Midterm Exam (CS 561-A) - Spring 2021 (Mon., 4/12/2021) 6:30-9 PM ET.

Due Apr 12 at 9:05pm **Points** 100 **Questions** 9
Available Apr 12 at 6:35pm - Apr 12 at 9:05pm about 3 hours
Time Limit 150 Minutes

Instructions

IMPORTANT: Please read the following instructions carefully before starting the exam.

This exam is worth 100 points (30% of your grade), and you have up to 150 minutes (2.5 hours) to finish the exam. This exam is closed book, closed notes and without looking up on-line. Write your answers "clearly."

Please note/remember the following points while taking the exam:

- Although you're taking the exam in your own space, please keep all of the reference material closed, including the textbooks, notes, SQL HW answers, on-line access, etc. The exam is to test your knowledge and understanding of the topics, NOT your ability to look up and search!
- Please make sure you are alone in your space -- this is NOT a group exam.
- Questions (especially those involving key concepts) are not necessarily complex, however, I'm looking for answers "in your own words" (vs., taking answers from exercise questions, slides, etc.). For that reason, I expect everyone's answers to be "unique" (i.e., not resembling someone else's answers).
- When writing SQL queries:
  - Use appropriate indentations, etc. to make sure your queries are easy to read and understand.
  - Please use only the standard SQL features we covered in class -- do **NOT** use any other functions, etc. you might have used in your HW assignment. 50% of the total points for the question will be deducted if you use those non-standard syntactic features.
  - You are only allowed to use the simple syntax of func(x) for ALL aggregate functions (e.g., sum(quant)) -- i.e., do **NOT** use any other variations of syntax such as CASE . . . WHEN . . . THEN inside aggregate functions. These syntactic features are hiding implicit joins. 50% of the total points for the question will be deducted if you use those other variations of syntax.
  - Use appropriate JOINS as needed.

This quiz was locked Apr 12 at 9:05pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	147 minutes	66 out of 100

ⓘ Correct answers are hidden.

Score for this quiz: **66** out of 100
Submitted Apr 12 at 9:04pm
This attempt took 147 minutes.

Question 1

10 / 10 pts

Write an SQL query for the following:

For the (year, month) combinations, find the products of most (MAX) and least (MIN) purchase quantities, and the corresponding customer who made the purchases.

sales (prod, cust, yr, mo, day, state, quant)

year	month	max_prod	max_cust	min_prod	min_cust
2011	12	Apple	Boo	Banana	Dan
2017	4	Cherry	Claire	Grapes	Chae
. . .	. . .	. . .	. . .	. . .	. . .
2014	7	Ice	Helen	Ham	Emily

Your Answer:

```
with q1 as(
select year,month, max(quant) MAX_Q, min(quant) MIN_Q
from sales
group by year,month
),
q2 as(
select q1.year, q1.month, s.prod MAX_PROD,
s.cust MAX_CUST, q1.MIN_Q
from q1, sales s
where q1.year = s.year and q1.month = s.month
and MAX_Q = s.quant
),
q3 as(
select q2.year, q2.month, q2.MAX_PROD, q2.MAX_CUST, s.prod
MIN_PROD,
s.cust MIN_CUST
from q2, sales s
where q2.year = s.year and q2.month = s.month
and MIN_Q = s.quant
)
select year,month, MAX_PROD, MAX_CUST, MIN_PROD, MIN_CUST
from q3
```

Question 2

10 / 10 pts

Write an SQL query for the following:

For each (year, month) combination, find the “busiest” and the “slowest” days (those days with the most and the least total sales quantities of products sold). For each of the “busiest” and “slowest” days, find the corresponding “states” where those most and least sales quantities were sold.

sales (prod, cust, yr, mo, day, state, quant)

year	month	busiest_day	max_state	slowest_day	min_state
2011	7	4	NY	31	NY
2019	12	24	NJ	31	CT
. . .	. . .	. . .	. . .	. . .	. . .
2015	9	12	CT	24	NY

Your Answer:

```
with q1 as(
select year,month, max(quant) MAX_Q, min(quant) MIN_Q
from sales
group by year,month
),
q2 as(
select q1.year, q1.month, s.day BUSIEST_DAY,
s.state MAX_STATE, q1.MIN_Q
from q1, sales s
where q1.year = s.year and q1.month = s.month
and MAX_Q = s.quant
),
q3 as(
select q2.year, q2.month, q2.BUSIEST_DAY, q2.MAX_STATE, s.day
SLOWEST_DAY,
s.state MIN_STATE
from q2, sales s
where q2.year = s.year and q2.month = s.month
and MIN_Q = s.quant
)
select year,month, BUSIEST_DAY, MAX_STATE, SLOWEST_DAY,
MIN_STATE
from q3
```

Question 3

7 / 10 pts

Write an SQL query for the following:

For each customer, find the “busiest” date (year, month, day) and “slowest” date -- i.e., made the most/least numbers of sales transactions (i.e., maximum/minimum counts of sales transactions, not maximum/minimum quantities)

sales (prod, cust, yr, mo, day, state, quant)

cust	busy_yr	busy_mo	busy_dy	slow_yr	slow_mo	slow_dy
Sam	2020	6	27	2012	12	31
Mia	2017	11	17	2009	3	25
. . .	. . .	. . .	. . .	. . .	. . .	. . .
Emily	2011	9	23	2012	7	2

Your Answer:

```
with q1 as(
select cust, count(quant) MIN_Q, count(quant) MAX_Q
from sales
group by cust
),
q2 as(
select q1.cust, q1.MIN_Q,
s.month slow_MONTH, s.day slow_DAY, s.year slow_YEAR,
q1.MAX_Q
from q1, sales s
where q1.cust = s.cust
and MIN_Q = s.quant
),
q3 as(
select q2.cust, q2.MIN_Q, CONCAT_WS ('/', slow_MONTH, slow_DAY ,
slow_YEAR) slow_DATE, q2.MAX_Q,
s.month busy_MONTH, s.day busy_DAY, s.year busy_YEAR
from q2, sales s
where q2.cust = s.cust and q2.MAX_Q = s.quant
)
select cust CUSTOMER, slow_DATE, CONCAT_WS ('/', busy_MONTH,
busy_DAY , busy_YEAR) busy_DATE
from q3
```

incorrect logic: you compare quantity to count value

Question 4

10 / 10 pts

Write an SQL query for the following:

For each (cust, prod) combination, find the average sales quantities for ‘NY’, ‘NJ’, and ‘CT’, and display the averages sales quantities only if NJ average is greater than both NY and CT averages.

sales (prod, cust, yr, mo, day, state, quant)

cust	prod	ny_avg	nj_avg	ct_avg
Claire	Ham	48	98	15
Dan	Eggs	12	67	9
. . .	. . .	. . .	. . .	. . .
Chae	Butter	3	49	2

Your Answer:

```
with q1 as (
select distinct prod, cust
from sales
order by cust
),
q2 as (
select cust, prod, round(avg(quant)) NY_AVG
from sales
where state='NY'
group by cust, prod
order by cust
),
q3 as (
select cust, prod, round(avg(quant)) NJ_AVG
from sales
where state='NJ'
group by cust, prod
order by cust
),
q4 as(
select cust, prod, round(avg(quant)) CT_AVG
from sales
where state='CT'
group by cust, prod
order by cust
)
select * from q1
left join q2 using(cust,prod)
left join q3 using(cust,prod)
left join q4 using(cust,prod)
where NJ_AVG > NY_AVG or NJ_AVG > CT_AVG
```

Question 5

10 / 10 pts

Write an SQL query for the following:

For each customer and year between 2011 and 2020, find the total sales quantities (i.e., SUM) for the given year, and the total sales quantities for a decade prior in 2 separate columns -- For example, for 'Boo' and the year 2011, find total sales quantities for 2011 and 2001.

sales (prod, cust, yr, mo, day, state, quant)

cust	year	total_sales_quant_for_the_year	total_sales_quant_for_a_de
Boo	2011	29,243	14,253
Wally	2019	41,295	28,571
. . .	. . .	. . .	. . .
Sam	2017	149,206	90,214

Your Answer:

```
with q1 as(
select cust, year, sum(quant) total
from sales
group by cust, year
),
q2 as(
select cust, year, sum(total) total_sales_quant_for_the_year
from q1
where year between 2011 and 2020
group by cust, year
),
q3 as(
select cust, year, sum(total) total_sales_quant_for_a_year_prior
from q1
where year between 2011 and 2001
group by cust, year
)
select q1.cust, q1.year, total_sales_quant_for_the_year,
total_sales_quant_for_a_year_prior
from q1
left join q2 using(cust,year)
left join q3 using(cust,year)
```

Question 6

5 / 10 pts

Write the following query using:

1. WITH
2. NESTED SUB-QUERY

For each year and month combination, show the maximum and minimum quantities for apples and bananas only if the average quantities are greater than 100.

Your Answer:

1.

```
with q1 as(
select year, month, prod, max(quant) total, avg(quant) avg_quant
from sales
group by year, month, prod
),
q2 as(
select year, month, total max_apple
from q1
where q1.prod = 'APPLE'
),
q3 as(
select year, month, total max_banana
from q1
where q1.prod = 'BANANA'
)
select q2.year, q2.month, max_apple, max_banana
from q2,q1
left JOIN q3 using(year, month)
where q1.avg_quant > 100
```

2. select year, month , avg(quant) avg_q, max(quant) from

(select year, month, max(quant) ,avg(quant) from sales

group by year, month

where prod = 'Apple' and

prod = 'Banana') as max_sales(year, month, max(quant))

where avg_q > 100

1) Incorrect join logic results in duplicate rows (-2) 2) syntax and structure is incorrect (-4)

Question 7

2 / 15 pts

Use the schemas below to provide the following SQL queries:

- course (course_id, title)
- prereq (course_id, prereq_id)

(5 pts.) Rewrite the following query WITHOUT using JOIN syntax:

```
select * from course natural join prereq
```

(10 pts.) Rewrite the following query USING JOIN syntax:

```
select c.course_id, c.title, null
  from course c
 where not exists
 (
    select * from prereq p
   where c.course_id = p.course_id
 )
union all
select p.course_id, null, p.prereq_id
 from prereq p
 where not exists
 (
  select *
    from course c
   where p.course_id = c.course_id
 )
```

Your Answer:

1.

select * from course

intersect

select * from prereq

2.

(select c.course_id,c.title, null from course c

left outer join prereq on c.course_id = p.course_id)

natural join

(select p.course_id, null, p.prereq_id from prereq p

left outer join course on p.course_id = c.course_id)

1) Intersect is not the correct way to join the tables (-4) 2) This does not reflect what the prompt was asking for (-9)

Question 8

0 / 5 pts

Rewrite the following query using a nested sub-query:

```
select customer_name
  from borrower a natural join loan l
 where l.branch_name = 'Hoboken'
except
select customer_name
  from depositor d natural join account a
 where a.branch_name = 'Hoboken'
```

Your Answer:

```
select distinct customer_name
from borrower a, loan l
where a.loan_number = l.loan_number and b.branch_name =
'Hoboken'
and (branch_name, customer_name)
except
(select branch_name, customer_name
from depositor d, account a
where  d.account_number = a.account_number)
```

This is not a NSQ

Question 9

12 / 20 pts

Answer the following questions **in your own words** -- each question is worth 4 points:

1. Define what ***composite attribute type*** is. Why is a composite attribute problematic?
2. What is a ***foreign key***? And in what ways is it helpful to have foreign keys?
3. Is it true that “every ***relation*** has a primary key”? In either case of “YES” or “NO”, justify your answer.
4. How would you check to see if "***two given tables are identical***" using a SQL query? Write a sample SQL query and provide a brief explanation (of how the query works to check if the two given tables are identical).
5. Is it possible to ***convert any SQL query into an equivalent relational algebraic expression (assuming relational algebra can support multi-sets)***? In either case of “YES” or “NO”, justify your answer.

Your Answer:

1. Composite attribute type is said to have atomic data if whole value consists of single entity and not composite entity. It depends on the type of application or how data is used in business.

2. Foreign key is key to parents data which is used to identify your parents records.

- Foreign key is field in one table that helps to refer primary key in another table.

3. yes it is required to have primary key in every relation because primary key is used to uniquely identify rows in table. And if we need to perform any operation like join it is required to have primary key to form a relationship with another table.

4. For example select distinct * from tb1 and select distinct * from tb2

first check for duplicates row in tb1 and tb2 respectively. Then perform inner join operation on every column.

5. Yes, it is possible to convert SQL query in relational algebra because we can convert subquery into select, joins and projections which are features provided in relation algebra.

1) This does not answer the question (-4) 4) This would require a visual check on each table which is not the purpose of the question (-4)

Quiz Score: **66** out of 100