

CISC 7512X Final Exam

For the below questions, use the following schema definition.

```
customer(custid,username,fname,lname)
product(prodid,description,listedprice)
purchase(purchid,custid,timestamp)
purchaseitem(purchid,prodid,qty,price)
```

It is a schema for a store, with customers, products, and purchases that link customers to products. Each purchase can have multiple items, which are in purchaseitem table. 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 customer id of John Doe.
 - (a) select lname,fname from customer where fname='John' and lname='Doe'
 - (b) select custid from purchase where fname='John' and lname='Doe'
 - (c) select custid from customer where fname='John' and lname='Doe'
 - (d) select custid from customer inner join purchaseitem using(custid) where fname='John' and lname='Doe'
 - (e) Other:
2. (5 points) Find the average price of a product.
 - (a) select avg(price) from purchaseitem
 - (b) select avg(listedprice) from product
 - (c) select avg(qty*listedprice) from product
 - (d) select avg(price) from product
 - (e) Other:
3. (5 points) Find number of purchases by customer.
 - (a) select custid,count(*) from purchase natural inner join purchaseitem group by custid
 - (b) select prodid,count(*) from purchaseitem group by prodid
 - (c) select purchid,count(*) from purchaseitem group by purchid
 - (d) select custid,count(*) from purchase group by custid
 - (e) Other:
4. (5 points) Find names of all items ever purchased by 'John Doe'.
 - (a) select count(*) from customer a natural inner join purchase b natural inner join purchaseitem c where a.lname='Doe' and a.fname='John'
 - (b) select description from customer a natural inner join purchase b natural inner join purchaseitem c natural inner join product d where a.lname='Doe' and a.fname='John' group by description
 - (c) select description from customer a natural inner join product d where a.lname='Doe' and a.fname='John' group by description

- (d) select distinct description from customer a natural inner join product d where a.lname='Doe' and a.fname='John'
- (e) Other:
5. (5 points) Find all purchases that total more than \$1000.
- (a) select purchid from purchase a natural inner join purchaseitem b group by purchid having sum(qty*price) > 1000
- (b) select purchid from purchase a natural inner join purchaseitem b where qty*price > 1000 group by purchid
- (c) select purchid from customer a inner join purchase a natural inner join purchaseitem b where qty*price > 1000 group by purchid
- (d) select purchid from purchaseitem b where qty*price > 1000
- (e) Other:
6. (5 points) Find customers who have never purchased anything.
- (a) select a.* from customer a natural inner join purchase b where b.purchid is null
- (b) select a.* from customer a left join purchaseitem b on a.custid=b.custid where b.purchid=0
- (c) select a.* from customer a inner join purchase b on a.custid=b.custid where b.purchid > 0
- (d) select a.* from customer a natural left outer join purchase b where b.purchid is null
- (e) Other:
7. (5 points) Find top 10 customers who spent the most in 2016.
- (a) select top 10 custid from purchase a natural inner join purchaseitem b where timestamp >='20160101' and timestamp < '20170101'
- (b) select custid from purchase a natural inner join purchaseitem b where timestamp >='20160101' and timestamp < '20170101' order by sum(qty*price) desc
- (c) select custid,row_number() over (order by sum(qty*price) desc) rn from purchase a natural inner join purchaseitem b where timestamp >='20160101' and timestamp < '20170101' and rn <= 10
- (d) select custid,sum(qty*price) v from purchase a natural inner join purchaseitem b where timestamp >='20160101' and timestamp < '20170101' group by custid order by 2 desc limit 10
- (e) Other:
8. (5 points) What is the most appropriate index for customer.username field?
- (a) Btree Index
- (b) Bitmap Index
- (c) Clustered Index
- (d) Bitmap Clustered Index
- (e) Other:

9. (5 points) What is the most appropriate index for product.description field?
- (a) Btree Index
 - (b) Bitmap Index
 - (c) Clustered Index
 - (d) Bitmap Clustered Index
 - (e) Other:

10. (5 points) What is the most appropriate index for product.prodid field?
- (a) Btree Index
 - (b) Bitmap Index
 - (c) Clustered Index
 - (d) Bitmap Clustered Index
 - (e) Other:

11. (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 left join n b on b.n < sqrt(a.n)
group by a.n
having a.n=2 or min(a.n % b.n) > 0
```

- (a) Is invalid
 - (b) Will generate a list of numbers 1 to 1000
 - (c) Will create a table with all dates between 19000101 and 21000101
 - (d) Will output list of all prime numbers between 1 and 1000
 - (e) Other:
12. (5 points) Find average number of items per purchase.
- (a) select avg(purchase) from customer a natural inner join purchase b
 - (b) select avg(*) from customer a natural inner join purchase b where custid > 0
 - (c) select avg(cnt) from (select purchid,count(*) cnt from purchase a natural inner join purchaseitem b group by purchid) a
 - (d) select avg(sum(1.0)) over () from customer a
 - (e) Other:
13. (5 points) Find items that were bought on sale (listed price is higher than purchase price).
- (a) select * from product a natural inner join purchaseitem b where listedprice > price

- (b) `select * from product a natural inner join purchaseitem b group by purchid having listed-price > price`
 - (c) `select count(*) from product a natural inner join purchaseitem b group by purchid having listedprice > price`
 - (d) `select * from purchaseitem b where listedprice > price`
 - (e) Other:
14. (5 points) Find the last sale price for each item.
- (a) `select prodid,max(price) ls from purchaseitem order by timestamp`
 - (b) `select prodid,max(timestamp) over (partition by prodid order by price) ls from purchaseitem`
 - (c) `select prodid,last_value(price) over (partition by prodid order by timestamp) ls from purchaseitem`
 - (d) `select prodid,last_value(price) over (partition by prodid order by timestamp) ls from purchase p natural inner join purchaseitem pi`
 - (e) Other:
15. (5 points) Find percentage of purchases with above average amount.
- (a) `select row_number() over () / count(*) from purchase a inner join purchaseitem b where qty*price > avg(qty*price)`
 - (b) `select purchid,sum(qty*price) px, avg(sum(qty*price)) over () avgpx purchase a inner join purchaseitem b where px > avgpx`
 - (c) `select percentage(qty*price) from purchaseitem where qty*price > avg(qty*price)`
 - (d) `select sum(case when qty*price>avg() then 1.0 else NULL end) / sum(1.0) from purchase inner join purchaseitem`
 - (e) Other:
16. (5 points) Find all customers who purchased 'Gruntmaster 6000' during the first month of 2017.
- (a) `select * from customer where purchase = 'Gruntmaster 6000'`
 - (b) `select * from customer inner join purchase inner join purchaseitem where item='Gruntmaster 6000'`
 - (c) `select * from purchase inner join purchaseitem where description='Gruntmaster 6000'`
 - (d) `select distinct from purchaseitem inner join customer using(custid) having description='Gruntmaster 6000'`
 - (e) Other:
17. (5 points) Find customers who purchased 'Gruntmaster 6000' and also 'Gruntmaster 9000'.
- (a) `select * from customer where purchased in ('Gruntmaster 6000', 'Gruntmaster 9000')`
 - (b) `select * from customer inner join purchaseitem on custid and description in ('Gruntmaster 6000', 'Gruntmaster 9000')`
 - (c) `select custid from purchase where description in ('Gruntmaster 6000', 'Gruntmaster 9000')`

- (d) select custid from purchase a inner join purchaseitem inner join purchase b inner join purchaseitem where a.description='Gruntmaster 6000' and b.description='Gruntmaster 9000'
 - (e) Other:
18. (5 points) In general, on limited memory system, no indexes, and huge tables, what join type would perform best?
- (a) merge join.
 - (b) hash join.
 - (c) indexed lookup join.
 - (d) inner loop join.
 - (e) Other:
19. (5 points) For “customer inner join purchase”, and no indexes, most modern databases will perform:
- (a) merge join.
 - (b) hash join.
 - (c) indexed lookup join.
 - (d) inner loop join.
 - (e) Other:
20. (5 points) Partitions:
- (a) Are similar to views.
 - (b) Are similar to temporary tables.
 - (c) Allow for physical clustering of logically similar data.
 - (d) All of the above.
 - (e) Other: