



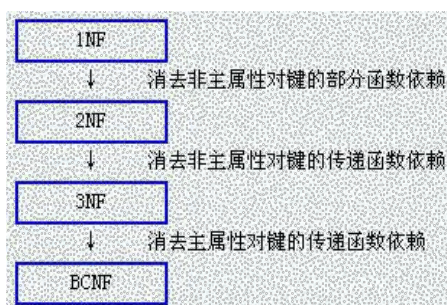
深大自考数据库系统原理复习要点

(内部资料, 请勿外传)

1. 数据库发展的 4 个阶段以及每个阶段的特点: 人工管理阶段, 文件系统阶段, 数据库阶段, 高级数据库阶段;
2. 文件系统的三个缺陷: 数据冗余性, 数据不一致性, 数据联系弱;
3. 数据抽象过程:
 - (1) 概念模型 (表达用户需求的模型, 抽象级别最高),
 - (2) 逻辑模型 (表达计算机实现的模型, 内部又分为层次模型, 网状模型, 关系模型),
 - (3) 外部模型 (表达用户使用的模型, 简化了用户观点),
 - (4) 内部模型 (表达物理结构的模型, 数据库最底层抽象, 描述数据库的存储存取方式);
4. 三层模式
 - (1) 外模式: 用户与数据库的接口
 - (2) 逻辑模式: 是数据库中全部数据的整体逻辑结构描述
 - (3) 内模式: 是数据库在磁盘存储方面的描述
5. 两级映像:
 - (1) 外模式/逻辑模式映像: (2) 逻辑模式/内模式;
6. 数据独立性:
 - (1) 物理数据独立性: 修改内模式时, 尽量不影响概念模式及外模式;
 - (2) 逻辑数据独立性: 修改概念模式时, 尽量不影响外模式和应用程序;
7. 数据库阶段的三件大事:
 - (1) IBM 推出层次模型 IMS;
 - (2) CODASYL 总结报告中提出网状模型;
 - (3) IBM 论文中提出关系模型;
8. DBMS 主要功能:
 - (1) 数据库的定义功能: DDL, 控制数据库的三级结构, 两级印象, 定义数据完整性约束;
 - (2) 数据库的操纵功能: DML, 负责检索和更新 (插入, 删除, 更新);
 - (3) 数据库的保护功能: 负责恢复, 并发控制, 完整性控制, 安全性控制;
 - (4) 数据库的维护功能: 负责数据载入, 转换, 转储;
 - (5) 数据字典: 数据库中存放三级结构定义的数据库称为数据字典;
9. DB, DBMS, DBS 三者关系: $DBS > DBMS > DB$
10. 物理存储介质: 高速缓冲存储器, 主存储器, flash, 磁盘, 磁带等;
11. DBMS 目标: 用户界面友好, 功能完善, 结构清晰, 高效率, 开放性;
12. 数据库生存期: 规划, 需求分析, 概念设计, 逻辑设计, 物理设计, 实现, 运行维护;
13. 规划阶段: 调查, 可行性分析, 确立总目标;
14. 需求分析阶段: 产生业务流程图, 系统流程图, 数据流图, 数据字典;
15. 概念设计阶段: 数据抽象, 局部概念设计, 全局概念设计;
16. 逻辑设计阶段: 概念模型转逻辑模型, 设计外模型, 设计 API, 评价和修正模型;
17. 物理设计阶段: 存储结构设计等;
18. 属性分类 (1) 简单属性和复合属性 (2) 单值属性和多值属性;
19. 键: 超键, 候选键, 主键, 外键; 【联系 SQL 语句】
20. 三类完整性约束: 实体完整性, 参照完整性, 用户定义完整性;
21. E-R 图怎么画? 【重点, 多做真题】



22. E-R 图转关系模式： (1) 1:1 (2) 1: N (3) M:N;
23. 一个实体的存在必须以另一个实体的存在为前提，前者称为弱实体，后者为强实体；
24. 子类实体继承了超类实体的所有属性；
25. 数据冗余：同一个数据在系统中重复出现；
26. 操作异常：由于数据冗余，在对数据操作时出现的各种异常
(结合书本 P68 来看)
- (1) 修改异常：修改了元组中一个老师的姓名，其他元组中该老师姓名未修改而导致报错；
- (2) 插入异常：应该插入的数据未被插入；
- (3) 删除异常：只想删除某一部分信息，却无奈整个元组信息删除。
27. 函数依赖是关键码概念的推广；
28. 函数依赖的定义 P71 (确定了 A, 则可以确定 B)
29. 函数蕴含：所有函数依赖的集合；
30. 闭包：函数依赖集+函数依赖本身；
31. 函数依赖推理规则 (背)：
- (1) 自反性 (2) 增广性 (3) 传递性 (4) 合并性 (5) 分解性
(6) 复合性 (7) 伪传递性 (8) 通用一致性
32. 推理规则的正确性是指：F 使用推理规则集推出的 FD 必定存在于 F+ 中；
推理规则的完备性是指：F+ 中的 FD 都能由 F 集中使用推理规则集导出；
33. 闭包计算规则 P75；注意课堂上讲的口诀!!!
34. 连接之后，仍能恢复为原本表的称为无损分解；否则称之为损失分解；
35. 分解出来多出来的元组称为寄生元组；
36. 模式分解能够消除数据冗余和操作异常的现象，但分解以后检索操作的时间代价将增大
(优缺点，简单了解)；
- 37.



38. 部分函数依赖：部分函数依赖是指存在组合关键字中的某一个关键字决定非关键字的情况 (一般地，所有的单关键字都符合第二范式)；
39. 消除了部分函数依赖，则成为完全依赖，或者左部不可约依赖；
40. 分解成 3NF 【重点】 (要诀：哪里出问题，就分解哪里)；
- (1) 第一范式分解成第二范式：(组合关键字 1, 组合关键字 2) -> 非关键字属性，将两个关键字分开；
- (2) 第二范式分解成第三范式：X->Y->Z, 将 Y 和 Z 分开；
41. 关系代数的五个基本操作：并差积选投；
42. 四个组合操作：交连自除；
43. 并操作和差操作需要元数相同；
44. 投影是垂直分割，选择是水平分割；
45. 自然连接：去除公共属性，重新组合；



- 46.除法：找出被除属性的元组，并去掉被除属性；
- 47.用关系代数表达查询语句；
- 48.关系代数转换规则 P104；
- 49.SQL 语言的四个部分：DDL，DML，嵌入式 SQL，DCL；
- 50.SQL 特点：只提供对数据库的操作功能；灵活而强大的查询功能；国际标准语言；核心词汇不多；
- 51.模式的创建：Create schema <模式名> authorization <用户名>
- 52.模式的撤销：Drop schema <模式名> [cascade | restrict]
- 53.SQL 的数据类型：int (integer) ,smallint,real,char,varchar,date,time
- 54.基本表的创建：

Create table <表名>

```
(
    <列名 类型>,
    <列名 类型>,
    ...,
    完整性约束
```

);

- 55.基本表的修改：

(1) 增加属性：Alter table <表名>

Add <列名><类型>

(2) 删除属性：alter table <表名>

drop <列名> [cascade | restrict]

- 56.基本表的撤销：Drop table <表名> [cascade | restrict]

- 57.索引的创建：

Creat [unique] index <索引名> on <表名>

Unique 表示每个索引值对应唯一的数据记录；

- 58.索引的撤销：Drop index <索引名>

- 59.查询语句基本结构：

Select <目标列名>

From <表名或者视图名>

[where <条件表达式>]

[group by<列名序列>]

[having <组条件表达式>] ! 满足 having 条件才能输出

[order by <列名 [asc | desc] >,...]

鉴于有些同学经常对什么时候要加 group by 有疑惑，在此特意解释下，group by 语句用于结合合计函数，根据一个或多个列对结果集进行分组。因此，如果用到聚合函数，例如 count, sum, avg 等，往往需要加上 group by 语句。

- 60.聚合函数：count,sum,avg,max,min (count * 会统计空值)

- 61.取别名 select 旧名 as 新名 from XX !as 可以省略

- 62.比较运算符：算数比较判断，确定范围，字符匹配，确定集合，唯一判断，空值判断

63. 两个等式 =some == IN ; <>all== NOT IN

- 64.单元组数据插入：Insert into <表名> [(<列名>)]



65.表的插入:

Insert into <表名 1>[<列名>]

Table <表名 2>

66.数据删除: Delete from <表名> [where <条件>]

67.数据修改:

Update <表名>

Set <列名>=<值表达式>

[where <条件>]

68.创建视图:

Create view <视图名> (<列表序列>)

As <select 查询语句>

[with check option] !允许用户修改视图

69.视图的撤销: Drop view <视图名>

70.DBMS 对 DB 的监控称为数据库的管理;

71.数据库管理主要有四个方面:

(1) 数据库的恢复 (2) 并发控制 (3) 完整性控制 (4) 安全性控制

72.DBS 运行的最小逻辑工作单位是“事务”,所有的数据库的操作都要以事务作为一个整体来执行或者撤销;

73.事务是构成单一逻辑工作单元的操作集合,要么完整的执行,要么完全不执行;

74.事务是以 begin transaction 开始,以 commit (成功执行,提交)或 rollback (执行失败,回滚)结束;

75.事务的 ACID 属性:原子性,一致性,隔离性,持久性;(一定要弄清楚每个性质的意义)

76.数据库的恢复策略:

(1) 日常做好两件事:转储(周期性对整个数据库进行拷贝),建立日志(建立日志数据库)

(2) 发生故障:

●灾难性故障:如磁头脱落,磁盘损坏,则装入最近一次拷贝的数据库备份到新磁盘,然后执行重做 REDO;

●未遭物理性破坏,但破坏了数据库的一致性(某些数据不正确),撤销 UNDO;

77.DBS 常见故障类型:

(1) 事务故障:

●预期性错误:如存款余额透支等, UNDO;

●非预期性错误:如运算溢出,数据错误等, UNDO;

(2) 系统故障:如硬件故障,掉电等,对未完成的事务 UNDO;而已提交的事务 REDO;

(3) 介质故障: REDO;

78.系统故障常被称为软故障,可以有系统自动恢复;而介质故障称为硬故障,还需要 DBA 执行配合;

79.检查点技术:在 DBS 运行时,DBMS 定时设置检查点,在检查点时刻才真正把对 DB 的修改写入到磁盘,并留下日志,以便恢复;

80.理解书本 P156 页的图:

只有到了检查点的时候才写入磁盘,如果检查点时没有做完就遇到故障,撤销;如果检查点时做完了,但还没有写入,重做;

81.并发是指在单处理机上利用分时方法,实行多个事务同时做;

82.并发操作带来的三个问题:丢失更新,读脏数据,不可重复读;



- 83.封锁技术主要分为两种：排他型封锁(X)和共享型封锁(S)；
- 84.排他锁自己可以读写，其他事务不允许读也不允许写；共享锁自己可以读写，其他事务可以读，但不允许写；
- 85.封锁相容矩阵：书本 P162
- 86.封锁对象的大小称为封锁的粒度；
 封锁粒度与系统的并发度和并发控制的开销密切相关。封锁的粒度越大，并发度越小，系统的开销也越小；相反，封锁的粒度越小，并发度越高，系统开销也越大；
- 87.封锁有三级协议，知道即可；
- 88.事务执行的次序称为调度；
- 89.如果有 n 个事务串行调度，则有 $n!$ 种不同的有效调度；
如果有 n 个事务并行调度，则并发调度的数目远大于 $n!$ ；
- 90.数据库中的完整性是指数据的正确性，有效性和相容性；
 其中正确性是指数据的合法性；有效性是指数据是否在定义的有效范围内；相容性是指表示同一事实的两个数据应该相同；
- 91.DBMS 中执行完整性检查的子系统成为完整性子系统，主要负责监督事务的执行，并采取相关措施阻止违反完整性规则的操作；
- 92.完整性约束分为三类：域约束，基本表约束，断言；
- 93.域约束，用 `create domain` 来定义新的域，还可以用 `check` 语句；
- 94.基本表约束主要有三种形式：候选键定义，外键定义和检查约束。在定义前加入 `constraint` 即可；
- 95.断言关键词：`assertion`；
- 96.触发器是一个能有系统自动执行对数据库修改的语句（类似句柄或事件）
- 97.触发器的三个部分：事件，条件，动作；
- 98.数据库的安全性是指保护数据库，防止数据泄露及破坏；
- 99.数据库的安全性级别：环境级（防止物理破坏），职员级（设置用户权限），OS 级（防止从 OS 层访问数据库），网络级（防止远程访问），DBS 级（检查用户身份及权限是否合法合规）；
- 100.用户使用数据库的权限分为两种：访问权限，修改数据库结构的权限；
- 101.访问数据权限四个：读，插入，修改，删除；
- 102.修改数据库结构的权限：索引，资源（创建新的关系），修改，撤销；
- 103.SQL 的四个安全性机制：视图，权限，角色，审计；
- 104.视图是从一个或者多个基本表导出的表，视图本身没有数据，不占空间；
- 105.视图机制的有点：数据安全性，逻辑数据独立性，操作简便性；
- 106.6 个用户权限：`select,insert,delete,update,references,usage`；
 References 是指用户在定义新的权限时，可以用外键；
 Usage 允许用户使用已定义的域；
- 107.用户授权：`Grant <权限> on<数据库元素> to<用户名>[with grant option] !转授权限`
 All privileges, 所有权限
- 108.用户是实际的人或者应用程序，而角色是指一组具有相同属性的用户；
- 109.一个用户可以参与多个角色，一个角色也可以授予多个用户；
 Grant <角色名> to <用户>
- 110.用于安全性目的的数据库日志，称为审计最终；
- 111.常用的安全性措施：强制存取控制，统计数据库的安全性，自然环境的安全性（硬件安全）；



112.SQL 语言的两种使用方式，一种是在终端交互下使用，称为交互式 SQL；另一种是嵌套在主程序中，称为嵌入式 SQL；

113.嵌入式 SQL 的使用规定：

（1）主程序中要区分 SQL 语句和主语言语句，所有 SQL 前加上 EXEC SQL；

（2）允许嵌入 SQL 语句应用主程序变量（共享变量），引用时这些变量前面要加冒号作为前缀表示，同时这些变量要用 declare 语句说明；

（3）SQL 是记录集合，主程序一次只能处理一个记录，需要用到游标机制，将集合操作变为单记录处理方式；

114.声明变量时，用 declare 命令，并且要指定声明的变量名和数据类型；

115.临时表和表变量是临时的，适合临时性的任务短时间保存数据以及不同的对象之间传递数据；

116.临时表分为局部临时表和全局临时表，局部临时表由单个#符号开头，而全局临时表作用于更广泛，用两个#符号开头；

117.表变量类似于临时表，它的优点在于它的内存中（速度快）

118.SQL server 2000 是一个分布式的关系型的数据库管理系统；

119.SQL Server 2000 提供的服务，MS SQL Server, SQL Server agent, Microsoft server services, MS dts；

120.企业管理器是用户管理 SQL Server 2000 的图形化界面；

121.T-SQL 中的核心是 DML；T-SQL 是拓展的 SQL，弥补 SQL 标准的不足；

122.T-SQL 中使用 use 命令来选定当前的数据库，用 exec 命令来调用存储过程；

123.T-SQL 中变量分为局部变量和全局变量，局部变量由用户定义和维护，由单个@字符开头；全局变量由系统定义和维护，以两个@开头，用户不能定义全局变量；

124.临时表和表变量适合用来在不同的对象之间传递数据，或者临时性的任务短时间保存数据；

125.局部临时表以#符号开头，临时表是创建并保存在 tempdb 数据库中；

126.全局临时表用##符号开头；

127.表变量类似于临时表，存储在内存中，读写速度快；

128.P182 系统目录：

\backup	存放备份文件
\binn	存放客户端和服务端可执行文件与 DDL 文件
\books	存放联机文档
\data	存放数据库文件，包括系统，实例，用户数据库
\ftdata	存放全文索引目录文件
\html	存放 html 文档
\log	存放日志文件
\install	存放安装方面的信息文件
\repldata	存放复制数据
\jobs	存放工作文件
\upgrade	存放升级工具

129.常用数据库：

(1)master:存放 SQL server 的系统信息，系统配置设置等；

(2)model:存放为用户创建数据库提供的模板数据库；

(3)msdb:存储任务计划信息，事件处理信息，备份恢复信息等；

(4)tempdb:存放所有临时表和临时的存储程序；



(5)northwind:提供范例数据库, 包含 northwind 公司的销售数据库

(6)pubs:也是提供范例数据库, 包含书籍出版公司的数据库范例。

130.流程控制语句: P212

131.PB9.0 是图形化的应用程序开发环境, 具有良好的数据库访问能力;

132.PB 9.0 特点:

- (1) 采用面向对象的变成方法和时间驱动的工作原理;
- (2) 支持跨平台开发, 具有开放的数据库连接系统;
- (3) 语句简单;
- (4) 人性化设计;
- (5) 独特的库文件画板和层次清晰的系统资源树;

133.开发空间的三个层次: workspace,target,library;

134.后缀名.pbl 是 PB 的库文件;

135.PB 9.0 的主要画板 (可能考简答):

- (1) 窗口画板: 用户可以在窗口画板中拖放可视化控件, 设置窗口大小, 类型, 样式等;
- (2) 应用画板: 用来创建和编辑应用对象, 可以对应用程序的 open, close, error 等事件编程和对象属性进行定制和修改;
- (3) 数据库画板: 用统一的界面来处理不同的数据库, 与数据库相关的绝大部分任务可以在数据库画板中完成;
- (4) 数据窗口画板: PB 中最强大的画板, 在其中创建的数据窗口对象可以实现对数据库的各种操作, 并按照不同风格显示数据, 可以提供多个工作区;
- (5) 数据管道画板: 用来实现数据库内, 数据库之间, 不同 DBMS 之间的快速数据转换;
- (6) 函数画板: 用来定义函数;
- (7) 库画板: 用于管理工作空间, 目标, PBL 库中的对象;
- (8) 菜单画板: 用来生成菜单对象以及关联的工具栏;
- (9) 工程画板: 用于生成可执行文件, 动态链接库, 组件和代理对象;
- (10) 查询画板: 用于定义数据窗口和数据管道等对象中可重用的 SQL 语句;
- (11) 结构画板: 用于定义程序中要使用的全局结构;
- (12) 用户对象画板: 用于创建用户对象

136. 概念设计是对用户数据需求的精确描述, 不涉及具体的实现细节;

137.面向对象类型的数据类型系统:

- (1) 基本类型: 指数, 浮点, 字符, 字符串, 布尔, 枚举;
- (2) 复合类型: P258 了解;

138.对象联系图:

- 椭圆代表对象类型 (相当于实体);
- 小圆圈代表属性的基本数据类型;
- 椭圆之间的边表示对象之间的引用;
- 单箭头表示属性值是单值;
- 双箭头表示属性值是多值;
- 双线箭头表示对象类型之间超类和子类的联系;
- 双向箭头表示两个属性之间的值的联系为逆联系;

139.UML 类图是统一建模语言, 适用于各类系统的建模;

140.类图, 了解关联元数的概念, 与关联有关的类的个数成为关联元数或者度数;

141. API 是应用程序设计接口;



142.公共 API 是为开发者提供的单一编程的接口，这样同一应用程序就可以访问不同的数据库服务器；

143.ODBC 技术使得应用程序和 DBMS 之间在逻辑上可以分离，使应用程序具有数据库无关性；

144.ODBC 定义了一个 API，每个应用程序利用相同的源代码就可访问不同的数据库系统，存取多个数据库中的文件；

145.ODBC 为应用程序提供了一套 CLI 函数库和基于 DLL 的运行支持环境；

146.CLI：调用层接口， DLL：动态链接库；

147.ODBC 的主要部分及功能 P268：

- (1) ODBC 数据库应用程序；
- (2) 驱动程序管理器；
- (3) DB 驱动程序；
- (4) 数据源；

148.SQL/CLI 是 ODBC 技术的延续，属于 SQL 标准，其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；

(如果书本的定义看不懂，可以参考百科解释：SQL/CLI 是应用程序访问数据库时执行 SQL 语句的调用级接口,该接口独立地执行 SQL 语句)。