

CISC 7510X Final Exam

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.

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.).

Email must arrive by midnight on 2020-12-17.

For the below questions, use the following schema definition.

```
customer(cid, fname, lname, dob, street, city, state, zip)
product(pid, description, listprice)
ordr(oid, tim, cid)
ordrdtls(odid, oid, pid, price, qty)
advertisement(aid, tim, cid, pid, type)
```

This is a schema for a store. The `customer` table has customer information, the `product` table has product information. The `ordr` table has information on individual customer orders, with `ordrdtls` table for products that are part of an order. Think of an online order, where a single purchase has multiple products. The `advertisement` is when we advertise product *pid* to customer *cid*.

1. (5 points) Find address of customers named John Doe.
 - (a)

```
select cid,fname,lname,dob
from customer
where (fname,lname)=('John','Doe')
```
 - (b)

```
select * from customer
where (lname,fname)=('John','Doe')
```
 - (c)

```
select cid,dob
from customer where fname='John' and lname='Doe'
```
 - (d)

```
select cid,street,city,state,zip
from customer where fname='John' and lname='Doe'
```
 - (e) Other:

2. (5 points) Find customers who have more than 10 orders?
 - (a)

```
select cid,count(*) from ordr
group by cid having count(*) > 10
```
 - (b)

```
select a.cid,count(*) from
customer a inner join ordr b using(cid)
group by a.cid where count(*) > 10
```
 - (c)

```
select b.cid,count(*)
from customer a left outer join ordr b
on a.cid=b.cid group by b.cid having count(*) > 10
```
 - (d)

```
select cid,sum(case when b.cid is not null then 1 else 0 end)
from customer a left outer join ordr b using (cid)
group by cid
having sum(case when b.cid is not null then 1 else 0 end) >= 10
```
 - (e) Other:

3. (5 points) Find customers who have less than 5 orders.

- (a)

```
select cid,count(*) from ordr
group by cid having count(*) < 5
```
- (b)

```
select a.cid,count(*)
from customer a inner join ordr b using(cid)
group by a.cid having count(*) < 5
```
- (c)

```
select a.cid,count(*)
from customer a left outer join ordr b on a.cid=b.cid
group by a.cid having count(*) < 5
```
- (d)

```
select cid,sum(case when b.cid is not null then 1 else 0 end)
from customer a left outer join ordr b using (cid)
group by cid having sum(case when b.cid is not null then 1 else 0 end) < 5
```
- (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 ordr
where state='NY' group by state
```
- (d)

```
with ordr_state as (
select cid, case when state='NY' then 1 else 0 end as in_state
from ordr natural inner join customer)
select b.in_state,count(*)
from customer a left outer join ordr_state b
on a.cid=b.cid
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) What product was advertised the most in 2020?

(a) select pid,count(*) cnt
 from advertisement
 where cast(tim as date) between '2020-01-01' and '2020-12-31'
 group by pid
 order by 2 desc
 limit 1

(b) select *
 from (
 select pid, dense_rank() over (order by count(*)) r
 from advertisement
 where cast(tim as date) between '2020-01-01' and '2020-12-31'
 group by pid
) a
 where r=1

- (c)

```
with stat as (
  select pid,count(*) cnt
  from advertisement
  where cast(tim as date) between '2020-01-01' and '2020-12-31'
  group by pid ),
  rnk as (
  select pid, dense_rank() over (order by cnt desc) r
  from stat )
select * from rnk where r=1
```
- (d)

```
select pid,count(*) cnt
from advertisement
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by pid
having dense_rank() over (order by cnt desc)=1
```
- (e) Other:

8. (5 points) Which customer got the most advertisements 2020?

- (a)

```
select cid,count(*) cnt
from advertisement
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by cid
order by 2 desc
limit 1
```
- (b)

```
select *
from (
  select cid, dense_rank() over (order by count(*)) r
  from advertisement
  where cast(tim as date) between '2020-01-01' and '2020-12-31'
  group by cid
) a
where r=1
```
- (c)

```
with stat as (
  select cid,count(*) cnt
  from advertisement
  where cast(tim as date) between '2020-01-01' and '2020-12-31'
  group by cid ),
  rnk as (
  select cid, dense_rank() over (order by cnt desc) r
  from stat )
select * from rnk where r=1
```
- (d)

```
select cid,count(*) cnt
from advertisement
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by cid
having dense_rank() over (order by cnt desc)=1
```
- (e) Other:

9. (5 points) Find all advertisements (the *aid*) that resulted in an order for advertised product within 10 minutes of advertisement.

- (a)

```
select distinct c.aid
from ordr a
inner join ordrdtls b using (oid)
inner join advertisement c using (pid,cid)
where a.tim between c.tim and c.tim+interval '10 minute'
```
- (b)

```
with allrecs as (
select aid, cid, pid, tim from advertisement
union all
select null, cid, pid, tim from ordr natural inner join ordrdtls
),
lastad as (
select a.*,
max( case when aid is not null then tim else null end)
over (partition by cid, pid order by tim) mxtim
from allrecs a
)
select *
from lastad
where aid is null and
mxtim between tim and tim+interval '10 minute'
```
- (c)

```
with wnd as (
select a.*,
lag(tim) over (partition by cid, pid order by tim) last_tim
from advertisement a
)
select *
from wnd a
inner join ordr using (cid)
natural inner join ordrdtls
where a.tim between a.tim and
a.last_tim and a.last_tim < a.tim+interval '10 minute'
```
- (d)

```
with o as (select tim as otim, pid, cid
from ordr natural inner join ordrdtls )
select *
from advertisement natural inner join o
where tim between otim and otim+interval '10 minute'
```
- (e) Other:

10. (5 points) What percentage of customers have more than 50 orders?

- (a)

```
with cnts as (
select cid,sum( case when b.cid is null then 0 else 1 end ) cnt
from customer a
inner join ordr b using (cid)
group by cid)
```

```
select 100*sum(case when cnt>50 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts
```

(b) with cnts as (
select cid,sum(case when b.cid is null then 0 else 1 end) cnt
from customer a
left outer join ordr b using (cid)
group by cid)
select 100*sum(case when cnt>50 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts

(c) with cnts as (
select cid,count(*) cnt
from ordr natural inner join ordrdtls
group by cid)
select 100*sum(case when cnt > 50 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts

(d) with cnts as (
select cid,count(*) cnt
from ordr
group by cid)
select 100*sum(case when cnt > 50 then 1.0 else 0.0 end)/sum(1.0) prcnt
from cnts

(e) Other:

11. (5 points) Count number of households with more than 2 customers (distinct addresses where more than 2 customers live).

(a) with hh as (
select 1 c
from customer
group by street, city, state, zip
having count(*) > 2
)
select sum(c) from hh

(b) select count(*)
from customer
group by street, city, state, zip
having count(distinct cid) > 2

(c) select count(*)
from customer a
inner join customer b using (street, city, state, zip)
inner join customer c using (street, city, state, zip)
where a.cid != b.cid and a.cid != c.cid and b.cid != c.cid

(d) select sum(1)
from customer
natural inner join ordr
group by street, city, state, zip
having count(distinct cid)>2

(e) Other:

12. (5 points) Identify situations when we advertise the same product to the same customer more than once within 60 minutes.

(a) `select pid,cid,min(tim),max(tim)
from advertisement
group by pid,cid
having max(tim) - min(tim) < interval '60 minute'`

(b) `select *
from advertisement a natural inner join advertisement b
where a.tim-b.tim < interval '60 minute'`

(c) `select *
from advertisement a
inner join advertisement b
using (pid,cid)
where b.tim between a.tim and interval '60 minute'`

(d) `with tm as (
select pid,cid,tim,lag(tim) over (partition by pid,cid order by tim) ltim
from advertisement)
select * from tm where tim-ltim < interval '60 minute'`

(e) Other:

13. (5 points) Identify customers who spent more than \$10,000 in 2020.

(a) `select cid,sum(price * qty)
from ordr a inner join ordrdttls b on a.oid=b.oid and a.cid=b.pid
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by cid
having sum(price * qty) > 10000`

(b) `select cid,sum(price * qty)
from ordr natural inner join ordrdttls
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by cid
having sum(price * qty) > 10000`

(c) `select cid,sum(listprice * qty)
from customer
natural inner join ordr
natural inner join ordrdttls
natural inner join product
where cast(tim as date) between '2020-01-01' and '2020-12-31'
group by cid
having sum(listprice * qty) > 10000`

(d) `select cid,max(price * qty)
from customer
natural inner join ordr
natural inner join ordrdttls
natural inner join product`

```
where cast(tim as date) between '2020-01-01' and '2020-12-31'  
group by cid  
having max(price * qty) > 10000
```

(e) Other:

14. (5 points) Identify customers who spent less than \$10,000 in 2020.

(a)

```
select cid  
from ordr a left outer ordrdtls b on a.oid=b.oid and a.cid=b.pid  
where cast(tim as date) between '2020-01-01' and '2020-12-31'  
group by cid  
having coalesce( sum(price * qty), 0) < 10000
```

(b)

```
select cid  
from ordr natural left outer join ordrdtls  
where cast(tim as date) between '2020-01-01' and '2020-12-31'  
group by cid  
having coalesce( sum(price * qty), 0) < 10000
```

(c)

```
select a.cid  
from customer a  
left outer join ordr b  
on a.cid=b.cid and  
cast(tim as date) between '2020-01-01' and '2020-12-31'  
natural left outer join ordrdtls  
group by a.cid  
having coalesce( sum(price * qty), 0) < 10000
```

(d)

```
select cid  
from customer  
natural left outer join ordr  
natural left outer ordrdtls  
natural left outer product  
where cast(tim as date) between '2020-01-01' and '2020-12-31'  
group by cid  
having max(price * qty) < 10000
```

(e) Other:

15. (5 points) Identify 5 most popular (maximum quantity purchased) products for each zip code.

(a)

```
with stats as (  
select zip, pid, sum(qty) tot  
from customer  
natural inner join ordr  
natural inner join ordrdtls  
group by zip,pid),  
rnk as (  
select a.*, dense_rank() over (partition by zip order by tot desc) r  
from stats a)  
select a.* from rnk a where r <= 5
```


- (b) with stats as (
 select zip, pid, count(distinct oid) cnt
 from customer
 natural inner join ordr
 natural inner join ordrdtls
 group by zip,pid),
 rnk as (
 select a.*,
 dense_rank() over (partition by zip order by cnt desc) r
 from stats a)
 select a.* from rnk a where r <= 5
- (c) select zip, pid, count(distinct oid) cnt
 from customer
 natural inner join ordr
 natural inner join ordrdtls
 group by zip,pid
 having dense_rank() over (partition by zip order by cnt desc) <= 5
- (d) select a.*
 from (
 select a.*, dense_rank() over (partition by zip order by tot) r
 from (
 select zip, pid, sum(qty) tot
 from customer
 natural inner join ordr
 natural inner join ordrdtls
 group by zip,pid) a) a where r <= 5
- (e) Other:

16. (5 points) Find customers who only buy stuff on sale (all purchases less than listed price).

- (a) select cid
 from ordr
 natural inner join ordrdtls
 natural inner join product
 group by cid
 having sum(qty*price) < sum(qty*listprice)
- (b) select cid,count(*)
 from ordr a
 left join ordrdtls b on a.oid=b.oid
 left join product c on b.pid=b.pid and b.price < c.listprice
 group by a.cid
 having count(*) > 1
- (c) select cid,count(*)
 from ordr
 natural inner join ordrdtls
 natural inner join product
 where price < listprice
 group by cid

```
(d) select cid
      from ordr
      natural inner join ordrdtls
      natural inner join product
      group by cid
      having max(case when price >= listprice then 1 else 0 end)=0
```

(e) Other:

17. (5 points) Find extreme customers, those who spend more money than 99% of other customers, and are at least 4-standard deviations away than the average dollars spent for all customers.

```
(a) select cid, sum(qty*price) as dlrs,
      avg(sum(qty*price)) over () avgdlrs,
      stddev(sum(qty*price)) over () sddlrs,
      row_number() over (order by sum(qty*price)) rn,
      count(*) cnt
      from ordr natural inner join ordrdtls
      group by cid
      having dlrs > avgdlrs+4*sddlrs and rn/cnt>0.99
```

```
(b) with stats as (
      select cid, sum(qty*price) as dlrs,
      avg(sum(qty*price)) over () avgdlrs,
      stddev(sum(qty*price)) over () sddlrs,
      row_number() over (order by sum(qty*price)) rn,
      count(*) cnt
      from ordr natural inner join ordrdtls
      group by cid)
      select *
      from stats
      where dlrs > avgdlrs+4*sddlrs and 1.0*rn/cnt>0.99
```

```
(c) with stats as (
      select cid, sum(qty*price) as dlrs
      from ordr natural inner join ordrdtls
      group by cid),
      prcnts as (
      select percentile_cont(0.99) within group (order by dlrs) as p99
      from stats),
      avg as (
      select avg(dlrs) a, stddev(dlrs) sd
      from stats)
      select *
      from stats cross join prcnts cross join avg
      where dlrs > 4*sd and dlrs>=p99
```

```
(d) select *
      from (
      select cid, sum(qty*price) as dlrs,
      avg(sum(qty*price)) over () avgdlrs,
      stddev(sum(qty*price)) over () sddlrs
```

```

    from ordr natural inner join ordrdtls
    group by cid) a
    where dlrs > avgdlrs+4*sddlrs

```

(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: