

NEW PLAN

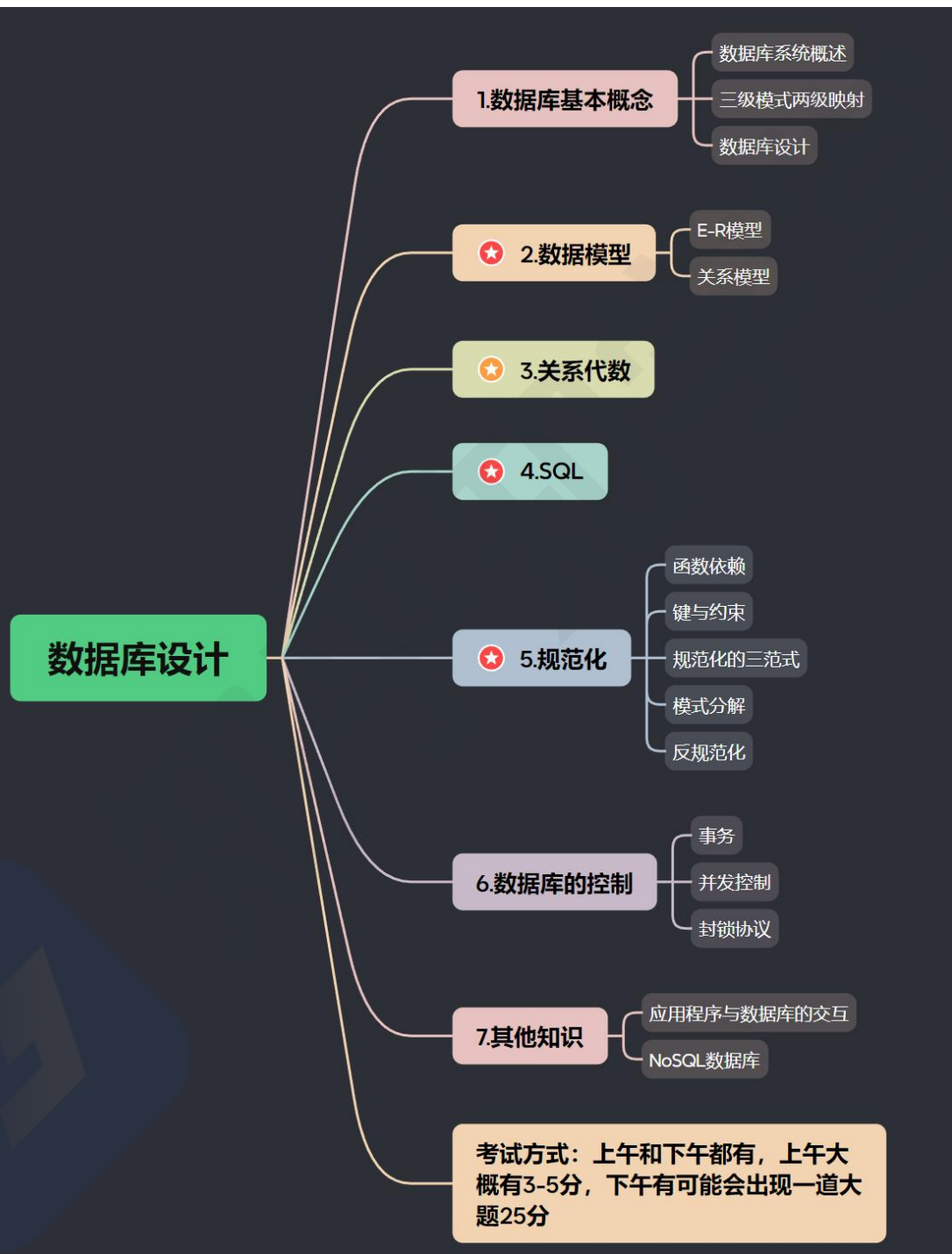
软考高级架构师

• — 段 新 征 程 •





大纲介绍



函数依赖



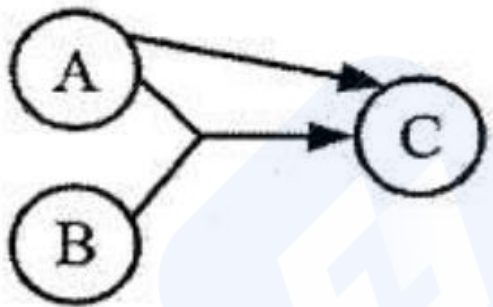
函数依赖： 给定一个X，能唯一确定一个Y，就称X决定(确定)Y，或者说Y依赖于X。

- 例如： $Y=X*X$ 函数，此时X能确定Y的值，但是Y无法确定X的值，比如 $x=2$ ， $y=4$ ，但是 $y=4$ 无法确定 $x=2$ 。

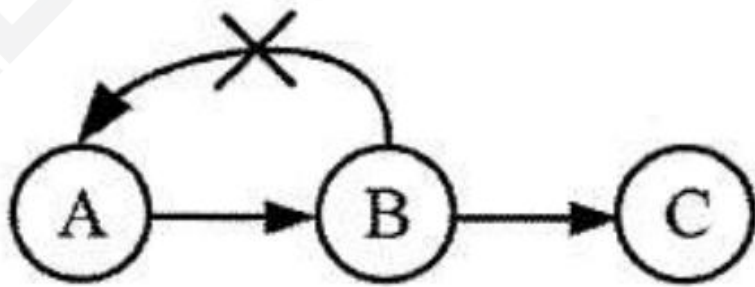


函数依赖又可扩展以下两种规则：

- **部分函数依赖：** A可决定C，(A, B)也可决定C，既然A都可以决定C了，那么要不要B其实都无所谓了，所有这种就称之为部分函数依赖。
- **传递函数依赖：** 当A和B不相同，A可决定B，B可决定C，则A可决定C，是传递函数依赖；若A和B相同，则不存在传递了，A就可以直接就可决定C。



部分函数依赖



传递函数依赖



函数依赖公理系统



函数依赖的公理系统是指一组用于推导和证明函数依赖的规则和公理集合。这些公理系统基本上是数学理论的扩展，包括以下公理和规则：

- 自反律：对于任意属性集合 X 和 Y ，有 $X \subseteq Y$ ，则 $X \rightarrow Y$ 。
- 增广律：对于任意属性集合 X 、 Y 和 Z ，如果 $X \rightarrow Y$ ，那么 $XZ \rightarrow YZ$ 。
- 合并律：对于任意属性集合 X 、 Y 和 Z ，如果 $X \rightarrow Y$ 和 $X \rightarrow Z$ ，那么 $X \rightarrow YZ$ 。
- 分解律：对于任意属性集合 X 、 Y 和 Z ，如果 $X \rightarrow YZ$ ，则 $X \rightarrow Y$ 和 $X \rightarrow Z$ 。
- 合成律：对于任意属性集合 X 、 Y 和 Z ，如果 $X \rightarrow Y$ 且 $Y \rightarrow Z$ ，那么 $XZ \rightarrow YZ$ 。
- 传递律：对于任意属性集合 X 、 Y 和 Z ，如果 $X \rightarrow Y$ 且 $Y \rightarrow Z$ ，那么 $X \rightarrow Z$ 。

键与约束



超键: 指能够唯一标识关系中每个元组的属性集合。换句话说, 超键中的属性组合可以保证每个元组在关系中都是唯一的。

举例: 假设我们有一个学生表, 包含以下属性: 学号、姓名、性别、出生日期, 那么, 以下属性集合都是超键: {学号}、{学号, 出生日期}、{姓名, 性别, 出生日期}, 因为这些属性组合都可以唯一标识每个学生。



候选键: 超键中不包含任何冗余属性的超键。换句话说, 候选键中的每个属性都是必需的, 用于唯一标识元组。

举例: 在上面的例子中, {学号} 是候选键, 因为它是唯一标识学生的最小属性集合。{姓名, 性别, 出生日期} 也是候选键, 但它包含冗余属性, 因为姓名和性别和出生日期可以由学号推断出来。



主属性: 是指除候选键以外的、具有代表意义的非重复且非衍生属性。

举例: 在上面的例子中, 姓名、性别和出生日期都是主属性。因为它们具有代表意义, 并且不是由其他属性推导出来的。



主键: 从候选键中选取的一个属性或属性集合, 作为表中元组的唯一标识符。

举例: 在上面的例子中, 我们可以选择 {学号} 作为主键。



总结:

- 超键: 能唯一标识记录的属性集合
- 候选键: 最小的超键
- 主属性: 除候选键外的具有代表意义的非重复且非衍生属性
- 主键: 任选一个候选键

键与约束



设计一张学生课程表，包含以下属性：学号、姓名、系名、课程名称、教师姓名



超键：是表中能唯一区分每条记录的数据项集合。该表有两个超键：

- 学号：能唯一标识每位学生
- 姓名+系名+课程名称：也能唯一标识每门课程



候选键：是表中的最小超键，用于关联其他表或保证数据完整性。该表的候选键是学号。因为学号作为最小的超键，既能唯一标识每位学生，又适合作为主键与其他表建立关联。



主属性：是除候选键之外的非重复且非衍生的属性集合。该表的主属性为：

- 姓名：标识学生的名称，非重复且非从其他属性导出
- 系名：标识学生的系别，非重复且非从其他属性导出
- 课程名称：标识选修课程的名称，非重复且非从其他属性导出
- 教师姓名：标识任课教师的姓名，非重复且非从其他属性导出



键与约束



外键：是指一个表中的属性，它引用另一个表中的主键。外键用于建立表之间的关系。



实体完整性约束：即主键约束，主键值不能为空，也不能重复。



参照完整性约束：即外键约束，外键必须是其他表中已经存在的主键的值，或者为空。



用户自定义完整性约束：自定义表达式约束，如设定年龄属性的值必须在0到180之间。

键与约束

雇员表:

雇员编号	姓名	性别	职位	部门编号
e1	张三	男	经理	d1
e2	李四	女	开发员	d1
e3	王五	男	测试员	d2

部门表:

部门编号	部门名
d1	技术部
d2	销售部
d3	产品部



超键: 雇员编号, 姓名, 部门编号, 部门名(这些属性的任意组合可以唯一标识一行记录)



候选键: 雇员编号(可以唯一标识一行记录, 并且没有它的任何属性子集具有同样的特征)



主键: 雇员编号(从候选键中选择的最简单属性, 用于唯一标识一行记录)



外键: 部门编号(在雇员表中指向部门表的主键, 用于两张表之间的联系)



主属性: 雇员编号, 姓名, 职位(决定雇员实体的属性)

范式

 **第一范式1NF：**要求数据库表中的所有字段都是不可分割的原子值。通俗地说，第一范式就是表中不允许有小表的存在。比如，对于如下的员工表，就不属于第一范式：

员工编号	员工姓名	出生日期	薪资/月		所属部门
			基本工资/月	补贴/月	
1	王红	19900908	9000	1000	101
...	...	...	...	...	

范式

例：用一个单一的关系模式学生来描述学校的教务系统：学生(学号，学生姓名，系号，系主任姓名，课程号，成绩)

依赖关系(学号->学生姓名，学号->所在系，所在系>系主任姓名，(学号，课程号)->成绩)

学号	学生姓名	所在系	系主任姓名	课程号	成绩
201102	张明	计算机系	章三	04	70
201103	王红	计算机系	章三	05	60
201103	王红	计算机系	章三	04	80
201103	王红	计算机系	章三	06	87
201104	李青	机械系	王五	09	79
...	...	...	...	...	...



第二范式：在1NF的基础上，要求数据库表中的每个非主属性完全依赖于候选键。也就是说，一个表只描述一件事物，通俗地说，就是表中不能存在联合主键，按照定义，上面的学生表就不满足2NF，因为学号不能完全确定成绩(每个学生可以选多门课)。



解决方案：将学生表分解为：

- 学生(学号，学生姓名，系编号，系名，系主任)
- 选课(学号，课程号，成绩)。

每张表均属于2NF。



第三范式：在2NF的基础上，要求数据库表中的每个非主属性不依赖于其它非主属性。也就是说，数据表中的每一列都和主键直接相关，而不依赖于其它列。



继续上面的实例，学生关系模式就不属于3NF，因为学生无法直接决定系主任和系名，是由学号 $\rightarrow$ 系编号，再由系编号 $\rightarrow$ 系主任，系编号 $\rightarrow$ 系名，因此存在非主属性对主属性的传递依赖，

解决方案：将学生表进一步分解为：



- 学生(学号，学生姓名，系编号)
- 系(系编号，系名，系主任)
- 选课(学号，课程号，成绩)

每张表都属于3NF。



BC范式(BCNF): 规范化数据库设计的一种方法, 它对关系型数据库中的表进行分解, 其符合第三范式(3NF), 同时尽量避免数据冗余和不一致性, 提高数据的可靠性和完整性。



假设仓库管理关系表(仓库ID, 存储物品ID, 管理员ID, 数量), 且有一个管理员只在一个仓库工作; 一个仓库可以存储多种物品。此关系模式已经属于了3NF, 那么这个关系模式是否存在问题呢? 我们来看以下几种操作:

- 删除异常: 当仓库被清空后, 所有"存储物品ID"和"数量"信息被删除的同时, "仓库ID"和"管理员ID"信息也被删除了。
- 插入异常: 当仓库没有存储任何物品时, 无法给仓库分配管理员。
- 更新异常: 如果仓库换了管理员, 则表中所有行的管理员ID都要修改。

解决方案: 把仓库管理关系表分解为二个关系表:



- 仓库管理: (仓库ID, 管理员ID);
- 仓库: (仓库ID, 存储物品ID, 数量)。

这样的数据库表是符合BCNF范式的, 消除了删除异常、插入异常和更新异常。

 给定关系模式 $R(U, F)$, $U=\{A, B, C, D\}$, $F=\{AB \rightarrow C, CD \rightarrow B\}$ 。关系 R (), 且分别有()。

A.只有1个候选关键字ACB

B.只有1个候选关键字BCD

C.有2个候选关键字ACD和ABD

D.有2个候选关键字ACB和BCD

A.0个非主属性和4个主属性

B.1个非主属性和3个主属性

C.2个非主属性和2个主属性

D.3个非主属性和1个主属性

候选关键字的求法：根据依赖集，找出从未在右边出现过的属性，必然是候选键之一，以该属性为基础，根据依赖集依次扩展，看能否遍历所有属性，将无法遍历的加入候选键中。



设有关系模式 $R(E, N, M, L, Q)$ ，其函数依赖集为 $F=\{E \rightarrow N, EM \rightarrow Q, M \rightarrow L\}$ 。则关系模式 R 达到了(), 该关系模式()

A.1NF

B.2NF

C.3NF

D.BCNF

A.无需进行分解，因为已经达到了3NF

B.无需进行分解，因为已经达到了BCNF

C.尽管不存在部分函数依赖，但还存在传递依赖，所以需要进行分解

D.需要进行分解，因为存在冗余、修改操作的不一致性、插入和删除异常

T H E E N D

功不唐捐，玉汝于成！

开 启 新 征 程

