

# Lecture 5: 关系数据理论

## 函数依赖

### 引入

#### 引入

- 这里面的数据有大量的冗余，此时插入数据和删除数据都很麻烦；
- 例:请评价教学表的关系模式**有哪些问题？是否合理？**

教学(学号, 姓名, 年龄, 系名, 系主任, 课程名, 成绩)

| 学号    | 姓名 | 年龄 | 系名  | 系主任 | 课程名  | 成绩 |
|-------|----|----|-----|-----|------|----|
| 98001 | 李华 | 21 | 计算机 | 王民  | C语言  | 90 |
| 98001 | 李华 | 21 | 计算机 | 王民  | 高等数学 | 80 |
| 98002 | 张平 | 22 | 计算机 | 王民  | C语言  | 65 |
| 98002 | 张平 | 22 | 计算机 | 王民  | 高等数学 | 70 |
| 98003 | 陈兵 | 21 | 数学  | 赵敏  | 高等数学 | 95 |
| 98003 | 陈兵 | 21 | 数学  | 赵敏  | 离散数学 | 75 |

- 实质上是某些数据依赖导致各种操作异常；

## 函数依赖基础概念

### 概念：函数依赖基础概念

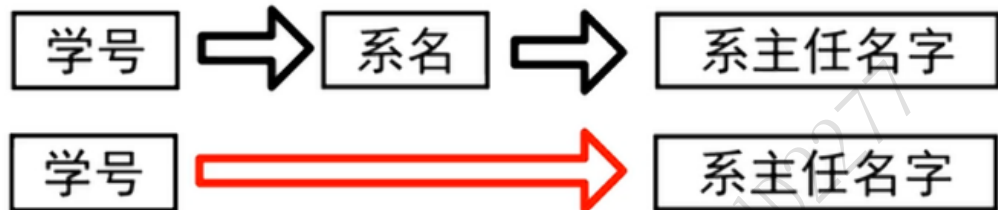
- 概念：
  - 比如 A 属性值相同时，B 属性值必然相等；
  - 比如：学号确定姓名，学号确定所在系，记作： Sno-NSname, Sno->Sdept
  - 在关系数据库中，如果在一个表中，属性集 X 的值可以唯一确定属性集 Y 的值，则称 Y 函数依赖于 X；
  - 记作  $X \rightarrow Y$
  - 比如如果 StudentID 可以唯一确定 Name，则记作 StudentID  $\rightarrow$  Name；
- 概念：平凡函数依赖
  - 如果 Y 是 X 的子集（即  $Y \subseteq X$ ），则称  $X \rightarrow Y$  为平凡函数依赖；
  - 平凡函数依赖总是成立的，因为它没有提供有意义的信息；
  - 如果  $X=\{StudentID, Name\}$   $X=\{StudentID, Name\}$ ，则  $X \rightarrow Y$  是平凡函数依赖，因为  $Y \subseteq X$
- 概念：传递函数依赖

在关系模式  $R(U)$  中，对于  $U$  的子集  $X$  和  $Y$ ：

◆如果  $X \rightarrow Y$  ( $Y \not\subseteq X$ )， $Y \not\rightarrow X$ ， $Y \rightarrow Z$ ， $Z \not\subseteq Y$ ，  
则称  $Z$  对  $X$  **传递函数依赖**。

记为： $X \xrightarrow{\text{传递}} Z$ 。

◆ 例如：学生信息(学号, 系名, 系主任名字)



## 范式

### 基础概念

**概念：**什么是范式

- 概念：
  - 评价关系模式“好坏”的理论标准就是范式；
  - 范式 (NormalForm) 是设计数据库结构中所要遵循的规则和指导方式，表示的是关系模式的规范化程度；
  - 范式由低到高：1 NF, 2 NF, 3 NF, BCNF, 4 NF, 5 NF
- 概念：关系模式的规范化
  - 关系模式的规范化：把一个低一级的关系模式分解为高级关系模式的过程；
  - 规范化程度越高，数据冗余越小，数据更新 (增删改) 操作的复杂度越低；

### 第一范式

**概念：**第一范式

- 定义：
  - 如果一个关系模式  $R$  的所有属性都是不可分的基本数据项，则属于第一范式；
- 理解：
  - 列不可再分；
  - 第一范式是关系模式的最低要求，必须满足；
- 图示：
  - 下图中的出生日期不属于第一范式，因为其列可以再分；

| 姓名 | 出生日期 |   |    | 性别 |
|----|------|---|----|----|
|    | 年    | 月 | 日  |    |
| 张三 | 2019 | 8 | 7  | 男  |
| 李四 | 2000 | 7 | 20 | 女  |

## 第二范式

**概念：**第二范式

- 定义：
  - 如关系模式 R 属于 1NF，且每个非主属性完全函数依赖于候选码，则称 R 属于第二范式；
  - 即：第二范式要求表必须满足 1NF，并且所有非主键列必须完全依赖于整个主键，而不是部分依赖；
- 推论：
  - 单属性码的关系模式必然属于 2NF；
- 示例：
  - 假设有一个表存储学生选课信息，主键是（学生ID，课程ID）；
  - 如果表中还存储了学生姓名，而学生姓名只依赖于学生 ID，而不是整个主键（学生ID，课程ID），则不符合 2NF。应将学生姓名移到单独的学生表中；
- 代码：不符合 2NF
  - 假设有一个表存储学生选课信息，主键是（学生 ID，课程 ID），同时存储学生姓名。学生姓名只依赖于学生 ID，而不是整个主键，因此不符合 2NF；

```
CREATE TABLE StudentCourses (
  StudentID INT,
  CourseID INT,
  StudentName VARCHAR(50),
  PRIMARY KEY (StudentID, CourseID)
);
```

- 代码：符合 2NF

```
-- 学生表
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  StudentName VARCHAR(50)
);

-- 选课表
CREATE TABLE StudentCourses (
  StudentID INT,
  CourseID INT,
  PRIMARY KEY (StudentID, CourseID),
```

```
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
```

```
);
```

## 第三范式

**概念：** 第三范式

- 概念：
  - 如果关系模式符合第二范式，并且关系的任意非主属性都不传递函数依赖于任何候选码，则称关系属于第三范式；
  - 即：第三范式要求表必须满足 **2NF**，并且**所有非主键列之间不能有传递依赖**，即非主键列必须直接依赖于主键，而不是通过其他非主键列间接依赖；
- 理解：
  - 理解：若 R 属于 **3NF**，则每一个非主属性既不部分依赖于码也不传递依赖于码；
- 推论：
  - 双目的关系模式必然是属于 3 NF；
- 示例：
  - 假设有一个表存储学生信息，包含学生 ID、姓名、班级 ID 和班级名称。班级名称依赖于班级 ID，而班级 ID 又依赖于学生 ID，存在传递依赖，因此不符合 **3NF**；
- 代码：不符合 3 NF 的表

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    ClassID INT,  
    ClassName VARCHAR(50)  
);
```

- 代码：符合 **3NF** 的表

```
-- 班级表  
CREATE TABLE Classes (  
    ClassID INT PRIMARY KEY,  
    ClassName VARCHAR(50)  
);  
  
-- 学生表  
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    ClassID INT,  
    FOREIGN KEY (ClassID) REFERENCES Classes(ClassID)  
);
```

## BCNF 范式

**概念：** **BCNF** 范式

- 概念：
  - **BCNF** 是第三范式的强化版，要求表必须满足 **3NF**，并且对于每一个函数依赖  $X \rightarrow Y$ ，**X** 必须是超键（即 **X** 能够唯一标识一行）；
- 示例：
  - 假设有一个表存储学生选课信息，包含学生 ID、课程 ID 和教师 ID；
  - 假设每个课程只有一个教师，但一个教师可以教授多门课程；
  - 如果存在函数依赖（课程 ID  $\rightarrow$  教师 ID），而课程 ID 不是超键，则不符合 **BCNF**
- 代码：不符合 **BCNF** 的表

```
CREATE TABLE StudentCourses (  
    StudentID INT,  
    CourseID INT,  
    TeacherID INT,  
    PRIMARY KEY (StudentID, CourseID)  
);
```

- 代码：符合 **BCNF** 的表

```
-- 课程表  
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY,  
    TeacherID INT  
);  
  
-- 选课表  
CREATE TABLE StudentCourses (  
    StudentID INT,  
    CourseID INT,  
    PRIMARY KEY (StudentID, CourseID),  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);
```

## 总结

- **1NF**：确保每列是原子的。
- **2NF**：确保非主键列完全依赖于整个主键。
- **3NF**：确保非主键列之间没有传递依赖。
- **BCNF**：确保所有函数依赖的左部是超键