] 将学生的学号及他的平均成绩定义为一个视图

假设SC表中“成绩”列Grade为数字型

```
CREAT VIEW S_G(Sno, Gavg)
```

```
AS
```

```
SELECT Sno, AVG(Grade)
```

```
FROM SC
```

```
GROUP BY Sno;
```



建立视图（续）

363

□ 不指定属性列

[例3.90] 将Student表中所有女生记录定义为一个视图

```
CREATE VIEW F_Student(F_Sno, name, sex, age, dept)
```

```
AS
```

```
SELECT *
```

```
FROM Student
```

```
WHERE Ssex='女' ;
```

缺点：

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



二、删除视图

364

□ 语句的格式:

DROP VIEW <视图名>;

- 该语句从数据字典中删除指定的视图定义
- 如果该视图上还导出了其他视图，使用**CASCADE**级联删除语句，把该视图和由它导出的所有视图一起删除
- 删除基表时，由该基表导出的所有视图定义都必须显式地使用**DROP VIEW**语句删除



删除视图(续)

365

[例3.91] 删除视图BT_S和IS_S1

```
DROP VIEW BT_S;          /*成功执行*/
```

```
DROP VIEW IS_S1;        /*拒绝执行*/
```

要删除IS_S1，需使用级联删除：

```
DROP VIEW IS_S1 CASCADE;
```

✓	108	20:17:52	Drop VIEW BT_S	0 row(s) affected
✓	127	20:23:43	Drop VIEW IS_S1	0 row(s) affected
✗	139	20:27:29	Error loading schem...	Error Code: 1356 View 'S-T.is_s2' references invalid table(s) or colum...



3.7 视图

366

3.7.1 定义视图

3.7.2 查询视图

3.7.3 更新视图

3.7.4 视图的作用



3.7.2 查询视图

367

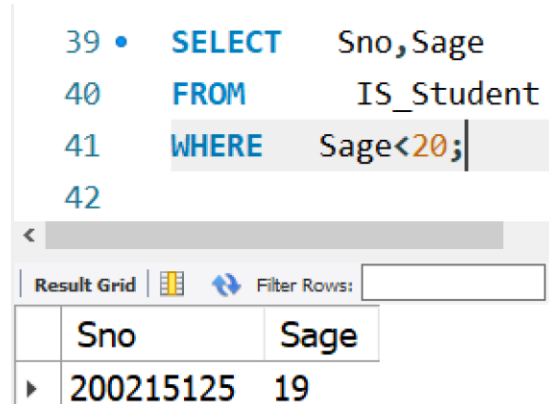
- 用户角度：查询视图与查询基本表相同
- RDBMS实现视图查询的方法（检查视图定义）
 - 视图消解法（View Resolution）
 - 进行有效性检查
 - 转换成等价的对基本表的查询
 - 执行修正后的查询



查询视图 (续)

[例3.92] 在信息系学生的视图中找出年龄小于20岁的学生。

```
SELECT Sno, Sage
FROM IS_Student
WHERE Sage<20;
```



IS_Student视图的定义 (参见视图定义例3.84)

[例3.84] 建立信息系学生的视图。

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

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200515125	张立	男	19	IS



查询视图（续）

369

[例3.92] 在信息系学生的视图中找出年龄小于20岁的学生。

```
SELECT Sno, Sage
FROM IS_Student
WHERE Sage<20;
```

视图消解转换后的查询语句为：

```
SELECT Sno,Sage
FROM Student
WHERE Sdept= 'IS' AND Sage<20;
```



查询视图（续）

370

[例3.93] 查询选修了1号课程的信息系学生

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

问：视图消解转换后的查询语句为？



查询视图（续）

371

- 视图消解法的局限
 - 有些情况下，视图消解法使用需注意。

[例3.94] 在S_G视图中查询平均成绩在90分以上的学生学号和平均成绩

```
SELECT *  
FROM S_G  
WHERE Gavg>=90;
```

S_G视图的子查询定义【例3.89】：

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




查询转换

372

错误:

```
SELECT Sno, AVG(Grade)
FROM SC
WHERE AVG(Grade)>=90
GROUP BY Sno;
```

正确:

```
SELECT Sno, AVG(Grade)
FROM SC
GROUP BY Sno
HAVING AVG(Grade)>=90;
```



查询转换

373

[例3.94]也可以用如下SQL语句完成

```
SELECT *  
FROM (SELECT Sno,AVG(Grade)  
FROM SC  
GROUP BY Sno) AS S_G(Sno,Gavg)  
WHERE Gavg>=90;
```



3.7 视图

374

3.7.1 定义视图

3.7.2 查询视图

3.7.3 更新视图

3.7.4 视图的作用



更新视图

375

□ WITH CHECK OPTION

- 对视图进行UPDATE, INSERT和DELETE操作时要保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）
 - 修改操作：自动加上Sdept= 'IS'的条件
 - 删除操作：自动加上Sdept= 'IS'的条件
 - 插入操作：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'



更新视图（续）

376

[例3.95] 将信息系学生视图IS_Student中学号200215122的学生姓名改为“刘辰”。

```
UPDATE IS_Student
SET Sname= '刘辰'
WHERE Sno= ' 200215122 ';
```

转换后的语句:

```
UPDATE Student
SET Sname= '刘辰'
WHERE Sno= ' 200215122 ' AND Sdept= 'IS';
```

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



更新视图（续）

377

[例3.96] 向信息系学生视图IS_S中插入一个新的学生记录：
200215129， 赵新， 20岁

```
INSERT  
INTO IS_Student  
VALUES('200215129', '赵新', 20);
```

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

转换为对基本表的更新：

```
INSERT  
INTO Student(Sno, Sname, Sage, Sdept)  
VALUES('200215129', '赵新', 20, 'IS');
```



更新视图（续）

378

[例3.97]删除信息系学生视图IS_Student中学号为200215129的记录。

```
DELETE
FROM IS_Student
WHERE Sno= ' 200215129 ';
```

转换为对基本表的更新：

```
DELETE
FROM Student
WHERE Sno= ' 200215129 ' AND Sdept= 'IS';
```

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



更新视图（续）

379

- 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

例： 3.89定义视图S_G为不可更新视图。

```
UPDATE S_G  
SET     Gavg=90  
WHERE  Sno='200215121';
```

这个对视图的更新无法转换成对基本表SC的更新



更新视图（续）

380

- 允许对**行列子集视图**进行更新
- 对其他类型视图的更新**不同系统有不同限制**
 - P127, DB2的规定



更新视图（续）

381

- DB2对视图更新的限制：
 - 若视图是由两个以上基本表导出的，则此视图不允许更新
 - 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
 - 若视图的字段来自集函数，则此视图不允许更新
 - 若视图定义中含有GROUP BY子句，则此视图不允许更新
 - 若视图定义中含有DISTINCT短语，则此视图不允许更新
 - 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
 - 一个不允许更新的视图上定义的视图也不允许更新



3.7 视图

382

3.7.1 定义视图

3.7.2 查询视图

3.7.3 更新视图

3.7.4 视图的作用



3.7.4 视图的作用

383

- 1. 视图能够简化用户的操作
 - 简化查询等操作
- 2. 视图使用户能以多种角度看待同一数据
 - 便于多用户共享同一数据库时
- 3. 视图对重构数据库提供了一定程度的**逻辑独立性**
 - 有时数据库的逻辑结构改变了，但是应用程序不用变（改视图定义）
- 4. 视图能够对机密数据提供安全保护
 - 机密数据不出现在低权限用户的视图中
- 5. 适当的利用视图可以更清晰的表达查询
 - 例，P129，“查询每个同学的获得最高成绩的课程号”



什么时候使用视图

384

- 经常用到的查询，或较复杂的联合查询应当创立视图，这是会优化性能的
- 涉及到权限管理方面，比如某表中的部分字段含有机密信息，不应当让低权限的用户访问到的情况，这时候给这些用户提供一个适合他们权限的视图，供他们阅读自己的数据就行了。
- 当一个查询逻辑复杂，且别的查询需要查询这个查询的结果时用视图。还有就是敏感数据



第三章 关系数据库标准语言SQL

385

3.1 SQL概述

3.2 学生-课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 空值的处理

3.7 视图

3.8 小结



第三章 关系数据库标准语言SQL

386

- SQL可以分为数据定义、数据查询、数据更新、数据控制四大部分
- SQL是关系数据库语言的工业标准。大部分数据库管理系统产品都能支持SQL92,但是许多数据库系统只支持SQL99、SQL2008和SQL2011的部分特征，至今尚没有一个数据库系统能够完全支持SQL99以上的标准。