

# 4735 数据库系统原理简答题

2009 / 09 / 19

## 第一章 数据库系统基本概念

1. 文件系统中的数据独立性是指\_\_\_\_\_独立性。(设备)
2. 在数据库方式下的信息处理中，\_\_\_\_\_占据了中心位置。(数据)
3. DBMS 是位于\_\_\_\_\_和\_\_\_\_\_之间的一层数据管理软件。(用户 OS)
4. 数据模型不仅描述数据本身的特点，还要描述\_\_\_\_\_。(数据之间的联系)
5. DBS 中，用户的数据和磁盘中的数据之间转换由\_\_\_\_\_实现。(DBMS)
6. 在层次、网状模型中，用\_\_\_\_\_导航数据；而在关系模型中，用\_\_\_\_\_导航数据。(指针 关键码)
7. 数据库的三级模式结构是对\_\_\_\_\_的三个抽象级别。(数据)
8. DBS 中存放三级结构定义的 DB 称为\_\_\_\_\_。(数据字典)
9. DBS 的全局结构体现了其\_\_\_\_\_结构。(模块功能)
10. DBMS 为应用程序运行时开辟的 DB 系统缓冲区，主要用于\_\_\_\_\_和\_\_\_\_\_。(数据传输 模式转换)
11. 层次模型用\_\_\_\_\_型结构来表示实体间的联系。(树)
12. 在数据的人工管理阶段，程序与数据是\_\_\_\_\_的关系。(一一对应)
13. 定义数据库的安全性和完整性的工作由\_\_\_\_\_完成。(DBA)
14. 数据独立性的好处是\_\_\_\_\_。(数据存储方式的变化不会影响到应用程序的使用)
15. 数据库的三级体系结构使用户能抽象地使用数据，不必关心\_\_\_\_\_。(数据在计算机中的表示和存储)

## 第二章 数据库设计和 ER 模型

1. ER 数据模型一般在数据设计的\_\_\_\_\_阶段使用。(概念设计)
2. “为哪些表，在哪些字段上，建立什么样的索引”这一设计内容应该属于数据库设计中的\_\_\_\_\_阶段。(物理设计)
3. 数据模型是用来描述数据库的结构和语义的，数据模型有概念数据模型和结构数据模型两类，ER 模型是\_\_\_\_\_。(概念数据模型)
4. 数据实施阶段包括两项重要的工作，一项是数据的\_\_\_\_\_，另一项是应用程序的编码和调试。(载入)
5. ER 图向关系模型转化要解决的问题是如何将实体和实体之间的联系转换成关系模式，如何确定这些关系模式的\_\_\_\_\_。(属性和键)
6. 数据库的物理设计是对一个给定的\_\_\_\_\_模型选取一个最合适应用环境的物理结构的过程。(逻辑数据)
7. 数据库设计中，将分 ER 图集成时，主要任务是增补\_\_\_\_\_。(各局部 ER 之间的联系)
8. 数据库应用系统设计中逻辑设计的主要内容是把 ER 模型的\_\_\_\_\_和\_\_\_\_\_转换为关系模式。(实体 联系)
9. ER 方法是设计\_\_\_\_\_的方法。(概念数据模型)
10. 现实世界到机器世界过渡的中间层次是\_\_\_\_\_。(概念模型)
11. 概念设计的目标是产生反映\_\_\_\_\_的数据库概念结构，即概念模式。(企业组织信息需求)
12. 在 DBD 中，子类具有一个重要的性质：\_\_\_\_\_。(继承性)
13. DBD 的逻辑设计分成两大部分：\_\_\_\_\_和\_\_\_\_\_。(DB 逻辑结构设计 应用程序设计)
14. 关系模型用\_\_\_\_\_表示实体之间的联系。(关键码)


15. DBS 的维护工作由\_\_\_\_\_承担。


(DBA)

### 第三章 关系模式设计理论

1. 关系模式的操作异常问题往往是由\_\_\_\_\_引起的。(数据冗余)
2. 如果  $X \rightarrow Y$  和  $Z \sqsubseteq Y$  成立, 则  $X \rightarrow Z$  成立, 这条推理则称为\_\_\_\_\_。(分解性)
3. 在关系模式  $R(U)$  中,  $X \in U, Y \in U, X \rightarrow Y$ , 如果  $X$  的某一真子集  $X'$  都有  $X' \rightarrow Y$ , 则称为\_\_\_\_\_。(Y 部分函数依赖于 X)
4. 包含在所有候选键中的属性称为\_\_\_\_\_, 所有候选键之外的属性称为\_\_\_\_\_。(主属性 非主属性)
5. 在关系模式  $R(U)$  中,  $X, Y \in U$ , 若  $X \rightarrow Y, Z = U - X - Y$ , 则\_\_\_\_\_。(X  $\rightarrow$  Y)
6. 若关系模式  $R$  中没有非主属性, 关系模式  $R \in$  \_\_\_\_\_ 范式。(3NF)
7. 若关系模式  $R$  已属于第一范式, 且其中的每一非主属性都\_\_\_\_\_于主键, 则  $R$  属于第二范式。  
(完全函数依赖)
8. 如果关系  $R$  中只有一个键, 且该键是单属性, 那么  $R$  至少属于\_\_\_\_\_ 范式。(2NF)
9. 被  $F$  逻辑蕴涵的函数依赖的全体称为\_\_\_\_\_, 记为\_\_\_\_\_。(F 的闭包  $F^+$ )
10. 关系规范化过程的实质是\_\_\_\_\_。(对关系模式不断分解的过程)
11. "不能从已知  $FD$  集使用推理规则导出的  $FD$  不在  $F^+$  中", 这是推理规则的\_\_\_\_\_性。(完备)
12. 在关系模式  $R$  中, 能函数决定所有属性的属性组, 称为模式  $R$  的\_\_\_\_\_。(超键)
13. 消除了非主属性对候选键局部依赖的关系模式, 称为\_\_\_\_\_ 模式。(2NF)
14. 两个函数依赖集  $F$  和  $G$  等价的充分必要条件是\_\_\_\_\_。(F = G)
15. 关系  $R$  的每个非主属性既不部分依赖候选键, 又不传递依赖于候选键,  $R \in$  \_\_\_\_\_ 范式。(第三)
16. 消除了每一属性对候选键传递依赖的关系模式称为\_\_\_\_\_ 模式。(BCNF)
17. 在关系模式  $R(U)$  中,  $X, Y \in U$ , 若  $X \rightarrow Y$ , 且  $XY = U$  (或  $Y \sqsubseteq X$ )。则称  $X \rightarrow Y$  为\_\_\_\_\_。(平凡多值依赖)
18. 一个关系模式属于 4NF, 它必定属于\_\_\_\_\_。(BCNF)

### 第四章 关系运算

1.  连接操作由关系代数的\_\_\_\_\_操作和\_\_\_\_\_操作组合而成。(笛卡尔积 选择)
2. 若在关系中置换属性的次序, 可用的关系代数运算是\_\_\_\_\_。(投影)
3. 在关系  $R$  和  $S$  自然连接时, 把原该舍弃的元组放在结果关系中, 这种操作称为\_\_\_\_\_。(外连接)
4. 查询优化是指系统对关系代数表达式进行优化组合, 它的目的是\_\_\_\_\_。(提高系统效率)
5. 自然连接要求被连接的两个关系具有\_\_\_\_\_。(一个或多个相同的属性名)
6. 域关系演算类似与元组演算, 域变量的变化范围是\_\_\_\_\_, (某个值域)  
元组变量的变化范围是\_\_\_\_\_。(一个关系)
7. 在关系代数运算中, 最浪费时间的两个运算是\_\_\_\_\_和\_\_\_\_\_。(笛卡尔积 连接运算)
8. 安全运算是指不产生\_\_\_\_\_和\_\_\_\_\_的运算。(无限关系 无穷验证)
9. 若  $D_1 = \{a_1, a_2, a_3\}, D_2 = \{1, 2, 3\}$ , 则  $D_1 \times D_2$  集合共有元组\_\_\_\_\_个。(9)
10. 设  $W = R \bowtie S$ , 且  $W, R$ 、和  $S$  的元组个数分别为  $p, m$  和  $n$ , 那么三者之间应满足的条件是\_\_\_\_\_  
( $P \leq (m \times n)$ )
11. 关系中没有行序的原因是\_\_\_\_\_。(关系被定义为一个集合)

12. 关系模型的基本数据结构是\_\_\_\_\_，其数据库存储时基本组织方式是\_\_\_\_\_。(关系(或二维表) 文件)
13. 实体完整性规则是对\_\_\_\_\_的约束，参照完整性规则是对\_\_\_\_\_的约束。(主键 外键)
14. 关系代数的基本操作是\_\_\_\_\_。(U - × Π Q)
15.  连接操作由\_\_\_\_\_等基本操作组合而成。(× Q)
16. 自然连接操作由\_\_\_\_\_等基本操作组合而成。(× Π Q)
17. 除法操作由\_\_\_\_\_等基本操作组合而成。( - Π Q)
18. 等式  $R \bowtie S = R \times S$  成立的条件是\_\_\_\_\_。(R 和 S 没有公共属性)
19. 对关系进行垂直分割的操作称为\_\_\_\_\_，对关系进行水平分割的操作称为\_\_\_\_\_。(投影 选择)
20. 关系代数是以\_\_\_\_\_为运算对象的一组高级运算的集合。(关系)

## 第五章节 SQL 语言

1. SQL 语言的使用方式有两种，一种是\_\_\_\_\_，另一种是\_\_\_\_\_。(交互式 嵌入式)
2. 视图是一种虚表，它是从\_\_\_\_\_导出的表。(基本表)
3. 索引的用途是\_\_\_\_\_。(快速查询)
4. 在 SQL 语言中，创建索引时保留字 UNIQUE 的作用是\_\_\_\_\_。  
(使此索引的每一索引值只对应于唯一的记录值)
5. SQL 语言的数据查询是介乎于\_\_\_\_\_和\_\_\_\_\_之间的一种语言。  
(关系代数 关系演算)
6. SQL 中表结构的修改命令是\_\_\_\_\_。(ALTER TABLE)
7. SQL 的集合处理方式与主语言单记录的处理方式之间用\_\_\_\_\_来协调。(游标)
8. 如果在查询结果中不允许出现重复元组，应在 SELECT 语句中增加保留字\_\_\_\_\_。(DISTINCT)
9. DELETE 删除的最小单位是\_\_\_\_\_。(一个完整的元组)
10. 在 SQL 中一个关系对应于一个\_\_\_\_\_。(基本表)
11. 删除索引时，系统会同时从\_\_\_\_\_中删除有关该索引的描述。(数据字典)
12. 在游标处于\_\_\_\_\_时，可以修改或删除游标指向的元组。(活动状态)
13. SELECT 语句进行查询时，如果要列出所查表中的所有列的简便方法是列名表用\_\_\_\_\_表示。(\*)
14. 在 SELECT 语句中，HAVING 子句必须跟在\_\_\_\_\_子句后面。(GROUP BY)
15. SQL/PSM 提供了顺序、条件分支和\_\_\_\_\_三种流程控制语句。(循环)
16. SQL 中，用户有两种：\_\_\_\_\_和\_\_\_\_\_。(应用程序 终端用户)
17. SQL 中，创建了一个 SQL 模式，就是定义了\_\_\_\_\_。(一个存储空间)
18. SQL 中，外模式一级数据结构的基本单位是\_\_\_\_\_。(视图)
19. 操作“元组 IN (集合)”的语义是\_\_\_\_\_。(若元组在集合中，其值为 true 否则为 false)
20. 嵌入式 SQL 的预处理方式，是指预处理程序先对源程序进行扫描，识别出\_\_\_\_\_，并处理成主语言的\_\_\_\_\_形式。(SQL 语句 函数调用)
21. 为保证嵌入式 SQL 的实现，通常 DBMS 制造商提供一个\_\_\_\_\_，供编译时使用。(SQL 函数定义库)
22. SQL 语句嵌入在 C 语言程序时，必须加上前缀标识\_\_\_\_\_和结束标志\_\_\_\_\_。(EXEC SQL 分号)
23. 关键字 ASC 和 DESC 分别表示\_\_\_\_\_和\_\_\_\_\_的含义。(升序 降序)
24. 权限授予的命令是\_\_\_\_\_。(GRANT)
25. 视图可以定义在\_\_\_\_\_基本表上。(1 个或 n 个)
26. 数据库中只存放视图的\_\_\_\_\_而不存放视图的\_\_\_\_\_。视图的定义存放在\_\_\_\_\_。

(定义 记录数据 数据字典)

27. 游标实际上是一个\_\_\_\_\_。当游标指向一组元组中的一个时，该元组被主语言处理。(指针)
28. 基本表结构的修改用\_\_\_\_\_关键字，基本表内容的修改用\_\_\_\_\_关键字。(ALTER UPDATE)
29. 定义卷游标与游标相比多了关键字\_\_\_\_\_。(SCROLL)
30. 在 SELECT 语句中可用谓词 UNIQUE 来测试结果集合中是否有\_\_\_\_\_存在。(重复元素)

## 第六章节 数据库管理

1. 事务运行过程中，由于种种原因，使事务未运行到正常终止点就被撤消，这种情况称为\_\_\_\_\_。(事务故障)
2. 事务的原子性由 DBMS 的\_\_\_\_\_子系统来实现的。(事务管理)
3. 数据库的并发操作带来三个问题是：丢失更新问题、读脏数据问题和\_\_\_\_\_。(不可重复读问题)
4. 封锁技术中基本的两种封锁是排他型封锁和\_\_\_\_\_。(共享型封锁)
5. 若事务 T 对数据 A 加上\_\_\_\_\_锁，则允许 T 读取和修改 A，其他任何事务都不允许对 A 加任何类型的锁，直到 T 释放 A 上的锁。(X)
6. 在数据库技术中，把未提交的随后被撤消的数据称为\_\_\_\_\_。(脏数据)
7. S 锁解决了丢失更新问题，但同时又可能会引起\_\_\_\_\_问题。(死锁)
8. SQL2 中，程序开始时默认的事务存取模式是\_\_\_\_\_。(READ WRITE)
9. SQL 中用\_\_\_\_\_语句定义新的域。(CREATE DOMAIN)
10. 触发器的动作事件条件用\_\_\_\_\_子句定义，它可以是任意的条件表达式。(WHEN)

## 第七章节 SQL Server 2000 简介及应用

1. SQL Server 2000 是分布式的关系型数据库管理系统，具有\_\_\_\_\_体系结构。(客户/服务器)
2. SQL Server 2000 中，实际上用户连接的数目上限是\_\_\_\_\_个。(32767)
3. SQL Server 2000 的版本主要有企业版、标准版、\_\_\_\_\_和开发版 4 个版本。(个人版)
4. SQL Server Agent 服务包括作业、\_\_\_\_\_和操作员三部分。(警告)
5. 服务管理器在启动\_\_\_\_\_服务后才能进行数据库操作。(SQL Server)
6. 企业管理器提供遵从\_\_\_\_\_的用户界面。(Microsoft 管理控制台)
7. 一个存储过程可以声明零个或多个参数，参数名以 \_\_\_\_\_开头。( @ )
8. 在 SQL Server 2000 中，每个表最多有 3 个触发器，分别用于 Insert 、Update 和 \_\_\_\_\_(Delete)
9. 导入和导出数据可以完成多个数据库之间的\_\_\_\_\_。(数据转换和转移)
10. 用户访问 SQL Server 数据库中的数据时，经过的第一个验证过程是\_\_\_\_\_。(身份验证)
11. 用户在混合验证模式下使用\_\_\_\_\_用户登录 SQL Sever ，必须提供登录名和密码。(SQL 授权)
12. 在 SQL Server 中，将一组具有相同权限的用户组织在一起称为\_\_\_\_\_。(角色)
13. T—SQL 语言中局部变量的作用域是\_\_\_\_\_。(当前的批处理)
14. T—SQL 中用于循环结构的流程控制语句是\_\_\_\_\_。(while 语句)
15. 创建局部临时表必须使用由 \_\_\_\_\_开头的表名。( # )

## 第八章 PowerBuilder 9.0 简介及应用

1. PowerBuilder 是一种企业级\_\_\_\_\_和多层次体系结构开发工具。(数据库前端应用)
2. PB 问世于 1991 年,最初是由\_\_\_\_\_公司开发。(Powersoft)
3. PB 采用面向对象的编程方法和\_\_\_\_\_的工作原理。(事件驱动)
4. PB9.0 的开发空间的三个层次是 Workspace、Target 和\_\_\_\_\_。(Library)
5. Target (目标) 用于描述加入到工作空间中的\_\_\_\_\_。(应用)
6. PowerBuilder 库文件的扩展名为\_\_\_\_\_。( .pbl)
7. \_\_\_\_\_用于显示对开发人员做出的操作响应,(输出窗口)
8. PB9.0 有两组工具栏: PowerBar 和\_\_\_\_\_。(PainterBar)
9. 数据窗口画板通过提供多个\_\_\_\_\_来完成与数据窗口对象相关的工作。(工作区)
10. \_\_\_\_\_用于定义程序中使用的全局结构。(结构画板)
11. 数据窗口是一个\_\_\_\_\_,它包含了对数据库中的数据进行特定操作的信息。(对象)
12. 在事件的脚本中,置 dw-l 的过滤条件为空的语句是\_\_\_\_\_。(dw-l.setfilter("") )
13. 打开窗口 w-main 的语句是\_\_\_\_\_。(open(w-main))
14. PB9.0 中,连接数据库时用\_\_\_\_\_命令。(Connect )
15. PB 是基于\_\_\_\_\_应用的快速数据库前端开发工具。(客户机/服务器 )

## 第九章 数据库技术的发展

1. 对象联系图是对\_\_\_\_\_的扩展。(ER 图)
2. 面向对象技术中,数据类型系统由基本类型、复合类型和\_\_\_\_\_三部分组成。(引用类型)
3. 相同类型元素的有序集合,并且允许有重复的元素,称为\_\_\_\_\_。(列表类型)
4. 复合类型中的数组、列表、包、集合统称为\_\_\_\_\_。(汇集类型或批量类型)
5. 引用类型可以避免数据结构的\_\_\_\_\_。(无穷嵌套)
6. 传统的关系模型中,每一个关系模式都具有\_\_\_\_\_性质。(1NF)
7. 结构类型是\_\_\_\_\_元素的有序集合。(不同类型)
8. 子类具有\_\_\_\_\_性。(继承性)
9. 在面向对象模型中的对象相当于 ER 模型中的\_\_\_\_\_。(实体)
10. 目前面向对象开发的行业标准语言是\_\_\_\_\_。(统一建模语言 或 UML)
11. 关联的端部,也就是与关联相连的类,称为\_\_\_\_\_。(关联角色)
12. ODBC 技术使得\_\_\_\_\_与 DBMS 之间在逻辑上可以分离。(应用程序)
13. API 的中文含义是\_\_\_\_\_。(应用程序设计接口)
14. ODBC 技术为应用程序提供了一套 CLI 函数库\_\_\_\_\_。(基于 DLL 的运行支持环境)
15. ODBC 规范定义的驱动程序有两种类型:即单层驱动程序和\_\_\_\_\_。(多层驱动程序)
16. 数据源是驱动程序与\_\_\_\_\_连接的桥梁。(DBS)
17. CLI 定义了一套可以从宿主语言调用的公共函数,每一个函数具有一个名称和一个\_\_\_\_\_,(要求的算法)
18. 使用 SQL/CLI 时,函数调用 SQL 语句将作为\_\_\_\_\_参数动态地创建和传递。(字符串)
19. 驱动程序管理器用于连接各种 DBS 的\_\_\_\_\_。(DBMS 驱动程序)
20. 数据库应用系统开发工具可以在客户机上通过\_\_\_\_\_接口与诸如 Oracle、SQL Server 等服务器端的数据库相连。(ODBC)