[**数据库系统概论-任务01**](http://www.edu2act.net/team/shu-ju-ku-yuan-li1/tasks/1816/)

1)       硬件平台及数据库

     a)       足够大的内存

     b)       足够大的磁盘

     c)       系统具有较高的通道能力

2)       软件

     a)       数据库管理系统

     b)       操作系统

     c)       具有与数据库接口的高级语言及其编译系统

     d)       以数据库管理系统为核心的应用开发工具

     e)       数据库应用系统

3)       人员

     a)       数据库管理员 DBA

     b)       系统分析人员

     c)       数据库设计人员

     d)       应用程序员

     e)       用户

                                i.            偶然用户

                              ii.            简单用户

                             iii.            复杂用户

任务01-2.       请说明数据库系统的三级模式两级映像，并详细说明每一级模式的特点。

1) 外模式

     a)局部数据的逻辑结构和特征的描述

     b)与某一应用有关的数据的逻辑表示

     c)一个数据库可以有多个外模式，但一个应用程序只能使用一个外模式

2)模式

     a)全体数据的逻辑结构和特征的描述

     b)是中间层，既不涉及数据的物理存储细节和硬件环境，又与具体的应用程序、所使用的应用开发工具、高级程序设计语言无关

     c)一个数据库只有一个模式

      定义模式时不仅要定义数据的逻辑结构，还要定义数据之间的关系，例如完整性

3)内模式

     a)数据物理结构和存储方式的描述

     b)数据在数据库内部的表示方式

     c)一个数据库只有一个内模式

4)外模式/模式映像

     a)每一个外模式，数据库系统都有一个外模式／模式映象，定义外模式与模式之间的对应关系

     b)保证了数据的逻辑独立性

       当模式改变时，数据库管理员修改有关的外模式／模式映象，使外模式保持不变

       应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

5)模式/内模式映像

     a)定义了数据全局逻辑结构与存储结构之间的对应关系。

     b)保证了数据的物理独立性

       当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式／内模式映象，使模式保持不变

       应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性

[**数据库系统概论-任务02**](http://www.edu2act.cn/team/shu-ju-ku-yuan-li1/tasks/1830/)

任务02-1.       请简述关系模型的完整性约束规则，并详细介绍在参照完整性中，外码可以取什么样的值。

关系模型中有三类完整性约束：实体完整性、参照完整性和用户自定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，应有关系系统自动支持。用户定义的完整性是应用领域需要遵循的约束条件，体现了具体领域中的语义约束。

1、实体完整性规则：若属性（指一个或一组属性）A是基本关系R的主属性，则A不能取空值。所谓空值就是“不知道”或“不存在”或“无意义”的值。①实体完整性规则是针对基本关系而言的；一个基本表通常对应现实世界的一个实体集。②现实世界中的实体是可区分的，即它们有某种唯一性标识。③相应地，关系模型中以主码作为唯一性标识。④主码中的属性即主属性不能取空值，如果主属性取空值，则说明存在某个不可标识的实体，即存在不可区分的实体，这与②点相矛盾，所以这个规则成为实体完整性。

2、参照完整性：若属性（或属性组）F是基本关系R的外码，它与基本关系S的主码Ks相对应（基本关系R和S不一定是不同的关系），则对于R中每个元组在F上的值必须是：①或者取空值（F中的每个属性值均为空值）；②或者等于S中某个元祖的主码值。

3、用户定义的完整性：就是针对某一具体关系数据库的约束条件，它反映了某一具体应用所设计的数据必须满足的语义要求。

外码可以取得值：①空值（F的每个属性均为空值）；②等于S中某个元祖的主码值。例如：对于课本例2.1，学生关系中每个元组的“专业号”属性只能取空值（表示尚未给该学生分配专业）或者非空值（此时该值必须是专业关系中某个元祖的“专业号”值，表示该学生不可能分配到一个不存在的专业中。即被参照关系“专业”中一定存在一个元祖，他的主码值等于该参照关系“学生”中的外码值）。

任务02-2.     ***（请把本题的答案写在纸上，拍照上传）*** 设有下列关系模式：STUDENT(SNO,SNAME,AGE,SEX,DNO)其中，SNO表示学号，SNAME表示姓名，AGE表示年龄，SEX表示性别，DNO表示院系号。

SC(SNO,CNO,GRADE)，其中SNO表示学号，CNO表示课程号，GRADE表示成绩。

COURSE(CNO,CNAME)，其中CNO表示课程号，CNAME表示课程名。

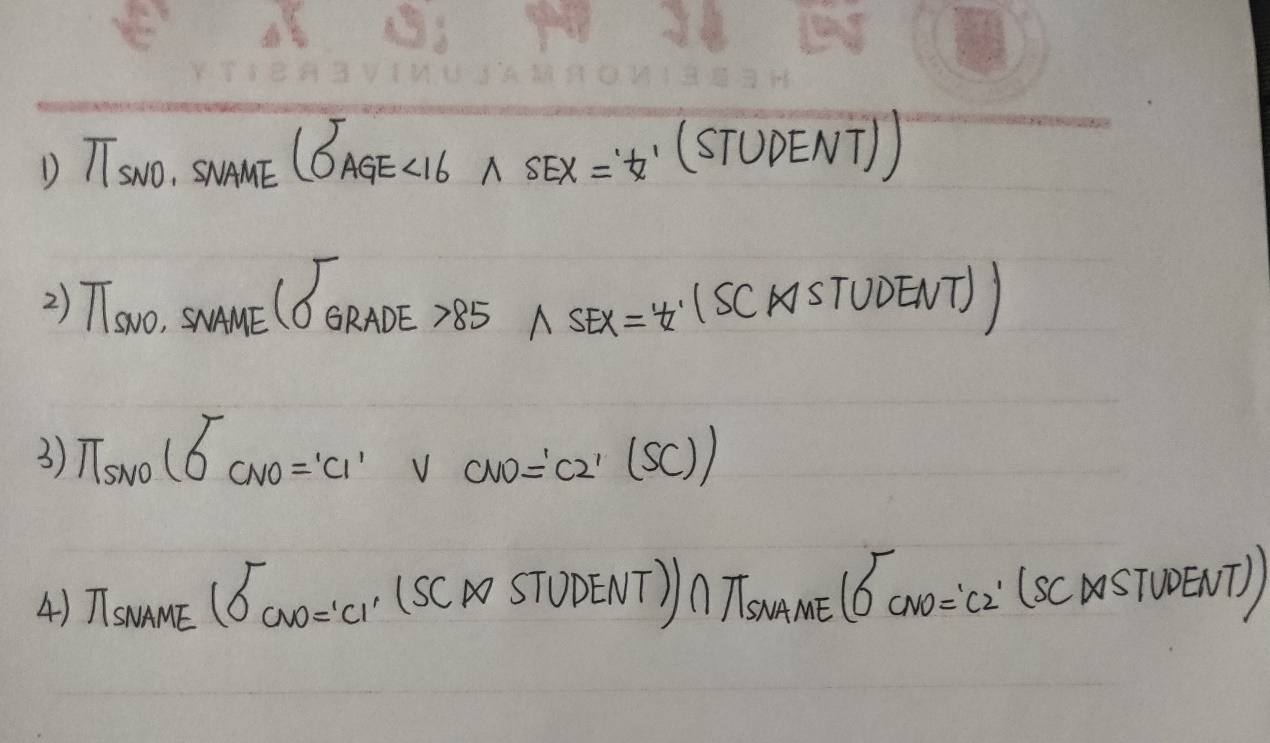
请用关系代数表示下列查询：

1）   检索年龄小于16的女学生的学号和姓名。

2）   检索成绩大于85分的女学生的学号、姓名。

3）   检索选修课程号为C1或C2的学生的学号

4）   检索至少选修了课程号为C1和C2的学生的学生姓名。



[**数据库系统概论-任务04**](http://www.edu2act.cn/team/shu-ju-ku-yuan-li1/tasks/1869/)

1.   设有如图所示的学生关系student：



试问student是否属于3NF? 为什么?若不是，它属于第几范式? 并将其规范化为3NF?

student1表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生号 | 学生名 | 年龄 | 性别 | 系号 |
| 201901 | 王语涵 | 16 | 女 | Soft3 |
| 201902 | 张璐婷 | 19 | 女 | Soft1 |
| 201903 | 张明远 | 20 | 男 | Soft2 |
| 201904 | 潘临 | 21 | 男 | Soft3 |

student2表：

|  |  |
| --- | --- |
| 系号 | 系名 |
| Soft1 | 机器学习 |
| Soft2 | Html5 |
| Soft3 | 大数据与智能设备 |

//任务名：任务04-1

//姓名：王亚红

//学号：2017011743

//班级：2017级2班

//日期：2019-5-10

student不属于3NF。

原因：student表中存在传递依赖：学生号->系号，系号->系名，系号不能决定学生号，系号不属于学生号，所以学生号->（传递）系名，所以student不属于3NF。

Student属于2NF，因为student表中不存在表中套表的情况，而且不存在非主属性对码的部分依赖。

将其规范化为3NF，将student表分解成两个表：

student1（学生号，学生名，年龄，性别，系号）

student2（系号，系名）

2.   设有如图所示的关系R



（不考虑重名）

试问R属于3NF? 为什么?若不是，它属于第几范式? 并如何规范化为3NF？

//任务名：任务04-2

//姓名：王亚红

//学号：2017011743

//班级：2017级2班

//日期：2019-5-10

R不属于3NF

原因：R表中的职工名为主码，职工名->单位号，单位号->单位名，而且单位号不属于职工名，单位号不能决定职工名，所以存在非主属性“单位名”对候选关键字“职工名”的传递函数依赖，所以R不是3NF。

R1（职工级别号，职工名，年龄，性别，单位号）

R2（单位号，单位名）

R1表：

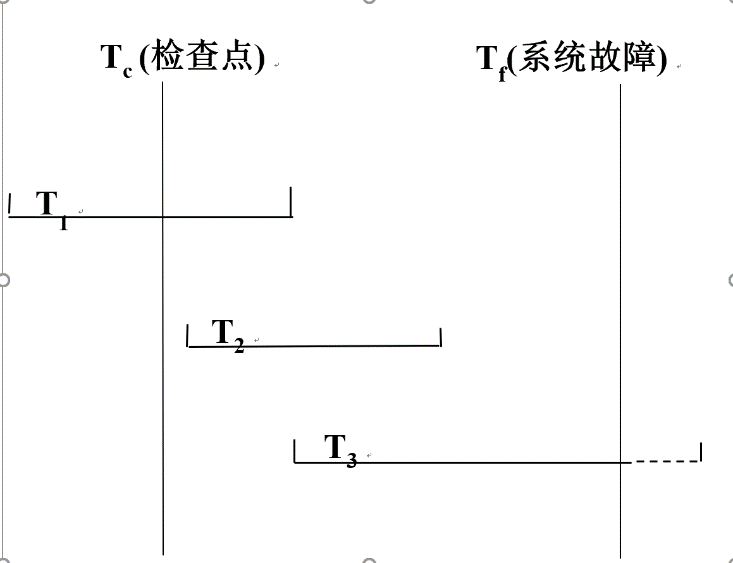
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 职工级别号 | 职工名 | 年龄 | 性别 | 单位号 |
| T1 | 丁一 | 20 | 男 | Soft3 |
| T2 | 张二 | 25 | 女 | Soft1 |
| T3 | 李三 | 38 | 女 | Soft3 |
| T4 | 王四 | 25 | 男 | Soft3 |

R2表：

|  |  |
| --- | --- |
| 单位号 | 单位名 |
| Soft1 | 机器学习 |
| Soft3 | 大数据与智能设备 |

[**数据库系统概论-任务06**](http://www.edu2act.cn/team/shu-ju-ku-yuan-li1/tasks/1895/)

1.数据库在Tf时刻发生故障时，T1、T2、T3三个事务的状态如下图所示，数据库的日志带有检查点，请给出故障恢复的策略。



//任务名：任务06-1

//姓名：王亚红

//学号：2017011743

//班级：2017级2班

//日期：2019-6-12

①事务T1：事务T1在检查点之前开始执行，在检查点之后系统故障之前提交。

对于事务T1的故障恢复策略：T1在检查点之后才提交，它对数据库所做的修改在故障发生时可能还在缓冲区中，尚未写入数据库，所以要REDO。

②事务T2：事务T2在检查点之后开始执行，在故障点之前提交。

对于事务T2的故障恢复策略：T2在检查点之后才提交，它对数据库所做的修改在故障发生时可能还在缓冲区中，尚未写入数据库，所以要REDO。

③事务T3：事务T3在检查点之后开始执行，在故障点时还未完成。

对于事务T3的故障恢复策略：T5在故障发生时还未完成，所以予以撤销。

1.试说明并发操作时可能带来哪几类数据不一致？用什么方法能避免各种不一致的情况？

//任务名：任务06-2

//姓名：王亚红

//学号：2017011743

//班级：2017级2班

//日期：2019-6-12

（1）并发操作带来的数据不一致性包括：

①丢失修改：两个事务T1和T2读入同一数据并修改，T2的提交结果破坏了T1提交的结果，导致T1的修改被丢失

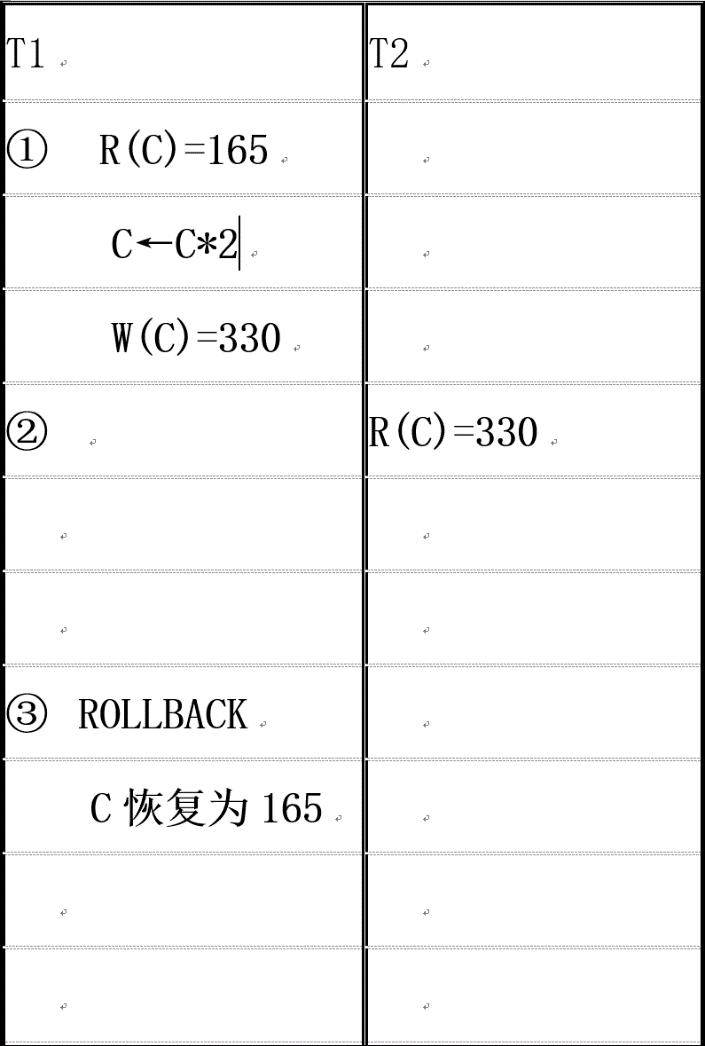
②不可重复读：不可重复读是指事务T1读取数据后，事务T2执行更新操作，使T1无法再现前一次读取结果。

③读“脏”数据：指的是事务T1修改某一数据，并将其写回磁盘；事务T2读取同一数据后，T1由于某种原因被撤销；这时T1已修改过的数据恢复原值，T2读到的数据就与数据库中的数据不一致；T2读到的数据就为“脏”数据，即不正确的数据。

（2）避免各种不一致的情况的方法：

数据不一致性是由于并发操作破坏了事务的隔离性，所以避免各种不一致的情况的方法就是并发控制。并发控制指的是用正确的方式调度并发操作，是一个用户事务的执行不受其他事务的干扰，从而避免造成数据的不一致性。并发控制的主要技术包括：有封锁、时间戳、乐观控制法和多版本并发控制等。

2.设有两个事务T1，T2，其并发操作如下图所示，执行结果是什么？有什么问题？原因何在？



//任务名：任务06-3

//姓名：王亚红

//学号：2017011743

//班级：2017级2班

//日期：2019-6-12

（1）执行结果是T1读到的C=165，T2读到的C=330.

（2）问题：数据库中C的值为165，而T2读到的C为330，读到了不正确的数据。

（3）原因：T1将读到的C的值改为了330，写入C=330，而后T2读取数据库中的数据，即C=330,；后来T1因为某种原因被撤销（ROLLBACK），其之前的修改操作不再产生作用，C值恢复为原值165，但是T2中读取到的C仍然是330，与数据库中的内容不一致，T2读到的数据是“脏”数据，级不正确的数据。