

1. What are the differences between B Tree and B+ tree? Why do we use B+ tree in DBMS?

2. There are four relationships:

Student(sid, sname, did, sex, birthdate, gpa)

Course(cid, cname, preid, did, semester)

*/*preid indicates the course number of the pre-taken course*/*

Enroll(sid, cid, grade)

Dept(did, dname, phdstunum)

*/*phdstunum indicates the number of doctoral students in the department/*

Write an SQL statement that expresses the following query requirements (must be expressed in SQL statements):

(1) Use the join query to find the names of students who scored 90 or above in more than one course in the fall semester.

(2) Use the nested query to find the names of students who scored 90 or above in more than one course in the fall semester.

(3) Use the nested query with correlation to find the names of students who scored 90 or above in more than one course in the fall semester.

(4) Select the name and the corresponding course number of the student whose grade of each course is not lower than the average score of the course.

(5) Query the cid, cname and average GPA of students for each course offered by the Computer Department.

1. What are the differences between B Tree and B+ tree? Why do we use B+ tree in DBMS?

In the B+ tree, keys are the indexes stored in the internal nodes and records are stored in the leaf nodes. In B tree, keys cannot be repeatedly stored, which means that there is no duplication of keys or records. In B+ tree, the leaf nodes are linked to each other to provide the sequential access. In the B tree, leaf nodes are not linked to each other.

The B+ tree is a balanced binary search tree. B+ tree ensures that all leaf nodes remain at the same height. In the B+ tree, the leaf nodes are linked using a link list. Therefore, a B+ tree can support random access as well as sequential access.

2. There are four relationships:

Student(sid, sname, did, sex, birthdate, gpa)

Course(cid, cname, preid, did, semester)

*/*preid indicates the course number of the pre-taken course*/*

Enroll(sid, cid, grade)

Dept(did, dname, phdstunum)

*/*phdstunum indicates the number of doctoral students in the department/*

Write an SQL statement that expresses the following query requirements (must be expressed in SQL statements):

(1) Use the join query to find the names of students who scored 90 or above in more than one course in the fall semester.

```
SELECT sname
FROM Student S, Course C, Enroll E
WHERE S.sid=E.sid AND E.cid=C.cid AND E.grade>90 AND C.semester=' fall' ;
```

(2) Use the nested query to find the names of students who scored 90 or above in more than one course in the fall semester.

```
SELECT sname
FROM Student S
WHERE S.sid IN ( SELECT E.sid
FROM Course C, Enroll E
WHERE E.cid=C.cid AND E.grade>90 AND C.semester=' fall' );
```

(3) Use the nested query with correlation to find the names of students who scored 90 or above in more than one course in the fall semester.

```
SELECT sname
FROM Student S
WHERE EXISTS ( SELECT E.sid
FROM Course C, Enroll E
WHERE S.sid=E.sid AND E.cid=C.cid AND E.grade>90 AND C.semester=' fall' );
```

(4) Select the name and the corresponding course number of the student whose grade of each course is not lower than the average score of the course.

```
SELECT S.sname, E.cid
FROM Enroll E, Student S
WHERE E.sid = S.sid AND
S.sid NOT IN ( SELECT sid
FROM Enroll E1
WHERE grade < (SELECT AVG(grade)
FROM Enroll E2
WHERE E2.cid = E1.cid));
```

```
SELECT S.sname, E.cid
FROM Enroll E, Student S
WHERE E.sid = S.sid AND NOT EXISTS
( SELECT E1.cid
FROM Enroll E1
WHERE E1.sid = S.sid AND
grade < (SELECT AVG(grade)
FROM Enroll E2
WHERE E2.cid = E1.cid));
```

(5) Query the cid, cname and average GPA of students for each course offered by the Computer Department.

```
SELECT C.cid, C.cname, AVG(S.gpa)
FROM Student S, Course C, Enroll E, Dept D
WHERE S.sid = E.sid AND E.cid = C.cid AND C.did = D.did
AND D.dname='Computer Department'
GROUP BY C.cid, C.cname;
```