

数据库系统作业

姓名	学号
苏家明	15352296

chapter 1

Q1:举出适合用文件系统而不是数据库系统的例子；再举出适合用数据库系统的应用例子。

要回答这两个问题，我们首先得说出文件系统和数据库系统两者的差异：

- ① 文件系统用文件将数据长期保存在外存上，数据库系统用数据库统一存储数据；
- ② 文件系统程序和数据有一定的联系，数据库系统中的程序和数据分离；
- ③ 文件系统用操作系统中的存取方法对数据进行管理，数据库系统用DBMS统一管理和控制数据；
- ④ 文件系统实现以文件为单位的数据共享，数据库系统实现以记录和字段为单位的数据共享。

因此数据库系统比文件系统至少应该多出高级用户接口、并发控制和访问控制三项功能。综上，数据的备份和软件（系统）产生的临时数据比较适合使用文件系统，另外简单且固定的系统也适合使用文件系统，因为他们的数据结构相对简单而且一般不会面临多用户并发访问。

至于适合数据库系统的例子：现代的信息系统如人事管理系统、库存管理系统、作业调度系统、设备管理系统等都适合用数据库系统，因为面临多用户访问比较多，而且数据结构比较复杂，对数据的操作（尤其是修改操作）比较复杂，因此适合使用数据库管理系统。

Q2:数据库管理系统的主要功能有哪些？

- 1. 数据库定义功能
- 2. 数据存取功能
- 3. 数据库运行管理
- 4. 数据库的建立和维护功能

Q3：试述数据模型的概念、数据模型的作用和数据模型的三个要素。

- 概念：数据模型也是一种模型，它是现实世界数据特征的抽象。
- 作用：在数据库中用数据模型来抽象、表示和处理现实世界中的数据和信息
- 三要素：数据结构、数据操作、完整性约束

教材1.9：定义并解释概念模型中一下术语：

- 实体：客观存在并可相互区分的事物
- 实体集：同一类型的实体的集合
- 实体型：具有相同属性的实体必然具有相同的特征和属性，用实体名及其属性集合来抽象刻画同类实体，称为实体型
- 实体联系：不同实体集之间的联系（一对一、一对多等）
- 关系：一个关系通常对应一张表
- 属性：关系（表）中的一列，给一个属性起名（列名）为属性名
- 码：某个属性组，可以唯一确定一个元组
- 域：具有相同数据类型的数据集合
- 元组：关系（表）中的一行
- 分量：元组中的一个属性
- 关系模式：对关系的描述，一般由关系名和若干属性组成

chapter 2

Q:试述关系模型的完整性规则。在参照完整性中，为什么外部码属性的值也可以为空？什么情况下才可以为空？

实体完整性：属性A是基本关系R的主属性，则属性A不能取空值。若属性（或属性组）F是基本关系的外码，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上的值必须为：，或者取空值（F的每个属性均为空值）；或者等于S中某个元组的主码值。总的来说，即属性F本身不是主属性，则可以取空值，否则不能取空值。

Q：简述等值连接与自然连接的区别和联系。

连接运算符是“=”的连接运算称为等值连接，它是从关系R与S的广义笛卡尔积中选取A、B属性值相等的那些元组。

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。

Q:试述数据库系统三级模式结构，这种结构的优点是什么？

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成。数据库系统的三级模式是对数据的三个抽象级别，它把数据的具体组织留给DBMS管理，使用户能逻辑地抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方

式。为了能够在内部实现这三个抽象层次的联系和转换，数据库管理系统在这三级模式之间提供了两层映像：外模式／模式映像和模式／内模式映像，正是这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

Q:什么叫数据与程序的物理独立性?什么叫数据与程序的逻辑独立性?为什么数据库系统具有数据与程序的独立性。

数据库的存储结构改变了(例如选用了另一种存储结构)，由数据库管理员对模式／内模式映像作相应改变，可以使模式保持不变，从而应用程序也不必改变。保证了程序与数据的物理独立性,简称数据的物理独立性。

当需要改变模式时(例如增加新的关系、新的属性、改变属性的数据类型、改变数据间的联系等)，由数据库管理员对各个外模式／模式的映像作相应改变，而使外模式保持不变，从而不必修改或重写应用程序。而应用程序是依据数据的外模式编写的，保证了数据与程序的逻辑独立性。简称数据的逻辑独立性。

特定的应用程序是在外模式描述的数据结构上编制的，它依赖于特定的外模式，与数据库的模式和存储结构独立。不同的应用程序有时可以共用同一个外模式。数据库的二级映像保证了数据库外模式的稳定性，从而从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改。

Q：比分析C/S，B/S体系结构的优缺点

C/S结构：

- 优点：充分利用两端硬件环境的优势，发挥了客户端的处理能力，很多工作可以在客户端处理后再提交给服务器，可以有效降低系统的通信开销。
C/S结构的缺点是只适用于局域网，客户端需要安装专用的客户端软件，升级维护不方便，并且对客户端的操作系统一般也会有一定限制。
- 缺点：只适用于局域网，客户端需要安装专用的客户端软件，升级维护不方便，并且对客户端的操作系统一般也会有一定限制。

B/S结构：

- 优点：简化了客户端，客户端只要安装通用的浏览器软件即可。因此，只要有一台能上网的计算机就可以在任何地方进行操作而不用安装专门的客户应用软件。这样不但可以节省客户机的硬盘空间与内存，实现客户端零维护，而且使系统的扩展非常容易。其次是简化了系统的开发和维护。系统的开发者无须再为不同级别的用户设计开发不同的应用程序，只需把所有的功能都实现在应用服务器(Web服务器)上，并就不同的功能为各个级别的用户设置权限就可以了。
- 缺点：应用服务器(Web服务器)端处理了系统的绝大部分事务逻辑，从而造成应用服务器运行负荷较重。其次是客户端浏览器功能简单，许多功能不能实现或实现起来比较困难。例如，通过浏览器进行大量的数据输入就比较困难和不便。

Exercise 6.15

- $\{t \mid \exists s \in \text{works } (t[\text{person_name}] = s[\text{person_name}] \wedge s[\text{company_name}] = \text{"First Bank Corporation"})\}$
- $\{t \mid \exists r \in \text{employee } \exists s \in \text{works } (t[\text{person_name}] = r[\text{person_name}] \wedge t[\text{city}] = r[\text{city}] \wedge r[\text{person_name}] = s[\text{person_name}] \wedge s[\text{company_name}] = \text{"First Bank Corporation"})\}$
- $\{t \mid t \in \text{employee} \wedge (\exists s \in \text{works } (s[\text{person_name}] = t[\text{person_name}] \wedge s[\text{company_name}] = \text{"First Bank Corporation"} \wedge s[\text{salary}] > 10000)))\}$
- $\{t \mid \exists e \in \text{employee } \exists w \in \text{works } \exists c \in \text{company } (t[\text{person_name}] = e[\text{person_name}] \wedge e[\text{person_name}] = w[\text{person_name}] \wedge w[\text{company_name}] = c[\text{company_name}] \wedge e[\text{city}] = c[\text{city}])\}$
- $\{t \mid \exists l \in \text{employee } \exists m \in \text{manages } \exists r \in \text{employee } (l[\text{person_name}] = m[\text{person_name}] \wedge m[\text{manager_name}] = r[\text{person_name}] \wedge l[\text{street}] = r[\text{street}] \wedge l[\text{city}] = r[\text{city}] \wedge t[\text{person_name}] = l[\text{person_name}])\}$
- $\{t \mid \exists w \in \text{works } (w[\text{company_name}] \neq \text{"First Bank Corporation"} \wedge t[\text{person_name}] = w[\text{person_name}])\}$
 $\{t \mid \exists e \in \text{employee } (t[\text{person_name}] = e[\text{person_name}] \wedge \neg \exists w \in \text{works } (w[\text{company_name}] = \text{"First Bank Corporation"} \wedge w[\text{person_name}] = t[\text{person_name}]))\}$

$(w[\text{company_name}] = \text{"First Bank Corporation"} \wedge w[\text{person_name}] = t[\text{person_name}]))\}$

- $\{t \mid \exists w \in \text{works } (t[\text{person_name}] = w[\text{person_name}] \wedge \forall s \in \text{works } (s[\text{company_name}] = \text{"Small Bank Corporation"} \Rightarrow w[\text{salary}] > s[\text{salary}]))\}$
- $\{t \mid \forall s \in \text{company } (s[\text{company_name}] = \text{"Small Bank Corporation"} \Rightarrow \exists r \in \text{company } (t[\text{company_name}] = r[\text{company_name}] \wedge r[\text{city}] = s[\text{city}]))\}$