CISC 7510X Midterm Exam

For the below questions, use the following schema definition.

```
patient(pid,fname,lname,dob,email,street,city,state,zip)
doctor(did,fname,lname)
appointment(aid,pid,did,fromtim,totim,room)
patientstate(tim,pid,aid,did,state)
```

It’s a schema for a medical office. We have `patient` records, `doctor` records, `appointment`s for patients to see doctors at a certain time and location. These appointments could be created months in advance.

When a patient arrives for an appointment, medical office creates a record in the `patientstate` table with state=`’A’`. Other values for state are: `’D’` patient is being seen by a doctor, `’L’` patient left exam room, and `’X’` patient left the office.

1. (5 points) Find dob of patient Bob Johnson.

   (a) `select dob from patient where fname=’Bob’ and lname=’Johnson’;`

   (b) `select * from patient where (fname,lname)=(’Bob’,’Johnson’);`

   (c) `select dob from patient where name = ’Bob Johnson’;`

   (d) `select fname, lname from patient where fname=’Bob’ and lname=’Johnson’;`

   (e) Other:

2. (5 points) Find all patients (pid) who have appointment on November 1st, 2022.

   (a)
   ```
   select a.* from patient a
   inner join appointment b
   on a.pid=b.pid
   where fromtim>=cast(’2022-11-01’ as date) and
   fromtim<cast(’2022-11-02’ as date);
   ```

   (b)
   ```
   select pid from appointment
   where fromtim>=cast(’2022-11-01’ as date) and
   fromtim<cast(’2022-11-02’ as date);
   ```

   (c)
   ```
   select pid from patientstate
   where tim>=cast(’2022-11-01’ as date) and
   tim<cast(’2022-11-02’ as date) and
   state=’A’;
   ```

   (d)
   ```
   select p.pid from appointment
   natural inner join patient p
   where fromtim>=cast(’2022-11-01’ as date) and
   fromtim<cast(’2022-11-02’ as date);
   ```

   (e) Other:

3. (5 points) Find number of patients who showed up at medical office on November 1st, 2022.

(a)
```
select count(*) from appointment
where fromtim>=cast('2022-11-01' as date) and
fromtim<cast('2022-11-02' as date);
```

(b)
```
select count(a.*) from patient a
natural inner join appointment b
where fromtim>=cast('2022-11-01' as date) and
fromtim<cast('2022-11-02' as date);
```

(c)
```
select count(*) from patientstate
where state='A' and
tim>=cast('2022-11-01' as date) and
tim<cast('2022-11-02' as date);
```

(d)
```
select sum(case when state='D' then 1 else 0 end)
from patientstate
where tim>=cast('2022-11-01' as date) and
tim<cast('2022-11-02' as date);
```

(e) Other:

4. (5 points) For each state, find average age of patients.

(a) `select state, avg(dob) from patient group by state;`

(b) `select avg(dob) from patient group by state;`

(c) `select state, avg(age(dob)) from patient group by state;`

(d)
```
select state, avg(extract(years from age(dob))) from patient
group by state;
```

(e) Other:

5. (5 points) How many patients are currently being seen by a doctor?

(a)
```
select sum(case when state='D' then 1
when state='L' then -1
else 0 end)
from patientstate;
```

(b)
```
select count(*)
from patientstate where state='D';
```

(c)
```
select count(case when state='D' then 1 else 0 end)
from patientstate
where state='D';
```

(d)
```
select sum(case when state='D' then 1
when state='X' then -1
else 0 end) / sum(1.0) as prcnt
from patientstate
where tim=now();
```

(e) Other:

6. (5 points) Find all doctors (did) who have ever seen John Doe.

(a)
```
select a.did
from doctor a
inner join patientstate b
on a.pid=b.pid and b.state=’D’
where fname=’John’ and lname=’Doe’;
```

(b)
```
select distinct b.did
from patient a
inner join patientstate b
on a.pid=b.pid and b.state=’D’
where fname=’John’ and lname=’Doe’;
```

(c)
```
select distinct b.did
from patient a
inner join appointment b
on a.pid=b.pid
where fname=’John’ and lname=’Doe’;
```

(d)
```
select distinct did
from patientstate
where state=’D’ and fname=’John’ and lname=’Doe’;
```

(e) Other:

7. (5 points) Has patient John Jackson ever been seen by doctor Jack Johnson?

(a)
```
select ’Y’
from patient a
cross join doctor c
where a.did=c.did
(a.fname,a.lname)= (’John’,’Jackson’) and
(c.fname,c.lname)= (’Jack’,’Johnson’);
```

(b)
```
select 1
from patient a
inner join patientstate b
on a.pid=b.pid
inner join doctor c
on b.did=c.did
where
(c.fname,c.lname)= (’John’,’Jackson’) and
(a.fname,a.lname)= (’Jack’,’Johnson’);
```

(c)
```
select 1
from patient a
inner join patientstate b
on a.pid=b.pid
inner join doctor c
on b.did=c.did
where
(a.fname,a.lname)= (’John’,’Jackson’) and
(c.fname,c.lname)= (’Jack’,’Johnson’);
```

(d)
```
select 1
from patient a
inner join patientstate b
on a.pid=b.did
inner join doctor c
on b.pid=c.did
where
(a.fname,a.lname)= (’John’,’Jackson’) and
(c.fname,c.lname)= (’Jack’,’Johnson’);
```

(e) Other:

8. (5 points) List all future appointments for John Doe.

(a)
```
select b.*
from patient a
inner join appointment b
on a.pid=b.pid
where fname=’John’ and lname=’Doe’;
```

(b)
```
select lead(appointment) over () dt
from appointment
where name=’John Doe’;
```

(c)
```
select b.*
from appointment a
left outer join patient b
on a.pid=b.pid and b.fromtim>=now()
where (fname,lname)=(’Doe’,’John’);
```

(d)
```
select b.*
from patient a
inner join appointment b
on a.pid=b.pid and b.fromtim>=now()
where fname=’John’ and lname=’Doe’;
```

(e) Other:

9. (5 points) List all patients (pid) who had *more* than 10 appointments in 2022.

(a)
```
select pid
from appointment
where fromtim >= cast(’2022-01-01’ as date) and
fromtim < cast(’2023-01-01’ as date)
group by pid
having count(*) > 10;
```

(b)
```
select pid
from patient a
left outer join appointment b
on a.pid=b.pid
where fromtim >= cast(’2022-01-01’ as date) and
```

```
fromtim < cast(’2023-01-01’ as date)
group by pid
having count(*) > 10;
```

(c)

```
select a.pid
from patient a
left outer join appointment b
on a.pid=b.pid
where b.fromtim >= cast(’2022-01-01’ as date) and
b.fromtim < cast(’2023-01-01’ as date)
group by a.pid
having count(b.aid) > 10;
```

(c)

```
select b.*
from patient
left outer join appointment b
on a.pid=b.pid
left outer join doctor c
on b.did=c.did
where b.fromtim >= cast(’2022-01-01’ as date) and
b.fromtim < cast(’2023-01-01’ as date)
group by a.pid
having count(b.aid) > 10;
```

(e) Other:

10. (5 points) List all patients (pid) who had *less* than 10 appointments in 2022.

(a)

```
select pid
from appointment
where fromtim >= cast(’2022-01-01’ as date) and
fromtim < cast(’2023-01-01’ as date)
group by pid
having count(*) < 10;
```

(b)

```
select a.pid
from patient a
left outer join appointment b
on a.pid=b.pid
where b.fromtim >= cast(’2022-01-01’ as date) and
b.fromtim < cast(’2023-01-01’ as date)
group by a.pid
having count(b.aid) < 10;
```

(c)

```
select a.pid
from patient a
inner join appointment b
on a.pid=b.pid
where b.fromtim >= cast(’2022-01-01’ as date) and
b.fromtim < cast(’2023-01-01’ as date)
```

```
group by a.pid
having count(b.aid) < 10;
```

(d)

```
select *
from patient a
natural left outer join appointment b
where b.fromtim between
cast(’2022-01-01’ as date) and cast(’2023-01-01’ as date)
group by a.pid
having count(*) < 10;
```

(e) Other:

11. (5 points) List names of doctors who have no appointments for 2023.

(a)

```
select *
from appointment b
on b.fromtim>=cast(’2023-01-01’ as date) and
b.fromtim<cast(’2024-01-01’ as date)
where count(*)=0;
```

(b)

```
select a.fname, a.lname
from doctor a
left outer join appointment b
on a.did = b.did and b.fromtim>=cast(’2023-01-01’ as date)
and
b.fromtim<cast(’2024-01-01’ as date)
group by a.did
having count(*) = 0;
```

(c)

```
select a.fname, a.lname
from doctor a
left outer join appointment b
on a.did = b.did and b.fromtim>=cast(’2023-01-01’ as date)
and
b.fromtim<cast(’2024-01-01’ as date)
where b.aid is null;
```

(d)

```
select a.fname, a.lname
from doctor a
inner join appointment b
on a.did = b.did and b.fromtim>=cast(’2023-01-01’ as date)
and
b.fromtim<cast(’2024-01-01’ as date)
where b.aid is null;
```

(e) Other:

12. (5 points) Find zip code with most patients.

(a)

```
select zip,count(*) cnt from patient group by zip;
```

(b)

```
select zip,count(*) cnt from patient group by zip
order by 2 desc limit 1;
```

(c)
```
with stats as (
select zip,count(*) cnt, max( count(*) ) maxcnt
from patient group by zip),
select zip
from stats
where cnt = maxcnt;
```

(d)
```
with stats as (
select zip,count(*) cnt from patient group by zip),
mxcnt as (select max(cnt) as cnt from stats)
select zip
from stats natural inner join mxcnt;
```

(e) Other:

13. (5 points) Find the sickest patient: has been seen by most doctors.

(a)
```
with stats as (
select pid, count(distinct did) cnt,
max(count(distinct did)) over () maxcnt
from patientstate
where state=’D’
group by pid
)
select pid from stats where cnt = maxcnt;
```

(b)
```
with stats as (
select pid, count(distinct did) cnt
from patientstate
where state=’D’
group by pid
),
stats2 as (
select max(cnt) mx from stats)
select pid
from stats natural inner join stats2
on a.cnt=b.mx;
```

(c)
```
with stats as (
select pid, count(distinct did) cnt
from patient a
inner join patientstate b
on a.pid=b.pid
inner join doctor c
b.did=c.did
where state=’D’
group by pid
),
mxcnt as (
select max(cnt) cnt from stats)
select pid from stats natural inner join stats2;
```

(d)
```
select pid, count(distinct did) cnt
from patient
natural inner join patientstate
natural inner join doctor
where state=’D’
group by pid
having count(distinct did) >=
all(select count(distinct did)
from patientstate where state=’D’ group by pid);
```

(e) Other:

14. (5 points) What fraction of appointments are late (the patient does not show up on time, or does not show up at all).

(a)
```
select count(case when a.fromtim < coalesce(b.tim,a.totim)
then a.aid else null end)/count(*)
from appointment a
left outer join patientstate b
on a.aid=b.aid and b.state=’A’;
```

(b)
```
select sum(case when a.fromtim < coalesce(b.tim,a.totim)
then 1.0 else 0.0 end)/sum(1.0)
from appointment a
left outer join patientstate b
on a.aid=b.aid and b.state=’A’;
```

(c)
```
select count(*) / sum (1.0) as fraction
from appointment a
left outer join patientstate b
on a.aid=b.aid and b.state=’A’
where a.fromtim < coalesce(b.tim,a.totim);
```

(d)
```
with stats as (
select aid, max(case when fromtim < coalesce(b.tim,a.totim)
then 1.0 else 0.0 end) lt
from appointment a
left outer join patientstate b
on a.aid=b.aid and b.state=’A’
group by a.aid
)
select 100.0*sum(lt)/sum(1.0)
from stats;
```

(e) Other:

15. (5 points) Find all patients who arrived without an appointment.

(a)
```
select a.*
from patient a left outer join appointment b on a.pid=b.pid
where b.aid is null;
```

(b)
```
select *
from patient a inner join join appointment b on a.pid=b.pid
where b.aid is null;
```

(c)
```
select *
from patientstate where aid is null and state=’A’;
```

(d)
```
select *
from patient a inner join patientstate b
on a.pid=b.pid where b.aid is null ;
```

(e) Other:

16. (5 points) Find all instances when a room is double-booked (have more than one appointment at the same time).

(a)
```
select room
from appointment
group by fromtim, totim
having count(*) > 1;
```

(b)
```
select a.*, b.*
from appointment a
inner join appointment b
on a.room = b.room
where a.fromtim between b.fromtim and b.totim or
b.fromtim between a.fromtim and a.totim;
```

(c)
```
with stats as (
select room, fromtim as tim, 1 cnt from appointment
union all
select room, totim as tim, -1 cnt from appointment
),
stats2 as (
select room, tim, sum(cnt) cnt
from stats
group by room, tim
)
select *
from stats2
where cnt>1;
```

(d)
```
with stats as (
select room, fromtim as tim, 1 cnt from appointment
union all
select room, totim as tim, -1 cnt from appointment
),
stats2 as (
select room, tim, sum(cnt)
over (partition by room order by tim) cnt
from stats
) select * from stats2 where cnt>1;
```

(e) Other:

17. (5 points) Which doctors have the worst average waiting time? (the time between state=’A’ and state=’D’ is waiting).

(a)
```
with stats as (
select pid, aid, did,
max(case when state=’D’ then tim else null end) -
min(case when state=’A’ then tim else null end) duration
from patientstate
group by pid, aid, did
),
stats2 as (
select did, avg(duration) avgdurr,
max( avg(duration) ) over () maxavgdurr
from stats
group by did
)
select did
from stats2
where avgdurr = maxavgdurr;
```

(b)
```
select did
from appointment
where waitingtime >= max(state=’D’ - state=’A’);
```

(c)
```
with stats as (
select pid, a.totim - a.fromtim as duration
from appointment
),
stats2 as (
select max( avg(duration) ) over () maxavgdurr from stats
)
select *
from stats
cross join stats2
where duration = maxavgdurr;
```

(d)
```
with stats as (
select a.did, avg(a.tim - b.tim) duration
from patientstate a
inner join patientstate b
using(pid, aid, did)
where a.state=’A’ and b.state=’D’
),
mxdur as (
select max(duration) duration from stats
)
select did
from stats natural inner join mxdur;
```

(e) Other:

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

    (a) hash join.

    (b) merge join.

    (c) inner loop join.

    (d) indexed lookup join.

    (e) Other:

19. (5 points) Below query is identical to: `select a.*,b.val from T1 a left outer join T2 b on a.key=b.key and a.val!=b.val`

    (a) `select a.*,b.val from T1 a inner join T2 b on a.key=b.key and a.val!=b.val`

    (b)
    ```
    with TMP as (select a.*,b.val from T1 a left outer join T2
    b on a.key=b.key where a.val!=b.val)
    select a.* from TMP where a.val!=b.val
    ```

    (c)
    ```
    with TMP as (select a.*,b.val from T1 a inner join T2 b on
    a.key=b.key
    where a.val!=b.val)
    select a.*,b.val from T1 a left outer join TMP b on a.key=b.key
    ```

    (d) All of the above queries are identical.

    (e) None of the queries are identical to the question.

20. (5 points) The below code (tip: write out the first few output numbers):

    ```
    with recursive n(n) as (
        select 1 n union all
        select n+1 from n where n<1000
    )
    select a.n
    from n a
    where a.n % 2 > 0 and a.n % 3 = 0
    ```

    (a) Is invalid

    (b) Will create a table with all primes between 1 and 1000

    (c) Will produce all prime numbers between 1 and 1000

    (d) Will generate a list of numbers 1 to 1000

    (e) Other: