

CISC 7512X Midterm Exam

For the below questions, use the following schema definition.

```
customer(cid, fname, lname, street1, stree2, city, state, zip, dob, ssn)
policy(pid, cid, tim, typeid, premium, amnt, termid)
incident(iid, pid, tim, cost, covered)
```

It's a schema for an insurance company: Customers pay (premium) for policies that cover a particular type (typeid) of event during a particular term (termid) for upto a certain dollar value (amnt).

Incidents (during the policy) have a cost, and may be covered or not.

e.g. Bob pays \$1k for a \$1m policy on lightning strikes for 2021.

customer: has Bob's information policy: type is lightning, term is 2021, premium=\$1000, amnt is \$1000000.

Bob gets an electric shock from an outlet (insurance company does not pay; not lightning): incident: type = electric shock, cost \$300, covered=no (false).

Bob gets a bit hurt by lightning (insurance company covers costs): incident: type = lightning, cost \$170000, covered=yes.

Bob gets very badly hurt by lightning (insurance company pays out max amnt): incident: type = lightning, cost \$1000000, covered=yes.

1. (5 points) Find time when policy 235235 was created.

- (a)

```
select tim
from policy
where cid=235235
```
- (b)

```
select tim
from policy
where pid=235235
```
- (c)

```
select *
from customer
where cid=236235
```
- (d)

```
select *
from policy
left outer join customer
using (cid)
where pid=235235
```
- (e) Other:

2. (5 points) How many customers bought policies in January 2021?

- (a)

```
select count(*) as cnt
from customer
natural inner join policy
natural left outer join incident
where tim>=cast('2021-01-01' as date) and
tim<cast('2021-02-01' as date)
```
- (b)

```
select a.cid,count(*) as cnt
from customer a
inner join policy b
```

```
on a.cid=b.pid
where tim>=cast('2021-01-01' as date) and
tim<cast('2021-02-01' as date)
group by a.cid
```

(c)

```
select sum ( count(*) ) over ( ) cnt
from customer a
inner join policy b
on a.cid=b.cid
where tim>=cast('2021-01-01' as date) and
tim<cast('2021-02-01' as date)
group by a.cid
```

(d)

```
select count(distinct cid) cnt
from policy
where tim>=cast('2021-01-01' as date) and
tim<cast('2021-02-01' as date)
```

(e) Other:

3. (5 points) What percentage of customers had policies during termid=235235?

(a)

```
select 100.0*sum(case when termid=235235 then 1.0 else 0.0 end)/sum(1.0) prcnt
from policy a
```

(b)

```
with dcust as (
select distinct cid
from policy
where termid=235235
)
select 100.0*sum(cid)/sum(1.0) prcnt
from dcust a
```

(c)

```
with dcust as (
select a.cid, min(pid) m
from customer a
left outer join policy b
on a.cid=b.cid and b.termid=235235
group by a.cid
)
select 100.0*sum(case when m is not null then 1.0 else 0.0 end)/sum(1.0) prcnt
from dcust
```

(d)

```
select 100.0*sum(case when b.termid is not null then 1.0 else 0.0 end)/sum(1.0)
prcnt
from customer a
left outer join policy b
on a.cid=b.cid and b.termid=235235
```

(e) Other:

4. (5 points) Find policies that have not had any incidents.

(a)

```
select a.*
from policy a
inner join incident b
```

```
using (pid)
where b.iid is null
```

- (b)

```
select a.*
from policy a
natural left outer join incident b
where b.iid is null
```
- (c)

```
select a.*
from policy a
left outer join incident b using (pid)
where b.pid is null
```
- (d)

```
select a.*
from policy a
cross join incident b
where a.pid=b.pid and b.iid is null
```
- (e) Other:

5. (5 points) Find average time (in seconds) between policy creation and first incident.

- (a)

```
select avg( extract( epoch from b.tim - a.tim) )
from policy a
inner join incident b
on a.pid = b.pid
```
- (b)

```
with diff as (
select extract(epoch from min(b.tim) - a.tim) d
from policy a
natural inner join incident b
group by a.pid,a.tim
)
select avg(d)
from diff
```
- (c)

```
with diff as (
select extract(epoch from min(b.tim) - a.tim) d
from policy a
inner join incident b using (pid)
group by a.pid,a.tim
)
select avg(d)
from diff
```
- (d)

```
with evts as (
select pid, tim from policy
union all
select pid, tim from incident
),
diff as (
select extract(epoch from lag(tim) over (partition by pid order by tim) - tim)
d
from evts
)
```

```
select avg(d)
from diff
```

(e) Other:

6. (5 points) Identify the most popular policy type (most distinct customers).

(a) with stats as (
select b.pid, count(a.cid) cnt, max(count(a.cid)) over () maxcnt
from customer a
natural left outer join policy b
natural left outer join incident c
group by b.pid
)
select *
from stats
where cnt = maxcnt

(b) with stats as (
select typeid, count(distinct cid) cnt, max(count(distinct cid)) over () maxcnt
from policy
group by typeid
)
select *
from stats
where cnt = maxcnt

(c) with stats as (
select typeid, sum(1) cnt
from policy
group by typeid
),
stats2 as (
select typeid, rank() over (order by cnt) rnk
from stats
)
select *
from stats2
where rnk=1

(d) with stats as (
select typeid, sum(1) cnt
from policy
group by typeid
),
stats2 as (
select typeid, dense_rank() over (order by cnt) rnk
from stats
)
select *
from stats2
where rnk=1

(e) Other:

7. (5 points) Identify customers with most policies for term=124.

(a) with stats as (
select cid, rank() over (order by count(distinct pid) desc) r
from policy
where termid=124
group by cid
)
select *
from stats
where r=1

(b) with stats as (
select a.cid, count(distinct b.pid) cnt
from customer a
natural inner join policy b
where termid=124
group by a.cid
) ,
mx as (
select max(cnt) m
from stats
)
select *
from stats cross join mx
where m = cnt

(c) with stats as (
select cid, count(*) cnt, min(count(*)) over () mx
from policy
where termid=124
group by cid
)
select *
from stats
where cnt = mx

(d) select cid
from policy
where termid=124
group by cid
having count(cid) >= all(select count(cid) from customer)

(e) Other:

8. (5 points) Customers may have more than one incident. What's the average number of covered incidents do customers with policies have?

(a) with stats as (
select count(*) cnt
from policy a
natural inner join incident b
where b.covered = true

- ```

group by a.cid
)
select avg(cnt) from stats

```
- (b) with stats as (
- ```

select a.cid, sum(case when b.iid is not null then 1 else 0 end) cnt
from policy a
left outer join incident b
on a.pid=b.pid and b.covered = true
group by a.cid
)
select avg(cnt) from stats

```
- (c) with stats as (
- ```

select a.cid, sum(case when b.iid is not null then 1 else 0 end) cnt
from policy a
left outer join incident b
on a.pid=b.pid
where b.covered = true
group by a.cid
)
select avg(cnt) from stats

```
- (d) with stats as (
- ```

select distinct cid, iid
from policy
natural left outer join incident
where covered = true
),
cnts as (
select count(*) cnt
from stats
group by cid
having count(*) is not null
)
select avg(cnt) from cnts

```
- (e) Other:
9. (5 points) What percentage of incidents are covered?
- (a) select 100.0 * sum(case when covered is true then 1 else 0 end) /
sum(case when covered is false then 1 else 0 end) prcnt
from policy a
inner join incident b
on a.pid=b.pid
- (b) select 100.0 * (select count(*) from incident where covered=true)/
(select count(*) from incident where covered=false) prcnt
- (c) select 100.0*sum(case when a.covered=b.covered then 1.0 else 0.0 end)/sum(1.0)
prcnt
from incident a cross join incident b
where a.pid!=b.pid

(d) `select 100.0*sum(case when covered=true then 1.0 else 0.0 end)/sum(1.0) prcnt
from incident a`

(e) Other:

10. (5 points) Find unusually expensive terms (termid) (terms that are 2 standard deviations more expensive than the average cost).

- (a) `with stats as (
select termid, sum(b.cost) as cost
from policy a
natural inner join incident b
where b.covered = true
group by termid
)
,
stats2 as (
select a.*
avg(cost) over () avgcost,
stddev(cost) over () sdcost
from stats a
)
,
stats3 as (
select *
from stats2
where cost >= avgcost+2*sdcost
)
select *
from stats3`
- (b) `with stats as (
select termid, sum(b.cost) as cost,
avg(sum(b.cost)) over () avgcost,
stddev(sum(b.cost)) over () sdcost
from policy a
inner join incident b
on a.pid=b.pid and b.covered = true
group by termid
)
select *
from stats
where cost >= avgcost+2*sdcost`
- (c) `with tcost as (
select termid, sum(b.cost) as cost
from policy a
inner join incident b
on a.pid=b.pid and b.covered = true
group by termid
)
select a.*
from tcost a
group by termid, cost
having cost >= avg(cost) + 2*stddev(cost)`

```
(d) with tcost as (
  select termid, sum(case when b.covered = true then b.cost else 0 end) as cost
  from policy a
  inner join incident b
  on a.pid=b.pid
  group by termid
  having sum(case when b.covered = true then b.cost else 0 end) > 0
),
stats as (
  select avg(cost) avgcost, stddev(cost) sdcost
  from tcost
)
select termid
from tcost a natural inner join stats
group by termid
having max(cost) >= max(avgcost)+2*max(sdcost)
```

(e) Other:

11. (5 points) Find unusually profitable terms (terms that are 2 standard deviations more profitable than average: profit is policy premium - payed out costs).

```
(a) with profits as (
  select termid, sum(a.premium) - coalesce(sum(b.cost),0) profit,
  avg( sum(a.premium) - coalesce(sum(b.cost),0) ) over () avgprofit,
  stddev( sum(a.premium) - coalesce(sum(b.cost),0) ) over () sdprofit
  from policy a
  left outer join incident b
  on a.pid=b.pid and b.covered=true
  group by termid
)
select *
from profits
where profit >= avgprofit+2*sdprofit
```

```
(b) with stats as (
  select termid, sum(a.premium) - sum(b.cost) profit,
  from policy a
  natural inner join incident b
  where b.covered = true
  group by termid
),
stats2 as (
  select a.*,
  avg( profit ) over () avgprofit,
  stddev( profit ) over () sdprofit
  from stats a
),
stats3 as (
  select *
  from stats2
  where profit >= avgprofit+2*sdprofit
```



```

)
select *
from stats3
(c) with tstat as (
select termid, sum(a.premium) -
sum(case when b.covered = true then b.cost else 0 end) as profit
from policy a
inner join incident b
on a.pid=b.pid
group by termid
having sum(a.premium) - sum(case when b.covered = true then b.cost else 0 end)
> 0
),
stats as (
select avg(profit) avgprofit, stddev(profit) sdprofit
from tstat
)
select termid
from tstat a natural inner join stats
group by termid, profit, avgprofit, sdprofit
having profit >= avgprofit + 2* sdprofit
(d) with revenue as (
select termid, sum(premium) rev
from policy
group by termid
),
costs as (
select termid, sum(cost) as cost
from policy natural inner join incident
group by termid
),
profit as (
select a.termid, a.rev - coalesce(b.cost,0) profit
from revenue a
left outer join costs b
on a.termid=b.termid
),
stats as (
select avg(profit) avgprofit, stddev(profit) sdprofit
from profit
)
select *
from profit a cross join stats
where profit >= avgprofit + 2* sdprofit

```

(e) Other:

12. (5 points) What is the geometric mean cost of typeid=2 policy?
geometric mean of x_1, \dots, x_n is defined as $(\prod_{i=1}^n x_i)^{\frac{1}{n}}$

(a) `select avg(cost)`

```

from policy a
inner join incident b
on a.pid=b.pid and b.covered=true
where a.typeid=2

```

- (b)

```

select sum(cost)/sum(1.0)
from policy a
inner join incident b
on a.pid=b.pid and b.covered=true
where a.typeid=2

```
- (c)

```

select exp ( avg ( ln( cost ) ) )
from policy a
inner join incident b
on a.pid=b.pid and b.covered=true
where a.typeid=2

```
- (d)

```

select exp ( sum ( ln( cost ) ) ) / sum(1.0)
from policy a
inner join incident b
on a.pid=b.pid and b.covered=true
where a.typeid=2

```
- (e) Other:

13. (5 points) Which single-event would be worst for the insurance company for termid=123 (which typeid would cost the most).

- (a)

```

with stats as (
select typeid, sum(amnt) amnt, max( sum(amnt) ) over () mx
from policy
where termid=123
group by typeid
)
select *
from stats
where amnt = mx

```
- (b)

```

with stats as (
select typeid, rank() over (order by sum(amnt) desc) r
from policy
where termid=123
group by typeid
)
select *
from stats
where r=1

```
- (c)

```

select typeid
from policy
where termid=123
group by typeid
having sum(amnt) >= all( select sum(amnt) from policy group by typeid )

```
- (d)

```

select typeid, sum(b.cost) tot
from policy a

```

```

inner join incident b
on a.pid = b.pid and b.covered = true
group by typeid
order by 2 desc
limit 1

```

(e) Other:

14. (5 points) Identify customers ages 30-45 who don't have typeid=234 insurance (term-life-insurance).

(a)

```

select a.*
from customer a
natural left outer join policy b
where b.typeid=234 and
extract(year from age(a.dob)) between 30 and 45 and
b.pid is null

```

(b)

```

select a.*
from customer a
left outer join policy b
on a.cid=b.cid and b.typeid=234
where extract(year from age(a.dob)) between 30 and 45 and
b.pid is null

```

(c)

```

select a.*
from customer a
inner join policy b
on a.cid=b.cid and b.typeid=234
where extract(year from age(a.dob)) between 30 and 45 and
b.pid is null

```

(d)

```

select a.*
from customer a
full outer join policy b
on a.cid=b.cid and b.typeid=234
where extract(year from age(a.dob)) between 30 and 45 and
b.pid is null

```

(e) Other:

15. (5 points) Identify most expensive age group for the insurance company (across all policies and terms). Age groups: 10-25, 26-45, 46-65, >65.

(a)

```

with stats as (
select
case when age between 10 and 25 then '10-25'
when age between 26 and 45 then '26-45'
when age between 46 and 65 then '46-65'
when age > 65 then '>65'
end as agegrp, sum(c.cost) costs, max( sum(c.cost) ) over () mx
from (select a.*, extract(year from age(dob)) as age from customer a) a
inner join policy b on a.cid=b.cid
inner join incident c on b.pid=c.pid and c.covered=true
group by agegrp

```

```

)
select *
from stats
where costs = mx
(b) with stats as (
select
case when age between 10 and 25 then '10-25'
when age between 26 and 45 then '26-45'
when age between 46 and 65 then '46-65'
when age > 65 then '>65'
end as agegrp, sum(c.cost) costs, max( sum(c.cost) ) over () mx
from (select a.*, extract(year from age(dob)) as age from customer a) a
natural inner join policy b
natural inner join incident c
where c.covered=true
group by case when age between 10 and 25 then '10-25'
when age between 26 and 45 then '26-45'
when age between 46 and 65 then '46-65'
when age > 65 then '>65'
end
)
select *
from stats
where costs = mxn
(c) with custage as (
select a.*, extract(year from age(dob)) as age
from customer a
),
custgrp as (
select a.*,
case when age between 10 and 25 then '10-25'
when age between 26 and 45 then '26-45'
when age between 46 and 65 then '46-65'
when age > 65 then '>65'
end as agegrp
from custage a
)
select agegrp, sum(c.cost) costs
from custgrp a
inner join policy b on a.cid=b.cid
inner join incident c on b.pid=c.pid and c.covered=true
group by agegrp
having sum(c.cost) >= all( select sum(c.cost) from incident)
(d) with custage as (
select a.*, extract(year from age(dob)) as age
from customer a
),
custgrp as (
select a.*,

```

```

case when age between 10 and 25 then '10-25'
when age between 26 and 45 then '26-45'
when age between 46 and 65 then '46-65'
when age > 65 then '>65'
end as agegrp
from custage a
),
stats as (
select agegrp,sum(c.cost) costs, rank() over (order by sum(cost) desc) rnk
from custgrp a
inner join policy b on a.cid=b.cid
inner join incident c on b.pid=c.pid and c.covered=true
group by agegrp
)
select *
from stats
where rnk=1

```

(e) Other:

16. (5 points) Identify suspicious and potentially fraudulent customers; customers who have unusually high chance of incidents. For each customer, calculate number (not dollar amount, but count) of covered incidents, and pick out customers that are 4 standard deviations higher than the mean.

(a) with stats as (

```

select a.cid, count(b.*) cnt,
avg(count(b.*)) over () avgcnt,
stddev(count(b.*)) over () sdcnt
from policy a
natural left outer join incident b
where b.covered = true
group by a.cid
)
select *
from stats
where cnt >= avgcnt+4*sdcnt

```

(b) with stats as (

```

select a.cid, count(b.*) cnt,
avg(count(b.*)) over () avgcnt,
stddev(count(b.*)) over () sdcnt
from policy a
left outer join incident b
on a.pid=b.pid and b.covered = true
group by a.cid
)
select *
from stats
where cnt >= avgcnt+4*sdcnt

```

(c) with stats as (

```

select a.cid, sum(case when b.pid is not null then 1 else 0 end) cnt

```

```

from policy a
left outer join incident b
on a.pid=b.pid and b.covered = true
group by a.cid
),
stats2 as (
select a.*,
avg(cnt) over () avgcnt,
stddev(cnt) over () sdcnt
from stats a
)
select *
from stats2
where cnt >= avgcnt+2*sdcnt

```

(d) with stats as (
select a.cid, sum(case when b.pid is not null then 1 else 0 end) cnt
from policy a
left outer join incident b
on a.pid=b.pid and b.covered = true
group by a.cid
),
stats2 as (
select
avg(cnt) avgcnt,
stddev(cnt) sdcnt
from stats a
)
select *
from stats cross join stats2
where cnt >= avgcnt+2*sdcnt

(e) Other:

17. (5 points) Identify suspicious and potentially fraudulent customers; more than 20 different customers (potentially with variation in names and dob), that are registered under the same address and who have a policy of the same type and term. (e.g. "Bob Johnson", "Bob Johns", "Bob Jonson", etc., taking out fire insurance on their house with same address).

(a) select street1, stree2, city, state, zip, typeid, termid
from customer
natural inner join policy
group by street1, stree2, city, state, zip, typeid, termid
having count(*) > 20

(b) with stats as (
select a.*,
count(*) over (partition by street1, stree2, city, state, zip, typeid, termid
order by fname,lname) cnt
from customer a
inner join policy b
on a.cid=b.cid

```
)  
select *  
from stats  
where cnt > 20
```

- (c)

```
select *  
from customer  
natural inner join (  
select street1, stree2, city, state, zip, typeid, termid  
from customer  
natural inner join policy  
group by street1, stree2, city, state, zip, typeid, termid  
having count(distinct concat(fname,lname,dob)) > 20 ) a
```
- (d)

```
with addr as (  
select street1, stree2, city, state, zip, typeid, termid, count(*) cnt  
from customer a  
inner join policy b using (cid)  
group by street1, stree2, city, state, zip, typeid, termid  
having count(*) > 20  
)  
select *  
from customer  
natural inner join addr
```
- (e) Other:

18. (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.

19. (5 points) Find customers who have more than 10 policies?

- (a)

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

```
select a.cid,count(*)  
from customer a  
inner join policy b using(cid)
```

```
group by a.cid
where count(*) > 10
```

- (c)

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

20. (5 points) Find customers who have less than 5 policies.

- (a)

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

```
with outerunion as (
select cid,count(*) cnt
from policy
group by cid
)
select *
from outerunion
where cnt < 5
```
- (c)

```
select a.cid,count(*)
from customer a
inner join policy b
using(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 policy b
using (cid)
group by cid
having sum(case when b.cid is not null then 1 else 0 end) < 5
```
- (e) Other: