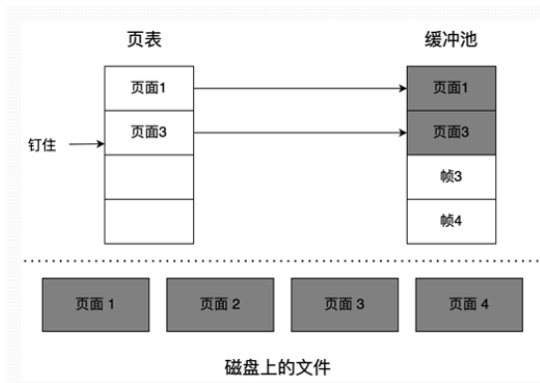


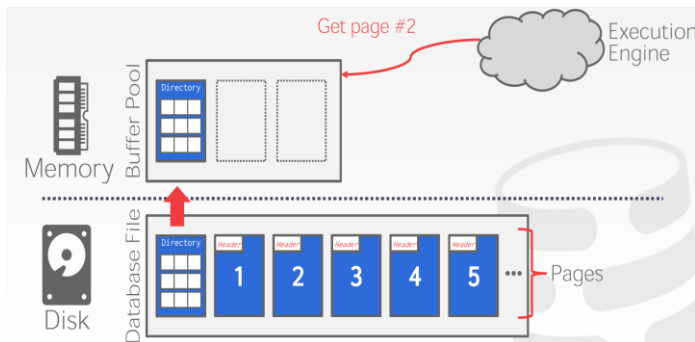
第 2 次测验

1. 画出 DBMS 缓冲池的结构示意图，并结合缓冲池的内部结构说明一般情况下执行引擎访问数据页面过程。说明其中什么情况下执行引擎读数据页面会触发磁盘页面的写操作？

解：



- ✓ 执行引擎在语句处理过程中需要使用某个数据页时，会向缓冲池提出请求；
- ✓ 缓冲池管理器负责将该页从磁盘读入内存，并向执行引擎提供该页在内存中的指针；
- ✓ 当执行引擎操作那部分内存时，缓冲池管理器必须确保该页面始终驻留在那片内存区域中。



分两种情况：

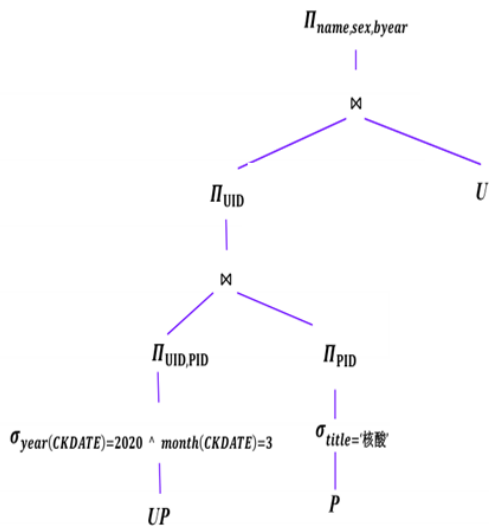
- 1) 若数据页面没加载到缓冲池中，当执行引擎发出一个请求访问，如第 2 页，而缓冲池中又没有第 2 页时，缓冲池首先将文件目录加载到内存中，找到第 2 页的物理位置，获取它，返回一个内存中的第 2 页指针给执行引擎，执行引擎根据此地址将第 2 页内容从外存加载到内存缓冲池中。
- 2) 当缓冲池空间已满，此时又有新的页面加载请求时，系统根据缓冲池替换策略算法将相关“脏”页面写回磁盘。

2. 假设某医院体检科有多种体检项目，其数据库中有用户关系 $U(\text{UID}, \text{NAME}, \text{BYEAR}, \text{SEX}, \text{FLAG})$ 记录来检查的每个用户的身份证号、姓名、出生年份、性别和用户类别，关系 $P(\text{PID}, \text{TITLE}, \text{PRICE})$ 记录每种体检项目的编号、名称及其费用，关系 $UP(\text{UID}, \text{PID}, \text{CKDATE})$ 记录用户参加了某种体检项目的一次检查的日期。假设数据库的函数 $\text{YEAR}(\text{UPDATE})$ 和 $\text{MONTH}(\text{UPDATE})$ 分别计算日期数据的年份和月份，请分别用一条 SQL 语句完成下列数据操纵需求。

- 查询 2020 年 3 月做过项目名为“核酸”检测的用户姓名、性别和出生年份
- 画出优化后的语法树

解：

```
select distinct name, sex, byear
from U, P, UP
where U.UID = UP.UID and P.PID = UP.PID and
      Title = '核酸' and
      year(CKDATE)=2020 and month(CKDATE)=3
order by byear desc;
```



3 系统崩溃时日志文件记录内容如下表所示，试写出系统重启后恢复处理步骤及恢复操作，并指明 A、B、C、D 恢复后的值分别是多少？

- <T0 ,start>
- <T1 ,start>
- <T0 ,A,3,10>
- <T2 ,start>
- <T0 ,commit>
- <T1 ,A,7,14>
- <T2 ,B,1,12>
- <checkpoint>
- <T1 ,A,14, 18>
- <T3 ,start>
- <T1 ,commit>
- <T2 ,B,13, 35>
- <T3 ,A,21,65>
- <T4 ,start>
- <T3 ,D,7,19>
- <T3 ,commit>
- <T4,C,14, 28>

解：在重新开始文件里找到最近检查点地址，发现在设立检查点时活动事务，将其放入 active list 队列{T1,T2}。
 将 UNDO 队列置为{T1,T2}。从检查点记录开始正向扫描日志文件，遇到 T3 开始，T3 加入 UNDO 队列；遇到 T1 提交，将 T1 从 UNDO 队列移到 REDO 队列；遇到 T4 开始，T4 加入 UNDO 队列；遇到 T3 提交，将 T3 从 UNDO 队列移到 REDO 队列；最终 UNDO 队列为{T2,T4}，REDO 队列为{T1,T3}。
 按照日志文件的正向顺序，依次 REDO 属于 T1、T3 的记录。
 按照日志文件的反向顺序，依次 UNDO 属于 T2、T4 的记录。
 最后结果 A=65,B=1,C=14,D=19

4. 阅读下表所示事务 T1 和 T2 的并发调度序列，假设数据库初始状态为 A=3，B=6。完成下列小题：
 (1) 该调度中两个事务是否满足 2PL 协议，为什么？
 (2) 该表中的事务和调度存在哪些不合理的地方，请逐一说明。

时间	事务 T1	事务 T2
1		Slock (A)
2		Num1=Read(A=3);
3	Slock(B) Num2=Read(B=6) Xlock(A)Wait...	

4		Unlock(A)
5	Xlock(A) Write(A=Num2+3);	
6		Xlock(B)
7		Write(B=5*Num1);
8	Unlock(A); COMMIT;	
9		Unlock(B); COMMIT;

解 1) 不满足 2PL 协议。事务 T2 在第 4 时间放锁后又在第 6 时间加锁。

2) 第一个不合理处为 T1 没有释放 B 锁

第二个不合理处为 T2 在第 6 时间不能对数据 B 加上写锁，因为数据 B 在第 3 时间已被 T1 加上读锁。