

第三章 关系数据库标准语言 SQL



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

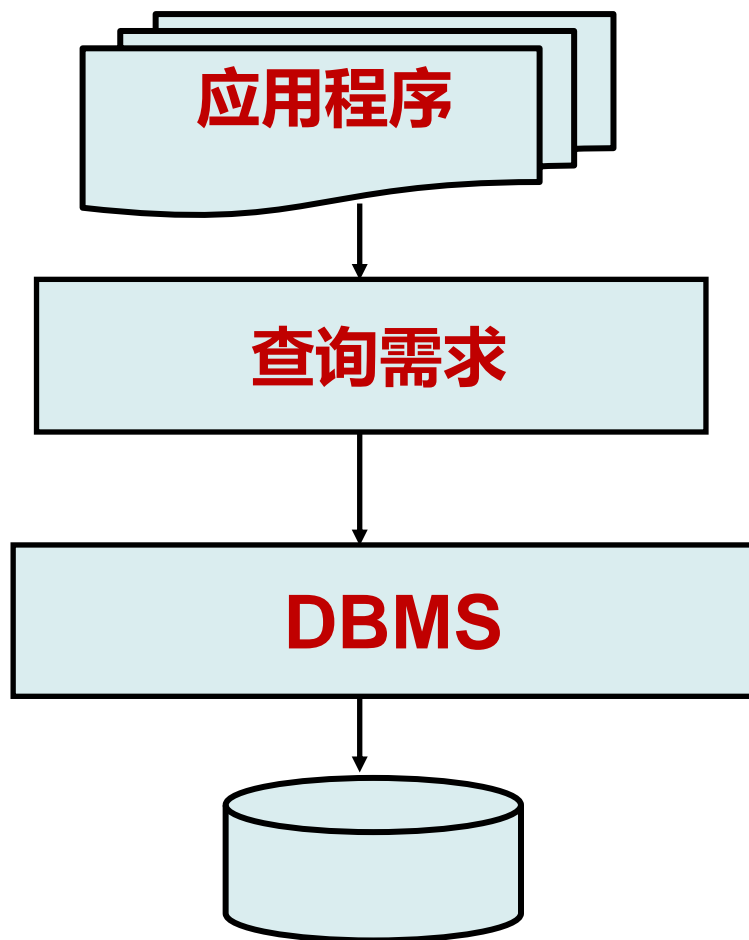
3.6 空值的处理

3.7视图



SQL的引入动机

问题：为什么需要SQL语言？



如何描述查询需求？

查询需求描述**自动化**
查询需求描述**自然化**

- 采用关系代数表达式描述查询需求能够满足**自动化**，但是还不够**自然化**
- SQL是一种比关系代数表达式**更加**自然化的查询需求描述语言
- 系统**自动解决**从SQL到代数表达式的转换

SQL的产生与发展

- 1972年，IBM公司开始研制实验型关系数据库管理系统SYSTEM R，其配备的查询语言称为SQUARE (Specifying Queries As RelationalExpression)，语言中使用了较多的数学符号。
- 1974年，Boyce和Chamberlin把SQUARE修改为SEQUEL (Structured English Query Language)语言。后来SEQUEL简称为SQL (Structured Query Language)，即“结构式查询语言”，现在SQL已经成为一个标准。



SQL标准的进展过程

标准	发布日期
■ SQL/86	1986年
■ SQL/89	1989年
■ SQL/92	1992年
■ SQL99	1999年
■ SQL2003	2003年
■ SQL2008	2008年
■ SQL2011	2010年
■	



SQL的组成及特点

SQL语言从功能上可以分为四部分：数据查询（Data Query）、数据操纵（Data Manipulation）、数据定义（Data Definition）和数据控制（Data Control）。

- 数据定义语言，即SQL DDL，用于定义SQL模式、基本表、视图、索引等结构。
- 数据操纵语言，即SQL DML，数据操纵分成数据查询和数据更新两类。其中数据更新又分成插入、删除和修改三种操作。
- 数据控制语言，即SQL DCL，对基本表和视图的授权、完整性规则的描述、事务控制等内容。



SQL的特点

1.综合统一

- 可以独立完成数据库生命周期中的全部活动
 - 建立数据库，定义关系模式，插入数据
 - 对数据库中的数据进行查询和更新
 - 数据库重构和维护
 - 数据库安全性、完整性控制等
- 用户数据库投入运行后，可根据需要随时逐步修改模式，不影响数据的运行。
- 数据操作符统一



SQL的特点

2.高度非过程化

- 非关系数据模型的数据操纵语言“面向过程”，必须制定存取路径
- SQL只要提出“做什么”，无须了解存取路径。
- 存取路径的选择以及SQL的操作过程由系统自动完成。



SQL的特点

3.面向集合的操作方式

- 非关系数据模型采用面向记录的操作方式，操作对象是一条记录
- SQL采用集合操作方式
 - 操作对象、查找结果可以是元组的集合
 - 一次插入、删除、更新操作的对象可以是元组的集合



SQL的特点

4.以同一种语法结构提供多种使用方式

- SQL是独立的语言

能够独立地用于联机交互的使用方式

- SQL又是嵌入式语言

SQL能够嵌入到高级语言（例如C，C++，Java）程序中，供程序员设计程序时使用



程序实例

```
int main(void)                                /*C语言主程序开始*/
{
    int    count = 0;
    char  yn;                                  /*变量yn代表yes或no*/
    printf("Please choose the department name(CS/MA/IS): ");
    scanf("%s", deptname);                    /*为主变量deptname赋值*/
    EXEC SQL CONNECT TO TEST@localhost:54321 USER
        "SYSTEM" /"MANAGER";                /*连接数据库TEST*/
    EXEC SQL DECLARE SX CURSOR FOR /*定义游标*/
        SELECT Sno, Sname, Ssex, Sage /*SX对应语句的执行结果
*/
        FROM Student
        WHERE SDept = :deptname;
    EXEC SQL OPEN SX;                        /*打开游标SX便指向查询结果的第一行*/
```



SQL的特点

5.语言简洁 易学易用

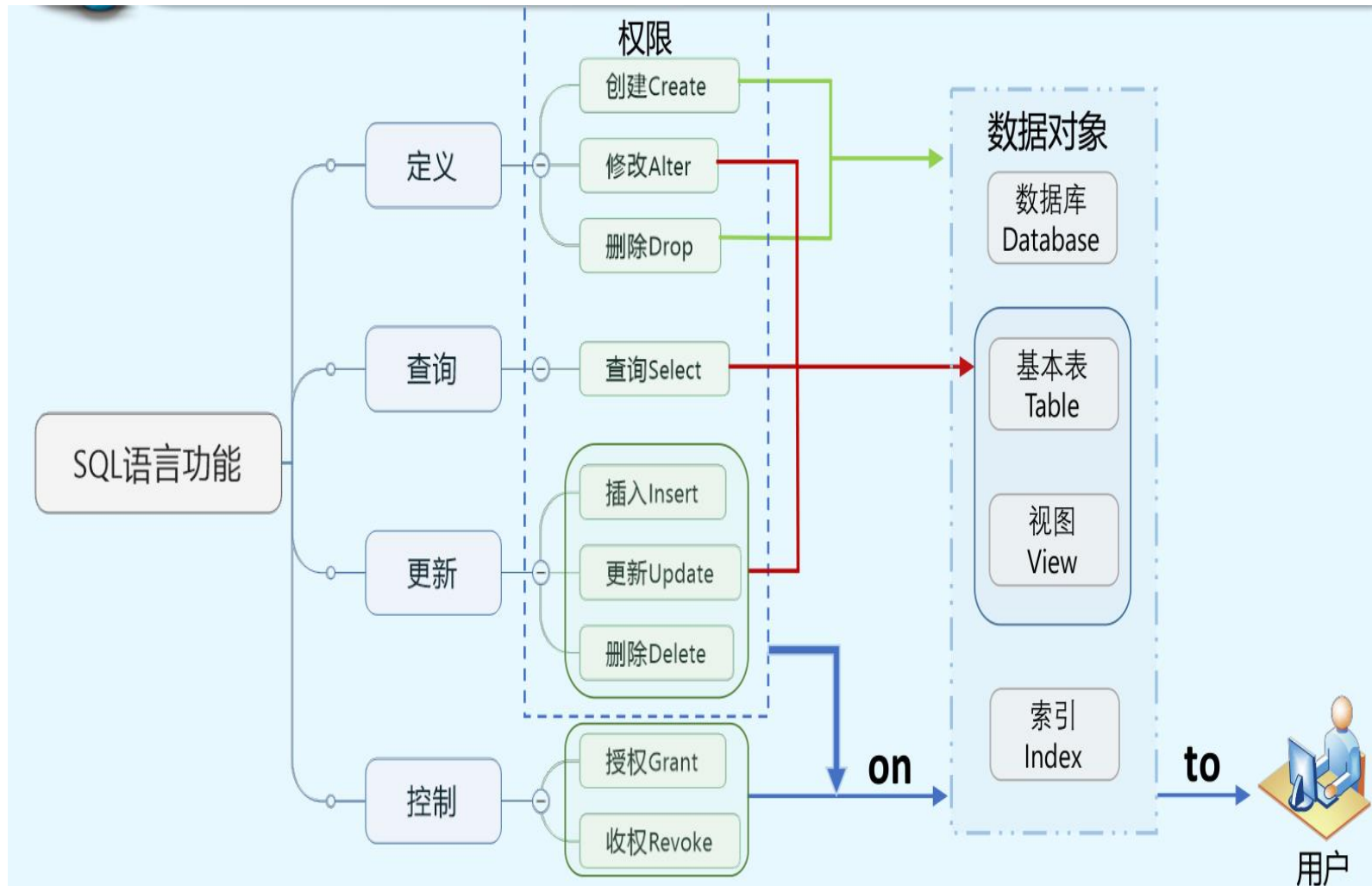
- SQL功能极强，完成核心功能只用了9个动词。

表 3.1 SQL 语言的动词

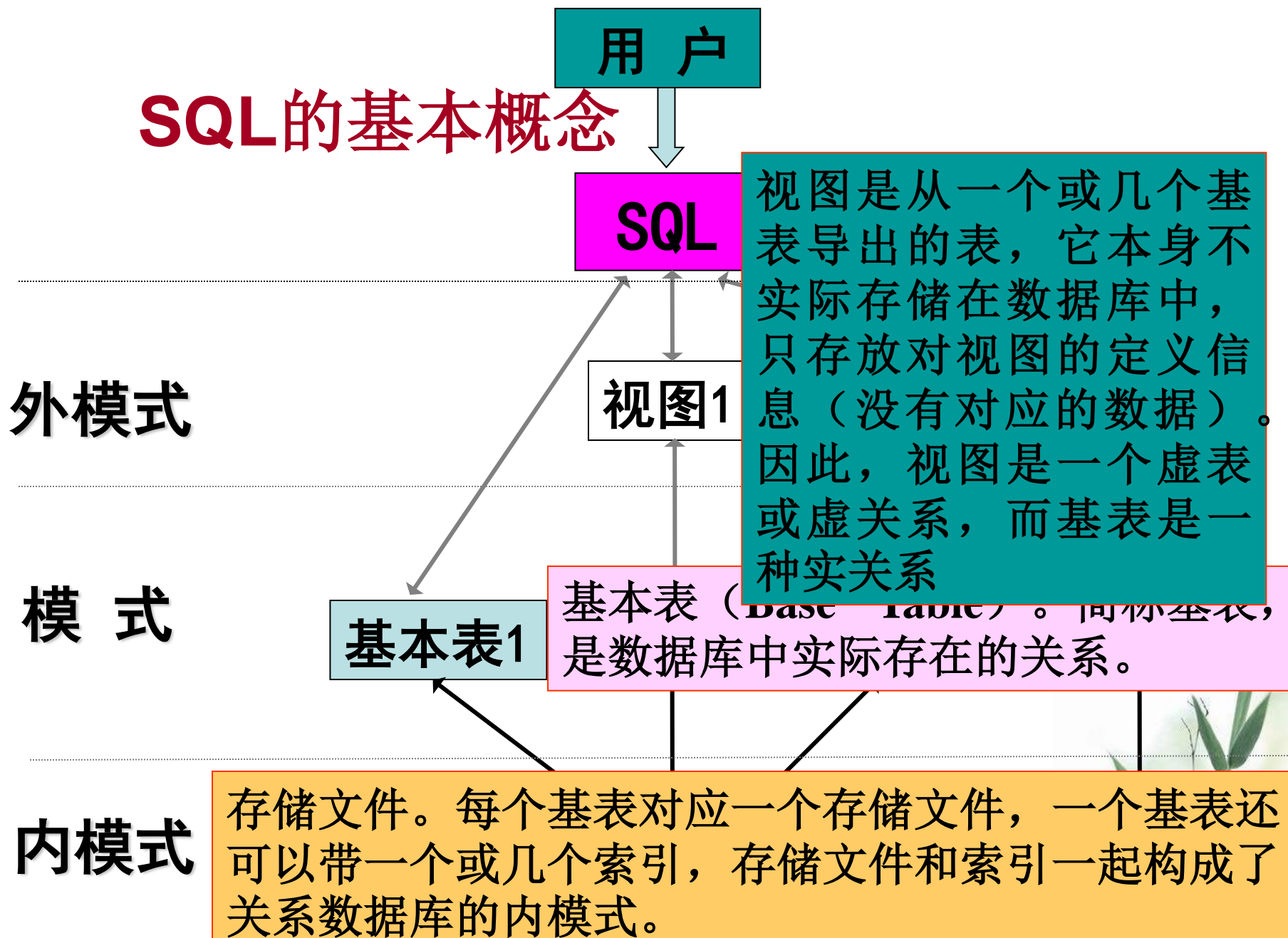
SQL 功 能	动 词
数 据 查 询	SELECT
数 据 定 义	CREATE, DROP, ALTER
数 据 操 纵	INSERT, UPDATE DELETE
数 据 控 制	GRANT, REVOKE



SQL的特点



SQL的基本概念



SQL的基本概念（续）

- 基本表
 - 本身独立存在的表
 - SQL中一个关系就对应一个基本表
 - 一个(或多个)基本表对应一个存储文件
 - 一个表可以带若干索引
- 存储文件
 - 逻辑结构组成了关系数据库的内模式
 - 物理结构是任意的，对用户透明
- 视图
 - 从一个或几个基本表导出的表
 - 数据库中只存放视图的定义而不存放视图对应的数据
 - 视图是一个虚表
 - 用户可以在视图上再定义视图



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



3.2 学生-课程 数据库

- 学生-课程模式 S-T :

学生表: Student(Sno, Sname, Ssex, Sage, Sdept)

课程表: Course(Cno, Cname, Cpno, Ccredit)

学生选课表: SC(Sno, Cno, Grade)



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

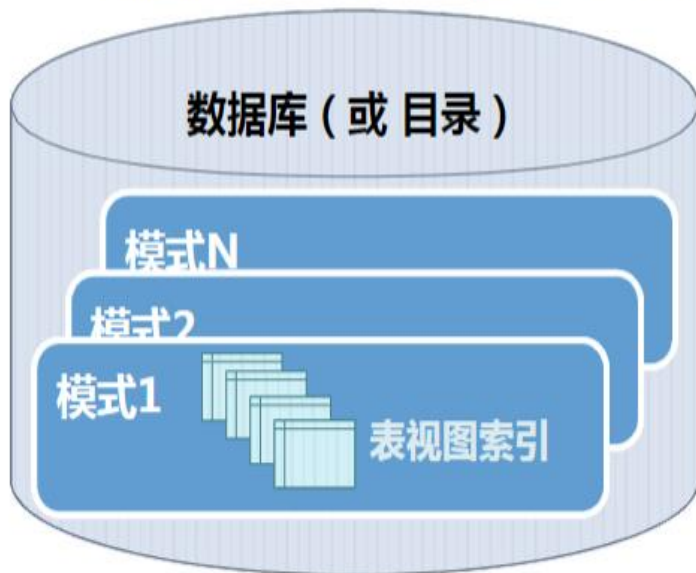
3.7视图



数据库对象命名机制

• 关系数据库系统为抽象对象提供了什么样的命名空间？

类似于文件系统层次空间，现代DBMS提供了一个层次化的数据库对象命名机制，包括：数据库（或目录）、模式、表。



- 一个RDBMS的实例中，可建立多个数据库；
- 一个数据库中可建立多个模式；
- 一个模式下可建立多个表、视图和索引等数据库对象。

3.3 数据定义

SQL的数据定义功能: 模式定义、表定义、视图和索引的定义

表 3.3 SQL 的数据定义语句

操 作 对 象	操 作 方 式		
	创 建	删 除	修 改
模式	CREATE SCHEMA	DROP SCHEMA	
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视 图	CREATE VIEW	DROP VIEW	
索 引	CREATE INDEX	DROP INDEX	ALTER INDEX



3.3 数据定义

3.3.1 模式的定义与删除(略)

3.3.2 基本表的定义、删除与修改

3.3.3 索引的建立与删除



3.3.2 基本表的定义、删除与修改

- 定义基本表

SQLDDL不仅允许定义一组关系，也要说明每个关系的信息

- 每个关系的模式
- 每个属性的值域
- 完整性约束
- 每个关系的安全性和权限
- 每个关系需要的索引集合



基本表的定义

- 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>]) ;

- 语义

定义一张名为<表名>的表，其属性及属性类型由（）说明。

- 完整性

完整性是说明所定义表必须满足的约束条件，分为列级完整性和表级完整性。

若完整性约束条件只涉及到该表的一个属性列，则称列级完整性

若完整性约束条件只涉及到该表的多个属性列，则称表级完整性

表级完整性必须定义在表级上

列级完整性既可以定义在列级也可以定义在表级



表定义示例

[例] 建立“学生”表Student，学号是主码，姓名取值唯一。

```
CREATE TABLE Student
```

```
( Sno CHAR(9) PRIMARY KEY, /* 列级完整性约束条件*/
```

```
  Sname CHAR(20) UNIQUE, /* Sname取唯一值*/
```

```
  Ssex CHAR(2),
```

```
  Sage SMALLINT check(Sage<120 AND Sage>15),
```

```
  Sdept CHAR(20) );
```

主码

候选码

完整性定义

[例] 建立一个“课程”表Course

```
CREATE TABLE Course
```

```
( Cno CHAR(4) PRIMARY KEY,
```

```
  Cname CHAR(40),
```

```
  Cpno CHAR(4),
```

```
  Ccredit SMALLINT,
```

```
  FOREIGN KEY (Cpno) REFERENCES Course(Cno)
```

```
);
```

先修课

Cpno是外码
被参照表是Course
被参照列是Cno

表定义示例

[例] 建立一个“学生选课”表SC

```
CREATE TABLE SC
```

```
( Sno CHAR(9),
```

```
  Cno CHAR(4),
```

```
  Grade SMALLINT,
```

```
  PRIMARY KEY (Sno, Cno),
```

```
    /* 主码由两个属性构成，必须作为表级完整性进行定义*/
```

```
  FOREIGN KEY (Sno) REFERENCES Student(Sno),
```

```
    /* 表级完整性约束条件，Sno是外码，被参照表是Student */
```

```
  FOREIGN KEY (Cno) REFERENCES Course(Cno),
```

```
    /* 表级完整性约束条件，Cno是外码，被参照表是Course*/
```

```
  check(Grage<=100 and Grade>=0)
```

```
);
```



二、数据类型

- SQL中域的概念用数据类型来实现
- 定义表的属性时 需要指明其数据类型及长度
- 选用哪种数据类型
 - 取值范围
 - 要做哪些运算



数据类型

- SQL中表属性的域用数据类型来实现

数据类型	含义表定义示例
CHAR (n)	长度为n的定长字符串
VARCHAR (n)	最大长度为n的变长字符串
INT	长整数（也可以写作INTEGER）
SMALLINT	短整数
NUMERIC (p, d)	定点数，由p位数字（不包括符号、小数点）组成，小数后面有d位数字
REAL	取决于机器精度的浮点数
Double Precision	取决于机器精度的双精度浮点数
FLOAT (n)	浮点数，精度至少为n位数字
DATE	日期，包含年、月、日，格式为YYYY-MM-DD
TIME	时间，包含一日的时、分、秒，格式为HH:MM:SS

修改基本表

SQL语言用ALTER TABLE语句修改基本结构表，其一般格式为：

ALTER TABLE <表名>

[ADD <新列名> <数据类型> **[完整性约束]**

[ADD <表级完整性约束>

[DROP [COLUMN] <列名> **[CASCADE | RESTRICT]**

[DROP CONSTRAINT <完整性约束名> **[CASCADE | RESTRICT]**

[ALTER COLUMN <列名> <数据类型> **];**



修改基本表（续）

[例]向Student表增加“入学时间”列，其数据类型为日期型。

```
ALTER TABLE Student ADD S_entrance DATETIME;
```

- 不论基本表中原来是否已有数据，新增加的列一律为空值。

Sno	Sname	Ssex	Sage	Sdept	S_entrance
1	李勇	男	18	CS	NULL
2	刘晨	男	17	CS	NULL
3	李敏	女	18	MA	NULL

[例]将年龄的数据类型由字符型（假设原来的数据类型是字符型）改为整数。

```
ALTER TABLE Student ALTER COLUMN Sage INT;
```

[例]增加课程名称必须取唯一值的约束条件。

```
ALTER TABLE Course ADD UNIQUE(Cname);
```



删除基本表

DROP TABLE <表名> [RESTRICT| CASCADE] ;

■ **RESTRICT**: 删除表是有限制的。

- 欲删除的基本表不能被其他表的约束所引用
- 如果存在依赖该表的对象，则此表不能被删除

■ **CASCADE**: 删除该表没有限制。

- 在删除基本表的同时，相关的依赖对象一起删除

Sc(sno,cno,grade)

Student(sno,sname,sage.....)



删除基本表(续)

[例] 删除Student表

```
DROP TABLE Student CASCADE ;
```

- 基本表定义被删除，数据被删除
- 表上建立的索引、视图、触发器等一般也将被删除



删除基本表（续）

[例] 若表上建有视图，选择RESTRICT时表不能删除

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

```
DROP TABLE Student RESTRICT;  
--ERROR: cannot drop table Student because other  
  objects depend on it
```



删除基本表（续）

[例]如果选择**CASCADE**时可以删除表，视图也自动被删除

```
DROP TABLE Student CASCADE;
```

```
--NOTICE: drop cascades to view IS_Student
```

```
SELECT * FROM IS_Student;
```

```
--ERROR: relation " IS_Student " does not exist
```

注意:不同**DBMS**产品在遵循**SQL**标准的基础上具体实现细节和处理策略上会与标准有差别.



3.3 数据定义

- 3.3.1 模式的定义与删除
- 3.3.2 基本表的定义、删除与修改
- 3.3.3 索引的建立与删除



索引的建立与删除

- 什么是索引?

索引是DBMS提供的一个对象，与关系表相关。系统通过维护索引，可以加快对表的查找。索引属于内模式范畴。

- 谁可以建立索引?

- DBA 或 表的属主（即建立表的人）
- DBMS一般会自动建立以下列上的索引

PRIMARY KEY

UNIQUE

- 如何使用索引?

SQL用户并不直接使用索引。当用户发出SQL请求时，DBMS自动选择是否使用索引以及使用哪些索引



索引

- 常用的索引技术

B+树索引 索引属性值组成B+树，具有动态平衡的优点

HASH索引 索引属性值分桶，具有查找速度快的特点

顺序索引 索引属性值排序，可二分查找

- 几点说明

- 用户定义表时，DBMS自动建立索引，采用B+树，还是HASH索引由具体的DBMS来决定
- 索引是关系数据库的内部实现技术，属于内模式的范畴
- 从功能上考虑，DBMS为用户提供了CREATE INDEX语句定义索引，但只能定义唯一索引、非唯一索引或聚簇索引



建立索引使用CREATE INDEX语句，其一般格式为：

```
CREATE [UNIQUE][CLUSTERED] INDEX <索引名>  
ON <表名>(<列名>[<次序>][, <列名>[<次序>]]...);
```

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。每个<列名>后面还可以用<次序>指定索引值的排列次序，可选ASC（升序，默认）或DESC（降序）。

UNIQUE表明此索引的每一个索引值只对应唯一的数据记录。

CLUSTERED表示要建立的索引是聚簇索引。



索引的类型

(1) 聚簇索引(或称为聚集索引)

- 聚簇索引在数据表中按照物理顺序存储数据。因为在表中只有一个物理顺序，所以在每个表中只能有一个聚簇索引。
- 默认情况下，SQL Server为PRIMARY KEY约束所建立的索引为聚簇索引。在语句 **CREATE INDEX** 中使用 **CLUSTERED**选项建立簇索引。



例

```
CREATE CLUSTERED INDEX STUSNAME ON  
Student(SNAME);
```

将会在Student表的SNAME（姓名）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。



(2) 唯一值索引

- 唯一索引可以确保所有数据行中任意两行的被索引列不包括NULL在内的重复值。
- 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



例：为学生-课程数据库中的Student， Course， SC三个表建立索引。

```
CREATE UNIQUE INDEX Stusno ON Student(Sno);
```

```
CREATE UNIQUE INDEX Coucho ON Course(Cno);
```

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

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



- 修改索引

alter index <旧索引名> rename to <新索引名>;

例： 将SC表的SCNO索引名改为SCSNO

alter index SCNO rename to SCSNO ;



3、删除索引

一般格式为

DROP INDEX <索引名>;

例 删除Student表的STUSNAME索引。

DROP INDEX STUSNAME;

删除索引时，系统会同时从数据字典中删去有关该索引的描述。

