# 如何求关系模式中的键

### 1. 确定关系模式中的候选键

#### （1）列出所有的属性

假设关系模式 R(A, B, C, …, Z)，我们需要找出哪些属性组合可以唯一标识一条记录。

#### （2）找出函数依赖（Functional Dependencies, FDs）

函数依赖是指某些属性的值可以唯一决定其他属性。例如：

A → B, C（A 的值唯一确定 B 和 C）

(X, Y) → Z（X 和 Y 共同决定 Z）

这些函数依赖可以通过业务规则、数据分析或需求文档获取。

#### （3）计算候选键

闭包法（Closure Method）

定义： 给定属性集 XXX，它的闭包 X+X^+X+ 是根据函数依赖可以推导出的所有属性。

步骤：

选定一个属性或属性集 XXX 作为起始候选。

计算 X+X^+X+（基于 FDs 逐步添加属性）。

如果 X+X^+X+ 包含关系模式的所有属性，则 XXX 是超键。

检查最小性：

如果 XXX 仍然包含多余属性，尝试移除某些属性，确保最小性。

若移除某个属性后仍然能唯一标识所有属性，则去掉该属性。

示例 给定关系 R(A, B, C, D, E)，函数依赖：

(A, B)⁺ = {A, B, C, D, E}（覆盖所有属性）

候选键 = (A, B)

### 3. 确定主键

主键是从候选键中选出的一个，通常选择：

较少的属性个数（减少存储开销）。

较稳定的属性（避免经常修改）。

业务逻辑上最合适的属性。

例如：

候选键有 (A, B), (C, D)，可以选 (A, B) 作为主键。

如果 A 是学号，B 是课程号，且学号是稳定的，则更可能选择 (A, B)。

# 最小函数依赖集计算是否可以简化

## ****1. 直接合并拆分右侧和冗余依赖检查****

在遍历 FDs 时，**同时进行右侧拆分**和**冗余依赖检查**，避免拆分后再遍历一次。

**做法**：

* **拆分右侧**：如果右侧有多个属性，直接拆分为单个属性依赖。
* **删除冗余依赖**：计算闭包，看是否可以由其他 FDs 推导出来。

## ****2. 直接简化左侧****

在简化左侧时，不必逐个尝试删除，而是**一次性计算闭包**来判断哪些属性是冗余的。

**做法**：

* 对于每个 FD X→YX \to YX→Y，尝试移除 XXX 中的某个属性 AAA。
* 计算剩下的部分 (X−A)+(X - A)^+(X−A)+，看是否仍能推出 YYY：
  + 若可以，则 AAA 是冗余的，去掉 AAA。
  + 否则，保留 AAA。

# BCF左部为什么是superkey，不是key

### ****（1）如果左部仅仅是候选键****

候选键是一个**最小**的唯一标识属性集，但如果 BCNF 仅仅要求**左侧必须是候选键**，可能仍然会存在**冗余依赖**。举例说明：

#### ****示例 1：候选键不满足 BCNF****

关系模式：

R(A,B,C)R(A, B, C)R(A,B,C)

函数依赖：

1. A→BA \to BA→B
2. B→CB \to CB→C
3. 候选键是 AAA，因为 AAA 能唯一确定 B,CB, CB,C（即 A+={A,B,C}A^+ = \{A, B, C\}A+={A,B,C}）。

这里：

* A→BA \to BA→B 是一个有效的 FD，AAA 是候选键，符合 3NF。
* B→CB \to CB→C 使得 BBB 决定了 CCC，但 BBB 不是超键，只是一个普通属性。

如果只要求**左部是候选键**，那么 B→CB \to CB→C 依赖会破坏 BCNF，因为：

* BBB 不能唯一标识元组（即 BBB 不是超键）。
* 但 B→CB \to CB→C 是一个非平凡的函数依赖。

### ****（2）为什么 BCNF 需要左部是超键****

BCNF 解决了**候选键不够强**的问题。BCNF 规定：

**每一个非平凡函数依赖** X→YX \to YX→Y **中，**XXX **必须是超键，而不只是候选键。**

如果要求**每个 FD 左侧是超键**，则：

* **不会出现类似** B→CB \to CB→C **这样的问题**，因为 BBB 不是超键，会导致分解。
* 确保所有依赖都依赖于**整个键**，不会有部分依赖（Partial Dependency）或传递依赖（Transitive Dependency）。

在 BCNF 规范化过程中，我们会拆分：

1. B→CB \to CB→C 违反 BCNF，因此拆分：
   * R1(B,C)R\_1(B, C)R1​(B,C) ，BBB 是主键。
   * R2(A,B)R\_2(A, B)R2​(A,B) ，AAA 是主键。

最终，所有依赖的左部都是超键，满足 BCNF。