CISC 7510X Final Exam

Multiple-choice exam: select the answer that best answers the question. Each question is worth 5-points. You get 1-point for leaving the answer blank. If no answer is appropriate, write your own (the `e` answer).

For the below questions, use the following schema definition.

```
customer(cid, name, email, dob)
ordr(oid, cid, tim)
ordritem(oid, pid, qty)
product(pid, name, price)
```

It’s a schema for a store: `customer` table has customer informatioin, `ordr` has order information (links to customer via `cid`), and `ordritem` (links to `ordr` via `oid`) has order line-items and quantity and link to `products` via `pid`.

1. Find customer id of John Doe.

   (a) `select cid from customer where name=’John Doe’`

   (b) `select name from customer where fname=’John’ and lname=’Doe’`

   (c) `select * from customer where name=’John Doe’`

   (d) `select cid from customer where fname=’John’ and lname=’Doe’`

   (e) Other:

2. Find name and price of product 42.

   (a) `select pid, price from product where pid=42`

   (b) `select name, price from product having pid=42`

   (c) `select * from product with pid=42`

   (d) `select name, price from product where pid=42`

   (e) Other:

3. Find customer id for order 42.

   (a)
   ```
   select cid from customer a
   inner join ordr b on (cid) where oid=42
   ```

   (b)
   ```
   select cid from customer a
   left outer join ordr b using(cid) where oid=42
   ```

   (c) `select cid from ordr where oid=42`

   (d) `select name, cid, dob from ordr where oid=42`

   (e) Other:

4. Find customer name for order 42.

   (a)
   ```
   select * from ordr a
   natural inner join customer b using a.cid=b.cid where oid=42
   ```

   (b)
   ```
   select name from ordr a
   inner join customer b on a.cid=b.cid where oid=42
   ```

(c) `select name from ordr where oid=42`

(d)
```
select name from customer a
inner join ordr b on a.cid=b.cid and cid=42
```

(e) Other:

5. Find customers who are older than 42-years-old.

(a)
```
select * from customer
where extract(year from age(dob)) > 42
```

(b) `select * from customer where dob > 42`

(c) `select * from customer where dob > ’1978-01-01’`

(d) `select * from customer where age > 42`

(e) Other:

6. What fraction of customers are older than 42-years-old.

(a) `select fraction(customer age > 42) from customer`

(b)
```
select sum(case when dob > 42 then 1.0 else 0.0 end)/sum(1.0)
from customer
```

(c)
```
select sum(case when dob < ’1978-01-01’ then 1.0 else 0.0 end)/sum(1.0)
from customer
```

(d)
```
select sum(case when extract(year from age(dob)) > 42 then 1.0
else 0.0 end)/sum(1.0) from customer
```

(e) Other:

7. Find all customers who have a `@hotmail.com` email address.

(a) `select * from customer where email = ’@hotmail.com’`

(b)
```
select * from customer
where substr(email, position(’@’ in email), 100) = ’hotmail.com’
```

(c) `select * from customer where email like ’%@hotmail.com’`

(d) `select * from customer with email matches regex’*hotmail.com’`

(e) Other:

8. Count number of customers for each email domain name.

(a) `select email,count(*) from customer group by email`

(b)
```
with domain as (
select substr(email, position(’@’ in email), 100) domain from customer )
select domain,count(*) from domain group by domain
```

(c) `select substring(email from ’@(.*)$’) as domain,count(*) from customer`

(d) `select substr(email,5,100) domain,count(*) from customer group by 1`

(e) Other:

9. Find customers (cid) with *more* than 10 orders.

(a)
```
select cid from ordr group by cid having count(*) > 10
```

(b)
```
select cid from ordr having count(*) > 10
```

(c)
```
select cid from ordr where count(*) > 10
```

(d)
```
select * from ordr group by cid having count(*) > 10
```

(e) Other:

10. Find customers (cid) with *less* than 10 orders.

(a)
```
select cid from ordr group by cid having count(*) < 10
```

(b)
```
select a.cid
from customer a
inner outer join ordr b on a.cid=b.cid
group by a.cid having count(b.cid) < 10
```

(c)
```
select a.cid
from customer a
natural outer join ordr b
where count(b.cid) < 10
```

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

(e) Other:

11. Find names of all products that John Doe has ever purchased.

(a)
```
select distinct product.name
from customer
natural inner join ordr
natural inner join product
where customer.name='John Doe'
```

(b)
```
select distinct d.name
from customer a, ordr b, ordritem c, product d
where a.cid=b.cid and c.pid=d.pid and a.name='John Doe'
```

(c)
```
select distinct d.name
from customer a
inner join ordr b on a.cid=b.cid
inner join ordritem c on b.oid=c.oid
inner join product d on c.pid=d.pid
where a.name='John Doe'
```

(d)
```
select distinct name from product a where name='John Doe'
```

(e) Other:

12. Which customers (cid) bought more than 10 coffees.

(a)
```
select a.cid
from customer a
inner join ordr b on a.cid=b.cid
inner join ordritem c on b.oid=c.oid
inner join product d on c.pid=d.pid and c.name=’coffee’
where sum(b.qty) > 10
```

(b)

```
select a.cid
from ordr a
inner join ordritem b on a.oid=b.oid
inner join product c on b.pid=c.pid and c.name=’coffee’
group by a.cid
having sum(b.qty) > 10
```

(c)

```
select a.cid
from ordr a
natural inner join ordritem b
natural inner join product c
where c.name=’coffee’
group by a.cid
having count(*) > 10
```

(d)

```
select a.cid
from ordr a
inner join ordritem b on a.oid=b.oid
inner join product c on b.pid=c.pid and c.name=’coffee’
group by a.cid
having sum(b.qty) > 10
```

(e) Other:

13. A query:

```
select distinct a.*
from customer a
left outer join ordr b
on a.cid=b.cid
where b.tim >= ’2022-12-20’
```

will return:

(a) customers who have one or more orders on or after 2022-12-20.

(b) all customers, with matching orders on or after 2022-12-20.

(c) customers who have zero or more orders on or after 2022-12-20.

(d) customers who have zero orders on or after 2022-12-20.

(e) Other:

14. 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)
```
select a.*,b.val from T1 a
left outer join T2 b
on a.key=b.key
where 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)
```
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
```

15. (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<=3 or min(a.n % b.n) > 0
```

(a) Will generate a list of numbers 1 to 1000

(b) Will generate a list of odd numbers less than 3.

(c) Will output list of all prime numbers between 1 and 1000

(d) Is invalid

(e) Other:

16. If our system has plenty of memory, what join performs best?

(a) merge join

(b) hash join

(c) outer join

(d) inner loop join

(e) Other:

17. If our system has very little memory (compared to the data), what join performs best?

(a) merge join

(b) outer join

(c) hash join

(d) inner loop join

(e) Other:

18. What's usally the worst performing join type?

(a) hash join

(b) merge join

(c) outer join

(d) inner loop join

(e) Other:

19. For the query:

```
select * from T1 a
inner join T2 b
on a.value >= b.start and a.value < b.end
```

most databases will perform:

(a) hash join

(b) merge join

(c) inner loop join

(d) outer join

(e) Other:

20. Bitmap index is appropriate when:

(a) Number of distinct values is very large.

(b) Number of distinct values is very small.

(c) Number of database records is very large.

(d) Number of database records is very small.

(e) Other: