***DBS*.数据库系统：**由数据库、数据库管理系统、应用程序、数据库管理员和用户组成。特点是数据共享、数据独立、减少数据冗余、数据安全和数据完整。***DBA*数据库管理员：**数据库设计、规划、协调的人员, 最高特权的用户。**DBA作用：**模式定义，存储结构及存取方法定义，数据访问权限，日常维护。***MIS*管理信息系统**。***DBMS.*数据库管理系统**：系统软件，对数据库进行统一管理和控制。**DBMS特点：**(1)用户接口，非过程数据库语言(2)查询处理与优化策略(3)并发控制，解决冲突(4)恢复功能，数据一致性状态 (5)完整性约束检查，实体完整性、引用完整性、域完整性、用户定义完整性(6)访问控制。访问权限，安全性问题。**DBMS组成：**数据定义语言***DDL***，数据操纵语言***DML***，数据库运行控制程序***DCL***，实用程序。***DAO*：** [数据访问对象](http://baike.baidu.com/view/185335.htm)是第一个[面向对象](http://baike.baidu.com/view/125370.htm)的接口。***RDO*（远程[数据对象](http://baike.baidu.com/view/178571.htm" \t "_blank)）：** 是[微软](http://baike.baidu.com/view/2353.htm)的一个[应用程序接口](http://baike.baidu.com/view/592964.htm)，让写Windows应用程序的程序师能够进入微软和其他的数据库供给者的数据库。***ODBC：***开放数据库互连。**数据管理发展的三个阶段：**人工管理阶段，文件系统阶段，数据库系统阶段。**列出文件处理系统和数据库系统的四个主要区别：**(1)两个都是由一个相关联的的数据的集合和一组数据用以访问数据的程序组成的。DBS:物理和逻辑访问数据；文件:物理访问数据；(2)DBS:被授权的程序可以访问存储的逻辑数据，数据可以被多个程序访问，减少数据的冗余；文件:一个程序所写的数据不可能被另一个程序访问；(3)DBS设计为更灵活的访问数据，文件处理系统则被设计为特定的程序访问数据；(4)DBS允许多个用户同时访问相同的数据。文件处理系统一般只允许一个或多个程序同时访问不同的数据文件。在文件处理系统中一个文件只有在两个程序都只有只读权限的时候才可以被两个程序并行访问。**数据模型四要素：**数据的结构、数据的联系、数据的操作、数据的约束。**数据模型：**概念~(ER模型、面向对象模型)，逻辑~(层次模型、网状模型、关系模型)，物理~。**数据抽象：**逻辑层、视图层、物理层。**映射程序：**外模式/模式映象：外模式和模式之间的对应关系，映象定义通常包含在各外模式中；模式/内模式映象：数据逻辑结构与存储结构之间的对应关系。**数据独立性？数据库系统如何实现数据独立性？**数据独立性是指应用程序与数据之间相互独立、互不影响，及数据结构的修改不会引起应用程序的修改数据独立性包括**物理独立性**和**逻辑独立性**。物理独立性是指数据库物理结构的变化时不必修改现有的应用程序。逻辑独立性是指数据库逻辑结构变化时不需要改变应用程序。 数据独立性是由DBMS的三级模式和二级映像来实现的。数据库系统通常采用**外模式、模式**和**内模式**三级结构，数据库管理系统在这三级模式之间提供了外模式/模式和模式/内模式两层映像。当整个系统要求改变模式时，由DBMS对各个外模式/模式映像做相应的修改，使外模式保持不变，从而使基于外模式的应用程序保持不变，从而保证了数据的逻辑独立性。当数据的存储结构改变时，由DBMS对模式/内模式映像进行修改，可以使模式保持不变，从而使应用程序也不必改变，保证了数据的物理独立性。**试述ER模型、层次模型、网状模型、关系模型和面向对象模型的主要特点：**ER模型直接表示实体类型及实体间联系，与计算机系统无关，充分反映用户的需求，用户容易理解。层次模型的数据结构为树结构，记录之间联系通过指针实现，查询较快，但DML属于过程化的，操作复杂。网状模型的数据结构为有向图，记录之间通过指针实现，查询较快，并且容易实现M:N联系，但DML属于过程化语言，编程较复杂。关系模型的数据结构为二维表格，容易为初学者理解，记录之间联系通过关键码实现DML属于非过程化语言，编程较简单。面向对象模型能完整描述现实世界的数据结构，具有丰富的表达能力，能表达嵌套、递归的数据结构，但涉及的知识面较广，用户较难理解。**关系模式：**由关系名、属性、域、属性与域之间的映象、完整性约束、属性间函数依赖义。一般只取关系名、属性名表示。**码：**能够唯一确定一个元组的最小属性集。**外码：**本关系的属性集，另一关系的码。**全码：**表中找不出码, 所有属性组成。**域完整性约束:**属性值应是域中的值，属性的值能否为null，由语义决定。**实体完整性约束：**关系的主码中的属性值不能为空。**引用完整性约束:**不同关系之间或同一关系的不同元组间的约束。若关系R中有一个外码(相对于关系S)，则R中每个元组的外码的值必须满足(1)或者取空值(2)或者等于S中某个元组的主码值。R与S可以是同一个关系**。举例说明参照完整性对数据有什么要求？**如果关系R2的外码Fk与关系R1的主码PK相对应，则R2中的每一个元组的FK值或等于R1中的某个元组的PK值或等于空值。**如何理解空值？**空值就是表示”无意义“，当实体在某个属性上没有值时设为null；或者表示”值未知“，即值存在，但目前没有获得该信息；当空值参与运算时，结果为空值。关系代数：基于集合运算表示表的操作。关系演算：基于谓词逻辑表示表的操作。**五种关系代数基本运算：**并、差、笛卡尔积、选择、投影。**运算定义式：**并:r∪s = {t|t∈r∨t∈s} 差: r-s={t|t∈r∧t ∉ s｝交:r∩s={t|t∈r∧t∈s} 笛卡尔积:r×s={tr⌒ts|tr∈r∧ts∈s} 条件连接: r∞s AθB =σAθB(r×s) 除:r÷s={tr[X]|tr∈r∧S ⊆ Yx} **将关系代数非基本运算用基本运算表示：**交、连接、除..r∩s=r-(r-s)..r÷S=ΠX(r)-ΠX(ΠX(r)×S-r)**…自然连接定义式：**r∞s=｛tr⌒ts[¬ B]|tr∈r∧ts∈S∧tr[B]=ts[B]} **关系的分类：**基本分类-连接成分：包括两个输入关系、连接条件、连接类型；连接条件：决定两个关系中哪些元组相互匹配，以及连接结果中出现哪些属性；连接类型：决定如何处理与连接条件不匹配的元组。**视图与表的区别：**表是实实在在得保存数据的实体，写入的数据都保存在表中，而视图是不保存数据的，也没有数据。表可以建立各种触发器，可以建立索引，可以建立主健、约束等。但是视图不能建立这些对象(视图可以建立替代触发器。表和视图可以更新，但是视图的更新受到约束。**视图更新：**from子句中只有一个数据库关系。Select子句中只包含关系的属性名，不包含任何表达式、聚集或distinct声明。任何没有出现在select子句中的属性可以取空值；即这些属性上没有not null约束，也不构成主码的一部分。查询中不含有group by或having子句。**断言**：定义：CREATE ASSERTION <断言名> CHECK <条件>.断言是谓词，表达数据库总应该满足的条件。**触发器：**触发器是一条语句，当对数据库做修改时，它自动被系统执行。定义：指明什么条件下触发器被执行，指明触发器执行的动作是什么。作用：警示；满足特定条件时自动执行某项任务。事件：insert、delete、update。**嵌入式SQL语句与主语言之间的通信过程：**1．数据库工作单元与源程序工作单元之间通信主要包括：(1)向主语言传递SQL语句的执行状态信息，使主语言能够据此控制程序流程(2)主语言向SQL语句提供参数(3)将SQL语句查询数据库的结果交主语言进一步处理。2．嵌入式SQL与主语言之间通信的实现：SQL语句通过SQL通信区SQLCA向主语言传递SQL执行状态信息。主语言通过自己的主变量向SQL语句输入数据参数。两者数据传送通过主变量和游标实现。3．SQL通信区：SQL语句执行后，系统将当前工作状态送到SQL通信区SQLCA中。应用程序从SQLCA中取出这些状态信息，据此决定接下来执行的语句。4．主变量：在SQL语句中使用的主语言程序的变量称为主变量。**在嵌入式SQL中什么情况下DML语句不必涉及游标操作？**insert，delete和update语句；对于select语句，如果已知查询结果肯定是单值时。**游标的作用：**游标是一种临时的数据库对象，即可以用来存放在数据库表中的数据行副本，也可以指向存储在数据库中的数据行的指针。游标提供了在逐行的基础上操作表中数据的方法。游标的一个常见用途就是保存查询结果，以便以后使用。游标的结果集是由select语句产生，当select语句的结果中包含多个元组时，使用游标可以逐个存取这些元组。**需要游标的数据操作：**当select语句的结果中包含多个元组时，使用游标可以逐个存取这些元组。**宿主语言：**将数据操作语言嵌入到某种高级语言中使用，则这种高级语言称为~。**自含语言：**可经独立使用且不依赖于其他任何程序设计语言的数据操纵语言称为~。**属性继承：**高层实体集的属性被低层实体集自动继承；低层实体集特有的性质仅适用于某个特定的低层实体集.**弱实体集：**如果一个实体集的所有属性都不足以形成主码，则称这样的实体集为弱实体集。**为什么使用弱实体集？**1.避免数据冗余（强实体集码重复），以及因此带来的数据的不一致性2.弱实体集反映了一个实体对其它实体依赖的逻辑结构3.弱实体集可以随它们的强实体集的删除而自动删除4.弱实体集可以物理地随它们的强实体集存储。**弱实体集与存在依赖：**弱实体集必然存在依赖于强实体集，存在依赖并不总会导致一个弱实体集，从属实体集可以有自己的主码。**分辨符：**弱实体集中用于区别依赖于某个特定强实体集的属性集合，也称作部分码。**概括与特殊化：**概括与特殊化是个互逆的过程，在E-R图中的表示方法是相同的；特殊化强调同一实体集内不同实体之间的差异，概括强调不同实体集之间的相似性；反映了数据库设计的不同方法。**四种异常现象：**冗余度、插入异常、删除异常、更新异常。**函数依赖：**设R=(U)表示一个关系模式，属性集合U={A1,A2,…,An}，X和Y是U的子集，对R=(U)下的任一个关系r的任意两个元组t1,t2，如果t1[X]=t2[X]，则t1[Y]=t2[Y]，称属性集X与属性集Y存在函数依赖，记作X→Y。**逻辑蕴涵：**设F是关系模式R的一个函数依赖集，X、Y是R的属性子集，如果从F中的函数依赖能够推出X→Y，则称F逻辑蕴涵X→Y。**候选码：**K→U,K完全决定U,候选码的任何真子集都不函数决定U。**超码：**K’→U,K’部分决定U,K’真子集函数决定U。**全码：** R=(U)中不存在任何依赖,所有属性组成全码。**第二范式：**若关系模式R∈1NF并且每个非主属性都完全函数依赖于R的码，则R∈2NF。**Armstrong公理：自反律**如果 Y ⊆ X ⊆ U，则F 蕴涵X→Y**增补律**如果X→Y，且Z ⊆ U，则XZ→YZ成立**传递律**如果X→Y和Y→Z，则X→Z成立**合并规则**如果X→Y，X→Z，则有X→YZ**分解规则**如果X→Y，且 Z ⊆ Y ，则X→Z成立**伪传递规则**如果X→Y，WY→Z，则XW→Z**函数依赖集的闭包：**关系模式R=(U)中，F及F所蕴涵的全体函数依赖称为F的闭包，记为F+。**第三范式：**关系模式R=(U)的每个非主属性都不部分依赖也不传递依赖于码，则称R满足第三范式，记为R∈3NF ***BCNF：***如果关系模式R=(U)的所有非平凡函数依赖α→β，α部分都包含R的一个候选码，则称R属于BC范式，记作R∈BCNF。**3NF与BCNF的差别：**3NF 要消除的是非主属性对码的部分依赖和传递依赖，而BCNF要消除的是主属性对码的部分依赖或传递依赖。如果R∈3NF，检查所有决定因素是否都包含候选码，若是，R∈BCNF。 **正则覆盖：**满足下列条件的函数依赖集F称为正则覆盖，记作Fc: (1)Fc与F等价(2)Fc中任何函数依赖都不含无关属性(3)Fc中函数依赖的左半部都是唯一的。**简述函数依赖与多值依赖的联系与区别：**函数依赖规定某些元组不能出现在关系中，也称为相等产生依赖；多值依赖要求某种形式的其他元组必须在关系中，称为元组产生依赖。X→Y的有效性仅决定于X、Y属性集上的值；X→→Y的有效性与属性范围有关。**全部性约束：**确定高层实体集中的一个实体是否必须属于某个概括的至少一个低层实体集. **表级继承：**允许对象有多重类型，而不必象类型继承那样要求具有一个最明确类型.**事务.ACID特性：**事务是访问并可能更新各种数据项的一个程序执行单元。事务具有ACID四种特性，A原子性：事务中包含的所有操作要么全做，要么全不做；C一致性：事务的隔离执行必须保证数据库的一致性；I隔离性：系统必须保证事务不受其他并发执行事务的影响；D持久性：一个事务一旦提交之后，它对数据库的影响必须是永久的。**锁的概念：**事务T对某个数据对象操作之前，先向系统发出请求，对其加锁。加锁成功后，事务T开始对该数据对象进行操作，在事务T释放它的锁之前，其它的事务不能更新此数据对象。加锁不成功，事务T等待。**死锁发生是好事还是坏事？如何解除死锁状态**？在DBS运行时，死锁状态是我们不希望发生的，因此死锁的发生本身是一件坏事。但是坏事可以转换为好事。如果我们不让死锁发生，让事务任意并发做下去，那么有可能破坏DB中数据，或用户读了错误的数据。从这个意义上讲，死锁的发生是一件好事，能防止错误的发生。 在发生死锁后，系统的死锁处理机制和恢复程序就能起作用，抽取某个事务作为牺牲品，把它撤销，做rollback操作，使系统有可能摆脱死锁状态，继续进行下去。**一级封锁协议：**事务T在修改数据Q之前必须先对其加X锁，直到事务结束才释放。**二级封锁协议：**事务T在修改数据Q之前必须先对其加X锁，直到事务结束才释放。事务T在读取数据Q之前必须先对其加S锁，读完后即释放S锁。二级封锁协议可防止丢失修改，防止读“脏”数据，但不能保证可重复读。**三级封锁协议：**事务在修改数据Q之前必须先对其加X锁，直到事务结束才释放。事务在读取数据Q之前必须先对其加S锁，直到事务结束才释放S锁。三级封锁协议可防止丢失修改，防止读“脏”数据，保证可重复读。**两阶段封锁协议：**对任何数据读写之前，先要获得对该数据的封锁，释放一个封锁之后，事务不再加任何锁。两段：增长阶段，申请加锁阶段；缩减阶段，释放锁阶段。事务开始为增长阶段，一旦释放锁，进入缩减阶段。**解决活锁：**先来者先执行。**死锁：**两个或两个以上的事务都在等待其中另一个事务解除封锁，它才能继续执行下去，结果任何一个事务都无法继续执行。**活锁：**封锁技术可能产生某个事务可能永远处于等待状态，得不到执行的机会，这种现象称为活锁。**死锁发生是好事还是坏事？如何解除死锁状态**？在DBS运行时，死锁状态是我们不希望发生的，因此死锁的发生本身是一件坏事。但是坏事可以转换为好事。如果我们不让死锁发生，让事务任意并发做下去，那么有可能破坏DB中数据，或用户读了错误的数据。从这个意义上讲，死锁的发生是一件好事，能防止错误的发生。 在发生死锁后，系统的死锁处理机制和恢复程序就能起作用，抽取某个事务作为牺牲品，把它撤销，做rollback操作，使系统有可能摆脱死锁状态，继续进行下去。**时间戳定义：**事务启动时间。事务Ti的时间戳记为 TS（Ti）。**时间戳协议：**Ti读数据Q时，(1)若TS(Ti)<W-TS(Q)，撤消Ti并重新启动 (2)若TS(Ti)>W-TS(Q)，执行读操作，R-TS(Q)= MAX(R-TS(Q),TS(Ti))；Ti写数据Q时(1)若TS(Ti)＜ R-TS（Q）,撤消Ti并重新启动 (2)若TS(Ti)＜W-TS(Q)，撤消Ti并重新启动(3) 其它情况，执行写操作，W-TS(Q)=TS(Ti)。**数据仓库：**是一个将从多个数据源中收集来的信息以统一模式存储在单个站点上的仓储(或归档)。一旦收集完毕，数据会存储很长时间，允许访问历史数据。**数据挖掘：**数据挖掘就是应用一系列技术从大型数据库或数据仓库中提取人们感兴趣的信息和知识，这些知识或信息是隐含的，事先未知而潜在有用的，提取的知识表示为概念、规则、规律、模式等形式。数据挖掘是一类深层次的数据分析。**数据挖掘的方法和技术：**信息论方法、集合论方法、人工神经网络方法、遗传算法、统计分析方法、文本挖掘、可视化技术。**数据关联：**是数据库中存在的一类重要的可被发现的知识，若两个或多个变量间存在着某种规律性，就称为关联。关联分析的目的就是找出数据库中隐藏的关联网。**关联规则：**关联规则是如下形式的逻辑蕴涵：A⇒ B，其中A，B是项集，A∈I，B∈I，A ∩ B=Ф。**分类的主要算法：**(1)决策树算法(2)贝叶斯分类算法(3)神经网络分类算法(4)遗传算法(5)粗糙集分类算法。***Apriori*算法（找频繁项集算法）：**首先产生频繁1-项集L1，然后是频繁2-项集L2，… Lr， 直到某个r使得Lr为空，这时算法停止。在第k次循环中， Lĸ-1进行自身连接， 通过Apriori的性质进行剪枝，产生候选K项集Ck，再对Ck中的每个元素按支持度选取，产生频繁项集Lĸ。**文本挖掘：**文本挖掘有时也被称为[文字探勘](http://zh.wikipedia.org/wiki/%E6%96%87%E5%AD%97%E6%8E%A2%E5%8B%98)、文本[数据挖掘](http://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98)等，大致相当于[文字分析](http://zh.wikipedia.org/wiki/%E6%96%87%E5%AD%97%E5%88%86%E6%9E%90)，一般指文本处理过程中产生高质量的[信息](http://zh.wikipedia.org/wiki/%E4%BF%A1%E6%81%AF)。**查询选修且仅选修数据库和数据结构的学生姓名**Π姓名(学生∞(Π学号，课程号（选课）÷Π课程号（σ课程名=“数据库”∨ 课程名=“数据结构”（课程））-Π学号(σ课程名<>“数据库”∧课程名<>“数据结构”（课程∞选课）)))**3NF ⊂ 2NF** 反证:若R∈3NF,但R∉2NF,则按2NF定义,一定有非主属性部分依赖于码**.**设X为R的码,则存在X的真子集X′,以及非主属性Z(ZX’),使得X’→Z**.**于是在R中存在码X,属性组X’,以及非主属性Z(ZX’),使得X→X’, X’→Z,X’→X成立,这与R∈3NF矛盾.所以R∈2NF**BCNF ⊂ 3NF** 反证:若R∈BCNF,但R∉3NF,则按3NF定义,一定有非主属性对码的传递依赖,于是存在:R的码X,属性组Y,以及非主属性Z（ZY）,使得X→Y ,Y→ Z,Y→X成立.由Y→Z，按BCNF定义,Y含有码,于是Y→X成立,这与Y→X矛盾. 所以R∈3NF.**4NF ⊂ BCNF** 证明:假设R属于4NF,但不属于BCNF,那么有X→A存在且X不是超码,如果XA=U,显然X是超码;如果XA<>U, 根据X→A成立,有X→→A成立,此时X不是超码,违反了4NF条件.因此R必是BCNF.