CISC 7510X Midterm Exam

Answers must be emailed in plain text (no formatting, no attachments). Email *must* have your *full name* at the *top*. Answers to questions must be clearly marked (question number before each answer), and be in sequence (question 1 should come before question 2, etc.).

For the below questions, use the following schema definition.

```
customer(cid, fname, lname, street,city,state,zip, dob)
service(sid, sdate, cid, cost, pckgid )
invoice(iid, idate, cid, dueamnt )
payment(pid, pdate, cid, payamnt, iid )
```

This is a schema for a phone network. Customers sign up for a service package, get billed monthly, make payments, etc.

The `customer` table has all the customers: `cid` is customer id.

The `service` table has services that were provided to customers: `sid` is a unique service identifier, `sdate` is date when service was provided, `cid` identifies the customer that recieved the service. `cost` is how much the customer was charged, and `pckgid` is what service package the customer got.

The `invoice` table has the monthly invoices that the company sends out. The `idate` is the invoice date, `cid` is the customer who got invoiced, and `dueamnt` is how much the customer needs to pay.

The `payment` table records customer's payments. `pdate` is when the payment was made. `cid` identifies customer who made the payment, and `payamnt` is how much they paid. The `iid`, if populated, identifies which invoice the customer is paying.

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.

1. (5 points) Find address of customers named John Doe with date-of-birth December 31, 1999.

   (a)
   ```
   select * from service
   where (fname,fname) = ('John','Doe') and dob='December 31, 1999'
   ```

   (b)
   ```
   select * from customer
   where (fname,fname) = ('John','Doe') and dob='December 31, 1999'
   ```

   (c)
   ```
   select cid,dob from customer
   where fname='John' and lname='Doe' and dob=cast('1999-12-31' as date)
   ```

   (d)
   ```
   select cid,street,city,state,zip from customer
   where fname='John' and lname='Doe' and dob=cast('1999-12-31' as date)
   ```

   (e) Other:

2. (5 points) Find customers who have recieved more than 2 services.

   (a)
   ```
   select cid
   from service
   group by cid having count(*) > 2
   ```

   (b)
   ```
   select cid
   from customer
   natural inner join service
   group by cid having count(*) > 2
   ```

   (c)
   ```
   select cid
   from customer
   natural inner join service
   natural inner join invoice
   natural inner join payment
   group by cid having count(*) > 2
   ```

   (d)
   ```
   select cid from service
   natural inner join invoice
   natural inner join payment
   group by cid having count(*) > 2
   ```

   (e) Other:

3. (5 points) Find customers who have recieved less than 2 services.

(a)
```
select cid
from service
group by cid having count(*) < 2
```

(b)
```
select cid
from customer
natural left outer join service
group by cid having count(sid) < 2
```

(c)
```
select cid
from customer
natural left join service
natural inner join invoice
natural inner join payment
group by cid having count(*) < 2
```

(d)
```
select cid from service
natural left outer join invoice
natural left outer join payment
group by cid having count(*) < 2
```

(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 service
where type='NY' group by state
```

(d)
```
with srvcs as (
select cid, case when cost is not null then 1 else 0 end as flag
from service group by cid )
select a.state, count(cid)
from customer a
left outer join srvcs b
using (cid)
group by a.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) Identify customers who submit more than one paymnent for a single invoice.

(a)
```
select a.cid
from customer a
inner join invoice b
on a.cid=b.cid
inner join payment c
on a.cid=c.cid
group by a.cid,b.iid
having count(*) > 1
```

(b)
```
select distinct cid
from payment
where iid is not null
group by cid,iid
having count(*)>1
```

(c)
```
select a.cid
from invoice a
inner join payment b
on a.iid=b.iid
group by a.cid
having count(*) > 1
```

(d)
```
select a.cid
from payment a
left outer join invoice b
on a.iid=b.iid and a.cid=b.cid
where a.iid is not null
group by a.cid
having count(*) > 1
```

(e) Other:

8. (5 points) Create a table `newcustomers` of all new customers (those whose first service was within last 30 days).

(a)
```
create table newcustomers as
select cid
from customer natural inner join service
where extract( days from now() - min(sdate) ) < 30
```

(b)
```
create table newcustomers as
select cid
from service
where extract( days from now() - max(sdate) ) < 30
```

(c)
```
create table newcustomers as
select cid
from service
where extract( days from now() - sdate ) < 30
```

(d)
```
create table newcustomers as
select cid
from service
group by cid
having extract( days from now() - min(sdate) ) < 30
```

(e) Other:

9. (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.cid is null then 1 else 0 end) cnt
from customer a left outer join newcustomers b on a.cid=b.cid
group by zip
```

(c)
```
create table custgainbyzip as
select zip, sum( case when b.cid 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.cid=b.cid
group by a.zip
```

(e) Other:

10. (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:

11. (5 points) Each service extends 30 days, starting at `sdate`. How many customers had service `pckgid=12345` on July 4th, 2020?

    (a)
    ```
    with enddate as (
    select cid, pckgid, sdate, sdate + 30 as edate
    from service )
    select a.cid, count(*)
    from customer a
    inner join enddate b
    on a.cid=b.cid and cast('2020-07-04' as date) between b.sdate and b.edate
    where b.pckgid=12345
    ```

    (b)
    ```
    with enddate as (
    select cid, pckgid, sdate, sdate + 30 as edate
    from service )
    select count(distinct cid)
    from service a
    inner join enddate b
    on a.cid=b.cid and cast('2020-07-04' as date) between b.sdate and b.edate
    where b.pckgid=12345
    group by sid
    ```

    (c)
    ```
    select count(distinct cid)
    from service a
    where pckgid=12345 and cast('2020-07-04' as date) - sdate between 0 and 30
    ```

    (d)
    ```
    select count(*)
    from service a
    where pckgid=12345 and sdate between cast('2020-07-04' as date) and 30
    ```

    (e) Other:

12. (5 points) Select customer payments that do not have an invoice, or do not match the invoiced amount.

    (a)
    ```
    select a.cid, a.payamnt, b.dueamnt
    from payment a
    inner join invoice b
    using (cid,iid)
    where b.dueamnt != a.payamnt
    ```

    (b)
    ```
    select a.cid, a.payamnt, b.dueamnt
    from payment a
    left outer join invoice b
    on a.cid=b.cid and a.iid=b.iid
    where b.dueamnt is null or b.dueamnt != a.payamnt
    ```

    (c)
    ```
    select a.cid, a.payamnt, b.dueamnt
    from payment a
    natural inner join invoice b
    where b.dueamnt != a.payamnt or b.dueamnt is null
    ```

    (d)
    ```
    select a.cid, a.payamnt, b.dueamnt
    from payment a
    inner join invoice b
    on a.cid=b.cid and a.iid=b.iid and
    (b.dueamnt is null or b.dueamnt != a.payamnt)
    ```

    (e) Other:

13. (5 points) For each customer, calculate current unpaid balance.

    (a)
    ```
    with evts as (
    select cid,cost as amnt from service
    union all
    select cid,-payamnt as amnt from payment)
    select cid,sum(amnt) bal
    ```

```
from evts
group by cid
having sum(amnt)!=0
```

(b)
```
with evts as (
select cid,dueamnt as amnt from invoice
union all
select cid,-payamnt as amnt from payment )
select cid,sum(amnt) bal
from evts
group by cid
having sum(amnt)!=0
```

(c)
```
select (select sum(cost) from service) - (select sum(payamnt) from payment)
```

(d)
```
select a.cid, sum(case when b.iid is not null then -dueamnt
when c.iid is not null then payamnt else 0 end) bal
from service a
natural inner join invoice b
natural inner join payment c
group by a.cid
having sum(case when b.iid is not null then -dueamnt
when c.iid is not null then payamnt else 0 end) != 0
```

(e) Other:

14. (5 points) Find unpaid invoices (note that if a payment follows an invoice then that invoice is technically "paid").

(a)
```
select a.iid
from invoice a
left outer join payment b
on a.cid=b.cid and a.iid=b.iid
where b.pid is null
```

(b)
```
with stuff as (
select 0 t,idate,cid,iid, dueamnt from invoice union all
select 1 t,pdate,cid,iid, payamnt from payment),
lag as (
select a.*, lag(t) over (partition by cid order by idate) lt
from stuff a)
select *
from lag
where t=0 and lt=1
```

(c)
```
with mx as (
select cid,max(pdate) m from payment group by cid )
select a.*
from invoice a
left outer join mx b
on a.cid=b.cid and a.idate > b.m
```

(d)
```
with mx as (
select cid,max(pdate) m from payment group by cid)
select a.*
from invoice a
left outer join mx b
on a.cid=b.cid
where a.idate > b.m
```

(e) Other:

15. (5 points) Identify instances when a payment is made prior to service?

(a)
```
with stuff as (
select pdate, cid, payamnt, null from payment union all
select sdate, cid, null, cost from service ),
```

```
stats as (
select a.*,
sum(payamnt) over (partition by cid order by pdate) paysum,
sum(cost) over (partition by cid order by pdate) duesum
from stuff a)
select * from stats where paysum > duesum
```

(b)
```
with stuff as (
select pdate, cid, payamnt from payment union all
select sdate, cid, -cost from service ),
stats as (
select a.*,sum(payamnt) over (partition by cid order by pdate) bal
from stuff a)
select * from stats where bal>0
```

(c)
```
select a.*
from payment a
inner join service b
on a.cid=b.cid and a.pdate < b.sdate
```

(d)
```
select a.*
from payment a
left outer join service b
on a.cid=b.cid and a.pdate > b.sdate
where b.sid is null
```

(e) Other:

16. (5 points) Identify customers who have never made a payment.

(a)
```
select distinct cid
from customer a
left outer join payment b
using (cid)
where b.pid is null
```

(b)
```
select cid
from customer a
left outer join payment b
using (cid)
where b.pid is null
group by cid
having count(*) < 1
```

(c)
```
select a.*
from customer a
left outer join invoice b
on a.cid=b.cid
left outer join payment c
on b.cid=c.cid and b.iid=c.iid
where b.iid is null or c.pid is null
```

(d)
```
select cid from customer except
select cid from payment
```

(e) Other:

17. (5 points) From all the services that are currently active (not expired), identify 5 most popular packages (top 5 pckgid, ranked by count of customers).

(a)
```
with cnts as (
select pckgid,count(distinct cid) cnt from service
where extract(days from now() - sdate) <= 30
group by pckgid),
rnk as (
select a.*, dense_rank() over (order by cnt desc) r from cnts a)
select * from rnk where r<=5
```

(b)
```
with cnts as (
    select pckgid,count(distinct cid) cnt from service
    where extract(days from now() - sdate) <= 30
    group by pckgid),
    rnk as (
    select a.*, rank() over (order by cnt) r from cnts a)
    select * from rnk where r<=5
```

(c)
```
with cnts as (
    select pckgid,count(distinct cid) cnt from service
    where now() between sdate and sdate+30
    group by pckgid),
    rnk as (
    select a.*, row_number() over (order by cnt desc) rn from cnts a)
    select * from rnk where rn<=5
```

(d)
```
with cnts as (
    select pckgid,count(distinct cid) cnt from service
    where now() between sdate and sdate+30
    group by pckgid),
    rnk as (
    select a.*, count(*) over () c from cnts a)
    select * from rnk where c<=5
```

(e) Other:

18. (5 points) The below code (tip: run the code):

```
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 odd numbers divisible by 3.

(d) Will generate a list of all primes between 1 and 1000

(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 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
```

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

(c)
```
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
```

(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 $O(NlogN)$, where N is the number of records in both T1 and T2.

(b) Inner loop join, approximately $O(N^2)$, where N is the number of records in both tables.

(c) Sort merge join, approximately $O(N)$, where N is the number of records in both T1 and T2.

(d) Distributed hash join, approximately $O(N)$ to distribute data, and $O(NlogN)$ after distribution.

(e) Other: