

## CISC 7510X Final Exam

For the below questions, use the following schema definition.

```
customer(custid,lname,fname,street,city,state,zip,dob)
device(devid,custid,type)
device_onoff(ts,devid,pwr)
device_chnl(ts,devid,chnl)
schedule(start_ts,end_ts,chnl,showid)
```

This is a schema for a tv network. Customers own devices (of a certain type; phone is a device, cable-box is a device, etc.), and they can watch channels on those devices. Whenever a device comes online, we get a device\_onoff event, indicating device power state. Whenever a device channel changes, we get a device\_chnl event. There is also a schedule object, which tells us start and end times of shows on channels.

Pick the *best answer* that fits the question. Not all of the answers may be correct. If none of the answers fit, write your own answer. There are at most 2 questions where writing your own answer may be appropriate.

1. (5 points) Find address of customers named John Doe.
  - (a) select custid,fname,lname,dob from customer where (fname,lname)=(‘John’,‘Doe’)
  - (b) select \* from customer where (lname,fname)=(‘John’,‘Doe’)
  - (c) select custid,dob from customer where fname=‘John’ and lname=‘Doe’
  - (d) select custid,street,city,state,zip from customer where fname=‘John’ and lname=‘Doe’
  - (e) Other:
2. (5 points) Find customers who have more than 1 device.
  - (a) select custid,count(\*) from device group by custid having count(\*) > 1
  - (b) select a.custid,count(\*) from customer a inner join device b using(custid) group by a.custid where count(\*) > 1
  - (c) select b.custid,count(\*) from customer a left outer join device b on a.custid=b.custid group by b.custid having count(\*) > 1
  - (d) select custid,sum(case when b.custid is not null then 1 else 0 end) from customer a left outer join device b using (custid) group by custid
  - (e) Other:
3. (5 points) Find customers who have less than 4 devices.
  - (a) select custid,count(\*) from device group by custid having count(\*) < 4
  - (b) select a.custid,count(\*) from customer a inner join device b using(custid) group by a.custid having count(\*) < 4
  - (c) select a.custid,count(\*) from customer a left outer join device b on a.custid=b.custid group by b.custid having count(\*) < 4
  - (d) select custid,sum(case when b.custid is not null then 1 else 0 end) from customer a left outer join device b using (custid) group by custid having sum(case when b.custid is not null then 1 else 0 end) < 4

- (e) Other:
4. (5 points) Count of customers by state?
- (a) `select state,count(*) from customer group by state`
- (b) `select zip,count(*) from customer group by zip`
- (c) `select state,count(*)  
from customer natural inner join device  
where type='NY' group by state`
- (d) `with device_state as (  
select custid, case when pwr=1 then 'on' else 'off' end as state  
from device_onoff natural inner join customer)  
select b.state,count(*)  
from customer a left outer join device_state b  
on a.custid=b.custid  
group by b.state`
- (e) Other:
5. (5 points) Count of customers by age group, where age 0-30 is "A", 31-50 is "B", 51-70 is "C", and "D" for older.
- (a) `select extract(years from age(dob)) grp,count(*) from customer group by extract(years from age(dob))`
- (b) `with agegrp as ( select case when extract(years from age(dob))<=30 then 'A' when extract(years from age(dob))<=50 then 'B' when extract(years from age(dob))<=70 then 'C' else 'D' end g from age) select g,count(*) from agegrp`
- (c) `with age as (select extract(years from age(dob)) a from customer ), agegrp as ( select case when a<=30 then 'A' when a<=50 then 'B' when a<=70 then 'C' else 'D' end g from age) select g,count(*) from agegrp group by g`
- (d) `with age as (select age(dob) a from customer ), agegrp as ( select case when a<=30 then 'A' when a<=50 then 'B' when a<=70 then 'C' else 'D' end g from age) select g,count(*) from agegrp group by g`
- (e) Other:
6. (5 points) What percentage of customers live in NY tri-state area (NY,NJ,CT)?
- (a) `select 100*sum(case when state in ('NY','NJ','CT') then 1.0 else 0.0 end)/sum(1.0)  
from customer  
where state in ('NY','NJ','CT')  
group by state  
having state in ('NY','NJ','CT')`
- (b) `select 100*sum(case when state in ('NY','NJ','CT') then 1.0 else 0.0 end)/sum(1.0)  
from customer  
group by state`
- (c) `select 100*sum(case when state in ('NY','NJ','CT') then 1.0 else 0.0 end)/sum(1.0)  
from customer`

- (d) `select 100*sum(case when state in ('NY','NJ','CT') then 1.0 else 0.0 end)/sum(1.0)`  
`from customer`  
`where (case when state in ('NY','NJ','CT') then 'NYTRI' else 'NOT' end)='NYTRI'`  
`group by case when state in ('NY','NJ','CT') then 'NYTRI' else 'NOT' end`  
`having count(*)>0`
- (e) Other:
7. (5 points) Create a table **newcustomers** of all new customers (those who subscribed within last 30 days). Assume that customer turns on their device as soon as they subscribe.
- (a) `create table newcustomers as with firston as (`  
`select devid,min(ts) firston`  
`from device_onoff`  
`where pwr=1`  
`group by devid ),`  
`firstdevice as (`  
`select custid,min(firston) firstdevts`  
`from device a inner join firston b on a.devid=b.devid`  
`group by custid`  
`)`  
`select custid`  
`from firstdevice`  
`where extract( days from now() - firstdevts ) < 30`
- (b) `create table newcustomers as select custid`  
`from device natural inner join device_onoff`  
`where pwr=1`  
`group by custid`  
`having extract( days from now() - min(ts) ) < 30`
- (c) `create table newcustomers as select custid`  
`from customer`  
`where extract(days from now() - dob) < 30`
- (d) `create table newcustomers as select a.custid`  
`from customer a`  
`left outer join device b on a.custid=b.custid`  
`left outer join device_onoff c on b.devid=b.devid`  
`group by a.custid`  
`having extract( days from now() - max(ts) ) > 30`
- (e) Other:
8. (5 points) Create a table **custgainbyzip**, representing count of new customers (those subscribed within last 30 days) for each zip code.
- (a) `create table custgainbyzip as select zip,count(*) cnt`  
`from customer group by zip`
- (b) `create table custgainbyzip as`  
`select zip,sum(case when b.custid is null then 1 else 0 end) cnt`  
`from customer a left outer join newcustomers b on a.custid=b.custid`  
`group by zip`

- (c) create table custgainbyzip as  
 select zip, sum( case when b.custid is not null then 1 else 0 end)  
 over (partition by zip) cnt  
 from customer natural left outer join newcustomers b
- (d) create table custgainbyzip as select a.zip, count(\*) cnt  
 from customer a inner join newcustomers b on a.custid=b.custid  
 group by a.zip
- (e) Other:
9. (5 points) Zip codes can be ranked by new customer gains (see previous question). Find zip codes that are within the top 10 ranks.
- (a) select zip  
 from custgainbyzip  
 order by cnt desc  
 limit 10
- (b) select zip  
 from ( select zip, rank() over (order by cnt desc) rnk from custgainbyzip ) a  
 where rnk <= 10
- (c) select zip  
 from ( select zip, dense\_rank() over (order by cnt desc) rnk from custgainbyzip ) a  
 where rnk <= 10
- (d) select zip  
 from ( select zip, row\_number() over (order by cnt desc) rn from custgainbyzip ) a  
 where rn <= 10
- (e) Other:
10. (5 points) How many customers were watching channel 4 on 2020-03-30, at 8PM? (note, the device has to be on, and tuned to channel 4).
- (a) with poweron as (  
 select ts,devid, pwr, lead(ts) over (partition by devid order by ts) next\_ts  
 from device\_onoff ),  
 ch as (  
 select ts,devid, chnl, lead(ts) over (partition by devid order by ts) next\_ts  
 from device\_chnl )  
 select count(distinct custid)  
 from device a  
 inner join poweron b  
 on a.devid=b.devid and b.pwr=1 and  
 b.ts<= cast('2020-03-30 20:00:00' as timestamp) and  
 b.next\_ts > cast('2020-03-30 20:00:00' as timestamp)  
 inner join ch c on a.devid=c.devid and c.chnl=4 and  
 c.ts<= cast('2020-03-30 20:00:00' as timestamp) and  
 c.next\_ts > cast('2020-03-30 20:00:00' as timestamp)
- (b) select count(distinct custid)  
 from device a  
 inner join device\_onoff b on a.devid=b.devid and b.pwr=1 and

```
b.ts=cast('2020-03-30 20:00:00' as timestamp)
inner join device_chnl c on a.devid=c.devid and c.chnl=4 and
c.ts=cast('2020-03-30 20:00:00' as timestamp)
```

- (c) 

```
select count(distinct custid)
from device a
left outer join device_onoff b on a.devid=b.devid
left outer join device_chnl c on a.devid=c.devid
where b.pwr=1 and b.ts=cast('2020-03-30 20:00:00' as timestamp) and
c.chnl=4 and c.ts=cast('2020-03-30 20:00:00' as timestamp)
```
- (d) 

```
with events as (
select ts, devid, case when pwr=1 then 1 else -1 end pwr, null chnl from device_onoff
union all
select ts, devid, 0 pwr, chnl from device_chnl),
eventseq as (
select ts, devid,
sum(pwr) over (partition by devid order by ts) pwr,
chnl,
lead(ts) over (partition by devid order by ts) next_ts
from events)
select count(distinct custid)
from device a
inner join eventseq b
on a.devid=b.devid and b.pwr=1 and b.chnl=4 and
b.ts >= cast('2020-03-30 20:00:00' as timestamp) and
b.next_ts <= cast('2020-03-30 20:00:00' as timestamp)
```
- (e) Other:

11. (5 points) What percentage of customers have more than 2 devices?

- (a) 

```
with cnts as (
select custid,sum( case when b.custid is null then 0 else 1 end ) cnt
from customer a
inner join device b using (custid)
group by custid)
select 100*sum(case when cnt>2 then 1.0 else 0.0 end)/sum(1.0) prcnt from cnts
```
- (b) 

```
with cnts as (
select custid,sum( case when b.custid is null then 0 else 1 end ) cnt
from customer a
left outer join device b using (custid)
group by custid)
select 100*sum(case when cnt>2 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts
```
- (c) 

```
with cnts as (
select custid,sum( case when b.custid is null then 0 else 1 end ) cnt
from customer a
left outer join device b
on a.custid=b.custid
group by custid)
```

```
select 100*sum(case when cnt>2 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts
```

(d) with cnts as (  
select custid,count(\*) cnt  
from device  
group by custid)  
select 100\*sum(case when cnt>2 then 1.0 else 0.0 end)/sum(1.0) prcnt  
from cnts

(e) Other:

12. (5 points) It is possible there are errors in the schedule. Find instances when more than one show is scheduled for the same time on the same channel. Note, that start time could be different. e.g. an instance of the problem might look like: show1 runs from 1-2pm, and show2 runs from 1:15pm till 1:30pm on the same channel.

(a) select \*  
from schedule a  
inner join schedule b  
using (chnl,showid)  
where  
a.start\_ts between b.start\_ts and b.end\_ts

(b) select \*  
from schedule a  
inner join schedule b  
using (chnl,start\_ts,end\_ts)  
where a.showid != b.showid

(c) select \*  
from schedule a  
inner join schedule b  
on a.chnl=b.chnl and a.showid=b.showid and  
(a.start\_ts between b.start\_ts and b.end\_ts or  
b.start\_ts between a.start\_ts and a.end\_ts)

(d) with start\_end as (  
select 1 c, start\_ts as tim, chnl, showid from schedule  
union all  
select -1 c, end\_ts as tim, chnl, showid from schedule ),  
cnts as (  
select chnl,showid,tim, sum(c) over (partition by chnl order by tim) cnt  
from start\_end )  
select \* from cnts where cnt>1

(e) Other:

13. (5 points) Find empty gaps in the schedule that last longer than 1 second, for '2020-05-18'. e.g. show1 ends at 1:55pm, and next show on that tv channel starts at 2:00pm. If there are no shows on a TV channel for the entire day, then that entire day is an "empty gap".

(a) select \*  
from schedule a

```

inner join schedule b
on a.chnl=b.chnl and
extract(epoch from a.end_ts-b.start_ts) > 1
where cast(a.start_ts as date)=cast('2020-05-18' as date) and
cast(b.start_ts as date)=cast('2020-05-18' as date)

```

- (b) with nxstart as (  
 select a.\*, lead(start\_ts) over (partition by chnl order by start\_ts) next\_start  
 from schedule a  
 where cast(start\_ts as date)=cast('2020-05-18' as date) )  
 select \* from nxstart  
 where extract(epoch from next\_start-end\_ts) > 1
- (c) with chnls as (select chnl from schedule group by chnl),  
 sq as (  
 select cast('2020-05-18' as timestamp) ts, 1 c, chnl from chnls  
 union all  
 select cast('2020-05-19' as timestamp) ts, -1 c, chnl from chnls  
 union all  
 select start\_ts ts, 1 c, chnl from schedule  
 union all  
 select end\_ts ts, -1 c, chnl from schedule ),  
 cnts as (  
 select chnl,ts, sum(c) over (partition by chnl order by ts) cnt,  
 lead(ts) over (partition by chnl order by ts) next\_ts  
 from sq )  
 select \* from cnts where cnt=1 and extract(epoch from next\_ts-ts) > 1
- (d) with sq as (  
 select start\_ts ts, 1 c, chnl from schedule  
 where cast(start\_ts as date)=cast('2020-05-18' as date)  
 union all  
 select end\_ts ts, -1 c, chnl from schedule  
 where cast(start\_ts as date)=cast('2020-05-18' as date) ),  
 cnts as (  
 select chnl,ts, sum(c) over (partition by chnl order by ts) cnt  
 from sq )  
 select \* from cnts where cnt=0 and extract(epoch from ts) > 1
- (e) Other:

14. (5 points) Create a table `device_show(ts,devid,chnl,showid)`, which looks up the currently playing show on chnl for each device.

- (a) create table device\_show as  
 with dshow as (  
 select ts,devid,chnl,null showid from device\_chnl  
 union all  
 select start\_ts,null,chnl,showid from schedule),  
 grps as (  
 select a.\*, sum(case when showid is not null then 1 else 0 end)  
 over (partition by chnl order by ts

```

rows between unbounded preceding and current row) grp
from dshow a ),
mx as (
select ts,devid,chnl,
max(showid) over (partition by chnl,grp) showid
from grps)
select ts,devid,chnl,showid from mx where devid is not null

```

- (b) create table device\_show as  
with dshow as (  
select ts,devid,chnl,null showid from device\_chnl  
union all  
select start\_ts,null,chnl,showid from schedule),  
grps as (  
select a.\*, sum(case when showid is not null then 1 else 0 end)  
over (partition by chnl order by ts  
rows between unbounded preceding and current row) grp  
from dshow a ),  
mx as (  
select ts,devid,chnl,  
max(showid) over (partition by chnl,grp) showid  
from grps)  
select ts,devid,chnl,showid from mx where devid is not null  
union all  
select a.ts, b.devid, a.chnl, a.showid  
from grps a  
inner join grps b  
on a.devid is null and b.devid is not null and  
a.chnl=b.chnl and a.grp=b.grp+1
- (c) create table device\_show as  
select distinct a.ts,a.devid,a.chnl,b.showid  
from device\_chnl a  
inner join schedule b  
on a.chnl=b.chnl and  
a.ts between b.start\_ts and b.end\_ts
- (d) create table device\_show as  
select b.start\_ts,a.devid,a.chnl,b.showid  
from device\_chnl a  
inner join schedule b  
on a.chnl=b.chnl and  
a.ts between b.start\_ts and b.end\_ts
- (e) Other:

15. (5 points) Using the `device_show` table, find the top 10% most popular shows (watched by most customers).

- (a) with cnts as (  
select showid,count(distinct custid) cnt  
from device\_show a



```

inner join device b
on a.devid=b.devid
group by showid ),
rnks as (
select a.*, dense_rank() over (order by cnt desc) r
from cnts a)
select * from rnks where r<=10

```

- (b) with cnts as (

```

select showid,count(distinct custid) cnt
from device_show a
inner join device b
on a.devid=b.devid
group by showid ),
rnks as (
select a.*, rank() over (order by cnt)/count(*) over () r
from cnts a)
select * from rnks where r<=0.10

```
- (c) with cnts as (

```

select showid,count(distinct custid) cnt
from device_show a
inner join device b
on a.devid=b.devid
group by showid ),
rnks as (
select a.*, row_number() over (order by cnt desc)/sum(1.0) over () pr
from cnts a)
select * from rnks where pr<=0.10

```
- (d) with cnts as (

```

select showid,count(*) cnt
from device_show a
inner join device b
on a.devid=b.devid
group by showid ),
rnks as (
select a.*, row_number() over (order by cnt desc)/sum(1.0) over () pr
from cnts a)
select * from rnks where pr<=0.10

```
- (e) Other:

16. (5 points) Using the `device_show` table, build `customer_show(custid,showid,score)` table, which will have a record if a customer has ever watched a particular show, where `score` is count of how often customer's device is tuned to the tv show.

- (a) create table `customer_show` as select `custid,showid,count(*) score` from `device_show` natural inner join `device` group by `custid,showid`
- (b) create table `customer_show` as select `custid,showid,count(*) score` from `device` left outer join `device_show` using (`devid`) group by `custid,showid`

- (c) create table customer\_show as select custid,max(showid) showid,count(\*) score from device inner join device\_show using (devid) group by custid
- (d) create table customer\_show as select max(custid) custid,showid,count(\*) score from device inner join device\_show using (devid) group by showid
- (e) Other:
17. (5 points) For each customer, identify the most watched TV show.  
Create a table most\_watched(custid,showid).
- (a) create table most\_watched as select custid,max(showid) showid from customer\_show a group by custid
- (b) create table most\_watched as with rnk as ( select a.\*,row\_number() over (partition by custid order by score) r from customer\_show a ) select custid,showid from rnk where r=1
- (c) create table most\_watched as with rnk as ( select a.\*,dense\_rank() over (partition by custid order by score desc) r from customer\_show a ) select custid,showid from rnk where r=1
- (d) create table most\_watched as with rnk as ( select a.\*,dense\_rank() over (partition by custid order by score) r from customer\_show a ) select custid,showid from rnk where r=1
- (e) Other:
18. (5 points) The below code (tip: write out the first few output numbers):
- ```
with recursive n(n) as (
    select 2 n union all
    select n+1 from n where n<1000
)
select a.n
from n a inner join n b on b.n < sqrt(a.n)+1
group by a.n
having a.n=2 or min(a.n % b.n) > 0 order by 1
```
- (a) Is invalid
- (b) Will generate a list of numbers 1 to 1000
- (c) Will generate a list of all primes between 1 and 1000
- (d) Will generate a list of all odd numbers.
- (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) 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
- (b) 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
- (c) select a.\*,b.val from T1 a inner join T2 b on a.key=b.key and a.val!=b.val

- (d) All of the above queries are identical.
- (e) None of the queries are identical to the question.

20. (5 points) When you write:

```
select * from T1 a inner join T2 b on a.tim between b.start and b.end
```

what is the expected performance?

- (a) Hash join, approximately  $O(N \log N)$ , where  $N$  is the number of records in both T1 and T2.
- (b) Sort merge join, approximately  $O(N)$ , where  $N$  is the number of records in both T1 and T2.
- (c) Inner loop join, approximately  $O(N^2)$ , where  $N$  is the number of records in both tables.
- (d) Distributed hash join, approximately  $O(N)$  to distribute data, and  $O(N \log N)$  after distribution.
- (e) Other: