



UNIT 5 SQL概述与SQL定义语言

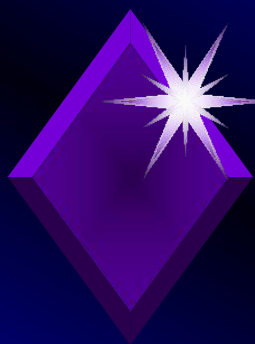


本讲主要目标



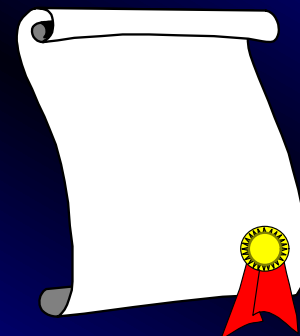
学完本讲后，你应该能够了解：

- 1、SQL是一个非过程化语言，使用者只需要说明“做什么”而不需要说明“怎么做”；
- 2、SQL是一个集定义、操作、查询和控制为一体的语言；
- 3、SQL的定义语言支持建立数据库三级模式结构；
- 4、如何使用Create语句创建模式、域、表和索引；
- 5、如何使用DROP语句删除模式、域、表和索引。



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引





DreamHome 租赁数据库

DreamHome 案例的部分关系模式：

- **Branch** (branchNo, street, city, postcode)
- **Staff** (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
- **PropertyForRent** (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
- **Client** (clientNo, fName, lName, telNo, prefType, maxRent)
- **PrivateOwner** (ownerNo, fName, lName, address, telNo)
- **Viewing** (clientNo, propertyNo, viewDate, comment)
- **Registration** (clientNo, branchNo, staffNo, dateJoined)



一个学生-课程数据库

S

学号 S#	姓名 SN	性别 SE	年龄 SA	所在系 SD
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

SC

学号 S#	课程号 C#	成绩 G
95001	C1	92
95001	C2	85
95001	C3	88
95002	C2	90
95002	C3	80

C

课程号 C#	课程名 CN	先行课 CP#	学分 CC
C1	数据库	C5	4
C2	数学		2
C3	信息系统	C1	4
C4	操作系统	C6	3
C5	数据结构	C7	4
C6	数据处理		2
C7	PASCAL语言	C6	4



一、SQL的特点、历史与功能

1、SQL是一个理想的数据库语言

- 存在一个SQL的ISO标准
- SQL集定义、操作、查询和控制为一体
- 易于学习
 - 是一个非过程性语言，用户只需要描述所需的信息(what)，不需给出获取该信息的具体过程(how)
 - 自由格式语言
 - SQL命令由标准英语单词所组成



一、SQL的特点、历史与功能

2、SQL的特点（见教材P66）

- 高度的非过程化语言
- 综合统一。数据描述、操纵、控制等功能一体化
- 面向集合的操作方式
- 两种使用方式，统一的语法结构
- 简洁易学

SQL功能	动词
数据查询 DQL	SELECT
数据定义 DDL	CREATE, DROP, ALTER
数据操纵 DML	INSERT, UPDATE, DELETE
数据控制 DCL	GRANT, REVOKE



一、SQL的特点、历史与功能

3、SQL的历史（见教材P65）

- 关系模型起源于1970年IBM San Jose研究室E. F. Codd发表的一篇论文
- 1974年，IBM San Jose研究室的D. Chamberlin定义了一种称为SQL的结构化英语查询语言，或称为SEQUEL，76年其修改版本SEQUEL/2出现
- SQL读为 ‘see-quel’，尽管官方的读法 ‘S-Q-L’
- SQL-87：ANSI和ISO发表了最初标准
- SQL-89：完整性增强特性
- SQL-92：做了大的修改，称为SQL2 or SQL/92
- SQL-99：支持面向对象数据管理的特性，通常称为SQL3
- SQL:2003:
- ...

SQL标准
是不确定
的

对标准
的遵守
程度



一、SQL的特点、历史与功能

3、SQL的历史（见教材P65）

- 关系模型起源于1970年IBM San Jose研究室E. F. Codd发表的一篇论文
- 1974年，IBM San Jose研究室的D. Chamberlin定义了一种称为SQL的结构化英语查询语言，或称为SEQUEL，76年其修改版本SEQUEL/2出现
- SQL读为 ‘see-quel’，尽管官方的读法 ‘S-Q-L’
- SQL-87：ANSI和ISO发表了最初标准
- SQL-89：完整性增强特性
- SQL-92：做了大的修改，称为SQL2 or SQL/92
- SQL-99：支持面向对象数据管理的特性，通常称为SQL3
- SQL:2003:
- ...

没有任何一个
DBMS支持标准
中的所有概念
和特性

对标准的
遵守
程度

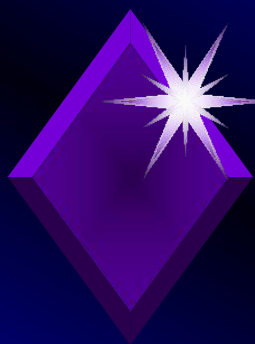


一、SQL的特点、历史与功能

4、SQL的功能：

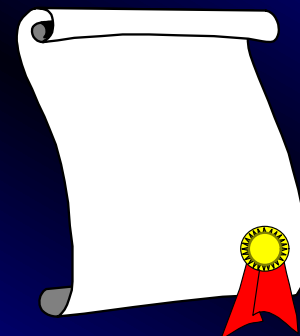
SQL语言是一个综合的、通用的、功能极强的关系数据库语言，它包括：

- DDL (Data Define Language)
数据定义语言：定义数据库结构
- DML (Data Manipulation Language)
数据操作语言：更新数据（如增、删、改）
- DQL (Data Query Language)
数据查询语言：查询数据
- DCL (Data Control Language)
数据控制语言：控制访问



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引





二、SQL的标准格式约定

1、SQL语句包括保留字和用户自定义字

➤ 保留字

- SQL语言的固定部分，有固定的含义。
- 必须准确拼写，不能分行拼写

➤ 用户自定义字（标识符）

- 用户自己定义
- 用于命名表、列、视图、索引等数据库对象



二、SQL的标准格式约定

2、SQL语言格式约定

- SQL是一种自由格式语言：它对每一行的字符数不作限定，也不要求一个SQL语句一定分成几行来写
- SQL的编译器不区分大小写；（字符常量除外）
- 以分号结束语句。



二、SQL的标准格式约定

3、增强SQL语句的可读性

- 用分行、对齐和缩进方法。
 - 分行：语句的每一个子句在新的一行开始书写
 - 对齐：每一子句与其它子句的开始处在同一列上
 - 缩进：如果子句由几个部分组成，则它们应当出现在不同的行，并在子句开始处使用缩进方式表示这种关系
- SQL的保留字使用大写。

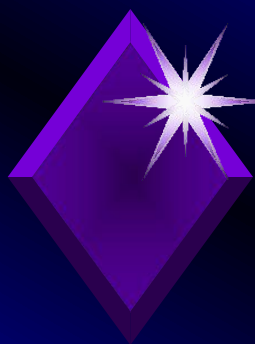


二、SQL的标准格式约定

4、用扩展的巴克斯范式(ABNF)定义SQL语句

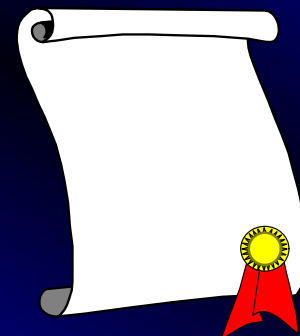
- **大写字母**用于表示保留字
- **小写字母**用于表示用户自定义字
- **竖线 (|)**用于表示从选项中进行选择
- **大括号 ({ })**用于表示可重复0到无数次的项
- **中括号 ([])**用于表示可选择项
- **省略号 (...)**用于表示该元素可重复0到多次

注：BNF中用 **尖括号 (< >)** 用于表示必选项
ABNF不再需要。



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引

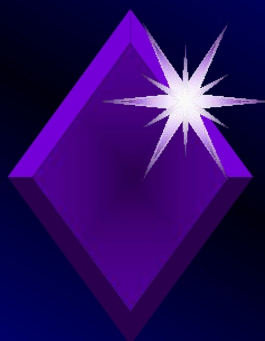




三、SQL的数据定义语言

1、SQL支持的数据库对象

- 关系和其它数据库对象是存在于**环境**中的；
- 每个环境由一个或多个**目录** (catalog) 组成，每个目录包含一个**模式** (schema) 集合；
- **模式** (schema)
 - 模式是数据库对象的命名集合
 - 模式中的对象可以是**表**、**视图**、**域**、**索引**等
 - 模式中的所有对象有相同的所有者



三、SQL的数据定义语言

2、主要的SQL DDL语句（见教材P68）

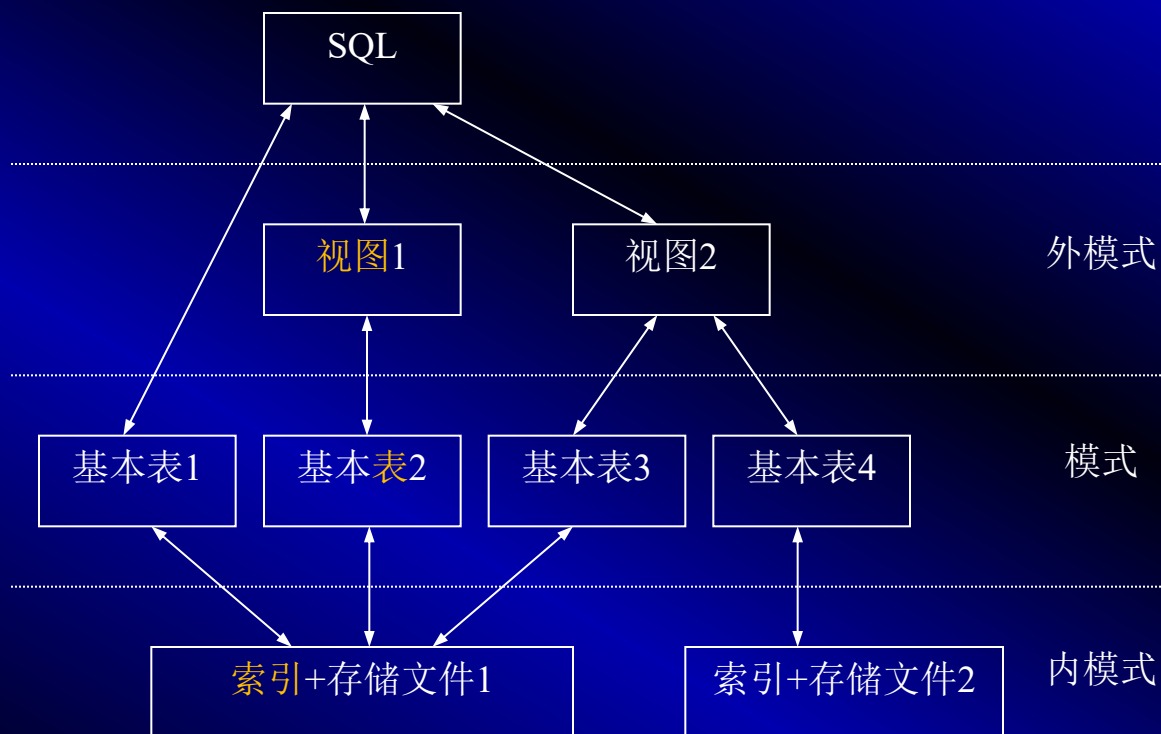
创建（CREATE）、修改（MODIFY）和删除（DROP）模式、域、基本表、视图、索引等数据库对象

Functions Objects	Create	Modify	Drop
SCHEMA	CREATE SCHEMA		DROP SCHEMA
DOMAIN	CREATE DOMAIN	ALTER DOMAIN	DROP DOMAIN
TABLE	CREATE TABLE	ALTER TABLE	DROP TABLE
INDEX	CREATE INDEX		DROP INDEX
VIEW	CREATE VIEW		DROP VIEW



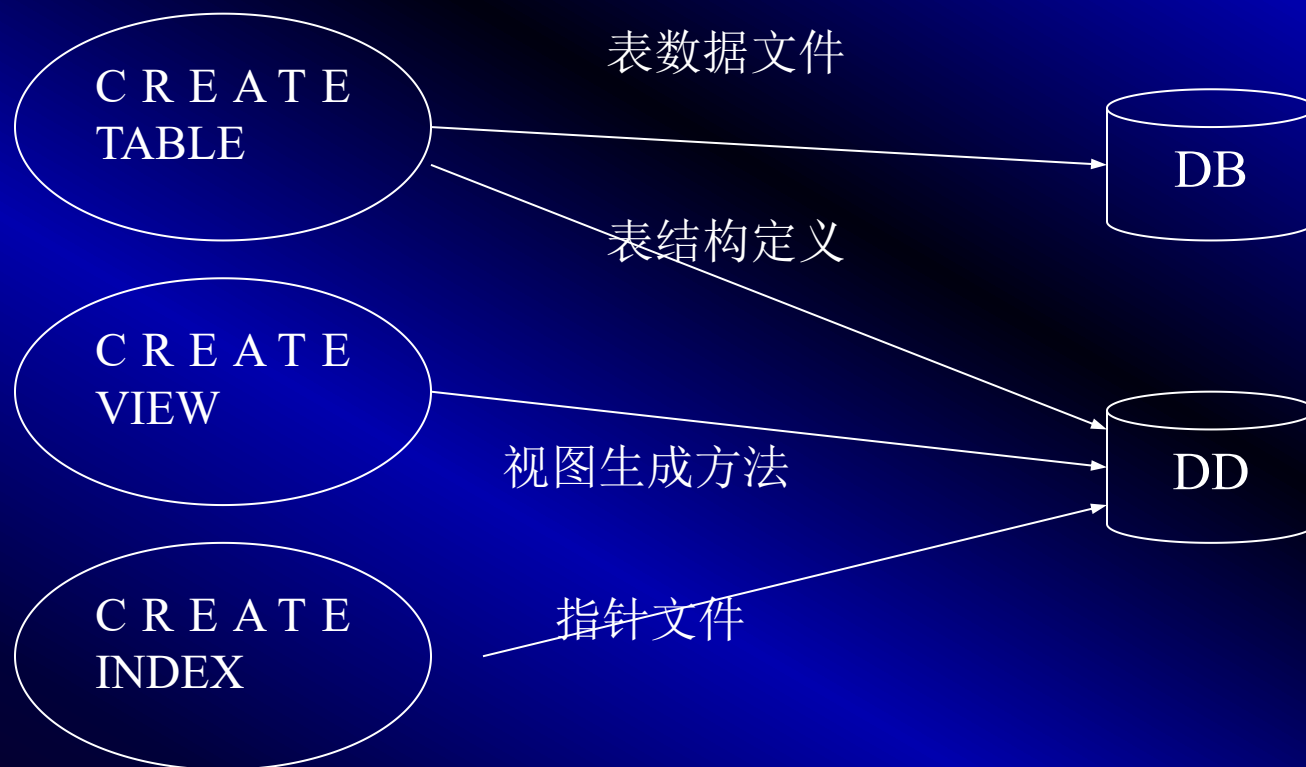
三、SQL的数据定义语言

3、DDL对数据库三级模式的支持（见教材P67）



三、SQL的数据定义语言

4、SQL的数据定义语句对DD和DB的影响





三、SQL的数据定义语言

5、创建和删除数据库（模式）

- 不同数据库产品的数据库创建过程是截然不同的. 多用户系统中, 创建数据库的权限通常为DBA保留. 单用户系统中, 通常系统安装和配置时会创建一个默认数据库, 其他数据库可以由用户需要时创建
- ISO标准并不指定如何创建数据库, 每种具体实现都有各自不同的方式
- 一个数据库中有一个或多个模式; **模式 (SCHEMA)** 是数据库对象的命名集合, 模式中所有对象有相同的所有者并且共享一些默认值



三、SQL的数据定义语言

- 创建数据库（模式）

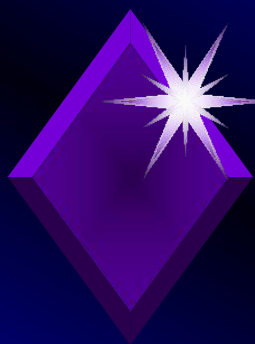
- **语法：**CREATE SCHEMA database-name
AUTHORIZATION user-identifier
[create-table-statement|create-view-statement|.....];

例：CREATE SCHEMA library_db AUTHORIZATION
Guoqing;

- 删除数据库（模式）

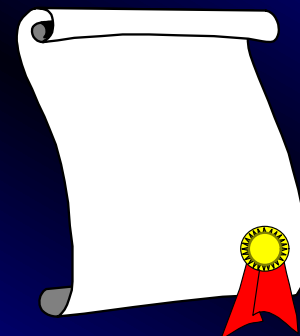
- **语法：**DROP SCHEMA database-name [RESTRICT | CASCADE];
- Restrict - 如果模式内没有对象，则删除该模式
- Cascade - 删除模式中所有对象的内容和定义

例：DROP SCHEMA library RESTRICT;



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型**
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引





四、SQL的数据类型

标识符（简单标识符）

- ◆ 标识符用于标识数据库中的对象，如表名、视图名、列名等
- ◆ 标识符所用的字符必须在字符集中存在
- ◆ ISO提供默认的字符集，包括大写字符A到Z、小写字符a到z，数字0到9和下划线（ ）字符
- ◆ 标识符有下列限制：
 - ◆ 保留字不能用作标识符
 - ◆ 不能长于128个字符
 - ◆ 必须以字母开头
 - ◆ 不能有空格



四、SQL的数据类型

ISO SQL数据类型（见教材P70）

Table 6.1 , data types.

Data type	Declarations			
boolean	BOOLEAN			
character	CHAR	VARCHAR		
bit	BIT	BIT VARYING		
exact numeric	NUMERIC	DECIMAL	INTEGER	SMALLINT
approximate numeric	FLOAT	REAL	DOUBLE PRECISION	
datetime	DATE	TIME	TIMESTAMP	
interval	INTERVAL			
large objects	CHARACTER LARGE OBJECT		BINARY LARGE OBJECT	



四、SQL的数据类型

- ① **布尔型**：包括TRUE, UNKNOWN, and FALSE值
- ② **字符型**：定义字符串，定长CHAR(n)，变长VARCHAR(n)
- ③ **位类型**：定义二进制位串，BIT(n), BIT VARYING(n)
- ④ **定点数**：

● **NUMERIC** [precision [, scale]]

precision (精度) 表示数全部的位数, scale (标度) 表示小数部分的位数

● **DECIMAL** [precision [, scale]]

● **INTEGER** 较大的正负整数

● **SMALLINT** 较小的正负整数



四、SQL的数据类型

⑤ 浮点数：非精确数字

- FLOAT [precision]
- REAL
- DOUBLE PRECISION

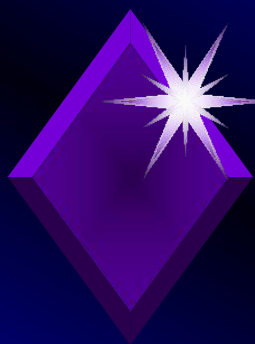
⑥ 日期时间型：一定精度的时间点，分为YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, and TIMEZONE_MINUTE

- DATE 存储日期 YEAR, MONTH, DAY
- TIME [timePrecision] [WITH TIME ZONE] 存储时间 HOUR, MINUTE, SECOND, 默认timePrecision为0（秒）
- TIMESTAMP [timePrecision] [WITH TIME ZONE] 存储日期时间，默认timePrecision为6（微秒）



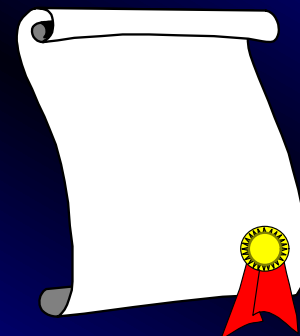
四、SQL的数据类型

- ⑦ **区间型**：表示一段时间。每个区间类型包括了字段的一个连续子集：YEAR, MONTH, DAY, HOUR, MINUTE, SECOND
- INTERVAL {{startField TO endField} singleDatetimeField}
- startField = YEAR | MONTH | DAY | HOUR | MINUTE
[(intervalLeadingFieldPrecision)]
- endField = YEAR | MONTH | DAY | HOUR | MINUTE | SECOND
[(fractionalSecondsPrecision)]
- singleDatetimeField = startField | SECOND
[(intervalLeadingFieldPrecision [,
fractionalSecondsPrecision])]
- ⑧ **大对象型**：SQL3标准定义了字符大对象CHARACTER LARGE OBJECT和二进制大对象BINARY LARGE OBJECT



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域**
- 六. 创建、删除与修改表
- 七. 创建与删除索引





五、创建和删除域

1、创建域

域用于定义列的合法值的集合

◆ 语法

```
CREATE DOMAIN DomainName [AS] dataType  
[DEFAULT defaultOption]  
[CHECK (searchCondition)]
```

◆ **功能：** 给域一个域名DomainName，一个数据类型，一个可选的默认值，一个可选的CHECK约束

◆ **例：**

```
CREATE DOMAIN SexType AS CHAR(2)  
DEFAULT 'M'  
CHECK (VALUE IN ('M', 'F'));  
sex SexType NOT NULL;
```



五、创建和删除域

2、删除域

◆ 语法

DROP DOMAIN DomainName [**RESTRICT** | **CASCADE**];

◆ 功能：

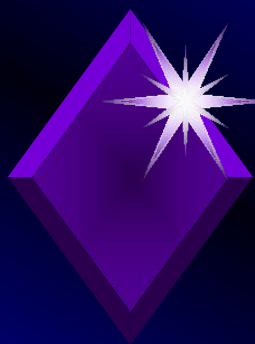
◆ 删除域

◆ 删除域时，使用**RESTRICT** | **CASCADE**进行约束

◆ 如果指定**RESTRICT**，且域用于现有的表中，则删除失败

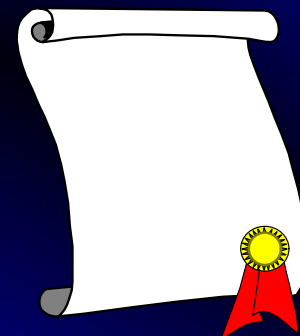
◆ 如果指定**CASCADE**，基于域的任一表中的列都会自动地改变为基本域类型，该域的**CHECK**约束或默认值都会被列的约束或默认值代替

◆ 例： **DROP DOMAIN** SexType;



本讲主要内容

- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引





六、创建、删除与修改表

(见教材P68-71)

1、创建表时，需要搞清楚的问题

- 此表包括那些列？
- 表名是什么？
- 各列名是什么？
- 各列的长度和数据类型是什么？
- 列是否允许取空值？
- 列是否取唯一值？
- 哪些列组成表的主键？
- 外键及被参照的关系是什么？



六、创建、删除与修改表

2、定义表语句的基本语法

```
CREATE TABLE TableName  
    ({columnName dataType [NOT NULL] [UNIQUE]  
    [DEFAULT defaultOption]  
    [CHECK (searchCondition)] }[,...]  
    [PRIMARY KEY (listOfColumns),]  
    {[UNIQUE (listOfColumns)] [,...]}  
    {[FOREIGN KEY (listOfFKColumns)  
    REFERENCES ParentTableName [(listOfCKColumns)],  
    [ON UPDATE referentialAction]  
    [ON DELETE referentialAction ]] [,...]}  
    {[CHECK (searchCondition)] [,...]});
```



六、创建、删除与修改表

定义表语句的基本语法：

CREATE TABLE 〈表名〉

(

 〈列名〉〈数据类型〉[NOT NULL | NULL] [UNIQUE]

 [, 〈列名〉〈数据类型〉[NOT NULL | NULL] [UNIQUE]]...

 [, PRIMARY KEY (〈列名〉[, 〈列名〉] ...)]

 [, FOREIGN KEY (〈列名〉[, 〈列名〉] ...) REFERENCES
 <被参照表名>]

) ;



六、创建、删除与修改表

3、定义表语句的功能

- 创建一个具有一列或多列的表，每列具有指定的数据类型；
- 同时可以定义该表有关的完整性约束
 - NOT NULL限定列上的值不允许为空；
 - DEFAULT定义列的缺省值；
 - UNIQUE不允许重复；
 - 主键列必须定义为NOT NULL
 - 外键子句可以定义可供采取的行为



六、创建、删除与修改表

4、定义表语句实例一

定义学生-课程数据库中的三个表

(1) 定义学生表S:

```
CREATE TABLE S
( S# CHAR (8) NOT NULL UNIQUE,
  SN CHAR (8) UNIQUE,
  SE CHAR (1) ,
  SA SMALLINT,
  SD CHAR (4) ,
  PRIMARY KEY (S#)
) ;
```



六、创建、删除与修改表

(2) 定义选课表SC:

```
CREATE TABLE SC
( S# CHAR (8) NOT NULL,
  C# CHAR (8) NOT NULL,
  G SMALLINT,
  PRIMARY KEY (S#, C#) ,
  FOREIGN KEY (S#) REFERENCES S (S#) ,
  FOREIGN KEY (C#) REFERENCES C (C#)
) ;
```

(3) 定义课程表C: (自己给出?)



六、创建、删除与修改表

(3) 定义课程表C:

```
CREATE TABLE C
( C# CHAR (8) NOT NULL,
  CN CHAR (30) ,
  CP# CHAR (8) ,
  CC SMALLINT,
  PRIMARY KEY (C#) ,
  FOREIGN KEY (CP#) REFERENCES C (C#)
) ;
```



六、创建、删除与修改表

5、定义表语句实例二

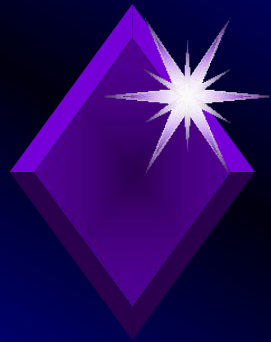
DreamHome 租赁数据库

- ① CREATE DOMAIN OwnerNumber AS VARCHAR(5)
CHECK (VALUE IN (SELECT ownerNo FROM PrivateOwner));
- ② CREATE DOMAIN StaffNumber AS VARCHAR(5)
CHECK (VALUE IN (SELECT staffNo FROM Staff));
- ③ CREATE DOMAIN PNumber AS VARCHAR(5);
- ④ CREATE DOMAIN PRooms AS SMALLINT;
CHECK(VALUE BETWEEN 1 AND 15);
- ⑤ CREATE DOMAIN PRent AS DECIMAL(6,2)
CHECK(VALUE BETWEEN 0 AND 9999.99);.



六、创建、删除与修改表

```
CREATE TABLE PropertyForRent (  
    propertyNo    PNumber        NOT NULL,  
    ....  
    rooms         PRooms          NOT NULL    DEFAULT 4,  
    rent          PRent           NOT NULL    DEFAULT 600,  
    ownerNo       OwnerNumber     NOT NULL,  
    staffNo       StaffNumber     CONSTRAINT StaffNotHandlingTooMuch  
    ....  
    branchNo      BranchNumber    NOT NULL,  
    PRIMARY KEY (propertyNo),  
    FOREIGN KEY (staffNo) REFERENCES Staff  
        ON DELETE SET NULL ON UPDATE CASCADE ....);  
.....
```



六、创建、删除与修改表

6、删除表语句

- **基本语法：**

DROP TABLE 〈表名〉 [**RESTRICT**|**CASCADE**] ;

- **功能：** 既删除DD中表的结构定义，也删除DB中的数据文件
- **RESTRICT**（缺省）：如果有其它对象依赖于该表，则拒绝进行DROP操作
- **CASCADE**：级联删除表及所有依赖于该表的对象
- **实例：** **DROP TABLE** PropertyForRent;



六、创建、删除与修改表

7、修改表语句

➤ 基本语法：

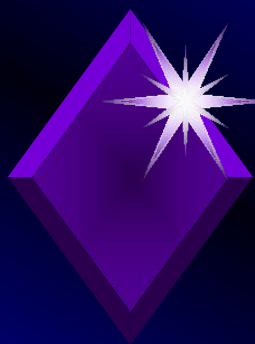
```
ALTER TABLE 〈表名〉  
[ADD 〈新列名〉 〈数据类型〉 {NULL}]  
[DROP 〈完整性约束条件〉 ]  
[MODIFY 〈列名〉 〈数据类型〉 ];
```

➤ 说明：

- ☆ 新增加的列为空值；
- ☆ 修改列定义可能破坏已有数据；
- ☆ 没有提供删除列的子句*

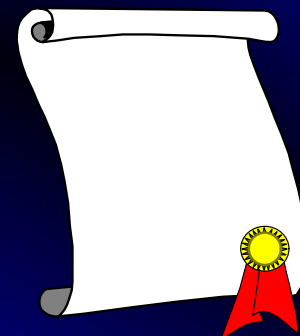
➤ 实例：删除学生姓名必须取唯一值的约束

```
ALTER TABLE S DROP UNIQUE (SN) ;
```



本讲主要内容

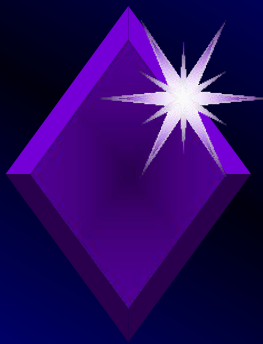
- 一. SQL的特点、历史与功能
- 二. SQL的标准格式约定
- 三. SQL的数据定义语言
- 四. SQL的数据类型
- 五. 创建和删除域
- 六. 创建、删除与修改表
- 七. 创建与删除索引





七、创建与删除索引

- **索引**是一种结构，它提供了基于一个或多个列值快速访问元组的方法。
- 索引的出现极大地**提高了查询的性能**。
- 但是，由于每一次更新基本关系时系统都可能会更新索引，所以将导致**额外的开销**。



七、创建与删除索引

1、创建索引（见教材P71-73）

➤ 基本语法：

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

➤ 说明：

- ❖ 一条语句建立一个索引；
- ❖ <次序>指定索引的排列次序 ---- ASC 或DESC，缺省为ASC；
- ❖ UNIQUE表明每个索引值只对应唯一的数据记录；
- ❖ CLUSTER是聚簇索引，即索引项的顺序与表中记录的物理顺序一致；一个基本表最多只能建立一个聚簇索引



七、创建与删除索引

2、创建索引实例

实例一：

在学生表S中按S#升序建唯一索引

```
CREATE UNIQUE INDEX S_S# ON S (S#) ;
```

实例二：

在SC中按S#升序和C#降序建唯一索引

```
CREATE UNIQUE INDEX SC_S#C#  
ON SC( S# ASC, C# DESC) ;
```



七、创建与删除索引

3、删除索引

➤ **基本语法：**

```
DROP INDEX 〈索引名〉；
```

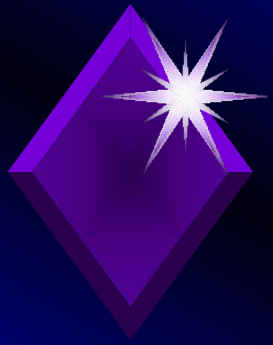
➤ **实例**

```
DROP INDEX S_S#；
```

数据量小的表不需要建立索引，建立会增加额外的索引开销。

数据变更需要维护索引，因此更多的索引意味着更多的维护成本。

更多的索引意味着也需要更多的空间。



Questions?





本讲主要目标



学完本讲后，你应该能够了解：

- 1、SQL是一个非过程化语言，使用者只需要说明“做什么”而不需要说明“怎么做”；
- 2、SQL是一个集定义、操作、查询和控制为一体的语言；
- 3、SQL的定义语言支持建立数据库三级模式结构；
- 4、如何使用Create语句创建模式、域、表和索引；
- 5、如何使用DROP语句删除模式、域、表和索引。

问题讨论

1、什么叫语言的非过程化？非过程化语言的优劣是什么？SQL是一个非过程化语言吗？

2、CREATE TABLE、CREATE VIEW、CREATE INDEX的执行对DB和DD的影响？

3、一般来说，建立索引可以提高查询效率，那么，索引建得越多越好吗？





练习

教材：《数据库系统原理教程》（第2版）

P112

1、

2、

