1、

数据模型是用来描述数据的一组概念和定义，如关系数据模型，它规定了属性、元组等概念来描述现实世界中的数据。

数据模式是对某一类数据的结构、联系和约束的描述。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，就是数据模式。

数据模型是对数据进行概念层次上的描述，是计算机中用来描述现实世界的方法。而数据模式是对特定数据集中数据的结构和联系的描述。数据模式是用给定的数据模型对具体数据的描述。

2、

(1)

SELECT S.sname FROM Sailors S, Boats B, Reserves R

WHERE S.sid=R.sid AND B.bid=R.bid AND R.bid>105 AND B.color=’white’

(2)

SELECT S.sname FROM Sailors S

WHERE NOT EXISTS

(SELECT B.bid FROM Boats B WHERE NOT EXISTS

(SELECT R.bid FROM Reserves R WHERE R.bid=B.bid AND R.sid=S.sid))

(3)

SELECT TEMPR.sid FROM

(SELECT R.sid, COUNT(R.bid) AS count

FROM Reserves R GROUP BY R.sid) AS TEMPR

WHERE TEMPR.count=(SELECT MAX(TEMPR.count) FROM TEMPR)

SELECT R.sid FROM Reserves R

GROUP BY R.sid HAVING COUNT(\*)=

(SELECT MAX(count) FROM

(SELECT COUNT(\*) AS count FROM Reserves R1 GROUP BY R1.sid))

(4) (平均次数是什么鬼)

SELECT R.sid, S.sname, avgcount=COUNT(R.bid)/(MAX(day)-MIN(day))

FROM Reserves R, Sailors S

WHERE S.sid=R.sid

GROUP BY R.sid

(5)

SELECT S.sname FROM Reserves R, Sailors S

WHERE S.sid=R.sid

GROUP BY R.bid, R.sid, S.sname

HAVING R.bid=205 AND COUNT(\*)=1

(6)

SELECT rating, MAX(count) FROM

(SELECT S.rating AS rating, R.sid, COUNT(\*) AS count

FROM Reserves R, Sailors S, Boats B

WHERE S.sid=R.sid AND B.bid=R.bid

GROUP S.rating, R.sid, B.color

HAVING B.color=’white’)

GROUP BY rating

~~SELECT S.rating, MAX(~~

~~SELECT COUNT(\*) FROM Reserves R, Sailors S1, Boats B~~

~~WHERE S1.sid=R.sid AND B.bid=R.bid~~

~~GROUP BY S1.rating, R.sid, B.color~~

~~HAVING B.color=’white’ AND S1.rating=S.rating)~~

~~FROM Sailors S~~

SELECT S.rating, COUNT(R.bid) AS cnt

FROM Reserves R, Sailors S

WHERE S.sid=R.sid AND R.color='white'

GROUP BY S.rating, R.sid

HAVING COUNT(R.bid) >= ALL(

SELECT COUNT(R1.bid)

FROM Reserves R1, Sailors S1

WHERE S1.sid=R1.sid AND R1.color='white'

GROUP BY S1.rating, R1.sid

HAVING S1.rating=S.rating)

3、

…

// 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;

4、

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。

一般在数据量较大的时候适合使用索引，在需要频繁进行查询、排序或分组操作的属性性建立合适的索引。并非在任何情况下使用索引都能优化查询效率，应该根据实际操作需求考虑。例如对于小文件上的顺序查找使用堆文件的形式的开销不大，使用索引反而可能会增加开销。因为索引本身需要占用一定的存储空间，而且维护索引也需要一定的开销。对于增删改操作频繁的属性上建立索引可能会起到反作用。

5、

判断并发事务运行正确性的标准是对并发事务运行的调度是否可串行化。

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。

事务对数据对象的加锁请求获准后，它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请求不能获准，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。

采用封锁法后可能会出现活锁、死锁等问题，其中必须解决的问题是由于事物之间的循环等待导致的死锁问题。

6、

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存储介质上的所有数据，恢复时在加载最近后备副本后需要重做最近后备副本以后提交的所有更新事务。

7、（Date类型怎么比较大小？）

(1)

CREATE TRIGGER sailor\_update\_check

AFTER UPDATE ON Sailors

REFERENCING NEW TABLE AS N

FOR EACH STATEMENT

WHEN(EXISTS(SELECT \* FROM N

WHERE YEAR(birth)>=1995))

INSERT

INTO YoungSailors

SELECT \* FROM N

WHERE YEAR(birth)>=1995;

(2)

CREATE TRIGGER day\_check

BEFORE INSERT ON Reserves

REFERENCING NEW AS N

FOR EACH ROW

WHEN( EXIST( SELECT \* FROM N

WHERE day<’2014-01-01’

OR day>’2015-01-12’) )

ROLLBACK;

8、

(1)

П年龄，性别（σ县名=Adair AND 1991=59(S)）

=

|  |  |
| --- | --- |
| 年龄 | 性别 |
| 2 | 2 |

(2)

R (连接c) T，c=（R.县名=T.县名）AND（T.人口=7558）

=

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 县名 | 土地面积 | 水域面积 | 性别 | 平均年龄 | 年龄范围 | 人口 | 种族 | 人口 |
|  |  |  |  |  |  |  |  |  |

(3)