简答题主要涉及知识点

**第一章**

1. **数据库相关四个概念：DATA，Database，DBMS，DBS**

DATA：是数据库中存储的基本对象，是描述事物的符号记录。

Database：长期储存在计算机内、有组织的、可共享的大量数据的集合。

DBMS：位于用户与操作系统之间的一层数据管理软件，是系统软件。

DBS：由数据库、数据库管理系统（及其应用开发工具）、应用程序和数据库管理员组成的存储、管理、处理和维护数据的系统。

1. **数据管理技术发展的三个阶段及其各自特点**

人工管理阶段：数据不保存，应用程序管理数据，数据不共享，数据不具有独立性。

文件系统阶段：数据可以长期保存，由文件系统管理数据，数据共享性差、冗余度大，数据独立性差。

数据库系统阶段：数据结构化，数据的共享性高、冗余度低且易扩充，数据独立性高，数据由数据库管理系统统一管理和控制。

1. **数据库系统的特点**
2. 数据结构化
3. 数据的共享性高、冗余度低且易扩充
4. 数据独立性高
5. 数据由数据库管理系统统一管理和控制
6. **数据库的特征**
7. 数据按一定的数据模型组织、描述和储存。
8. 冗余度最小。
9. 数据独立性较高。
10. 易扩展。
11. 可为各种用户共享。
12. **数据独立性的概念**
13. 数据独立性包括数据的物理独立性和逻辑独立性。
14. 物理独立性是指用户的应用程序与数据库中数据的物理存储是相互独立的，当数据的物理存储改变时应用程序不用改变。
15. 逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，数据的逻辑结构改变时应用程序不用改变。
16. **数据库管理系统的功能**
17. 数据定义功能
18. 数据组织、存储和管理
19. 数据操纵功能
20. 数据库的事务管理和运行管理
21. 数据库的建立和维护功能
22. 其他功能：数据库管理系统与网络中其他软件系统的通信，数据库管理系统之间的数据转换，异构数据库之间的互访和互操作等。
23. **数据模型的相关概念**

数据模型是对现实世界数据特征的抽象，使用数据模型来描述数据、组织数据和对数据进行操作。

1. **数据模型及其三要素**

数据模型分为概念模型、逻辑模型和物理模型。

三要素：数据结构、数据操作、数据的完整性约束条件。

1. **概念模型及其相关概念（结合关系模型、数据库中的相关术语对比，参考课件中我总结的表）**

概念模型也称信息模型，它是按用户的观点来对数据和信息建模，用于数据库设计。

用途：概念模型用于信息世界的建模，是现实世界到机器世界的一个中间层次，是数据库设计的有力工具，数据库设计人员和用户之间进行交流的语言。

实体：客观存在、可相互区别的事物。

属性：实体具有的某一特性。

码：唯一标识实体的属性集。

实体型：用实体名及其属性名集合来抽象和刻画同类实体。

实体集：同一类型实体的集合称为实体集。

联系：现实世界中事物内部以及事物之间的联系在信息世界中反映为实体（型）内部的联系和实体（型）之间的联系。可以分为一对一、一对多和多对多等几种类型。

|  |  |  |  |
| --- | --- | --- | --- |
| **概念模型术语** | **关系术语** | **数据库术语** | **一般表格的术语** |
| **实体名** | **关系名** | **数据表名** | **表名** |
| **实体型** | **关系模式** | **表结构** | **表头（表格的描述）** |
| **实体集** | **关系** | **数据表** | **（一张）二维表** |
| **实体** | **元组** | **记录** | **行** |
| **属性** | **分量** | **字段** | **列** |
|  | **非规范关系** |  | **表中有表（大表中嵌有小表）** |

1. **E-R模型（结合第七章相关内容）**

E-R模型也称E-R方法，是用E-R图来描述现实世界的概念模型。在E-R图中，实体型用矩形表示，属性用椭圆形表示，联系用菱形表示。

1. **了解常见数据模型**

层次模型、网状模型、关系模型、面向对象数据模型、对象关系数据模型、半结构化数据模型。

1. **关系模型及其概念，三要素，优缺点**

关系模型是最重要的一种数据模型，关系数据库系统采用关系模型作为数据的组织方式。

三要素：(1) 数据结构：关系：元组的集合。

1. 数据操作：查询、插入、删除和更新数据。
2. 关系的完整性约束条件：实体完整性约束，参照完整性约束，用户定义的完整性约束

优点：(1) 建立在严格的数学概念的基础上。

1. 概念单一：实体和各类联系都用关系来表示，对数据的检索结果也是关系。
2. 关系模型的存取路径对用户透明：具有更高的数据独立性、更好的安全保密性，简化了程序员的工作和数据库开发建立的工作

缺点：(1) 存取路径对用户透明，查询效率往往不如格式化数据模型。

1. 为提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的难度。
2. **数据库系统的结构**

从数据库应用开发人员角度看，数据库系统通常采用三级模式结构，是数据库系统内部的系统结构。

从数据库最终用户角度看，数据库系统的结构分为：单用户结构、主从式结构、分布式结构、客户-服务器、浏览器-应用服务器／数据库服务器多层结构等。

1. **三级模式**
2. 模式：也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式。模式的地位：是数据库系统模式结构的中间层。
3. 外模式：也称子模式或用户模式，是数据库用户使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。外模式的地位：介于模式与应用之间。模式与外模式的关系：一对多，外模式与应用的关系：一对多。
4. 内模式：也称存储模式，是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式。
5. **两层映射**
6. 外模式/模式映像：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。映象定义通常包含在各自外模式的描述中。

当模式改变时，数据库管理员对外模式／模式映象作相应改变，使外模式保持不变。应用程序是依据数据的外模式编写的，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。

1. 模式/内模式映像：定义了数据全局逻辑结构与存储结构之间的对应关系。数据库中模式／内模式映象是唯一的，该映象定义通常包含在模式描述中。

当数据库的存储结构改变时，数据库管理员修改模式／内模式映象，使模式保持不变，从而应用程序不受影响，保证了数据与程序的物理独立性，简称数据的物理独立性。

1. **数据独立性的概念（结合第一节）及实现（后面结合视图）**

视图对重构数据库提供了一定程度的逻辑独立性。

尽管数据库的逻辑结构改变了，但应用程序不必修改，因为新建立的视图定义为用户原来的关系，使用户的外模式保持不变，用户的应用程序通过视图仍然能够查找数据。

1. **数据库系统的组成**
2. 硬件平台及数据库
3. 软件：数据库管理系统，支持数据库管理系统运行的操作系统，与数据库接口的高级语言及其编译系统，以数据库管理系统为核心的应用开发工具。
4. 人员：数据库管理员，系统分析员和数据库设计人员，应用程序员，最终用户。

**第二章**

1. **关系模型及其相关概念**

域：域是一组具有相同数据类型的值的集合。

笛卡尔积：域上的一种集合运算，可表示为一张二维表，表中每行对应一个元组，每一列的值来自一个域。

基数：一个域允许的不同取值的个数称为这个域的基数。

候选码：若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码。

全码：最极端的情况，关系模式的所有属性组是这个关系模式的候选码，称为全码。

主码：若一个关系有多个候选码，则选定其中一个为主码。

主属性：候选码的诸属性称为主属性。

非主属性或非码属性：不包含在任何侯选码中的属性。

1. **关系的类型和性质**

类型：

1. 基本关系：也称为基本表或基表，是实际存在的表，是实际存储数据的逻辑表示。
2. 查询表：查询结果对应的表。
3. 视图表：由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据。

性质：

1. 列是同质的，每一列的值都来自同一个域。
2. 不同的列可出自同一个域，其中的每一列称为一个属性，不同的属性要给予不同的属性名。
3. 列的顺序无所谓，列的次序可以任意交换。
4. 任意两个元组的候选码不能相同。
5. 行的顺序无所谓，行的次序可以任意交换。
6. 分量必须取原子值。
7. **数据完整性约束（结合第五章）**
8. 实体完整性：若属性A是基本关系R的主属性，则属性A不能取空值

空值就是“不知道”或“不存在”或“无意义”的值。

关系模型的实体完整性在CREATE TABLE中用PRIMARY KEY定义。

1. 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上的值必须为：或者取空值（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。

关系模型的参照完整性在CREATE TABLE中用FOREIGN KEY定义哪些列为外码，用REFERENCES指明这些外码参照哪些表的主码。

1. 用户定义的完整性：针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求。

**第三章**

1. **SQL概述及基本概念**
2. SQL：结构化查询语言，是关系数据库的标准语言，SQL是一个通用的、功能极强的关系数据库语言。
3. SQL的特点：综合统一；高度非过程化；面向集合的操作方式；以同一种语法结构提供多种使用方式；语言简洁、易学易用。
4. SQL的基本概念：

基本表：本身独立存在的表，SQL中一个关系就对应一个基本表，一个（或多个）基本表对应一个存储文件，一个表可以带若干索引。

存储文件：逻辑结构组成了关系数据库的内模式，物理结构对用户是隐蔽的。

视图：从一个或几个基本表导出的表，数据库中只存放视图的定义而不存放视图对应的数据，是一个虚表，用户可以在视图上再定义视图。

1. **SQL语言中WHERE子句和HAVING子句的区别**

WHERE子句与HAVING短语的区别在于作用对象不同。WHERE子句作用于基本表或视图，从中选择满足条件的元组。HAVING短语作用于组，从中选择满足条件的组。

1. **SQL查询中的外连接**

以指定表为连接主体，将主体表中不满足条件的元组一并输出。

1. **视图的概念及作用**

概念：视图是从一个或几个基本表（或视图）导出的表，是一个虚表。数据库只存放视图的定义，不存放视图对应的数据。

作用：

1. 视图能够简化用户的操作
2. 视图使用户能以多种角度看待同一数据
3. 视图对重构数据库提供了一定程度的逻辑独立性
4. 视图能够对机密数据提供安全保护
5. 适当的利用视图可以更清晰的表达查询

**第四章**

1. **数据库安全性的概念**

数据库的安全性是指保护数据库以防止不合法使用所造成的数据泄露、更改或破坏。

1. **常用的数据库安全性技术（了解）**

用户身份鉴别，存取控制，自主存取控制方法，授权：授予与收回，数据库角色，强制存取控制方法。

**第五章**

**数据库的完整性及其实现**

**第七章**

1. **数据库设计过程及其相关概念**
2. 需求分析阶段：是否做得充分与准确，决定了构建数据库的速度和质量。
3. 概念结构设计阶段：通过对用户需求进行综合、归纳与抽象，形成一个独立于具体数据库管理系统的概念模型。
4. 逻辑结构设计阶段：将概念结构转换为某个数据库管理系统所支持的数据模型，并对其进行优化。
5. 物理结构设计阶段：为逻辑数据结构选取一个最适合应用环境的物理结构，包括存储结构和存取方法。
6. 数据库实施阶段：根据逻辑设计和物理设计的结果构建数据库，编写与调试应用程序，组织数据入库并进行试运行。
7. 数据库运行和维护阶段：经过试运行后即可投入正式运行，在运行过程中必须不断对其进行评估、调整与修改。
8. **设计步骤，各步骤需要解决的问题，形成的成果**
9. 需求分析：准确了解与分析用户需求（包括数据与处理）；主要成果是数据字典。
10. 概念结构设计：将需求分析得到的用户需求抽象为信息世界的结构；主要成果是形成概念模型（E-R图）。
11. 逻辑结构设计：把概念结构设计阶段设计好的基本E-R图转换为与选用数据库管理系统产品所支持的数据模型相符合的逻辑结构；主要成果是形成关系模型，形成数据的外模式。
12. 物理结构设计：为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构（包括存储结构和存取方法）；主要成果是建立索引，形成数据库内模式。
13. 数据库的实施：数据的载入，应用程序的编码和调试；主要成果是创建数据库模式。
14. 数据库的运行和维护：性能监测、转储/恢复、数据库重组与重构。
15. **E-R模型**

E-R模型是用E-R图来描述现实世界的概念模型。

构成E-R图的基本要素是实体型、属性和联系，其表示方法为：

实体型：用矩形表示，矩形框内写明实体名；

属性：用椭圆形表示，并用无向边将其与相应的实体连接起来；

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1 : 1，1 : n或m : n）。

1. **逻辑结构设计**

把概念结构设计阶段设计好的基本E-R图转换为与选用数据库管理系统产品所支持的数据模型相符合的逻辑结构。

步骤：转换为数据模型，关系规范化，模式优化，设计用户子模式。

**第十章**

1. **事务的概念**

事务是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。事务是恢复和并发控制的基本单位。

事务的ACID特性：原子性，一致性，隔离性，持续性。

1. **数据库恢复，故障的种类**
2. 数据库的恢复：

数据库管理系统必须具有把数据库从错误状态恢复到某一已知的正确状态(亦称为一致状态或完整状态)的功能。

恢复子系统是数据库管理系统的一个重要组成部分。

恢复技术是衡量系统优劣的重要指标。

1. 故障的种类：
2. 事务内部的故障——事务撤销（UNDO）
3. 系统故障——撤销故障发生时未完成的事务，重做（REDO）已完成的事务
4. 介质故障——重装数据库，然后重做已完成的事务
5. 计算机病毒
6. **恢复的实现技术**

数据转储，登记日志文件

**第十一章**

1. **在数据库中为什么要并发控制？**

数据库是共享资源，通常有多个事务在运行。当多个用户并发地存取数据库时就会产生多个事务同时存取同一数据的情况，若对并发操作不加控制就可能会存取和存储不正确的数据，破坏事务的一致性和数据库的一致性。所以数据库管理系统必须提供并发控制机制。

1. **并发操作可能会产生哪几类数据不一致？用什么方法能避免这些不一致的情况？**

丢失数据，不可重复读，读“脏”数据。

避免不一致的方法和技术就是并发控制，最常用的技术是封锁技术。也可以用其他技术，例如在分布式数据库系统中可以采用时间戳方法来进行并发控制。

1. **封锁技术及封锁协议**
2. 封锁技术：

封锁就是事务T在对某个数据对象例如表、记录等操作之前，先向系统发出请求，对其加锁。加锁后事务T就对该数据对象有了一定控制，在事务T释放它的锁之前，其他事物不能更新此数据对象。

1. 封锁协议：

一级封锁协议：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。

二级封锁协议：在一级封锁协议基础上增加事务T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。

三级封锁协议：在一级封锁协议的基础上增加事务T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。

1. **活锁与死锁**

活锁：数据资源释放时间不确定，导致某些事务长时间等待，得不到封锁的机会。

死锁：多个事务各自占有部分资源等待另一部分资源，资源需求出现回路，导致事务停顿得不到执行。

解决活锁：先来先服务

解决死锁：预防：一次封锁法、顺序封锁法

诊断与解除：超时法、等待图法