



# 完整性概述

## 总述

内容：数据库管理系统提供统一的数据保护功能来保证数据的安全可靠和正确有效。学习数据库完整性的基本概念。

- 一、完整性定义
- 二、完整性和安全性的关系
- 三、完整性子系统功能
- 四、完整性约束条件

## 一、完整性定义

数据库的完整性是指数据的正确性和相容性，DBMS必须提供一种功能来保证数据库中数据的完整性。

这种功能亦称为完整性检查，即系统用一定的机制来检查数据库中的数据是否满足规定的条件，这种条件在数据库中称为完整性约束条件，数据的约束条件是语义的体现，这些完整性约束条件将作为模式的一部分存入数据库中。

## 二、完整性和安全性的关系

完整性是为了防止数据库中存在不符合语义的数据，防止错误信息的输入和输出，即所谓垃圾进垃圾出（Garbage In Garbage Out）所造成的无效操作和错误结果。

安全性是保护数据库防止恶意的破坏和非法的存取。

完整性和安全性是密切相关的，特别从系统实现的方法来看，某一种机制常常既可用于安全性保护亦可用于完整性保证。

## 三、完整性子系统功能

数据库中数据发生错误，往往是由非法更新引起的。数据库完整性是通过DBMS的完整性子系统实现的，它有三个功能：

- ( 1 ) 提供定义完整性约束条件的机制；
- ( 2 ) 提供完整性检查的方法；
- ( 3 ) 违约处理。

## 四、完整性约束条件

完整性约束条件是加在数据之上的语义约束条件，完整性约束条件可以进行以下分类：

- ( 1 ) 值的约束和结构的约束
- ( 2 ) 静态约束和动态约束
- ( 3 ) 立即执行约束和延迟执行约束

## 四、完整性约束条件

### ( 1 ) 值的约束和结构的约束

值的约束是指对数据取值类型、范围、精度等的规定。

结构的约束指对数据之间联系的限制。数据库中同一关系的不同属性之间可以有一定的联系，从而应满足一定的约束条件，同时，由于数据库中数据是结构化的，不同的关系之间也可以有联系，因而不同关系的属性之间也可满足一定的约束条件。



## 四、完整性约束条件

### 值的约束

例如：对某个属性和属性组合规定某个值集，如职工年龄必须是大于或等于16、小于或等于65的整数或实数，若为实数，可规定精度为小数点后一位数字。

又如，规定年份是四位整数，月份是一至十二的整数。规定零件颜色是红、绿、黄、黑、白五种。规定某属性值的类型和格式。规定职工号第一个字符必须是字母，后面是四位数字。

## 四、完整性约束条件

### 值的约束

规定某属性的值的集合必须满足某种统计条件。

如任何职工的工资不得超过此部门平均工资的三倍，任何职工的奖金不得超过此部门平均工资的30%。

## 四、完整性约束条件

数据之间联系的约束

例如，任何一个码值唯一地确定关系的一个元组。因此候选码的值在关系中必须是唯一的。主码值必须非空（关系模型的实体完整性）。

## 四、完整性约束条件

数据之间联系的约束

一个关系中某属性的值集是同一关系或另一关系中某属性值集的子集。

如在课程关系C中，前修课pcno必须是课程号cno的子集。学生选课关系SC中，课程号cno必须是课程关系C中cno的子集。这二个例子是关系模型的参照完整性。即一个关系的外码的值集一定是相应的另一个关系上主码属性值集的子集。

## 四、完整性约束条件

### (2) 静态约束和动态约束

所谓静态约束是指对数据库每一确定状态的数据所应满足的约束条件。以上我们所讲的约束都属静态约束。

动态约束是指数据库从一种状态转变为另一种状态时新、旧值之间所应满足的约束条件。例如，当更新职工工资时要求新工资值不低于旧工资值。

## 四、完整性约束条件

### (3) 立即执行约束和延迟执行约束

立即执行约束是指在执行用户事务时，对事务中某一更新语句执行完后马上此数据所应满足的约束条件进行完整性检查。

延迟执行约束是指在整个事务执行结束后方对此约束条件进行完整性检查，结果正确方能提交。

## 小结

数据库的完整性是指数据的正确性和相容性，正确性指合法和有效，相容性指同一事实的两个数据应该一致。数据库是否具备完整性关系到DBS能否真实反映现实世界，因此DBMS提供完整性检查功能来保证数据库中数据的完整性。

The background is a collage of various digital and technological elements. It includes a grid of blue squares, a world map, a bar chart, a pie chart, and a large pixelated hand cursor pointing towards the bottom right. The overall color scheme is light blue and white.

谢谢！





# 实体完整性的实现

# 实体完整性的实现

## 总述

内容：关系数据库中完整性约束包括实体完整性、参照完整性和用户自定义的完整性。学习SQL中实体完整性的实现方法。

# 实体完整性的实现

- 一、实体完整性定义
- 二、实体完整性检查和违约处理

## 一、实体完整性定义

SQL中的完整性约束规则有主键约束、外键约束、属性值约束等多种形式。

主键约束（实体完整性）是数据中最重要的一种约束。在关系中主键值不允许空也不允许出现重复。

## 一、实体完整性定义

1) 主键子句 ( 表级 ) PRIMARY KEY (sno)

Create Table S

```
( sno CHAR(4),  
  sname CHAR(8),  
  age SMALLINT,  
  PRIMARY KEY(sno) );
```

2) 主键短语 ( 列级 )

```
sno CHAR(4) PRIMARY KEY
```

## 二、实体完整性检查和违约处理

插入或对主码列进行更新操作时，RDBMS按照实体完整性规则自动进行检查。包括：

- 1) 检查主码值是否唯一，如果不唯一则拒绝插入或修改；
- 2) 检查主码的各个属性是否为空，只要有一个为空就拒绝插入或修改。

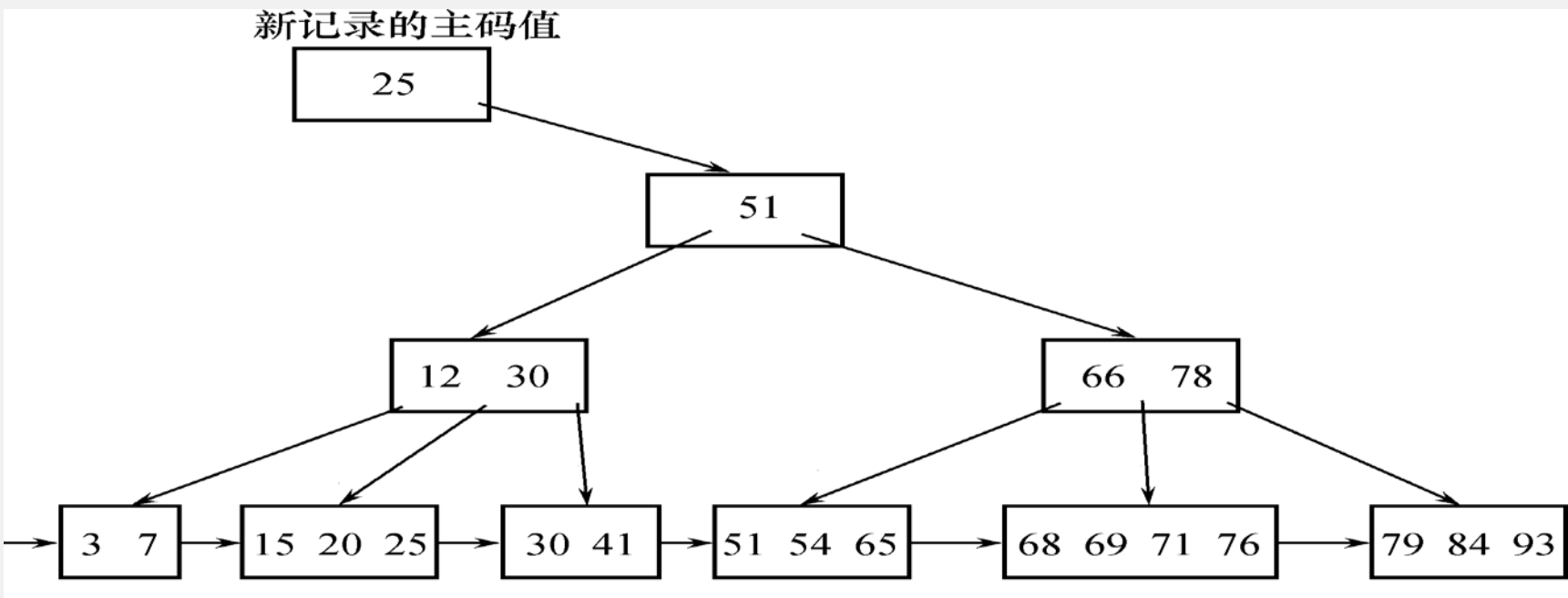
## 二、实体完整性检查和违约处理

检查记录中主码值是否唯一的一种方法是 进行全表扫描。



## 二、实体完整性检查和违约处理

检查记录中主码值是否唯一的另一种方法是 使用索引。





## 练习1

建立选课表SC ( sno,cno,grade ) 及其主键约束。

## 解答

建立选课表SC ( sno,cno,grade ) 及其主键约束。

```
CREATE TABLE SC  
( sno CHAR(4) ,  
  cno CHAR(4) ,  
  grade SMALLINT ,  
  PRIMARY KEY ( sno , cno ) ,  
)
```

The background is a collage of various digital and technological elements. It includes a grid of blue squares, a world map, a bar chart, a pie chart, and a large pixelated hand cursor pointing towards the bottom right. The overall color scheme is light blue and white.

谢谢！



# 参照完整性的实现

# 参照完整性的实现

## 总述

内容：关系数据库中完整性约束包括实体完整性、参照完整性和用户自定义的完整性。学习SQL中参照完整性的实现方法。

# 参照完整性的实现

- 一、参照完整性定义
- 二、参照完整性检查和违约处理

## 一、参照完整性定义

参照完整性指外键约束。

根据参照完整性规则，依赖关系中外键或者为空值，或者是基本关系（参照关系）中的该键的某个值。

例如在SC ( sno , cno , grade ) 中：

**FOREIGN KEY**(sno) **REFERENCES** S(sno)

**FOREIGN KEY**(cno) **REFERENCES** C(cno)

S和C称为基本关系或参照关系，SC称为依赖关系。

## 一、参照完整性定义

参照完整性可在建表时定义。

```
CREATE TABLE SC
```

```
( sno CHAR(4) NOT NULL,
```

```
  cno CHAR(4) NOT NULL,
```

```
  grade SMALLINT,
```

```
  PRIMARY KEY (sno,cno),
```

```
  FOREIGN KEY(sno) REFERENCES S(sno),
```

```
  FOREIGN KEY(cno)REFERENCES C(cno) )
```



## 二、参照完整性检查和违约处理

可能破坏参照完整性的情况及违约处理

被参照表 ( 例如S )	参照表 ( 例如SC )	违约处理
可能破坏参照完整性 ←	插入元组	拒绝/递归插入
可能破坏参照完整性 ←	修改外码值	拒绝/递归修改

sno	sname	age
s1	LI	17
s2	WANG	19

sno	cno	grade
s1	c1	90
s2	c2	89

## 二、参照完整性检查和违约处理

可能破坏参照完整性的情况及违约处理

被参照表 ( 例如S )	参照表 ( 例如SC )	违约处理
删除元组	可能破坏参照完整性	拒绝/级连删除/ 设置为空值/默认
修改主码值	可能破坏参照完整性	拒绝/级连修改/ 设置为空值/默认

sno	sname	age
s1	LI	17
s2	WANG	19

sno	cno	grade
s1	c1	90
s2	c2	89

## 二、参照完整性检查和违约处理

### 参照完整性违约处理

1 ) 拒绝(NO ACTION)执行 ;

默认策略

2 ) 级联(CASCADE)操作 ;

3 ) 设置为空值 ( SET-NULL ) 或默认值 ( SET-DEFAULT ) 。

对于参照完整性，除了应该定义外码，还应定义外码列是否允许空值。

## 二、参照完整性检查和违约处理

显式说明参照完整性的违约处理示例：

```
CREATE TABLE SC
(sno CHAR(4) ,
 cno CHAR(4) ,  grade SMALLINT ,
 PRIMARY KEY ( sno , cno ) ,
 FOREIGN KEY (sno) REFERENCES S (sno)
 ON DELETE CASCADE //级联删除SC表中相应的元组
 ON UPDATE CASCADE //级联更新SC表中相应的元组
 FOREIGN KEY (cno) REFERENCES C (cno)
 ON DELETE NO ACTION ) //当删除中C的元组造成了与SC表不一致时拒绝删除
)
```

## 练习

已知：学生关系( 学号 , 姓名 , 性别 , 专业号 )

专业关系 ( 专业号 , 专业名 , 最高分 ) ,

试述在专业关系中删除某一专业时，DBMS可能的处理策略。

# 参照完整性的实现

## 解答

已知：学生关系( 学号，姓名，性别，专业号)

专业关系( 专业号，专业名，最高分 )，

试述在专业关系中删除某一专业时，DBMS可能的处理策略。

专业号	专业名	最高分
m1	计算机	590
m2	电子	578
m3	土建	589

学号	姓名	性别	专业号
s1	LI	M	m1
s2	WANG	F	m2
s3	LIU	F	m2
s4	CHEN	M	m3

The background is a collage of various digital and technological elements. It includes a grid of blue squares, a world map, a bar chart, a pie chart, and a large pixelated hand cursor pointing towards the bottom right. The overall color scheme is light blue and white.

谢谢！



# 用户自定义完整性的实现



# 用户自定义完整性的实现

## 总述

内容：关系数据库中完整性约束包括实体完整性、参照完整性和用户自定义的完整性。学习SQL中用户自定义完整性的实现方法。

# 用户自定义完整性的实现

- 一、非空值约束
- 二、列值唯一约束
- 三、CHECK约束
- 四、元组上的约束

## 一、非空值约束

建表时在属性定义后加上关键字NOT NULL。

例1：在定义SC表时，说明sno、cno、grade属性不允许取空值。

Create Table SC

```
( sno CHAR(9) NOT NULL ,  
  cno CHAR(4) NOT NULL ,  
  grade SMALLINT NOT NULL ,  
  PRIMARY KEY (sno , cno) ) ;
```

如果在表级定义实体完整性，隐含了sno，cno不允许取空值，则在列级不允许取空值的定义就不必写了。

## 二、列值唯一约束

例2：建立部门表DEPT，要求部门名称dname列取值唯一，部门编号deptno列为主码。

```
Create Table DEPT
( deptno NUMERIC(2) ,
  dname CHAR(9) UNIQUE ,
  location CHAR(10) ,
  PRIMARY KEY (deptno)
) ;
```

## 三、CHECK约束

例3：S表的sex只允许取“男”或“女”。

Create Table S

```
( sno CHAR(9) PRIMARY KEY ,  
  sname CHAR(8) NOT NULL ,  
  sex CHAR(2) CHECK ( sex IN ( '男' , '女' ) ) ,  
  age SMALLINT ,  
  dept CHAR(20)  
);
```

## 四、元组上的约束

同属性值限制相比，元组级的限制可以设置不同属性之间的取值的相互约束条件。

例4：当学生的性别是男时，其名字不能以Ms.打头。

Create Table S

(sno CHAR(9) PRIMARY KEY ,

sname CHAR(8) NOT NULL ,

sex CHAR(2) ,

**CHECK** (sex='女' OR sname NOT LIKE 'Ms.% ')

## 四、元组上的约束

例5：学生的年龄在16到25之间，学号以“s”打头，后2位是数字。

Create Table S

( sno CHAR(3) PRIMARY KEY,

age SMALLINT,

sex CHAR(2) ,

**CHECK** ((age >= 16 and age <= 25)

or (sno LIKE 's[0-9][0-9]' )) );

The background is a collage of various digital and technological elements. It includes a grid of blue squares, a world map, a bar chart, a pie chart, and a large pixelated hand cursor pointing towards the bottom right. The overall color scheme is light blue and white.

谢谢！





# 完整性约束命名

# 完整性约束命名

## 总述

内容：在SQL的Create Table中可以定义主键约束、外键约束和属性约束，对完整性约束条件命名可以灵活地管理约束。本知识点学习SQL中完整性约束命名子句的用法。

# 完整性约束命名

- 一、完整性约束命名子句
- 二、完整性约束条件的修改

## 一、完整性约束命名子句

语法：

**CONSTRAINT** <完整性约束条件名>  
                    <完整性约束条件>

完整性约束条件：

PRIMARY KEY短语、FOREIGN KEY短语、  
CHECK短语等。

## 一、完整性约束命名子句

例1：建立学生表S，要求姓名不能取空值，年龄小于30，性别只能是“男”或“女”。

Create Table S

```
( sno char(4),  
  sname char(8) CONSTRAINT C1 NOT NULL,  
  age smallint CONSTRAINT C2 CHECK (age < 30),  
  sex char(2) CONSTRAINT C3 CHECK (sex IN( '男 ','女')),  
  CONSTRAINT SKey PRIMARY KEY(sno)  
);
```

## 二、完整性约束条件的修改

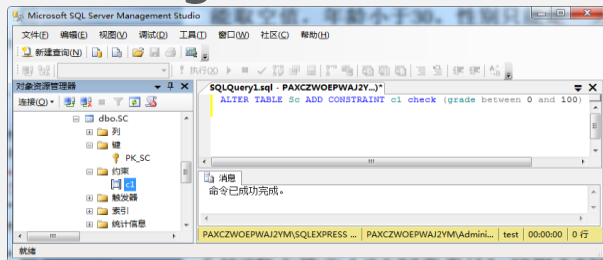
修改表中的完整性限制 (ALTER TABLE)

例2：为表S增加主键约束。

Alter Table S **ADD CONSTRAINT** a1 PRIMARY KEY (sno);

例3：为表SC增加成绩在0到100之间的限制。

Alter Table SC **ADD CONSTRAINT** c1 CHECK (grade between 0 and 100);



## 二、完整性约束条件的修改

修改表中的完整性限制 (ALTER TABLE)

例4: **取消**表S对年龄小于30的限制。

```
Alter Table S DROP CONSTRAINT C2;
```

## 二、完整性约束条件的修改

修改表中的完整性限制 (ALTER TABLE)

例5: **修改**表S中的约束条件, 要求年龄由小于30改为小于40。

先删除原来的约束条件, 再增加新的约束条件。

```
Alter Table S DROP CONSTRAINT C2;
```

```
Alter Table S ADD CONSTRAINT C2  
CHECK (age < 40);
```



## 小结

给完整性约束命名的好处是方便约束的维护、管理、记忆和使用。

在建表时未指定情况下用户可以通过Alter Table命令随时施加约束和修改表中的完整性限制。



谢谢！



# 触发器

## 总述

内容：数据库管理系统提供统一的数据保护功能来保证数据的安全可靠和正确有效。学习触发器的基本概念和用法。

- 一、触发器作用
- 二、触发器定义与删除
- 三、SQLServer中的触发器

## 一、触发器作用

触发器（ Triggers ）是一种特殊的存储过程，它在指定表中的数据发生变化时自动执行。导致触发器生效的操作包括Insert、Update 和 Delete等。

触发器类似约束，但比约束更灵活，可实施复杂的检查和操作。

触发器与表是紧密联系的，离开了表它将不复存在（这点与约束十分类似）。

## 一、触发器作用

( 1 ) 对数据进行复杂的检查，而这些检查是约束所不能完成的。

例如，当对选课 ( SC ) 表进行插入或修改时，要求有些课只有女生才能选，男生不能选。

( 2 ) 基于原始被触发表中的更新，对另一张表进行更改。

例如，删除学生 ( S ) 表中的记录时自动级联删除该学生在SC中的选课。

## 二、触发器定义与删除

### ( 1 ) 定义触发器

```
Create Trigger <触发器名> {Before|After} <触发事件> on <表名>  
    For Each{Row|Statement}    //行级或语句级  
    [ When<触发条件> ]  
    <触发动作体>
```

### ( 2 ) 触发器删除

```
Drop Trigger <触发器名> on <表名>
```



## 二、触发器定义与删除

例1：定义一个After行级触发器，为学生表S定义完整性规则，年龄不低于15岁，如果低于15岁自动改为15岁。

```
Create Trigger aa  After Insert or Update on S
For Each Row
as BEGIN
    IF NEW.age<15  THEN NEW.age=15;
    END IF;
END;
```

## 三、SQLServer中的触发器

Create Trigger <触发器名> on <表名>

{For | After | Instead of} { [ Insert ] [ , ] [ Update ] [ , ] [ Delete ] }

as <触发动作体>

- For Insert 表中进行插入操作时被激发；
- For Update 表中进行更新操作时被激发；
- For Delete 表中进行删除操作时被激发；
- After 在一个触发事件发生之后触发；
- Instead of 指定执行触发动作体而不是执行触发事件，从而替代触发事件的操作。

## 三、SQLServer中的触发器

例2：当删除学生表 S 中的某条记录时级连删除成绩表SC中相应的选课记录。

```
Create Trigger a on S for Delete  
as Delete SC  
From SC,Deleted  
Where Deleted.sno=SC.sno
```

在SQLServer中，当执行Delete时，被删除的行被放入一个逻辑表Deleted中，当执行Insert时，被插入的行被同时放入一个逻辑表Inserted中。

## 小结

触发器是实现数据库完整性的一个重要方法，功能非常强大。触发器不仅可以用于数据库完整性检查，还可用于数据库安全性以及实现应用系统的业务流程等功能。SQL允许用户建立自己的触发器，一个触发器只能在一个表上建立，一个表上可建多个触发器，由于一个触发器的动作可能激发另一个触发器，所以使用触发器要慎重。

The background is a collage of various digital and technological elements. It includes a grid of blue squares, a world map, a bar chart, a pie chart, and a large pixelated hand cursor pointing towards the bottom right. The overall color scheme is light blue and white.

谢谢！