# ****数据库知识点④****

**1.函数依赖的定义**

　　(a)平凡的函数依赖与非平凡的函数依赖

　　(b)函数依赖是语义范畴的概念

　　(c)函数依赖关系的存在与时间无关

　　(d)函数依赖可以保证关系分解的无损连接性

**2.函数依赖的基本性质 SCD(SNO,SN,AGE,DEPT,MN,CNO,SCORE)**

　　(a)投影性:一组属性函数决定它的子集

　　举例：(SNO,CNO)->SNO 和(SNO,CNO)->CNO

　　(b)扩张性: 若x->y且w->z 则有(x,w)->(y,z)

　　举例：SNO->(SN,AGE) DEPT->MN 则有(SNO,DEPT)->(SN,AGE,MN)

　　(c)合并性：若x->y 且x->z则有x->(y,z)

　　举例：SNO->(SN,AGE) SNO->(DEPT,MN) 则有SNO->(SN,AGE,DEPT,MN)

　　(d)分解性：若x->(y,z)则有x->y, x->z

**3.完全函数依赖与部分函数依赖**

　　1.如果X→Y，并且对于X的任何一个真子集X′,都有X′ Y，则称**Y对X完全函数依赖（Full Functional Dependency）**，记作 X -(f)->Y

　　2.如果对X的某个真子集X′，有X′→Y，则称**Y对X部分函数依赖（Partial Functional Dependency)** 记作X -(p)-> Y。

　　举例:

　　在关系模式SCD中

　　因为SNO -\->SCORE，且CNO -\-> SCORE，所以有：（SNO，CNO)-(f)->SCORE 。

　　而SNO→AGE，所以（SNO，CNO)-(p)->AGE

　　3.只有当决定因素是组合属性时，讨论部分函数依赖才有意义

　　4.当决定因素是单属性时，只能是完全函数依赖。

　　举例：在关系模式S（SNO，SN，AGE，DEPT），决定因素为**单属性**SNO，有SNO→（SN，AGE，DEPT），不存在部分函数依赖。

**4.传递函数依赖**

　　1.定义：

　　设有关系模式R（U），U是属性全集，X，Y，Z是U的子集，

　　若X→Y，但Y -\->X，而Y→Z（Y 不属于X，Z 不属于Y），则称**Z对X传递函数依赖（Transitive Functional Dependency）**，记作：X-(t)-> Z。

　　如果Y→X，则X <->Y，这时称Z对X**直接函数依赖，而不是传递函数依赖。**

　　举例：

　　例如，在关系模式SCD中，SNO→DEPTN，但DEPTN -\->SNO，而DEPTN→MN，则有SNO-(t)->MN。系主任是对学号的传递依赖

　　当学生不存在重名的情况下，有SNO→SN，SN→SNO，SNO <->SN，SN→DEPTN，这时DEPTN对SNO是直接函依赖，而不是传递函数依赖。

**5.总结函数依赖**

　　综上所述，函数依赖分为**完全函数依赖、部分函数依赖**和**传递函数依赖**三类，它们是规范化理论的依据和规范化程度的准则，下面我们将以介绍的这些概念为基础，进行数据库的规范设计。

**6.范式**

　　规范化的基本思想是**消除关系模式中的数据冗余，消除数据依赖中的不合适的部分，解决数据插入、删除时发生异常现象。**

　　这就要求关系数据库设计出来的关系模式要满足一定的条件。关系数据库的规范化过程中：不同程度的规范化要求设立的不同标准称为范式（Normal Form）。**规范化的程度不同，产生不同的范式**。满足最基本规范化要求的关系模式叫第一范式，在第一范式中**进一步满足**一些要求为第二范式，以此类推就产生了第三范式等概念。

　　每种范式都规定了一些限制约束条件。

**7.范式发展过程**

　　范式的概念最早由E.F.Codd提出。

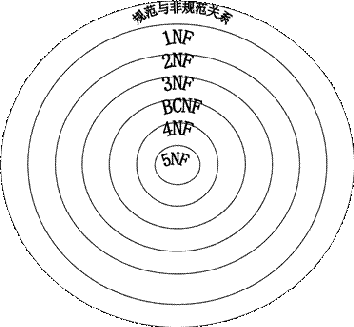
　　从1971年起，Codd相继提出了关系的三级规范化形式，即第一范式（1NF）、第二范式（2NF）、第三范式（3NF）。

　　1974年，Codd和Boyce以共同提出了一个新的范式的概念，即Boyce-Codd范式，简称BC范式。

　　1976年Fagin提出了第四范式，

　　后来又有人定义了第五范式。

　　至此在关系数据库规范中建立了一个范式系列：1NF,2NF,3NF,BCNF,4NF,5NF,一级比一级有更严格的要求。

[](http://images2015.cnblogs.com/blog/834545/201605/834545-20160524102250366-1483666681.gif)

**8.第一范式（First Normal Form）**

　　第一范式是最基本的规范形式，即关系中每个属性都是不可再分的简单项。

　　定义:

　　如果关系模式R，其所有的属性均为简单属性，即**每个属性域都是不可再分的**，则称R属于第一范式，简称1NF，记作R属于1NF。

　　然而，一个关系模式仅仅属于第一范式是不适用的。给出的关系模式SCD属于第一范式，但其具有大量的数据冗余，具有插入异常、删除异常、更新异常等弊端。

　　为什么会存在这种问题呢？

　　让我们分析一下SCD中的函数依赖关系，它的关系键是（SNO，CNO）的属性组合，所以有：

　　(SNO，CNO) -(f)->SCORE

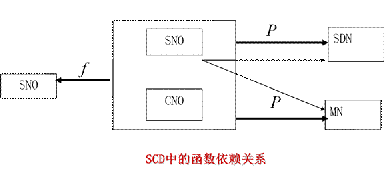
　　SNO→SN，(SNO,CNO)-(p)-> SN

　　SNO→AGE，(SNO,CNO)-(p)-> AGE

　　SNO→DEPT，(SNO,CNO)-(p)-> DEPT

　　SNO -(t)-> MN，(SNO,CNO)-(p)-> MN

　　我们可以用函数信赖图表示以上函数依赖关系，如下图所示。

[](http://images2015.cnblogs.com/blog/834545/201605/834545-20160524102252053-1510288301.gif)

**9.第二范式**

　　定义 如果关系模式RÎ1NF，且每个非主属性都**完全函数依赖**于R的每个关系键，则称R属于第二范式（Second Normal Form），简称2NF，记作R属于2NF。

　　分析：在关系模式SCD中，SNO，CNO为主属性，AGE，DEPT，MN，SCORE均为非主属性，经上述分析，存在非主属性对关系键的部分函数依赖，所以SCD不是2NF。

　　拆分：而将SCD分解成三个关系模式S(SNO,SN,AGE,DEPT)，D(DEPT,MN)，SC(SNO,CNO,SCORE)，其中S的关系键为SNO，D的关系键为DEPT，都是单属性，不可能存在部分函数依赖。

　　结果：而对于SC，(SNO，CNO)-(f)->SCORE。所以SCD分解后，消除了非主属性对关系键的部分函数依赖，S，D，SC均属于2NF。

　　经以上分析，可以得到两个结论：

　　1．从1NF关系中消除**非主属性**对**关系键**的**部分函数依赖**，则可得到2NF关系。

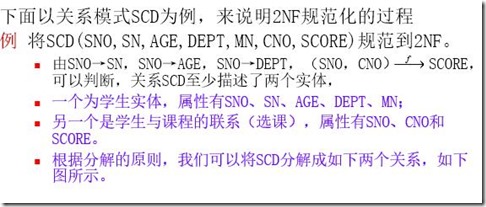
　　2．如果R的关系键为单属性，或R的全体属性均为主属性，则R属于2NF。

**10.1NF分解为2NF规则**

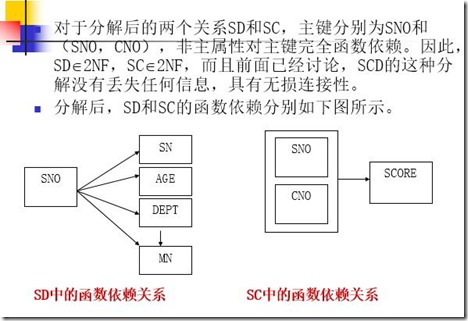
　　2NF规范化是指把1NF关系模式通过投影分解转换成2NF关系模式的集合。

　　分解时遵循的基本原则就是**“一事一表”**，让一个关系只描述一个实体或者实体间的联系。如果多于一个实体或联系，则进行投影分解。

**11.举例分析**

[](http://images2015.cnblogs.com/blog/834545/201605/834545-20160524102253381-558760776.jpg)

[](http://images2015.cnblogs.com/blog/834545/201605/834545-20160524102254913-759683178.jpg)

[](http://images2015.cnblogs.com/blog/834545/201605/834545-20160524102256944-1435220156.jpg)

　　改善之处：关系模式SD和SC在性能上比SCD有了显著提高。

　　(a)1NF的关系模式经过投影分解转换成2NF后，**消除了一些数据冗余**。

　　(b)分析图中SD和SC中的数据，可以看出，它们存储的冗余度比关系模式SCD有了较大辐度的降低。

　　(c)学生的姓名、年龄不需要重复存储多次。

　　(d)这样便可在一定程度上**避免数据更新所造成的数据不一致性的问题**。

　　(e)由于把学生的基本信息与选课信息分开存储，则学生基本信息因没选课而不能插入的问题得到了解决，插入异常现象得到了部分改善。

　　(f)同样，如果某个学生不再选修C1课程，只在选课关系SC中删去该该学生选修C1的记录即可，而SD中有关该学生的其它信息不会受到任何影响，也解决了部分删除异常问题。

[**上一篇数据库知识点③：http://www.cnblogs.com/zpfbuaa/p/5503200.html**](http://www.cnblogs.com/zpfbuaa/p/5503200.html)