all，unique，not in ，exists，Insert into values (, , ,)，is null

asc,desc

右端只有一个属性，左端不能有多余属性，不能有多余的函数依赖

名词解释和简答

数据

- 描述事物的符号记录。

事务的特性（3）

- 事务是用户定义的一个数据库操作序列，是不可分割的工作单位，具有ACID特性。
- A原子性：事务中包括的所有操作要么都做，要么都不做。
- C一致性：事务必须使数据库从一个一致性状态变到另一个一致性状态。
- I隔离性：一个事务内部的操作及使用的数据对并发的其他事务是隔离的。
- D持续性：事务一旦提交，对数据库的改变是永久的。

数据库并发操作通常会带来哪些问题？（2）

- 丢失修改
  - T1修改，T2又修改，于是把T1的**修改覆盖了**
- 不可重复读
  - T1读取数据后，T2执行更新操作，于是T1**读了旧数据**
- 读“脏”数据
  - T2修改数据，T1读，但是T2被撤销，于是T1**读了被撤回的数据**

正则覆盖(2)

- 满足下列条件的函数依赖集F称为正则覆盖，记作Fc：
- `后件是单属性` F中的函数依赖都是形如 $X \to A$的（即右边都只有一个属性）（注意，A是Attribute，只是一个属性，但是X可以是集合）。
- `没有多余函数依赖` F中不存在多余的函数依赖，凡是只要去掉一个，F^+就变了（该函数依赖集和原来不再等价了）。
- `没有多余属性` F中不存在多余的属性，凡是只要去掉一个（当然同时也要去掉其关联的函数依赖），F^+就变了（该函数依赖集和原来不再等价了）。

什么是饥饿现象

死锁及其解除

- 在发生死锁后，撤消某个事务，**回滚**，于是有可能打破死锁。

数据模型的三大要素

- 数据操作
- 数据结构
- 完整性约束

嵌入式SQL，什么情况下DML语句不涉及游标操作

- 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。

各模型：概念数据模型（例如ER）、逻辑数据模型（层次模型、网状模型、关系模型、面向对象模型）的主要特点（2）

- 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**联系**通过**指针实现**，查询较快，并且容易实现M:N 联系，但 DML 属于**过程化的**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

模型	数据结构	属性如何联系	是否过程化
层次模型	树	指针	过程化
网状模型	图	指针	过程化
关系模型	表	码	非过程化

人工管理阶段、文件系统阶段、数据库系统阶段

- 人工管理阶段：没有相应的文件系统，数据不具独立性，不容易共享
- 文件系统阶段：不支持并发访问，管理不统一
- 数据库系统阶段：数据安全性、并发访问性、并由DBMS统一管理

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。每个弱实体集必须与另一个称作标识或主实体集的实体集关联才能有意义。

完整性约束（2）

- 实体完整性：关系只有一个主码，且互异、不为空
- 参照完整性：取空值（F的每个属性值均为空值）或者等于S中某个元组的主码值。
- 用户定义完整性（域完整性）：类比强类型语言，即字段必须满足某种特定的数据类型或约束 CHECK、FOREIGN KEY 约束和DEFAULT、NOT NULL
- ```
  CREATE DOMAIN AGE SMALLINT
         CHECK((VALUE >= 15) AND VALUE <= 25))
  ```

空值的定义和运算（3）

- 表示“无意义”，或当前暂时“值未知”

数据库设计分哪几个阶段?

- 需求分析
- 概念结构设计
- 逻辑结构设计
- 物理结构设计
- 数据库实施
- 数据库运行和维护

可串行化调度和串行调度

- 串行调度：多个事务依次执行
- 可串行化：如果并发调度的结果与某一串行调度执行结果等价，则称**这个并发调度**是可串行化调度。

RAID

- 存取性能提高，并行传输
- 数据备份，更加安全
- 位级、块级拆分

索引

- 索引是一个单独的、物理的结构~~，它包括某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。~~
- 索引提供指向存储在表的指定列中的数据值的指针，然后根据指定的排序顺序对这些指针排序。
- 好处
  - 通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性
  - 加快操作速度
- 分类
  - 顺序索引和散列索引：如何给指针排位置？前者基于值的顺序排序，后者算哈希从而指定位置。
  - 聚集索引与非聚集索引：数据表的物理顺序与索引指定的顺序，前者相同，后者不同。
  - 稠密索引与稀疏索引：前者每个搜索码的值都有一个索引项，后者只为搜索码的某些值建立索引项。因此，只有使用聚集索引时才可使用稀疏索引。
  ```

- 主索引和辅助索引
  - 主索引
    - 将主文件分块，每一块对应一个索引项。每个存储快的第一条记录，又称为锚记录。
    - 主索引是有序文件。
    - 主索引是稀疏索引。
  - 辅助索引
    - 通常以层级的方式出现。即单值的一级索引和多值的二级索引（即中间桶）
    - 于是可以对字段（该字段非排序）的每一个不同值有一个索引项。（于是就可以通过引入副主索引的方法实现多值索引。）
  - 一个主文件仅有一个主索引，但可以有多个辅助索引；
  - 主索引通常建立**在主码/排序码**上面；
  - 可以利用主索引重新组织主文件数据，辅助索引不可以；

函数依赖和多值依赖的联系和区别

- 函数依赖：设X,Y是关系R的两个属性集合，当任何时刻R中的任意两个元组中的X属性值相同时，则它们的Y属性值也相同，则称X函数决定Y，或Y函数依赖于X。
- 函数依赖规定某些元组**不能出现**在关系中，多值依赖要求某种形式的其它元组**必须出现**在关系中。

数据库模型和实例

超码和候选码的异同

两阶段封锁协议（3）

- 两段锁协议要求每个事物分成两个阶段提出加锁和解锁申请。
  - 增长阶段：事物可以获得封锁，不能释放锁（只能获得）
  - 缩减阶段：事物可以释放锁，但不能获得新锁（只能释放）

一、二、三级封锁协议

- 一级：事务T在修改数据R之前必须先对其加X锁，直到事务结束才释放。
- 二级：一级封锁协议加上事务在读取数据R之前必须先对其加S锁，**读完后即可**释放S锁。二级封锁不仅可以解决“丢失修改”问题，而且可以解决读“脏“数据问题。
- 三级：一级封锁协议加上事务在读取数据R之前必须先对其加S锁，**直到事务结束**才释放。三级封锁协议不仅解决了“丢失修改”、读“脏”数据问题，而且进一步解决了“不可重复读”问题。

哪类视图是不可以更新的?

视图与表的区别(2)

- 视图是从一个或几个基本表导出的表，它与基本表不同，是一个虚表，数据库中只存放视图的定义。
- 一经定义就可以像基本表一样被操作，但是有一定的限制。
- 视图使用户能以多种角度看待同一数据，能够对机密数据提供安全保护。（1分）
- 视图对重构数据库提供了一定程度的逻辑独立性。（1分）

数据库系统的故障有哪些情况?

- 事务故障
- 系统故障
- 介质故障
- 前两类故障未破坏DB，但使其中某些数据变得不正确，可以通过故障恢复策略解决。但是介质故障破坏了DB，只能启用镜像备份磁盘，然后跑恢复策略解决。

数据库从故障中的恢复策略?

- 日志文件中**寄了前**已经提交的**事务**，将其事务标识记入**REDO**队列。
- 日志文件中**寄了时**尚未完成的**事务**，将其事务标识记入**UNDO**队列。
- 逐个撤销UNDO队列中的事务，逐个重做REDO队列中的事务。

DBS如何保证数据存储的稳定性?

- 通过数据备份和数据银行形式实现。

什么是对象关系模型

- 数据类型允许复合类型、引用类型。
- 类型和表都是可继承的。
- 查询时，属性可能是多值的。

有损分解有什么后果?

如果分解有损，即分解后的模式不能有效表示泛关系原来含有的全部信息，那么**分解没有价值**，有损分解是不可实际接受的

模式分解的重要评价指标?

- 无损分解和保持依赖

关系模式冗余?

- 数据冗余
- 插入异常
- 删除异常
- 这是因为两件事不应使用一个表来表示，需要进行模式分解。

实体、实体型、实体集、联系集?

- **客观存在**并可**相互区分的**事物叫实体
- 同型实体的集合称为实体集
- 实体名和属性名组成实体型
- 参与联系**的实体**组成联系集，可用参与联系实体的主码和联系上的属性构成属性，从而建表来表示

弱实体集在E-R图中如何表示?

- **双边框矩形表示**弱实体集。
- 弱实体集与其拥有者之间的联系称作标识性联系（**双边框菱形**表示）。
- 弱实体集必然存在依赖于强实体集（存在依赖即一对多）。
- 分辨符（也叫部分码），用**下划虚线**表示。
- 从联系集用**双线**（全部参与）连接弱实体集，用**箭头**（一对多联系）指向强实体集。

关系中的元组可以重复吗? 可以有先后顺序吗? 为什么? (2)

数据库管理员 (DBA) 的职责?

- 模式定义
- 模式、物理修改: 对逻辑结构和物理结构的修改
- 访问授权
- 日常维护

自然连接和等值连接的区别

- 自然连接是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列**去掉**。

物理独立性与逻辑独立性 (2)

- 数据独立性是指**应用程序和数据之间**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。
- 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

数据库三级模式两层映像结构是如何实现数据独立性的 (2)

- **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 (视图层)**、**模式 (逻辑层)** 和**内模式 (物理层)** 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。
    - 外模式: 用户看到的数据形式，例如视图;
    - 模式: DBA看到的数据形式，例如表;
    - 内模式: 磁盘上的存储形式。
    - 映射程序: 外模式/模式映象和模式/内模式映象。定义了相邻层之间的接口。
- 逻辑独立性的实现: 当整个系统要求改变模式时，由DBMS对各个外模式/模式映像作相应的修改，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。
- 物理独立性的实现: 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。

动态SQL和嵌入式SQL

- 动态sql: 通用程序设计语言可以通过函数或者方法来连接数据库服务器并与之交互。利用动态sql可以在运行时以字符串形式构建sql查询，提交产寻，然后把结果存入程序变量中，每次一个元组。动态sql的sql组件允许程序在运行时构建和提交sql查询。
- 嵌入式sql: 提供了另外一种使程序与数据库服务器交互的手段。嵌入式SQL必须在编译时全部确定，并交给预处理器。预处理程序提交sql语句到数据库系统进行变异和优化，然后将sql语句替换成相应代码和函数，最后调用程序语言的编译器进行编译

DB、DBS和DBMS (2)

- DBMS是位于用户与操作系统之间的具有数据定义、数据操纵、数据库的运行管理、数据库的建立和维护功能的一层数据管理软件。
- DBS是在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员和用户构成。
- DB是长期存储在计算机内的、有组织的、可共享的数据集合。

DBMS的功能

- 数据定义功能（DDL语言，例如表定义）
- 数据操纵功能（DML语言，例如SELECT）
- 数据库的运行管理：它包括并发控制，安全性检查，完整性约束条件的检查和执行，发生故障后的恢复等
- 数据库的建立和维护功能：初始数据的输入及转换，数据库的转储与恢复，数据库的重组功能和性能的监视与分析功能等

DBMS的组成

- 由三大语言组成。
- DDL
- DML（包括过程化、非过程化）
- DCL

DBMS的功能部件

- 查询处理器（DML编译器、DDL解释器、查询求值引擎）
- 存储管理器（权限及完整性管理器、文件管理器、缓冲管理器）

登记日志文件时必须遵循什么原则

- 登记的次序严格按并发事务执行的时间次序。（1分）
- 必须先写日志文件，后写数据库。（2分）

英文解释

- **MIS（Management Information System，管理信息系统）**是一个由人、计算机等组成的，能进行信息的收集、传递、储存、加工、维护和使用的系统。
- DAO (Data Access Objects 数据存取对象)是指位于业务逻辑和持久化数据之间实现对持久化数据的访问。通俗来讲，就是将数据库操作都封装起来。
- **RDO（Remote Data Objects）**远程数据对象是一个到ODBC的、面向对象的数据访问接口。
- **ODBC(Open Database Connectivity，开放数据库互连)**定义了一个API，应用程序用它来打开一个数据库连接，发送查询和更新，以及获取返回结果。
- JDBC：定义了Java程序连接数据库服务器的应用程序接口

数据库设计

判断3NF和BCNF，分解为3NF和BCNF（7）

保持依赖分解，无损连接分解（2）

证明3NF一定是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，使得X'→Z。于是在R中存在码X，属性组X'，以及非主属性Z，使得X→X'，X'→Z，X'→X成立，这与R∈3NF矛盾。 所以R∈2NF。

证明BCNF是3NF

反证：设R∈BCNF，但R∉3NF，则必然存在如下条件的函数依赖 X→Y（Y-/>X），Y→Z，其中X是键属性，Y是任意属性组，Z是非主属性。其中，Y→Z函数依赖的前件Y不是候选码，与BCNF范式的定义相矛盾， 所以R∈3NF。

证明4NF是BCNF

反证：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。

找候选码（4）

求正则覆盖（又称最小覆盖）

1、把右部分化为单属性，即对后键进行拆分，A->BC变成A->B，A->C。

2、去掉左部分的冗余属性，求前件的属性集闭包
例如XY→A，假设Y是多余的，看A是否属于(X)+，若是，则Y是多余属性，可以去掉。

3、去掉冗余的函数依赖，求前件的属性集闭包
从第一个函数依赖X→Y开始将其从F中去掉，然后在剩下的函数依赖中求X的闭包，看(X)+是否包含Y，若是，则去掉X→Y；否则不能去掉，依次做下去。

找无关属性

去掉后，求前件的属性集闭包，看是否和原来一样

求属性集闭包（2）

设关系模式R(A，B，C)上有一个多值依赖A→→B。如果已知R的当前关系中存在着三个元组(a,b1,c1)、(a,b2,c2)、(a,b3,c3)，那么这个关系中至少还应该存在哪些元组?

(a,b1,c2)(a,b1,c3)(a,b2,c1)(a,b2,c3)(a,b3,c1)(a,b3,c2)

（a b1 c2），(a b2 c1)，(a b1 c3)，(a b3 c1)，(a b2 c3)，(a b3 c2)

SQL、关系代数、关系演算

查询：select

（1）读取FROM子句的表或视图做笛卡尔积。

（2）WHERE子句找出满足条件表达式的元组

（3）按GROUP子句指定列名分组，值相等的元组为一组，每个组产生结果表中的一条记录。可在每组中用集函数。如果GROUP子句带HAVING短语，则只有满足指定条件的组才予输出。

（4）按select子句中给出的列名或表达式求值输出

（5）按ORDER子句列名的值升或降序

缺省为保留重复元组，也可用关键字all显式指明。若要去掉重复元组，可用关键字distinct或unique指明。

======

找出满足条件的字符串：列名 like “字符串”

%”匹配零个或多个字符

“_”匹配任意单个字符

=====

order by列名 [asc | desc]

=====

group by后面出现的那个属性也必须在select后面出现，否则报错。

Having是对已经分组处理的结果进行条件选择，而不是对原始数据选择后再分组。

Distinct 去掉重复，

先where,再 group，再 having

聚集函数忽略null，count（*）除外

=====

空值测试：

空值用is判断，where amount is null，不能用“=”。

找出姓名是空的老师：select PNAME from PROF where PAGE is null

=====

空集测试：

测试集合是否为空：exists，表示“某集合是否存在”

in后的子查询与外层查询无关，每个子查询执行一次，而exists后的子查询与外层查询有关，需要执行多次，称之为相关子查询。

用exists表示超集：若A为B的超集，则NOT EXISTS(B EXCEPT A)为TRUE

用exists表示“全称量词”：not exists(not exists)，不存在没有，例如不存在一门课该同学没有选过：

select SNAME from S where not exists (select CNO from C where not exists (select * from SC where SC.CNO=C.CNO and SC.SNO=S.SNO))

=====

重复测试：

unique（子查询）如果子查询结果中没有重复元组，则返回true

Eg：找出只教授一门课程的老师姓名：

select PNAME from PROF where unique(selec PNO from PC where PC.PNO=PROF.PNO)

（如果是“至少两门”就是not unique即可）

=====

集合并：union

集合交：intersect

集合差：except(minus)、

集合自动去除重复元组，如果需要保留，要用all显式说明。

```
=====
```

男生按姓名降序排列（2）

某门课成绩最好的同学的姓名（2）

同学所选课程总学分数，不包括不及格

没选c1课程的学生（7）（SQL，代数，演算）

至少选了学生s所选的所有课程的学生（5）（SQL，代数，演算）

所有同学都选了的课程

设每个职工可在多个公司工作，检索每个职工的兼职公司数目

和工资总数

列出在不同的商场同一种商品的最高售价和最低售价超过100元的商品

有两门以上优秀的学生

检索联华公司中低于本公司平均工资的职工工号和姓名

所有学生的成绩（SQL，代数）

修改：把没达到课程平均成绩的同学的成绩提高5%（5）

`Update 表名 set … =… where …`

删除：删除年龄大于 60 岁的职工有关元组

`Delete from 表名 where …`

插入：用SQL语句将（S2，P4，J6，400）插入关系中

`Insert into 表名 values ( , , , )`

建表：建立“供应商”S表(主码必须定义)

```
CREATE TABLE S
    (Sno CHAR(6) PRIMARY KEY,
    Sname CHAR(10),
    Status INT,
    City CHAR(20));
```

视图：建立一个有关女车间主任的职工号和姓名的视图，其结构如下：

NOANDNAME（ENO，ENAME）

```
CREATE VIEW NOANDNAME AS SELECT ...
```

事务

时间戳调度

视图可串行化调度

冲突可串行化调度（3）