CISC 7510X Final Exam

For the below questions, use the following schema definition.

```
student(sid,name)
faculty(fid,name)
class(clsid,name)
course(courseid,clsid,semester,year,room,fid)
registration(sid,courseid,grade)
```

The above is a schema for a school. Students have an id and name, faculty have an id and name, classes have an id and name. Course is an instance of a particular class, and registration links student to the course.

1. (5 points) Find names of students who are registered for Databases in Spring 2017.
   (a) select s.name from student s inner join registration r using (sid) inner join course cr using (courseid) inner join class cls using (clsid) where cr.semester='Spring' and cr.year=2017 and cls.name='Databases'
   (b) select name from student inner join registration using (sid) inner join course using (courseid) inner join class cls using (clsid) where semester='Spring' and year=2017 and name='Databases'
   (c) select name from student inner join registration on (studentid) where registration.year=2017 and registration.semester='Spring';
   (d) select name from student, class, course, registration where student.sid=registration.rid and class.clsid=registration.clsid and registration.year=2017 and registration.semester='Spring'
   (e) Other:

2. (5 points) Find total students in Spring 2017 semester.
   (a) select count(*) from student inner join registration where semester='Spring' and year=2017
   (b) select count(*) from student where semester='Spring' and year=2017
   (c) select count(distinct sid) from registration inner join course using (courseid) where semester='Spring' and year=2017
   (d) select count(distinct name) from registration where semester='Spring' and year=2017
   (e) Other:

3. (5 points) Which course has most students?
   (a) select max(count(*)) from registration group by courseid
   (b) select courseid from registration group by courseid having count(*) >= all(select count(*) from registration group by courseid)
   (c) select max(cnt) from (select count(*) cnt from registration group by courseid)
   (d) select max(cnt) from (select count(distinct sid) from registration group by courseid)
   (e) Other:

4. (5 points) Find all students with incomplete grades (grade is 'INC')
   (a) select grade from registration where grade='INC';

(b) select sid from registration where grade=’INC’;

(c) select sid from student a inner join registration b on a.sid=b.sid where grade=’INC’;

(d) select sid from student a, registration b where a.sid=b.sid and b.grade=’INC’;

(e) Other:

5. (5 points) Find students who never failed a single class.

(a) select sid from registration where grade <> ’F’

(b) select sid from registration where min(grade) <> ’F’

(c) select sid from student s left join registration r on s.sid=r.sid and r.grade=’F’ where r.sid is null

(d) select sid from student s inner join registration r on s.sid=r.sid and r.grade!=’F’

(e) Other:

6. (5 points) Find courses that have more than 20 students.

(a) select count(*) from registration where count(*) > 20

(b) select courseid from registration group by courseid having count(*) > 20

(c) select courseid from registration inner join class using (clsid) having count(*) > 20

(d) select courseid from registration left join class using (clsid) having count(*) > 20

(e) Other:

7. (5 points) What percent of students retakes a class they failed?

(a) select count(grade!=’F’)/count(grade=’F’) from registration

(b) select count(b.*)/count(a.*) from registration a inner join registration b where a.courseid=b.courseid and a.sid=b.sid and a.grade=’F’ and b.grade!=’F’

(c) with a as (select courseid,clsid,sid,grade from course a inner join registration b using (courseid)) select sum(case when b.sid is null then 0.0 else 1.0 end)/sum(1.0)*100 from a left outer join a on a.clsid=b.clsid and a.sid=b.sid and a.courseid!=b.courseid and a.grade!=’F’

(d) with a as (select courseid,clsid,sid,grade from course a inner join registration b using (courseid)) select sum(case when b.sid is null then 0.0 else 1.0 end)/sum(1.0)*100 from a left outer join a on a.clsid=b.clsid and a.sid=b.sid and a.courseid!=b.courseid and a.grade=’F’

(e) Other:

8. (5 points) Find students who only get ’A’s:

(a) select sid from registration group by sid having min(grade)=max(grade) and max(grade)=’A’

(b) select sid from registration where grade=’A’

(c) select sid from registration inner join student using (sid) where grade=’A’

(d) select sid from registration left join class using (clsid) group by max(grade) having grade=’A’

(e) Other:

9. (5 points) Faculty with most students:

(a) select fid,count(*) from course inner join registration using (courseid) group by fid order by 2 desc limit 1

(b) select fid from course inner join registration using (courseid) group by fid having count(*) >= all(select count(*) from course inner join registration using (courseid) group by fid)

(c) select fid,count(*) over (partition by fid) cnt from registration where cnt >= count(*)

(d) select fid,count(*) c,max(count(*)) over () c2 from course inner join registration using (courseid) group by fid

(e) Other:

10. (5 points) Which database runs under Hadoop

    (a) Postgres

    (b) Hive

    (c) Oracle

    (d) MySQL

    (e) Other:

11. (5 points) The key thing behind Hadoop is

    (a) Horizontal scalability

    (b) Vertical scalability

    (c) Skewed scalability

    (d) Northeast scalability

    (e) Other:

12. (5 points) Authorization is:

    (a) verifying the password

    (b) ensuring user only does allowed things

    (c) includes double factor authentication

    (d) is applied only once upon login.

    (e) Other:

13. (5 points) What is the most appropriate index for student.sid field?

    (a) Btree Index

    (b) Bitmap Index

    (c) Clustered Index

    (d) Bitmap Clustered Index

    (e) Other:

14. (5 points) What is the most appropriate index for student.name field?

    (a) Btree Index

    (b) Bitmap Index

(c) Clustered Index

(d) Bitmap Clustered Index

(e) Other:

15. (5 points) What is the most appropriate index for registration.grade field?

    (a) Btree Index

    (b) Bitmap Index

    (c) Clustered Index

    (d) Bitmap Clustered Index

    (e) Other:

16. (5 points) The below code:

```
create table dates as
with recursive mydates (tdate) as (
    select cast('19000101' as date) tdate
    union all
    select cast(tdate + interval '1 day' as date) tdate
    from mydates where tdate <= '21000101'
)
select * from mydates;
```

    (a) Is invalid.

    (b) Will return all dates between 19000101 and 21000101

    (c) Will create a table with all dates between 19000101 and 21000101

    (d) Will never return.

    (e) Other:

17. (5 points) For "student inner join registration", and no indexes, most modern databases will perform:

    (a) merge join.

    (b) hash join.

    (c) indexed lookup join.

    (d) inner loop join.

    (e) Other:

18. (5 points) In general, on limited memory system, no indexes, and huge tables, what join type would perform best?

    (a) merge join.

    (b) hash join.

    (c) indexed lookup join.

    (d) inner loop join.

(e) Other:

19. (5 points) Partitions:

    (a) Are similar to views.
    (b) Are similar to temporary tables.
    (c) Allow for physical clustering of logically similar data.
    (d) All of the above.
    (e) Other:

20. (5 points) select ... from a left join b on a.key=b.key where b.value=2

    (a) Will return everything from a table and only matching records from b table.
    (b) Will return everything from a table and all records from b table.
    (c) Will return only matching records from both left and right tables.
    (d) Will return everything from right table and all matching records.
    (e) Other: