

1

数据库管理系统（DBMS）的完整性控制机制主要包括以下三方面功能：

1. 实体完整性（Entity Integrity）

- 保证表中每一行都有唯一标识。
- 通常通过设置主键（Primary Key）来实现，主键不能为 NULL，且必须唯一。

2. 参照完整性（Referential Integrity）

- 保证两个表之间的引用关系是有效的。
- 例如，外键（Foreign Key）引用的主表中必须存在对应的记录。

3. 用户定义的完整性（User-defined Integrity）

- 由用户根据具体应用需求定义的业务规则，比如取值范围、数据格式、状态约束等。
- 通常使用 CHECK、TRIGGER、应用程序逻辑等方式实现。

2

违反参照完整性的情况：

假设有两个表：

- 父表（被引用表）： Department(Dno)
- 子表（引用表）： Study(Dno)

以下是几种可能的违规情况：

1. 插入子表数据时，引用的父表记录不存在

- 例如：插入 Study(Sno=101, Dno='D05')，但 Department 表中没有 D05。

2. 删除父表数据时，被子表引用

- 例如：删除 Department 表中的 D01，但还有很多 Study 表的记录引用 D01。

3. 更新父表主键时，子表存在引用该键的记录

- 例如：将 Department 的 D01 改为 D02，但 Study 中还有很多记录引用 D01。

处理策略（在定义外键时指定）：

1. **RESTRICT / NO ACTION（限制）**：不允许违反参照完整性的操作，直接拒绝执行。

2. **CASCADE (级联)**：操作父表时，自动对相关子表进行相同操作。

- 删除父表记录，则子表中对应记录也自动删除。
- 更新主键，则子表中外键也自动更新。

3. **SET NULL**：将子表中外键字段设置为 `NULL` 。

4. **SET DEFAULT**：将子表中外键字段设置为默认值。

3

```
DELIMITER //
```

```
CREATE TRIGGER insert_study_after_student
AFTER INSERT ON Student
FOR EACH ROW
BEGIN
    INSERT INTO Study (Sno, Dno, rank)
    VALUES (NEW.Sno, NULL, NULL);
END;
//
```

```
DELIMITER ;
```

4

1. 数据库结构说明

涉及四张表：

- **books**(编号, 书名, 出版社, 定价)
- **readers**(编号, 姓名, 读者类型, 已借数量)
- **borrowinf**(图书编号, 读者编号, 借期, 还期)
- **readertype**(类型编号, 类型名称)

2. 各类完整性约束设计

实体完整性：

- 每个表的主键（编号）不为空且唯一：
 - books(编号)
 - readers(编号)
 - readertype(类型编号)

参照完整性：

- readers(读者类型) → readertype(类型编号)
- borrowinf(图书编号) → books(编号)
- borrowinf(读者编号) → readers(编号)

用户自定义完整性：

- readers.已借数量 ≥ 0
- readers.读者类型 默认值为 3
- borrowinf.还期 \geq 借期

3. SQL 语句实现（以 MySQL 为例）

-- 创建图书表

```
CREATE TABLE books (  
    编号 INT PRIMARY KEY,  
    书名 VARCHAR(100) NOT NULL,  
    出版社 VARCHAR(100),  
    定价 DECIMAL(10, 2) CHECK (定价 >= 0)  
);
```

-- 创建读者类型表

```
CREATE TABLE readertype (  
    类型编号 INT PRIMARY KEY,  
    类型名称 VARCHAR(50) NOT NULL  
);
```

-- 创建读者表

```
CREATE TABLE readers (  
    编号 INT PRIMARY KEY,  
    姓名 VARCHAR(100) NOT NULL,  
    读者类型 INT DEFAULT 3,  
    已借数量 INT CHECK (已借数量 >= 0),  
    FOREIGN KEY (读者类型) REFERENCES readertype(类型编号)  
);
```

-- 创建借阅信息表

```
CREATE TABLE borrowinf (  
    图书编号 INT,  
    读者编号 INT,  
    借期 DATE NOT NULL,  
    还期 DATE NOT NULL,  
    PRIMARY KEY (图书编号, 读者编号, 借期),  
    FOREIGN KEY (图书编号) REFERENCES books(编号),  
    FOREIGN KEY (读者编号) REFERENCES readers(编号),  
    CHECK (还期 >= 借期)  
);
```