2022 年大连海事大学《数据库原理与应用》科目期末试卷 B（有答

案）

一、填空题

1、数据仓库是_____、_____、_____、_____的数据集合，支持管理的决策过程。

2、视图是一个虚表，它是从_____导出的表。在数据库中，只存放视图的_____，不存放视图对应的_____。

3、在设计局部 E-R 图时，由于各个子系统分别有不同的应用，而且往往是由不同的设计人员设计，所以各个局部 E-R 图之间难免有不一致的地方，称为冲突。这些冲突主要有_____、_____和_____3 类。

4、设某数据库中有商品表（商品号，商品名，商品类别，价格）。现要创建一个视图，该视图包含全部商品类别及每类商品的平均价格。请补全如下语句： CREATE VIEW V1（商品类别，平均价格）AS SELECT 商品类别，_____FROM 商品表 GROUP BY 商品类别；

5、数据库内的数据是_____的，只要有业务发生，数据就会更新，而数据仓库则是_____的历史数据，只能定期添加和刷新。

6、数据库系统是利用存储在外存上其他地方的_____来重建被破坏的数据库。方法主要有两种：_____和_____。

7、数据管理技术经历了_____、_____和_____3 个阶段。

8、如果多个事务依次执行，则称事务是执行_____；如果利用分时的方法，同时处理多个事务，则称事务是执行_____。

9、对于非规范化的模式，经过转变为 1NF，_____，将 1NF 经过转变为 2NF，_____，将 2NF 经过转变为 3NF_____。

10、某在 SQL Server 2000 数据库中有两张表：商品表（商品号，商品名，商品类别，成本价）和销售表（商品号，销售时间，销售数量，销售单价）。用户需统计指定年份每类商品的销售总数量和销售总利润，要求只列出销售总利润最多的前三类商品的商品类别、销售总数量和销售总

利润。为了完成该统计操作，请按要求将下面的存储过程补充完整。

二、判断题

11、在数据库设计中，数据流图是用来建立概念模型的。（）

12、全码的关系模式一定属于 BC 范式。（）

13、一个关系中不可能出现两个完全相同的元组是由实体完整性规则确定的。（）

14、可串行化的调度一定遵守两段锁协议。（）

15、在 SQL 中，ALTERTABLE 语句中 MODIFY 用于修改字段的类型和长度等，ADD 用于添加新的字段。（）

16、视图是观察数据的一种方法，只能基于基本表建立。（）

17、在一个关系中，不同的列可以对应同一个域，但必须具有不同的列名。（）

18、在关系模式中，候选码可以有多个，主码只能有一个。（）

19、在数据库表中，空值表示数值 0。（）

20、有出现并发操作时，才有可能出现死锁。（）

21、可以用 UNION 将两个查询结果合并为一个查询结果。（）

OBJ

22、文件系统的缺点是数据不能长期存储。（）

23、视图就是一个虚表，保存视图时，保存的是视图的定义。（）

24、函数依赖是多值依赖的一个特例。（）

25、在 SELECT 语句中，需要对分组情况满足的条件进行判断时，应使用 WHERE 子句。（）

三、选择题

26、设关系 R（A，B，C）和 S（B，C，D），下列各关系代数表达式不成立的是（）。A．πA（R）⋈πD（S）

B．R∪S

C．πB（R）∩πB（S）

D．R⋈S

27、下列关于数据库备份的叙述，错误的是（）。

A．数据库备份也受到数据库恢复模式的制约

B．数据库备份是一项复杂的任务，应该有专业的管理人员来完成

C．如果数据库很稳定就不需要经常做备份，反之要经常做备份

D．数据库备份策略选择应该综合考虑各方面因素，并不是备份做得越多越全就越好

29、DB、DBS 和 DBMS 三者的关系是（）。

A．DB 包括 DBS 和 DBMS

B．DBS 包括 DB 和 DBMS

C．DBMS 包括 DB 和 DBS

D．DBS 和 DBMS 包括 DB

30、执行语句 CREATE DATABASE Student 的结果是（）。

A．创建一个名为 Student 的数据库，包括数据文件和日志文件

B．运行失败，因为参数不完整

C．创建一个名为 Student 的数据库，但是只有默认的数据文件，无日志文件

D．为数据库 Student 创建一个名为 Student 的表

31、下列说法正确的是（）。

A．可以利用存储过程在当前数据库中创建固定数据库角色

B．当前数据库中的用户自定义角色可以用存储过程删除

OBJ

C．不能将数据库用户账户添加为当前数据库中角色的成员

D．Public 角色可以被删除

32、在 SQL Server 2000 中，下列安全控制方法最合理的是（）。

A．为计算机系每个教师授予每个表的 SELECT 权，为教务处全体人员授予每个表的 SELECT、INSERT、DELETE 和 UPDATE 权 B．为计算机系和教务处分别建立一个角色，将两个部门的每个职工设置为相应角色中的成员。将计算机系角色和教务处角色设置为此数据库的 db_datareader 角色中的成员，将教务处角色设置为此数据库的 db_datawriter 角色中的成员

C．为计算机系和教务处分别建立一个角色，将两个部门的每个职工设置为相应角色中的成员。为计算机系角色授予每个表的 SELECT 权，为教务处角色授予每个表的 SELECT、INSERT、DELETE 和

UPDATE 权

D．将计算机系和教务处的每个职工都设置为 db_owner 角色中的成员

33、对于联机事务处理系统和数据仓库系统中的数据，下列说法正确的是（）。

A．一般情况下，联机事务处理系统中的数据不能被修改，数据仓库系统中的数据可被修改 B．联机事务处理系统中的数据库规模一般大于数据仓库系统中的数据规模

C．联机事务处理系统中的数据一般按面向业务应用的方式组织，数据仓库系统中的数据一般按面向分析主题的方式组织

D．一般情况下，在安全性和一致性要求上，对数据仓库系统中数据的要求要高于对联机事务处理系统中数据的要求

OBJ

34、“年龄在 15 至 30 岁之间”这种约束属于 DBMS 的（）功能。

A．恢复

B．并发控制

C．完整性 D．安全性

35、在关系代数表达式的等价优化中，不正确的叙述是（）。

A．尽可能早地执行连接

B． 尽可能早地执行选择

C．尽可能早地执行投影

D．把笛卡尔积和随后的选择合并成连接运算

36、下列关于函数依赖的描述中，错误的是（）。

A．在函数依赖 A→B 中，A 称为决定因素

B．在关系 R 中，属性 B 依赖于属性 A，则说明当属性 A 的值确定之后，属性 B 的值也就随之确定 C．函数依赖具有传递性

D．在关系 R 中，如果属性 A 依赖于属性 B，这种依赖正式记作：A→B。

37、现有关系模式 R（学号，姓名，课程代码，课程名，课程成绩），对其进行分解将其规范化到第三范式，下列（）是最正确的规范化结果。

A．R1（学号，姓名）R2（课程代码，课程名，课程成绩）

B．R1（学号，姓名）R2（课程代码，课程名）R3（学号，课程代码，课程成绩）

C．R1（学号，姓名）R2（学号，课程代码，课程名，课程成绩）

D．R1（学号，姓名）R2（课程代码，课程名）R3（课程代码，课程成绩）

38、实现数据库并发控制的主要方法是（）。

A．授权 B．索引 C．日志 D．封锁

39、保护数据库，防止未经授权或不合法的使用造成的数据泄漏、非法更改或破坏。这是指数据的（）。

A．安全性 B．完整性 C．并发控制 D．恢复

40、数据仓库的模式中，最基本的是（）。

A．星座模式

B． 雪花模式 C．星型模式
D．以上都不对

41、OLAP 的核心是（）。

A．对用户的快速响应

B．互操作性

C．多维数据分析

D．以上都不是

四、简答题

42、使用数据库系统有什么好处？

43、试述关系模式的完整性规则。在参照完整件中，什么情况下外码属性的值可以为空值？
44、分析传统 RDBMS 在大数据时代的局限性。

45、简述传统数据库与数据仓库的区别

46、UNDO 操作和 REDO 操作各做些什么事情？

五、综合题

47、有学生表（学号，姓名，年龄，性别，系名，专业名，班号），设一个系可有多个专业，每个专业可有多个班，各班班号不重复，一个学生只在一个班学习。现经常需要按“系名”和“班号”进行查询，为提高查询效率，需要为“系名”和“班号”两个列建立一个非聚集索引，建立此索引有下列两种方法：

方法 1：索引列顺序为（系名，班号）。

方法 2：索引列顺序为（班号，系名）。

①这两种方法哪个更合理？请简要说明原因。

②针对你认为合理的方法，写出创建该索引的 SQL 语句。

48、请给出缓冲区管理中的一个淘汰算法。

49、设教学数据库的模式如下：

S（S#，SNAME，AGE，SEX）

SC（S#，C#，GRADE）

C（C#，CNAME，TEACHER）

试用多种方式定义下列完整性约束：
（1）在关系 S 中插入的学生年龄值在 16～25 之间。

（2）在关系 SC 中插入元组时，其 S#值和 C#值必须分别在 S 和 C 中出现。

（3）在关系 SC 中修改 GRADE 值时，必须仍在 0～100 之间。

（4）在删除关系 C 中一个元组时，首先要把关系 SC 中具有同样 C#值的元组全部删去。

（5）　在关系 S 中把某个 S#值修改为新值时，必须同时把关系 SC 中那些同样的 S#值也修改为新值。

参考答案

一、填空题

1、【答案】面向主题的；集成的；随时间变化的；非易失性

2、【答案】一个或几个基本表；定义；数据

3、【答案】属性冲突；命名冲突；结构冲突

4、【答案】AVG（价格）

【解析】SQL 中，AVG（字段名）函数用来计算一组记录中某个字段值的平均值。

5、【答案】动态变化；静态

6、【答案】冗余数据；后援副本；日志文件

7、【答案】人工管理；文件系统；数据库系统

8、【答案】串行；并行

9、【答案】使属性域变为简单域；消除非主属性对候选码的部分依赖；消除非主属性对候选码的传递依赖

10、【答案】TOP3；SUM（（销售单价－成本价）*销售数量）；DESC

二、判断题

11、【答案】错

12、【答案】对

13、【答案】错

14、【答案】错

15、【答案】错

16、【答案】错

17、【答案】对

18、【答案】对

19、【答案】错

20、【答案】对

21、【答案】对

22、【答案】错

23、【答案】对

24、【答案】对

25、【答案】错

三、选择题

26、【答案】B

【解析】A 项、D 项都是执行自行连接运算，当两个关系无公共属性时，自然连接就等同于笛卡尔积运算，因此，A 项、D 项都是正确的。关系的并、交、差运算要求两个关系是相容关系，即两个关系属性个数相等，且对应的属性来自同一个值域，R 与 S 不是相容关系，所以 B 项是错误的。

27、【答案】C

【解析】数据库系统总有可能出现故障，所以不管如何都需要一个合理的备份计划以防止出现故障造成数据库中数据的破坏。

28、【答案】A
【解析】“UPDATE 表名 SET”可以用来更改表中某个字段的值，如果出现分支选择情况，可以

使“CASE…WHEN…THEN”语句。

29、【答案】B

【解析】数据库系统（DBS DataBase System）数据库系统是一个引入数据库以后的计算机系统，它由计算机硬件（包括计算机网络与通信设备）及相关软件（主要是操作系统）、数据库（DB DataBase）、数据库管理系统（DBMS DataBase Management System）、数据库应用开发

系统和人员组成。

30、【答案】A

【解析】CREATE DATABASE dataname 可以用来创建数据库，其中 dataname 为数据库的名字，

默认包含数据文件和日志文件。

31、【答案】B

【解析】在 T-SQL 中，通过 sp-helpdbfixedrole 存储过程，查看 SQL

Server 固定数据库角色列表；利用 sp-addrolemember 存储过程将数据库用户的账户，添加为

当前数据库中数据库角色的成员；利用 sp-

droprolemember 存储过程从当前数据库的数据库角色中，删除数据库安全账户。但不能利用存储过程在当前数据库中创建固定数据库角色，排除 A 项。此外，SQL Server 包括两类具有隐含

权限的预定义角色，固定服务器角色和固定数据库角色。这些隐含权限不能授予其他用户账户。

如果有用户需要这些权限，则必须将其账户添加到这些预定义角色中，排除 C 项；Public 角色是 SQL Server 数据库管理系统中每个数据库都存在的特殊角色。它提供数据库中用户默认权限，每个数据库用户都自动是此角色的成员。所以 Public 角色不能被删除，排除 D 项；在 T-SQL 中，

通过 sp-helprole 存储过程，查看角色；利用 sp-addrole 存储过程在当前数据库创建新的数据

库角色；利用 sp-droprole 存储过程从当前数据库删除角色。B 项是正确的。

32、【答案】B

【解析】在数据库中，db_datareader 角色具有查询数据库中所有用户表以适用于计算机系和教

务处的权限；db_datawriter 具有更改数据库中所有用户表中数据以适用于教务处的权限。

33、【答案】C

【解析】联机事务处理系统一般没有复杂的查询和分析处理，按面向业务应用的方式组织数据，

数据仓库系统的特征在于面向主题、集成性、稳定性和时变性，一般按面向分析主题的方式组织

数据。

34、【答案】C

【解析】“年龄在 15 至 30 岁之间”是完整性约束中值的约束。

35、【答案】A

【解析】在关系代数表达式中，连接运算的结果常常是一个较大的关系。如果尽可能早地执行连

接，则运算得到的中间结果就

36、【答案】D

【解析】函数依赖，顾名思义，就是属性之间存在着类似于数学中函数的一种数据依赖关系。设

U{A1，A2，…，An}是属性集合， R（U）是 U 上的一个关系，X、Y 是 U 的子集。若对于 R（U）下的任何一个可能的关系，均有 X 的一个值对应于 Y 的唯一具体值，称 X 函数决定 Y，或者 Y 函

数依赖于 X，记作 X→Y，其中 X 称为决定因素。

37、【答案】B

【解析】如果一个关系模式 R 属于第一范式，且每个非主属性既不部分依赖于码又不传递依赖于码，则这个关系属于第三范式。同时在对关系进行规范化的过程中，对于关系的分解不是随意而

为的，需要考虑新关系与原关系在数据上的等价、在依赖上的等价，甚至是在数据和依赖上都等价，否则就很可能达不到规范化目的。

A 项中的分解，R1（学号，姓名）达到 3NF 要求，R2（课程代码，课程名，课程成绩）也达到

3NF 要求，但是它们与原关系 R 在数据上不能等价，在依赖上也不等价，因而不是最正确的规范化结果。

C 项中的分解，R1（学号，姓名）达到 3NF 要求，R2（学号，课程代码，课程名，课程成绩）

中由于存在：课程代码一课程名，即非码属性对码的部分依赖，因此，该 R2 只达到 1NF 要求，

没有达到 2NF 要求，当然也就没有达到 3NF 要求。
D 项中的分解，R1（学号，姓名）达到 3NF 要求，R2（课程代码，课程名），R3（课程代码，

课程成绩）也都达到 3NF 要求，但是它们与原关系 R 在数据上不能等价，在依赖上也不等价，

因而不是最正确的规范化结果。

只有 B 项的分解，R1（学号，姓名），R2（课程代码，课程名）， R3（学号，课程代码，课程

成绩）都达到 3NF 要求，同时分解后的关系与原关系在数据和依赖上都等价，因而是最正确的规范化结果。

38、【答案】D

【解析】数据库管理系统对事务的并发执行进行控制，以保证数据库一致性，最常用的方法是封锁的方法，即当一个事务访问某个数据项时，以一定的方式锁住该数据项，从而限制其他事务对该数据项的访问。

39、【答案】A

【解析】数据安全性是指防止未经授权或不合法的用户使用数据库。

40、【答案】C

41、【答案】C

【解析】OLAP 具有共享多维信息的快速分析的特征。

四、简答题

42、答：使用数据库系统的好处是由数据库管理系统的特点或优点决定的，比如：

（1）可以大大提高应用开发的效率。在数据库系统中，应用程序不必考虑数据的定义、存储和数据存取的具体路径，这些工作都由 DBMS 来完成。开发人员可以专注于应用逻辑的设计，而不必为数据管理的许多复杂的细节操心。

OBJ

（2）数据库系统提供了数据与程序之间的独立性。当应用逻辑发生改变，数据的逻辑结构需要改变时，DBA 负责修改数据的逻辑结构，开发人员不必修改应用程序，或者只需要修改很少的应

用程序，从而既简化了应用程序的编制，又大大减少了应用程序的维护和修改，方便用户的使用。
（3）使用数据库系统可以减轻数据库系统管理人员维护系统的负担。因为 DBMS 在数据库建立、运用和维护时对数据库进行统一的管理和控制，包括数据的完整性、安全性、多用户并发控制、故障恢复等，都由 DBMS 执行。

总之，使用数据库系统的优点很多，既便于数据的集中管理，控制数据冗余，提高数据的利用率和一致性，又有利于应用程序的开发和维护。

43、答：（1）关系模型的完整性规则是对关系的某种约束条件。关系模型中可以有三类完整性

约束：实体完整性、参照完整性和用户定义的完整性。

①实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

②参照完整性规则：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。

③用户定义的完整性是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据

必须满足的语义要求。

（2）在参照完整性中，外码属性值可以为空，它表示该属性的值尚未确定，但前提条件是该外码属性不是其所在参照关系的主属性。

44、答：关系数据库在大数据时代丧失了互联网搜索这个机会，其主要原因是关系数据库管理系统（并行数据库）的扩展性遇到了前所未有的障碍，不能胜任大数据分析的需求，关系数据管理模型追求的是高度的一致性和正确性，面向超大数据的分析需求。

45、答：传统数据库与数据仓库的区别如表

46、答：（1）在恢复操作中，REDO 操作称为重做，UNDO 操作称为撤消。如果数据库被破坏，利用日志文件执行 REDO 操作，将两个数据库状态之间的所有修改重新做一遍。这样，建立了新

的数据库，同时也没丢失对数据库的更新操作。

（2）REDO 处理的方法是正向扫描日志文件，重新执行登记的操作。如果数据库未被破坏，但某些数据可能不可靠，这时，可通过日志文件执行 REDO 操作，把已经结束的、不可靠的事务进

行 REDO 处理。

（3）UNDO 处理的方法是反向扫描日志文件，对每个 UNDO 事务的更新操作执行逆操作，即

对已插入的新记录执行删除操作，对已删除的记录重新插入，对已修改的数据库用旧值代替新值。

五、综合题

OBJ

47、答：①方法 1 更加合理，理由如下：更加有利于减少索引层次，提高查询效率；更加符合使用习惯；更加便于（系、班）进行统计。

② create index index_1 on 学生表（系名，班号）；

48、答：借助队列实现 LRU 页面替换算法作为缓冲区管理的淘汰算法。

定义缓冲区的数据结构如下：
完整的参考代码如下：

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
int len;

typedef struct LRU
{
    int data;
    int time;//计次数
} LRU;

typedef struct Queue
{
    LRU *pBase;//结构数组
    int front;//队列头
    int rear;//队列尾
}QUEUE;

void init(QUEUE *pQ)
{
    int N = len+1;
    pQ->pBase = (LRU*)malloc(sizeof(LRU ) * N);
    pQ->front = pQ->rear = 0;   //初始化为 0
}

int full_queue(QUEUE *pQ)
{
    int N = len+1;
    if((pQ->rear +1)%N == pQ->front)//循环队列
        return 1;
    else
        return 0;
}


int en_queue(QUEUE *pQ, int val)//入队前判断队列是否已满
{
    int N = len+1;
    if( full_queue(pQ) )
    {
        return 0;
    }
    else
    {
        pQ->pBase[pQ->rear].data = val;//压栈在队尾
        pQ->pBase[pQ->rear].time = 0;//初始化次数为 0
        pQ->rear = (pQ->rear+1) % N;
        return 1;
    }
}

int empty_queue(QUEUE *pQ)//1-->空    0-->非空
{
    int N = len+1;
    if(pQ->front == pQ->rear)
            return 1;
    else
            return 0;
}

int out_queue(QUEUE *pQ, int *pVal)//出队前判断队列是否为空
{
    int N = len+1;
    if(empty_queue(pQ))
    {
        return 0;
    }
    else
    {
        *pVal = pQ->pBase[pQ->front].data;//把出队的元素保存起来
        pQ->front = (pQ->front+1)%N;
        return 1;
    }
}
```

```
void add_time(QUEUE *pQ)
{
    int N = len+1;
    int i = pQ->front;
    while( i != pQ->rear)
    {
        pQ->pBase[i].time ++;
        i = (i + 1) % N;
        //printf("%d   %d", pQ->pBase[i].time, i);
    }
}

void Set_time_shot(QUEUE *pQ, int x)//若待入队元素与队中元素相同，将次数置为 0
{
    int N = len + 1;
    int i = pQ->front;
    while( i != pQ->rear)
    {
        if( pQ->pBase[i].data == x)
        {
            pQ->pBase[i].time = 0;
        }
        i = (i+1) % N;
    }
}

int Find_big_time(QUEUE *pQ)
{
    int N = len + 1;
    int i = pQ->front;
    int max_i = i;
    int max_time = pQ->pBase[pQ->front].time;
    while( i != pQ->rear)
    {
        if( pQ->pBase[i].time > max_time)
        {
            max_i = i; max_time = pQ->pBase[i].time;
        }
        i = (i+1) % N;
    }
    return max_i;
}


    void Replace_big_time(QUEUE *pQ, int x)//若待入队元素与队中元素不相同，替换元素，并将次数置为 0
    {
        int max_time = Find_big_time(pQ);
        printf("%d ", pQ->pBase[max_time].data);
        pQ->pBase[max_time].data = x;
        pQ->pBase[max_time].time = 0;
    }

    int same_queue(QUEUE *pQ, int x)//判断待入队元素是否与队中元素相同
    {
        int N = len+1;
        int i = pQ->front;
        while( i != pQ->rear)
        {
            if( pQ->pBase[i].data == x)
                return 1;
            i = (i+1) % N;
        }
        return 0;
    }

    int main(void)
    {
        char str[100];
        int val, data;
        int i, cnt = 0;
        scanf("%d", &len);
        scanf("%s", str);

        QUEUE Q;
        init(&Q);
        for(i=0; str[i] != '0'; i++)
        {
            val = str[i] - '0';
            if( empty_queue( &Q ) )//如果队列为空
            {
                en_queue(&Q, val);
            }
            else
            {
                    add_time(&Q);
                    if(full_queue(&Q))//如果队列已满
                    {
                            if( !same_queue(&Q, val))
                            {
                                Replace_big_time(&Q, val);
                            }
                            else
                            {
                                Set_time_shot(&Q, val);
                                cnt++;
                            }
                    }
                     else//如果队列没满也不为空
                      {
                            if( !same_queue(&Q, val))
                            {
                                en_queue(&Q, val);
                            }
                            else
                            {
                                Set_time_shot(&Q, val);
                                cnt++;
                            }
                      }
            }
        }
        printf("\n%d\n%d", cnt, strlen(str));
        return 0;
    }
```

49、答：（1）用检查子句定义：

CHECK（AGE BETWEEN 16 AND 25）

（2）在关系 SC 的定义中，用检查子句定义：

CHECK（S# IN（SELECT S# FROM S））CHECK（C# IN（SELECT C# FROM C））在关系 SC 的定义中，用外码子句定义：

FOREIGN KEY（S#）REFERENCES S（S#）

FOREIGN KEY（C#）REFERENCES C（C#）

（3）在关系 SC 的定义中，用检查子句定义：

CHECK（GRADE BETWEEN 0 AND 100）

（4）在关系 SC 中，可用外码子句定义：

FOREIGN KEY（C#）REFERENCES C（C#）ON DELETE RESTRICT；

（5）在关系 SC 中，可用外码子句定义：

FOREIGN KEY（S#）REFERENCES S（S#）ON UPDATE

CASCADE；