CISC 7510X Midterm Exam

For the below questions, use the following schema definition.

```
traveler(tid,fname,lname,ktn)
flight(fid,airline,flightno,srcairport,destairport,departtim,landtim,coachprice)
itinerary(iid,tid,timstamp)
itineraryitem(iid,fid,price,seat)
```

It is a schema for an airline, with travelers, flights, and itineraries that link travelers to flights. Each itinerary can have multiple flight legs, which are in itineraryitem 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. There are *more* than 2 questions where you have to write your own answer.

1. (5 points) Find traveler id (tid) of traveler 'John Doe'.

    (a) select lname,fname from traveler where fname='John' and lname='Doe'

    (b) select tid from itinerary where fname='John' and lname='Doe'

    (c) select tid from traveler where fname='John' and lname='Doe'

    (d) select tid from traveler inner join itineraryitem using(tid) where fname='John' and lname='Doe'

    (e) Other:

2. (5 points) Find the average price of a coach ticket.

    (a) select avg(price) from purchaseitem

    (b) select avg(coachprice) from flight

    (c) select avg(coachprice) from itinerary

    (d) select avg(price) from flight

    (e) Other:

3. (5 points) Find number of itineraries by traveler.

    (a) select tid,count(*) from itinerary natural inner join itineraryitem group by tid

    (b) select fid,count(*) from itineraryitem group by fid

    (c) select iid,count(*) from itineraryitem group by iid

    (d) select tid,count(*) from itinerary group by tid

    (e) Other:

4. (5 points) Find all flights for traveler 'John Doe'.

    (a) select count(*) from traveler a natural inner join itinerary b natural inner join itineraryitem c where a.lname='Doe' and a.fname='John'

    (b) select flightno,srcairport,departtim from traveler a natural inner join itinerary b natural inner join itineraryitem c natural inner join flight d where a.lname='Doe' and a.fname='John' group by flightno,srcairport,departtim

    (c) select flightno,srcairport,departtim from traveler a natural inner join itinerary d where a.lname='Doe' and a.fname='John' group by flightno,srcairport,departtim

1

(d) select distinct flightno,srcairport,departtim from traveler a natural inner join itinerary d where a.lname='Doe' and a.fname='John'

(e) Other:

5. (5 points) Find all itineraries that total more than $5000.

(a) select iid from itinerary a natural inner join itineraryitem b group by iid having sum(price) > 5000

(b) select iid from itinerary a natural inner join itineraryitem b where price > 5000 group by iid

(c) select iid from traveler a inner join itinerary a natural inner join itineraryitem b where price > 5000 group by iid

(d) select iid from itineraryitem b where price > 5000

(e) Other:

6. (5 points) Find travelers who have never purchased any flights.

(a) select a.* from traveler a natural inner join itinerary b where b.iid is null

(b) select a.* from traveler a left join itineraryitem b on a.tid=b.tid where b.iid=0

(c) select a.* from traveler a inner join itinerary b on a.tid=b.tid where b.iid > 0

(d) select a.* from traveler a natural left outer join itinerary b where b.iid is null

(e) Other:

7. (5 points) Find top 10 travelers who spent the most in 2014.

(a) select top 10 tid from itinerary a natural inner join itineraryitem b where timstamp >='20140101' and timstamp < '20150101'

(b) select tid from itinerary a natural inner join itineraryitem b where timstamp >='20140101' and timstamp < '20150101' order by sum(price) desc

(c) select tid,row_number() over (order by sum(price) desc) rn from itinerary a natural inner join itineraryitem b where timstamp >='20140101' and timstamp < '20150101' and rn <= 10

(d) select tid,sum(price) v from itinerary a natural inner join itineraryitem b where timstamp >='20140101' and timstamp < '20150101' group by tid order by 2 desc limit 10

(e) Other:

8. (5 points) What is the most appropriate index for traveler.lname 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 traveler.tid 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 itinerary.iid 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 odd numbers from 1 to 1000

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

    (e) Other:

12. (5 points) Find average number of flights per itinerary.

    (a) select avg(itinerary) from traveler a natural inner join itinerary b

    (b) select avg(*) from traveler a natural inner join itinerary b where tid > 0

    (c) select avg(cnt) from (select iid,count(*) cnt from itinerary a natural inner join itineraryitem b group by iid) a

    (d) select avg(sum(1.0)) over () from traveler a

    (e) Other:

13. (5 points) Find tickets that were bought on sale (listed price is higher than purchase price).

    (a) select * from flight a natural inner join itineraryitem b where coachprice > price

    (b) select * from flight a natural inner join itineraryitem b group by iid having coachprice > price

    (c) select count(*) from flight a natural inner join itineraryitem b group by iid having coachprice > price

   (d) select * from itineraryitem b where coachprice > price

   (e) Other:

14. (5 points) Find the latest sale price for each flight.

   (a) select distinct fid,max(price) ls from itineraryitem order by timstamp

   (b) select distinct fid,max(timstamp) over (partition by fid order by price) ls from itineraryitem

   (c) select distinct fid,last_value(price) over (partition by fid order by timstamp) ls from itineraryitem

   (d) select distinct fid,last_value(price) over (partition by fid order by timstamp) ls from itinerary i natural inner join itineraryitem ii

   (e) Other:

15. (5 points) Find percentage of itineraries with above average costs.

   (a) select row_number() over () / count(*) from itinerary a inner join itineraryitem b where price > avg(price)

   (b) select iid,sum(price) px, avg(sum(price)) over () avgpx itinerary a inner join itineraryitem b where px > avgpx

   (c) select percentage(price) from itineraryitem where price > avg(price)

   (d) select sum(case when price>avg() then 1.0 else NULL end) / sum(1.0) from itinerary inner join itineraryitem

   (e) Other:

16. (5 points) Find all travelers who were booked for flight 'ABC123' during the first month of 2015.

   (a) select * from traveler where flightno = 'ABC123'

   (b) select * from traveler inner join itinerary inner join itineraryitem where flightno='ABC123'

   (c) select * from itinerary inner join itineraryitem where flightno='ABC123'

   (d) select distinct * from itineraryitem inner join traveler using(tid) having flightno='ABC123'

   (e) Other:

17. (5 points) Find travelers who booked flight 'ABC123' and also 'XYZ789'.

   (a) select * from traveler where flightno in ('ABC123', 'XYZ789')

   (b) select * from traveler inner join itineraryitem on tid and flightno in ('ABC123', 'XYZ789')

   (c) select tid from itinerary where flightno in ('ABC123', 'XYZ789')

   (d) select tid from itinerary a inner join itineraryitem inner join itinerary b inner join itineraryitem where a.flightno='ABC123' and b.flightno='XYZ789'

   (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 "traveler inner join itinerary", 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) When should bitmap indexes be used?

    (a) When clustered indexes are not appropriate.

    (b) If a column has many distinct values.

    (c) Only on columns with few distinct values.

    (d) All of the above.

    (e) Other: