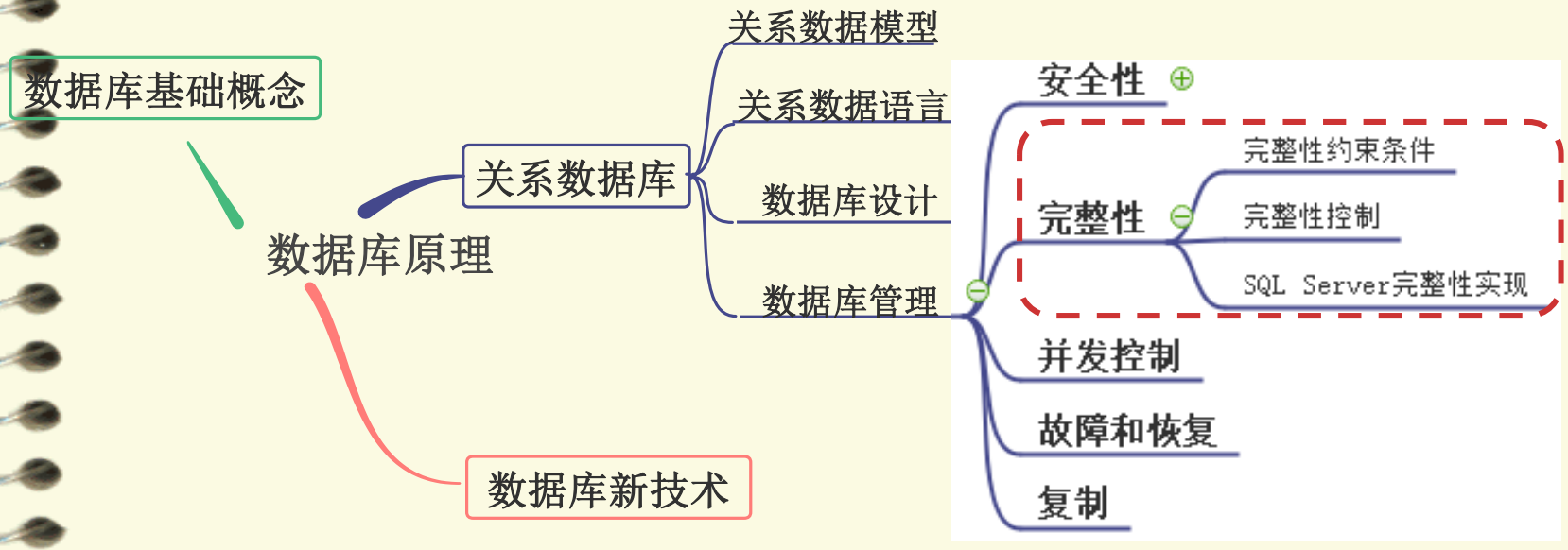
数据库原理

第7章 数据库管理

辽东学院 鲁 琴

本节要点


数据库基础概念
数据库原理
数据库新技术
关系数据库
关系数据模型
关系数据语言
数据库设计
数据库管理
安全性
完整性
完整性约束条件
完整性控制
SQL Server完整性实现
并发控制
故障和恢复
复制

什么是数据库的完整性

- 数据库的完整性是指数据的正确性和相容性，防止不合语义的数据进入数据库
  - 学生的年龄必须是整数，取值范围为14--29
  - 学生的性别只能是’男’或’女’
  - 学生的学号一定是唯一的
  - 学生所在的系必须是学校开设的系
- 数据库是否具备完整性关系到数据库系统能否真实地反映现实世界，因此维护数据库的完整性是非常重要的

完整性控制机制

（1）完整性约束条件定义机制

- 完整性约束条件是数据模型的一个重要组成部分,它约束了数据库中数据的语义
- DBMS应提供手段让用户根据现实世界的语义定义数据库的完整性约束条件，并把它们作为模式的一部分存入数据库中

（2）完整性检查机制

- 检查用户发出的操作请求是否违背了完整性约束条件

（3）违约反应

- 如果发现用户的操作请求使数据违背了完整性约束条件，则采取一定的动作来保证数据的完整性

2 完整性

2.1 完整性约束条件

2.2 完整性控制

2.3 SQL Server完整性实现

2.1 完整性约束条件

- 整个完整性控制都是围绕完整性约束条件进行的
- 完整性约束条件是完整性控制机制的核心

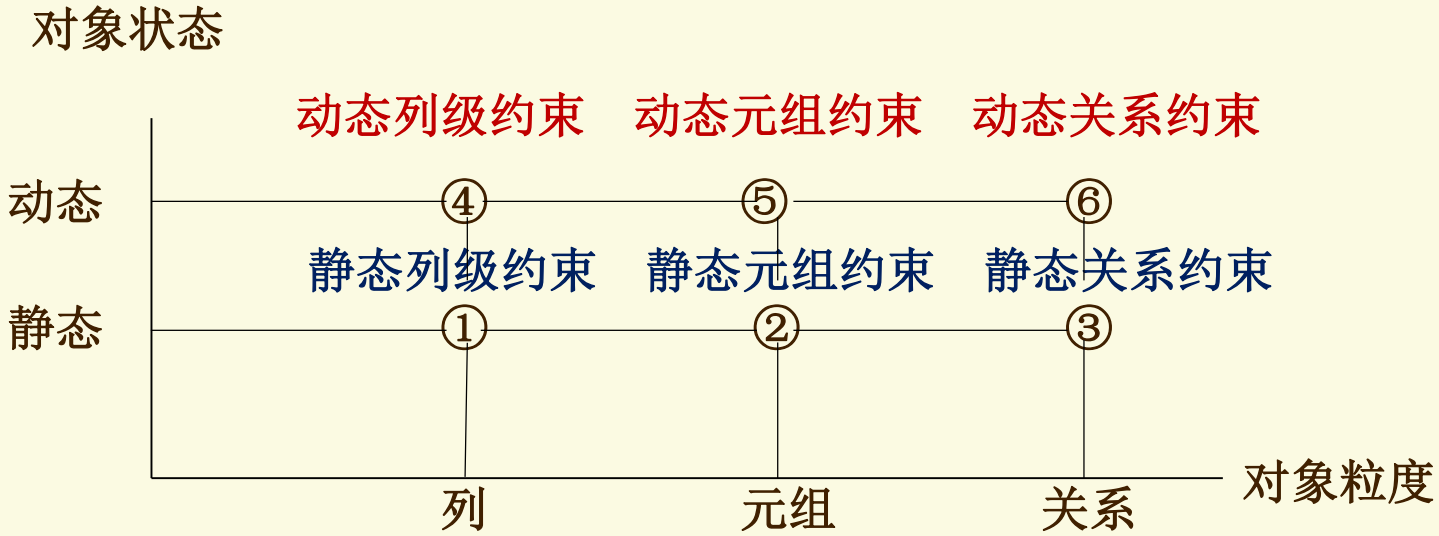
完整性约束条件作用的对象

- 列
  - 对属性的取值类型、范围、精度等的约束条件
- 元组
  - 对元组中各个属性列间的联系的约束
- 关系
  - 对若干元组间、关系集合上以及关系之间的联系的约束

对象的状态

- 静态
  - 对静态对象的约束是反映数据库状态合理性的约束
  - 这是最重要的一类完整性约束
- 动态
  - 对动态对象的约束是反映数据库状态变迁的约束

完整性约束条件分类


对象状态
动态列级约束
动态元组约束
动态关系约束
动态
静态
④
⑤
⑥
静态列级约束
静态元组约束
静态关系约束
①
②
③
列
元组
关系
对象粒度

(1) 静态列级约束(五类)

- 对一个列的取值域的说明
- 最常见、最简单、最容易实现

1) 对数据类型的约束

- 包括数据的类型、长度、单位、精度等
- 例：规定学生姓名的数据类型为字符串，长度为10

2) 对数据格式的约束

- 例：规定学号的格式为前两位表示入学年份，后四位为顺序编号

3) 对取值范围或取值集合的约束

- 例：性别的取值集合为{男,女}

4) 对空值的约束

- 空值表示未定义或未知的值，它与零值、空格不同
- 有的列允许空值，有的则不允许 例如规定成绩可以为空值

5) 其他约束 例：关于列的排序说明，组合列等

（2）静态元组约束

- 规定组成一个元组的各个列之间的约束关系
  - 订货关系中包含发货量、订货量等列，

    发货量不得超过订货量
  - 教师关系中包含职称、工资等列，

    教授的工资不得低于700元
- 静态元组约束只局限在单个元组上

（3）静态关系约束

- 在一个关系的各个元组之间或者若干关系之间
- 常见静态关系约束

  1) 实体完整性约束

  2) 参照完整性约束

  3) 函数依赖约束

  4) 统计约束

统计约束

- 定义某个字段值与一个关系多个元组的统计值之间的约束关系
  - 例：规定部门经理的工资不得高于本部门职工平均工资的5倍，不得低于本部门职工平均工资的2倍
  - 本部门职工的平均工资值是一个统计计算值

（4）动态列级约束

- 修改列定义或列值时应满足的约束条件
- 常见动态列级约束

1) 修改列定义时的约束

- 例: 规定将原来允许空值的列改为不允许空值时，如果该列目前已存在空值，则拒绝这种修改

2)修改列值时的约束

- 修改列值有时需要参照其旧值，并且新旧值之间需要满足某种约束条件
- 例: 职工工资调整不得低于其原来工资，学生年龄只能增长

（5）动态元组约束

- 修改某个元组的值时需要参照其旧值，并且新旧值之间需要满足某种约束条件
  - 例: 职工工资调整不得低于其原来工资+工龄*1.5

（6）动态关系约束

- 动态关系约束是加在关系变化前后状态上的限制条件

  例：事务一致性、原子性等约束条件

完整性约束条件小结

粒度 / 状态	列 级	元 组 级	关 系 级
静 态	列定义 ·类型 ·格式 ·值域 ·空值	元组值应满足的条件	实体完整性约束 参照完整性约束 函数依赖约束 统计约束
动 态	改变列定义或列值	元组新旧值之间应满足的约束条件	关系新旧状态间应满足的约束条件

2 完整性

2.1 完整性约束条件

2.2 完整性控制

2.3 SQL Server完整性实现

2.2 完整性控制

一、DBMS的完整性控制机制

二、关系系统三类完整性的实现

三、参照完整性的实现

一、DBMS的完整性控制机制

（1）定义功能

- 一个完善的完整性控制机制应该允许用户定义各类完整性约束条件

（2）检查功能

- 立即执行的约束(Immediate constraints)
  - 检查是否违背完整性约束的时机通常是在一条语句执行完后立即检查
- 延迟执行的约束(Deferred constrainsts)
  - 有时，完整性检查需要延迟到整个事务执行结束后再进行，如银行转账

（3）违约反应

- 拒绝该操作
- 其他处理方法

二、关系系统三类完整性的实现

- 违反实体完整性规则和用户定义的完整性规则的操作
  - 一般都是采用拒绝执行的方式进行处理
- 而对于违反参照完整性的操作，
  - 并不都是简单地拒绝执行
  - 有时还需要采取另一种方法，即接受这个操作，同时执行一些附加的操作，以保证数据库的状态仍然是正确的

三、参照完整性的实现

被参照关系（目标关系）（父关系）

参照关系（子关系）

- 例:职工－部门数据库包含两个表：

  部门(部门号，名称，经理名，电话)

  职工(职工号，姓名，年龄，职务，工资，部门号)

- RDBMS实现参照完整性时需要考虑以下3方面：

  （1）外码是否可以接受空值的问题

  （2）删除被参照关系的元组时的考虑

  （3）修改被参照关系中主码的考虑

（1）外码是否可以接受空值的问题

- 外码是否能够取空值是依赖于应用环境的语义的
- 在实现参照完整性时
  - 系统除了应该提供定义外码的机制
  - 还应提供定义外码列是否允许空值的机制

例1：在职工－部门数据库中，“职工”表关系包含有外码“部门号”

- 某一元组的这一列若为空值，表示这个职工尚未分配到任何具体的部门工作
- 这和应用环境的语义是相符的，因此“职工”表的“部门号”列应允许空值。

参照完整性的实现

主属性
不能为空值

例2：在学生－选课数据库中，Student关系为被参照关系，其主码为Sno;

SC为参照关系，外码为Sno。

- 若SC的Sno为空值，则表明尚不存在的某个学生，或者某个不知学号的学生，选修了某门课程，其成绩记录在Grade列中
- 这与学校的应用环境是不相符的，因此SC的Sno列不能取空值

（2）删除被参照关系的元组时的考虑

- 出现违约操作的情形
  - 需要删除被参照关系的某个元组，而参照关系有若干元组的外码值与被删除的被参照关系的主码值相对应
- 违约反应：
  - 级联删除（**CASCADES**）
  - 受限删除（**RESTRICTED**）
  - 置空值删除（**NULLIFIES**）

这三种处理方法，哪一种是正确的，要依应用环境的语义来定

违约反应策略

- 级联删除
  - 将参照关系中所有外码值与被参照关系中要删除元组主码值相对应的元组一起删除
- 受限删除
  - 只有当参照关系中没有任何元组的外码值与要删除的被参照关系的元组的主码值相对应时，系统才执行删除操作，否则拒绝此删除操作
- 置空值删除
  - 删除被参照关系的元组，并将参照关系中所有与被参照关系中被删除元组主码值相等的外码值置为空值

违约反应策略

例：要删除Student关系中Sno=950001的元组，

而SC关系中有4个元组的Sno都等于950001

- 级联删除：将SC关系中所有4个Sno=950001的元组一起删除。如果参照关系同时又是另一个关系的被参照关系，则这种删除操作会继续级联下去
- 受限删除：系统将拒绝执行此删除操作
- 置空值删除：将SC关系中所有Sno=950001的元组的Sno值置为空值

在学生选课数据库中

第一种方法和第二种方法都是对的，第三种方法不符合应用环境语义

（3）修改被参照关系中主码的考虑

- 级联修改
  - 修改被参照关系中主码值同时，用相同的方法修改参照关系中相应的外码值
- 受限修改
  - 拒绝此修改操作。只当参照关系中没有任何元组的外码值等于被参照关系中某个元组的主码值时，这个元组的主码值才能被修改
- 置空值修改
  - 修改被参照关系中主码值，同时将参照关系中相应的外码值置为空值

修改被参照关系中主码的考虑

- 例：学生95001休学一年后复学，这时需要将Student关系中Sno=95001的元组中Sno值改为96123。而SC关系中有4个元组的Sno=95001

- 级联修改：将SC关系中4个Sno=95001元组中的Sno值也改为96123。如果参照关系同时又是另一个关系的被参照关系，则这种修改操作会继续级联下去

修改被参照关系中主码的考虑

- **受限修改：只有SC中没有任何元组的Sno=95001时，才能修改Student表中Sno=950001的元组的Sno值改为96123**
- **置空值修改：将Student表中Sno=95001的元组的Sno值改为96123，而将SC表中所有Sno=95001的元组的Sno值置为空值**

在学生选课数据库中只有第一种方法（级联修改）是正确的。

参照完整性的实现结论

- RDBMS在实现参照完整性时，除了需要向用户提供定义主码、外码的机制
- 还需要向用户提供按照自己的应用要求选择处理依赖关系中对应的元组的方法。

SQL语言定义完整性约束条件

CREATE TABLE语句

ALTER TABLE语句

码

取值唯一的列

参照完整性

其他约束条件

2 完整性

2.1 完整性约束条件

2.2 完整性控制

2.3 SQL Server完整性实现

2.3 SQL Server的完整性

一、SQL Server中的实体完整性

二、SQL Server中的参照完整性

三、SQL Server中用户定义的完整性

一、SQL Server中的实体完整性

- **SQL Server在CREATE TABLE语句中提供了PRIMARY KEY子句，供用户在建表时指定关系的主码列。**
  - **在列级使用PRIMARY KEY子句**
  - **在表级使用PRIMARY KEY子句**

SQL Server中定义实体完整性

- 例1：在学生选课数据库中，要定义Student表的Sno属性为主码

```
CREATE TABLE Student
    (Sno   CHAR(8),
     Sname VARCHAR(20),
     Sage  INT,
     CONSTRAINT PK_SNO PRIMARY KEY (Sno));
```

SQL Server中定义实体完整性

- 例1：在学生选课数据库中，要定义Student表的Sno属性为主码

```
CREATE TABLE Student
        (Sno   CHAR(8) PRIMARY KEY ,
        Sname VARCHAR(20),
        Sage  INT);
```

SQL Server中定义实体完整性

- 例2：要在SC表中定义(Sno, Cno)为主码

```
CREATE TABLE SC
        (Sno   CHAR(8),
        Cno   CHAR(3),
        Grade DECIMAL(6,1),
        PRIMARY KEY (Sno, Cno));
```

SQL Server中的实体完整性

- 在用**PRIMARY KEY**语句定义了关系的主码后，每当用户程序对主码列进行更新操作时，系统自动进行完整性检查
  - 违约操作
    - 使主属性值为空值的操作
    - 使主码值在表中不唯一的操作
  - 违约反应
    - 系统拒绝此操作，从而保证了实体完整性

二、SQL Server中的参照完整性

- **SQL Server的CREATE TABLE语句允许用户定义参照完整性**
  - **用FOREIGN KEY子句定义哪些列为外码列**
  - **用REFERENCES子句指明这些外码相应于哪个表的主码**
  - **用ON DELETE CASCADE短语指明在删除被参照关系的元组时，同时删除参照关系中外码值等于被删除的被参照关系的元组中主码值的元组(没有默认为RESTRICT)**
  - **用ON UPDATE CASCADE短语指明在删除被参照关系的元组时，同时删除参照关系中外码值等于被删除的被参照关系的元组中主码值的元组(没有默认为RESTRICT)**

例1：建立表EMP表

```
CREATE TABLE EMP
        (Empno CHAR(4) PRIMARY KEY,
         Ename VARCHAR(10),
         Job VARCHAR(9),
         Mgr CHAR(4),
         Sal MONEY,
         Deptno CHAR(2),
         CONSTRAINT FK_DEPTNO    FOREIGN KEY (Deptno)
                 REFERENCES DEPT(Deptno)
                 ON DELETE CASCADE
                 ON UPDATE CASCADE ;
```

- 这时EMP表中外码为Deptno，它对应于DEPT表中的主码Deptno。
- 当要修改DEPT表中的Deptno值时，先要检查EMP表中有无元组的Deptno值与之对应
  - 若没有，系统接受这个修改操作
  - 否则，系统拒绝此操作

例1：建立表EMP表

```
CREATE TABLE EMP
      (Empno CHAR(4) PRIMARY,
       Ename VARCHAR(10),
       Job VARCHAR(9),
       Mgr CHAR(4),
       Sal MONDY,
       Deptno CHAR(2) CONSTRAINT FK_DEPTNO
        FOREIGN KEY REFERENCES DEPT(Deptno)
        ON DELETE CASCADE
        ON UPDATE CASCADE);
```

- 当要删除或修改DEPT表中某个元组时，系统要检查EMP表，若找到相应元组即将其随之删除或修改。
- 当要插入EMP表中某个元组时，系统要检查DEPT表，先要检查DEPT表中有无元组的Deptno值与之对应
  - 若没有，系统拒绝此插入操作
  - 否则，系统接受此操作

三、SQL Server中的用户定义的完整性

SQL Server中定义用户完整性的三种方法

- 用**CREATE TABLE**语句在建表时定义用户完整性约束
- 用**ALTER TABLE**语句添加用户完整性约束
- 通过触发器来定义用户的完整性规则

SQL Server中的用户定义的完整性（续）

1. 用CREATE TABLE语句在建表时定义用户完整性约束

- 可定义三类完整性约束
  - 列值非空（NOT NULL短语）
  - 列值唯一（UNIQUE短语）
  - 检查列值是否满足一个布尔表达式（CHECK短语）

SQL Server中的用户定义的完整性

例1：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 CHAR(2),
Dname VARCHAR(9) CONSTRAINT U1 UNIQUE,
Loc VARCHAR(10),
CONSTRAINT PK_DEPT PRIMARY KEY (Deptno));
```

其中 CONSTRAINT U1 UNIQUE 表示约束名为U1，该约束要求Dname列值唯一。

SQL Server中的用户定义的完整性（续）

例2： 建立学生登记表Student，要求学号在900000至999999之间，性别只能是‘男’或‘女’，姓名非空

```
CREATE TABLE Student
    (Sno    INT    CONSTRAINT C1 CHECK
                (Sno BETWEEN 10000 AND 99999),
     Sname  VARCHAR(20) CONSTRAINT C2 NOT NULL,
     Sage   INT,
     Ssex   VARCHAR(2)
                CONSTRAINT C4 CHECK (Ssex IN ('男', '女'));
```

SQL Server中的用户定义的完整性

例3：建立职工表EMP，要求每个职工的应发工资不得超过3000元。应发工资实际上就是实发工资列Sal与扣除项Deduct之和。

```
CREATE TABLE EMP
  (Eno    CHAR(4)
   Ename  VARCHAR(10),
   Job    VARCHAR(8),
   Sal    MONEY,
   Deduct MONEY,
   Deptno CHAR(2),
   CONSTRAINTS C1 CHECK (Sal + Deduct <=3000));
```

SQL Server中的用户定义的完整性（续）

例4： 为学生登记表Student，添加约束，年龄在14到40之间

ALTER TABLE Student ADD CONSTRAINT c_age

CHECK(Sage between 14 AND 40);

```
INSERT INTO Student(Sno,Sname,Sage,Ssex)
VALUES('16001','刘小',56,'IS');
```

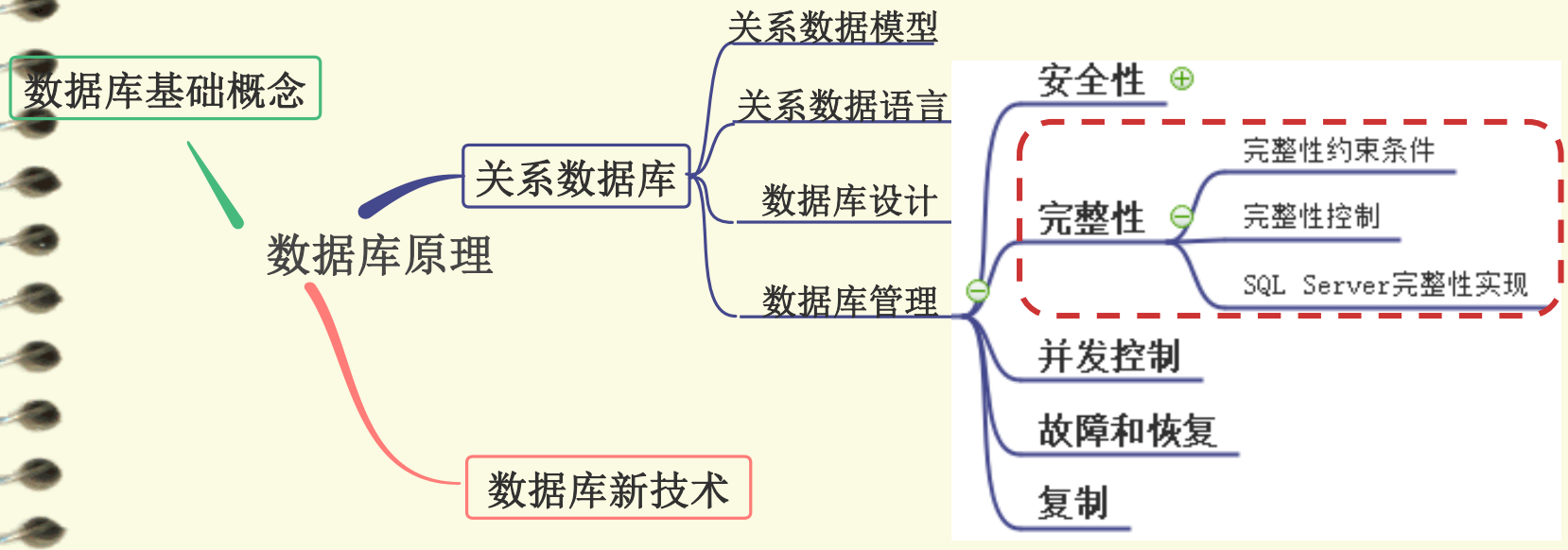
SQL Server中的用户定义的完整性

- 通过触发器来定义用户的完整性规则
  - 定义其它的完整性约束时，需要用数据库触发器（Trigger）来实现。
  - 数据库触发器是一类靠事件驱动的特殊过程，一旦由某个用户定义，任何用户对该数据的增、删、改操作均由服务器自动激活相应的触发子，在核心层进行集中的完整性控制。

定义数据库触发器的语句

- **CREATE [OR REPLACE] TRIGGER**

本节小结


数据库基础概念
数据库原理
数据库新技术
关系数据库
关系数据模型
关系数据语言
数据库设计
数据库管理
安全性
完整性
完整性约束条件
完整性控制
SQL Server完整性实现
并发控制
故障和恢复
复制

练习

- 职工—部门数据库包含两个关系模式：

  部门(部门号，名称，经理名，电话)

  职工(职工号，姓名，年龄，职务，工资，部门号)

- 用SQL语言定义两个关系模式，要求实现完整性约束：

  （1）定义每个关系的主码

  （2）定义参照完整性

  （3）定义职工年龄不得超过60岁