2019 年数据库复习整理

71116233 白丰硕

所谓整理，其实也是一个查漏补缺，梳理知识结构的过程，希望以后看到资料的同学可以认证对待专业知识，夯实基础！祝各位获得好的成绩~

1. 举例说明什么是数据模型？什么是数据模式？两者的关系和区别？

数据模型：

是用来描述数据的一组概念和定义。以文件系统为例,它所用的数据模型包含文件,记录和字段

数据模式：

以一定的数据模型对一个单位数据的类型、结构、相互的关系所进行的描述。例如学生信息记录可以定义为姓名、学号、性别等属性和关系的形式

区别：

数据模型是描述数据的手段,而数据模式是用给定的数据模型对具体的数据的描述

关系：

数据模式反映一个单位的各种事物的结构、属性、联系和约束，实质上是用数据模型对一个单位的模拟

2. 在 DBMS 中，通常采用多级数据模式，例如概念模式、外模式和内模式，简述数据库系统中的多级数据模式对数据独立性的影响。

外模式、概念模式、内模式，有效地组织、管理数据，提高了数据库的逻辑独立性和物理独立性。用户级对应外模式，概念级对应概念模式，物理级对应内模式，使不同级别的用户对数据库形成不同的视图。

数据独立性分为逻辑独立性和物理独立性.物理独立性是指内模式改变时,概念模式保持不变,逻辑独立是指概念模式改变时,外模式不变,从而使应用程序保持不变.当内模式改变时,DBMS 只要通过改变概念模式到内模式映射,即可使概念模式保持不变,从而实现了数据的物理独立性.而逻辑独立的实现正好相反.

数据独立性表示应用程序与数据库中存储的数据不存在依赖关系，包括逻辑数据独立性和物理数据独立性。

逻辑数据独立性是指局部逻辑数据结构(外视图即用户的逻辑文件)与全局逻辑数据结构(概念视图)之间的独立性。当数据库的全局逻辑数据结构(概念视图)发生变化(数据定义的修改、数据之间联系的变更或增加新的数据类型等)时，它不影响某些局部的逻辑结构的性质，应用程序不必修改。

物理数据独立性是指数据的存储结构与存取方法(内视图)改变时，对数据库的全局逻辑结构(概念视图)和应用程序不必作修改的一种特性，也就是说，数据库数据的存储结构与存取方法独立。

数据独立性的好处是，数据的物理存储设备更新了，物理表示及存取方法改变了，但数据的逻辑模式可以不改变。数据的逻辑模式改变了，但用户的模式可以不改变，因此应用程序也可以不变。这将使程序维护容易，另外，对同一数据库的逻辑模式，可以建立不同的用户模式,从而提高数据共享性,使数据库系统有较好的可扩充性,给 DBA 维护、

改变数据库的物理存储提供了方便。

3. 现代数据库怎么管理数据模式的？数据模型怎么影响系统性能？什么是结构化数据，半结构化数据，非结构化数据？

数据库系统划分为三个层次，称为三级模式，分别为概念模式，外模式，内模式，都存于数据目录中，是数据目录的最基本内容，DBMS 通过数据目录，管理和访问数据模式。

数据模型包含数据结构，数据操作，数据完整性约束，影响性能的因素主要是数据结构的复杂度和数据操作的可优化程度

结构化数据：数据整体结构化，通过数据模型描述　半结构化数据：单条记录内部的数据有结构，数据文件间无联系，整体无结构　非结构化数据：数据间，数据文件间都没有结构

4. 数据模式遵循的范式越高越好吗？

数据库的范式主要目的是防止数据冗余、更新异常、插入异常和删除异常，因此，如果达到了该目的也就可以了，但范式越高可能带来处理速度缓慢和处理逻辑复杂的问题，因此需要权衡考虑。并不是应用的范式越高越好，要看实际情况而定。应用的范式等级越高，则表越多。表多会带来很多问题：1 查询时要连接多个表，增加了查询的复杂度 2 查询时需要连接多个表，降低了数据库查询性能

5. 对于课程/学生和选课表的数据库，（1）关系模式的设计考虑了哪些问题？（2）表达每门课的先修情况，如何调整数据库的设计？设计方案和理由。

(1)考虑了完整性约束和规范化的问题。

① 数据库中各个关系表都有主键，主键的值是唯一的且不为空，满足了实体完整性约束。同时 enroll 表中定义了对 course 和 student 表的外键，满足了引用完整性约束。

② 关系中的属性都是原子的，因此满足第一范式的要求；同时各关系表中不存在部分函数依赖和传递函数依赖，因此该数据库的设计也满足第二和第三范式的要求。

(2)先修课程包括课程号、课程名、开课院系等属性。如果将先修课程情况直接作为属性附加到 course 关系表中，会导致关系中出现部分函数依赖，不符合二范式的要求。而且先修课程可能不止一门，上述的做法还会产生大量冗余。因此需要新增一个关系表表达先修课程情况。关系表设计如下：

cno	dname	cno1	dname1

其中 cno 和 dname 表示当前课程的课程号和开课院系，cno1 和 dname1 表示先修课程的课程号和开课院系。cno 和 dname、cno1 和 dname1 都是对 course 表的外键。

6. 简述 SQL 和关系代数的联系和区别

关系代数（以关系代数为基础的数据库语言是过程性的）、SQL、关系演算(元组关系演算和域关系演算)，它们的非过程化程度依次递增，主要应用领域也不同.SQL 是关系数据库的标准语言，关系代数和关系演算是它的理论基础

联系：关系代数是 sql 的理论基础。

区别：sql 是结构化查询语言，是数据库具体的技术标准和规范。关系代数是数学理论。

7. 嵌入式 sql

```
// 声明 SQL 通信区
EXEC SQL INCLUDE SQLCA;
// 定义宿主变量
EXEC SQL BEGIN DECLARE SECTION;
 float RATING;
 int NUM;
EXEC SQL END DECLARE SECTION;
// 连接数据库
EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;
// 使用游标读取数据
EXEC SQL DECLARE C1 CURSOR FOR
 SELECT rating, COUNT(sid)
 FROM Sailors
 GROUP BY rating
EXEC SQL OPEN C1;
while (TRUE)
{
 EXEC SQL FETCH C1 INTO :RATING, :NUM;
 if (SQLCA.SQLCODE == 10)
     break;
 if (SQLCA.SQLCODE < 0)
     break;
 （循环打印数据）
 ……
}
EXEC SQL CLOSE C1;
//关闭游标
```

8. R/S 的元组关系演算

$$r \div s = \{u \mid \text{对每一元组} v \in s, \text{都存在一元组} t \in r, \text{使得} t[Q]=u \text{ 且 } t[S]=v\}$$

$$r'(R') = \{u \mid \forall v(v \in s \rightarrow \exists t(t \in r \wedge t[Q]=u \wedge t[S]=v))\}$$

9. 相较层次和网状数据库系统，查询优化对关系数据库系统更为重要。你认为这句话对吗？给出理由。

对。层次和网状数据库系统的语言是过程性的，用户不仅要说明需要什么数据，还要告诉数据库系统获得这些数据的过程。而关系数据库系统的语言是非过程性的，用户只用说明需要什么数据，而如何获取由系统来实现。

对于同一个查询语句，对应的关系代数等价的不同表达式的查询效率有着很大的差异；集合操作不同的执行规则和策略也对查询效率有着很大的影响；同时关系数据在物理存储形式和存取方式和路径上都没有限制。因此对于关系数据库系统来说，查询优化就更为重要，它对系统的性能有着很大的影响。

所以查询优化对于关系数据库系统非常重要。

10. 简述索引对关系型数据库系统查询优化的意义；应该在什么时候使用索引？是不是正在任何情况下使用索引都能得到益处？举例说明。 **第二三问看 2 题**

关系型数据库查询优化的途径之一是依赖于存取路径的优化，而在关系型数据库中索引是用得最多的一种存取路径，建立合适的索引是实现查询优化的首要前提。索引提供了对数据的快速访问，根据操作建立合适的索引能够很大程度上优化存取路径，从而提高查询效率。

意义：1.大大加快数据的检索速度;2.创建唯一性索引，保证数据库表中每一行数据的唯一性;3.加速表和表之间的连接;4.在使用分组和排序子句进行数据检索时，可以显著减少询中分组和排序的时间。

只有当经常查询索引列中的数据时，才需要在表上创建索引。并非什么时候用索引都有益处，例如应用程序非常频繁地更新数据或磁盘空间有限时，则可能需要限制索引的数量。（索引占用磁盘空间，并且降低添加、删除和更新行的速度。）

11. 稠密索引是否一定能够提高针对索引属性查询的效率？

不一定。

① 如果是查询小文件中的全部或相当多的记录时，使用索引并不能提高查询效率，反而会因为索引查询增加开销；

② 如果稠密索引为次索引，且不是簇集索引，则再最低索引中，每个键值对应的不是一个地址而是一个地址集。很可能一个键值对应的多条记录分散在不同的物理块中；当一个键值对应的记录较多时，取这些记录时访问物理块的 I/O 开销反而会降低查询的效率。

12. 建立簇集索引的条件

P240

建立索引和不建索引的条件

P242

13. 如事务不遵守 ACID 准则，则对数据库产生何种后果？为什么一般不涉及数据库的程序中不提 ACID 准则？

若事务不遵守 ACID 准则，数据库中会产生脏数据，数据不一致、难以恢复等情况。不涉及数据库的程序不提 ACID 是因为它们很少需要满足数据库这样的高并发性和一致性需求。

一个事务是由应用程序中对数据库的一组操作序列组成的。如果事务不遵守 ACID 准则，则数据库中数据的完整性和一致性等就可能会因为事务的执行而遭到破坏。而一般不涉及数据库的程序不存在多用户之间数据的共享问题，所以在一般不涉及数据库的程序中不提 ACID 准则。

14. 从查询优化角度分析，为什么 SQL 查询 where 子句应尽量避免使用 OR？

为了加快查询速度，优化查询效率，主要原则就是应尽量避免全表扫描，尽量在 where 及 order by 涉及的列上建立索引。然而在 where 子句中使用 or 来当连接条件时，会导致引擎放弃使用索引而进行全表扫描，改用 Union 后，性能会大大提高。**P124**

15. 判断并发事务运行正确性标准是什么？封锁法的基本思想是什么？它是怎么样保证并发事务的正确执行的？采用封锁法以后必须解决的问题是什么？

（对于串行调度，各事务的操作没有交叉，也就没有互相干扰，当然不会产生并发引起的问题。可串行化调度和某一个串行化调度等价，所以它也不会产生并发所引起的问题。）

正确性准则：可串行化

封锁法的基本思想：

并发事务对同一数据对象操作前，向系统发出请求对操作对象进行加锁。从而强迫有冲突的事务按照抢到锁的次序执行

如何保证：

事务对某个数据对象的加锁请求获准后，该事务便对该对象有了一定的控制，在这个事务释放它的锁之前，其他事务对该数据对象的锁的请求需要根据相容矩阵进行锁的申请，如果没有申请到锁（证明有冲突），则无法对其进行操作，从而避免了访问冲突，保证并发事务正确执行。

需要解决的问题：活锁（先申请，先服务），死锁（等待-死亡，击伤-等待 P158）

16. 假设运行记录与数据库的存储磁盘具有独立失效模式，介质失效恢复时，对运行记录中上一检查点以前的已提交事务应该 redo 否？为什么？

介质失效恢复时，对运行记录中上一检查点以前的已提交的事务应该 redo。因为介质失效会丢失数据库中所有的数据，恢复时需要再加载最近的后备副本后，根据运行记录中的后像，重做最近后备副本以后提交的所有更新事务。因此最近一次检查点以前提交的事务也要做 redo 操作。

17. 阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

假设有两项业务对应的事务 T1、T2 与存款关系有关：

(1)转帐业务：T1(A，B，50)，从帐户 A 向帐户 B 转 50 元；

(2)计息业务：T2，对当前所有帐户的余额计算利息，余额为 X*1.01。

针对上述业务流程，回答下列问题：

【问题 1】（3 分）

假设当前帐户 A 余额为 100 元，帐户 B 余额为 200 元。有两个事务分别为 T1 (A，B，50)，T2，一种可能的串行执行为：

T1(A， B， 50) ->T2 结果：A= 50.5　B=252.5　A+B=303

请给出其它的串行执行次序和结果。

【问题 2】（8 分）

若上述两个事务的一个并发调度结果如下：

T1	T2
Read (A) A : =A-50 Write (A)	
	Read (A) A : =A*1.01 Write (A) Read (B) B : =B*1.01 Write (B)
Read (B) B : =B+50 Write (B)	

(1)上述调度是否正确，为什么？（3 分）

(2)引入共享锁指令 Slock()、独占锁指令 Xlock()和解锁指令 Unlock()，使上述调度满足两段锁协议，并要求先响应 T1 的请求。请给出一个可能的并发调度结果。（5 分）

【问题 3】（4 分）

若将计息业务 T2 改为对单个帐户的余额计算利息，即 T2 (A)余额为 A*1.01，请给出串行调度 T1 (A，B，50) ->T2 (A) ->T2 (B)和串行调度 T2 (A) ->T1 (A，B，50) ->T2 (B)的执行结果。

若将计息业务设计为对单个帐户的余额计算利息，这种方案是否正确，为什么？

正确答案：

【问题 1】（3 分）

T2—T1 (A，B，50) 结果：A= 51 B=252 A+B = 303

【问题 2】（8 分）

(1)调度不正确

结果为：A= 50.5B=252

原因： 与任何一个串行结果都不同。

(2)满足两段锁协议的调度：

T1（A,B,50）	T2
Xlock（A）	
Read（A）	
A=A-50	
Write（A）	
	Xlock（A）
Xlock（B）	等待
Read（B）	等待

B:=B+50	等待
Write（B）	等待
Unlock（A）	等待
Unlock（B）	等待
	Read（A）
	A:A*1.01
	Write（A）
	Xlock（B）
	Read（B）
	B:B*1.01
	Write（B）
	Unlock（A）
	Unlock（B）

【问题 3】（4 分）

三个事务的串行：

(1) T1 (A，B，50) ->T2 (A) ->T2 (B)结果：A= 50.5　B=252.5

(2)T2 (A) ->T1 (A，B，50) ->T2 (B)结果：A=51　B=252.5

不正确。计息业务设计为对单个帐户的余额计算利息，无法实现对所有帐户笺一计息，其间的转账会产生数据错误，会造成银行或客户的损失。

题考查对事务设计、并发控制的理解和掌握

两个事物 T1、T2 的串行执行只有两种方式：T1 执行完执行 T2（记为：T1->T2）和 T2 执行完执行 T1（记为：T2->T1），结合 A、B 的初值，即可计算出 T2->T1 的执行结果。

根据 A、B 的初值，按照给定的调度，获得执行结果为：A=50.5，B=252，与任何一个串行执行的结果都不同，为错误的调度，事实上会造成储户的无端损失。

引入两段锁协议后可保证调度的正确。根据锁类型和加解锁的要求，本题中所有的读取随后即要修改，对应了 SQL 中的 UPDATE 指令，可直接加 X 锁。

若将计息业务 T2 改为对单个账户的余额计算利息，根据提示的情况，调度结果可能存在不一定性，这样的事物设计是错误的。

18. 分布式数据库的全局死锁。

产生原因：由于系统提供的资源数比多个进程所需的资源数少，并且系统的资源分配策略和进程并发执行的速度不当。死锁是占有资源并申请资源的事务之间循环等待造成的。（举一个例子，T1 握有 T2 需要的资源的同时等待 T2 的握有的资源）

19. 试分析分布式数据库系统出现的技术背景和应用背景。它与后来出现的联邦式数据库系统的类似之处和本质区别是什么

背景：分布式处理技术的发展，当时网络宽还不足，因此出现了由网络连接的多台计算机共同协作解决大量数据的存储、管理、查询的需求，通过将数据就近存放提高访问效率。

本质区别：前者在物理上分布的，但逻辑上却是集中的。这种分布式数据库只适宜用途比较单一的、不大的单位或部门。而联邦式分布数据库系统在物理上和逻辑上都是分布的。由于组成联邦的各个子数据库系统是相对“自治”的，这种系统可以容纳多种不同用途的、差异较大的数据库，比较适宜于大范围内数据库的集成。

相似之处：分布式数据库系统的不同类别。是在集中式数据库系统的基础上发展来的。是数据库技术与网络技术结合的产物。包含分布式数据库管理系统(DDBMS)和分布式数据库(DDB)。

20. 数据查询和数据挖掘的区别联系？

P3

结构化，半结构化，非结构化数据

P4