1. 请简述数据库堆文件**的链表和页目录两种组织方式**中，空闲空间分别是如何组织的？

答：

**链表方式下**，当只有一个堆文件时，堆文件**头部设立一个header页面**，并存放**两个指针**，其中一个指针指向**空页列表（free list）头部（另一个指向数据页列表头部）**，而在**每个页面内部**，均记录**当前空闲的空间（slot）**；当有多个堆文件时，需要元数据记录文件中**有哪些页面**，以及**哪些页有空闲空间**。

（注：参考第八章课件P20）

**页目录方式下**， 堆文件中设立一类专门的页面（目录页），用于记录所有的数据页的存放位置，该目录也同时记录每个页面的空闲空间 信息（slot）。DBMS必须保持目录页与所有页的当前信息同步。

（注：参考第八章课件P21）

2. 请简述**面向元组型的页设计中槽页方案的基本思想**，并简要说明槽页方案相对于数组式的元组存储方案的好处。

答：每个页面都有页头，包含页面ID，还会保存页面内的Slot数组，记录已占用的“槽位（slot）”、这些“槽位（slot）”映射到特定元组开始位置的偏移量、以及上一次使用槽位的开始位置。

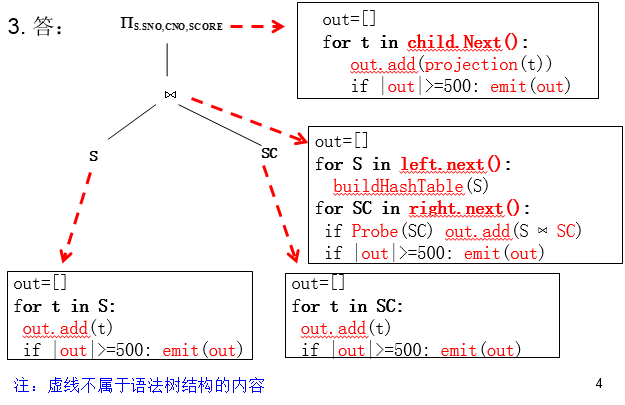
元组的存放则在页内倒序从页尾向页头依次存放。元组在内部的唯一标识符：可以使用页面ID和slot id（或偏移量），也可包含文件位置信息。

槽页方案元组**避免了删除元组时会产生碎片的问题，同时能更好的应对变长元组的存储**。

（注：参考第八章课件P26、P27）

3. 设有学生关系S(SNO, SNAME, SDEPT)存储学生的学号、姓名和所属院系，学生选课关系SC(SNO, CNO, SCORE)存储学生选课记录的学号、课号和成绩，现有查询的SQL语句“SELECT S.SNO, CNO, SCORE FROM S, SC WHERE S.SNO=SC.SNO”，请画出查询计划的语法树，请写出该查询语法树的处理模型采用向量模型时每个查询算子的伪代码（设算子中每次处理的一个batch的大小为500）。

（注：参考第八章课件P151、P152、P153，根据目标查询语句，得到其查询语法树，然后套用向量模型的处理流程和机制）



4.以下关系代数表达式的等价式是否成立？若否请给出关系实例说明之。

1) E1∞（E2 - E3）≡ E1∞E2 - E1∞E3

上述等价式成立

2) ΠA(r - s) ≡ ΠA(r) - ΠA(s)

上述等价式不成立

反例：

R:

|  |  |
| --- | --- |
| A | B |
| 1 | 2 |
| 1 | 3 |

S:

|  |  |
| --- | --- |
| A | B |
| 1 | 2 |
| 1 | 4 |

5.请简述为什么undo-list中事务的日志记录必须由后往前处理，而对于redo-list中的事务日记记录应该从前往后进行处理？

答：一个事务日志记录了该事物具有时间先后的若干动作，该动作序列表示事务的“应用语义”，任意更改这些动作顺序的undo或redo都会带来不正确的影响（数据一致性遭到破坏），

也就是说，恢复系统中的undo、redo机制保障了事务的原子性、持久性和一致性。

6.设有事务T1、T2如下。

|  |
| --- |
| T1 |
| Read(A) |
| Read(B) |
| If A=0 then B = B+ 1 |
| Write(B) |

|  |
| --- |
| T2 |
| Read(B) |
| Read(A) |
| If B=0 then A = A+ 1 |
| Write(A) |

请为T1、T2增加封锁和解锁指令，并以表格给出一个调度（两个事务都遵守两段锁协议），以上调度是否会导致死锁？

1）调度如下：

|  |  |  |
| --- | --- | --- |
| 时间 | T1 | T2 |
| 1 | RLock(A) |  |
| 2 | Read(A) |  |
| 3 | Xlock(B) |  |
| 4 | Read(B) |  |
| 5 | If A=0 then B = B+ 1 | RLock(B) |
| 6 | Write(B) | 等待。。。 |
| 7 | UnLock(B) | 等待。。。 |
| 8 | UnLock(A) | RLock(B) |
| 9 |  | Read(B) |
| 10 |  | Xlock(A) |
| 11 |  | Read(A) |
| 12 |  | If B=0 then A = A+ 1 |
| 13 |  | Write(A) |
| 14 |  | UnLock(B) |
| 15 |  | UnLock(A) |

2）会产生死锁的调度

|  |  |  |
| --- | --- | --- |
| 时间 | T1 | T2 |
| 1 | RLock(A) |  |
| 2 | Read(A) |  |
| 3 |  | RLock(B) |
| 4 |  | Read(B) |
| 5 | Xlock(B) |  |
| 6 | 等待。。。 | Xlock(A) |
| 7 | 等待。。。 | 等待。。。 |
| 8 | 等待。。。 | 等待。。。 |