CS585 Midterm exam

2016-10-14

Duration: 1 hour.

Last Name: _____

First Name: _____

Student ID: _____

Email: _____

Hello! There are 9 questions below (8 plus a bonus), one per page. Please read each question carefully before answering. You don't to elaborate on anything, so you won't need additional sheets.

The exam is CLOSED book/notes/devices/neighbors(!) but 'open mind' :)

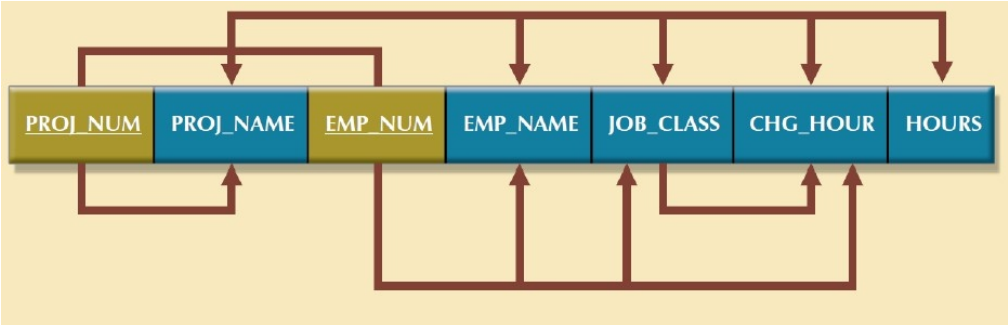
If you are observed cheating, or later discovered to have cheated in any manner, you will get a 0 on the test and also be reported to SJACS - so please don’t!

When we announce that the time is up, you NEED to stop writing immediately, and turn in what you have; if you continue working on the exam, we will not grade it (ie. you will get a 0).

Have fun, and good luck! Hope you do well.

Saty

Q1 (1+3=4 points). A 1NF table, such as the one shown below (we covered this in class on great detail), is analyzed to detect problems (related to unwanted dependencies), which are then systematically eliminated using a diagram such as the shown below (the table is converted to 2NF, then 3NF), in a process called 'normalization'.



a. What is the diagram (shown above) called?

b. How does the diagram aid in normalization? Explain briefly, using the above diagram (you can mark it up if you want).

Q2 (4 points). Two-phase locking (2PL) is a popular concurrency control scheme for managing transactions. Unfortunately, however, it cannot entirely prevent deadlocks from occurring. Using two transactions T1 and T2, show (using a sequence of events you come up with) how a deadlock can occur between them.

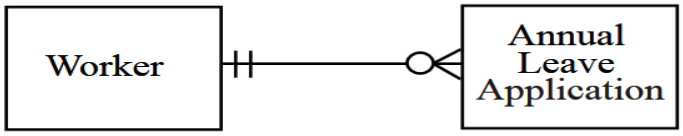
Q3 (4 points). In 1970, Ed Codd at IBM published a landmark paper that totally redefined the state of art of databases at the time. What was Codd's breakthrough? Explain, using your own words and diagrams. You need to have at least 4 sentences in your answer.

Q4 (2+2=4 points). For a while now, NASA has been conceptualizing a network called the Interplanetary Internet, which could come in handy 'someday' when we colonize Mars [when pigs fly out of our butts :)]. If that were to come to fruition, Eric Brewer's 'CAP theorem' would be highly relevant and applicable to such a distributed system of nodes. As per the CAP theorem, 'you can't always get what you want' (at least not C,A,P all at once, all equally guaranteed).

In an Interplanetary Internet, how would you rank C,A,P in terms of concerns? In other words, which would we worry about most, and relatively which, the least? You need to state why (justify your ordering).

Where might nodes be located, for an Interplanetary Internet? And, what disaster scenarios can you envision (that affect the network)?

Q5 (4 points). Consider the following relation:



As per the above, any employee in a certain company could file annual leave applications (submit vacation requests). The management institutes a new policy, stating that only full-time employees (not part-timers or contractors) can do so. How would you reflect the policy change via an updated diagram?

Q6 (2 points). Below is a table that tracks projects. There are several steps (phases) in each project (workorder), and for each step, its completion status is maintained (C means completed, A means awaiting completion).

```
CREATE TABLE Projects
(workorder_id CHAR(5) NOT NULL,
step_nbr INTEGER NOT NULL CHECK (step_nbr BETWEEN 0 AND 1000),
step_status CHAR(1) NOT NULL
CHECK (step_status IN ('C', 'A')),
PRIMARY KEY (workorder_id, step_nbr));
```

Here is some sample data conforming to the definition above:

```
Projects
workorder_id step_nbr step_status
================================
'O100'          0          ‘C’
'O100'          1          ‘A’
'O100'          2          ‘A’
'O200'          0          ‘A’
'O200'          1          ‘A’
'O300'          0          ‘C’
'O300'          1          ‘C’
```

Given the above, what does the following query do?

```
SELECT workorder_id
FROM Projects AS P1
WHERE step_nbr = 0
AND step_status = ‘C’
AND ‘W’ = ALL (SELECT step_status
FROM Projects AS P2
WHERE step_nbr <> 0
AND P1.workorder_id = P2.workorder_id);
```

Q7 (4 points). Here are a pair of tables – a PRODUCTS table that lists products a company sells, and SALES, which records sales of the products (each unit of a product that is sold, gets a separate row in SALES):

PRODUCTS(PRODUCT_ID, PRODUCT_NAME);
SALES(SALE_ID, YEAR, PRODUCT_ID, PRICE);

Consider the following three queries, we're calling them Q1, Q2, Q3. In Q2, fyi, 'SELECT 1' returns a 1, which we can ignore (it is not essential to our query).

```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
 JOIN
 PRODUCTS P
 ON (S.PRODUCT_ID = P.PRODUCT_ID)
GROUP BY S.PRODUCT_ID;

SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
WHERE EXISTS
(
 SELECT  1
 FROM PRODUCTS P
 WHERE P.PRODUCT_ID = S.PRODUCT_ID
)
GROUP BY S.PRODUCT_ID;

SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
WHERE PRODUCT_ID IN
(
 SELECT  PRODUCT_ID
 FROM PRODUCTS P
)
GROUP BY S.PRODUCT_ID;
```

Circle the correct choice below:

a. Q1, Q2, Q3 are all different (they produce different results)
b. Q1, Q2, Q3 are all identical
c. Q1 and Q2 are identical
d. Q1 and Q3 are identical
e. Q2 and Q3 are identical

Q8 (4 points). The following is a table that tracks sales made by salespeople across several districts (eg. at a car dealership).

```
CREATE TABLE SalesData
(district_nbr INTEGER NOT NULL,
sales_person CHAR(10) NOT NULL,
sales_id INTEGER NOT NULL,
sales_amt DECIMAL(5,2) NOT NULL);
```

Here is some conforming data:

```
SalesData
district_nbr sales_person sales_id sales_amt
=========================================
1          'Curly'         5          3.00
1          'Harpo'         11         4.00
1          'Larry'         1          50.00
1          'Larry'         2          50.00
1           'Larry'        3          50.00
1          'Moe'           4          5.00
2          'Dick'          8          5.00
2          'Fred'          7          5.00
2          'Harry'         6          5.00
2          'Tom'           7          5.00
3          'Irving'        10         5.00
3          'Melvin'        9          7.00
4          'Jenny'         15         20.00
4          'Jessie'        16         10.00
4          'Mary'          12         50.00
4          'Oprah'         14         30.00
4          'Sally'         13         40.00
```

Here are a pair of queries against the table shown earlier.

```
SELECT *
FROM SalesData AS S0
WHERE sales_amt IN (SELECT S1.sales_amt
FROM SalesData AS S1
WHERE S0.district_nbr = S1.district_nbr
AND S0.sales_amt <= S1.sales_amt
HAVING COUNT(*) <= 3)
ORDER BY S0.district_nbr, S0.sales_person, S0.sales_id,
S0.sales_amt;
```

```
SELECT DISTINCT district_nbr, sales_person
FROM SalesData AS S0
WHERE sales_amt <= (SELECT MAX(S1.sales_amt)
FROM SalesData AS S1
WHERE S0.district_nbr = S1.district_nbr
AND S0.sales_amt <= S1.sales_amt
HAVING COUNT(DISTINCT S0.sales_amt) <= 3);
```

Question: how are the two queries similar, and how are they different? In other words, what does each do, which makes them be alike and also distinct?

Bonus question (1 point).

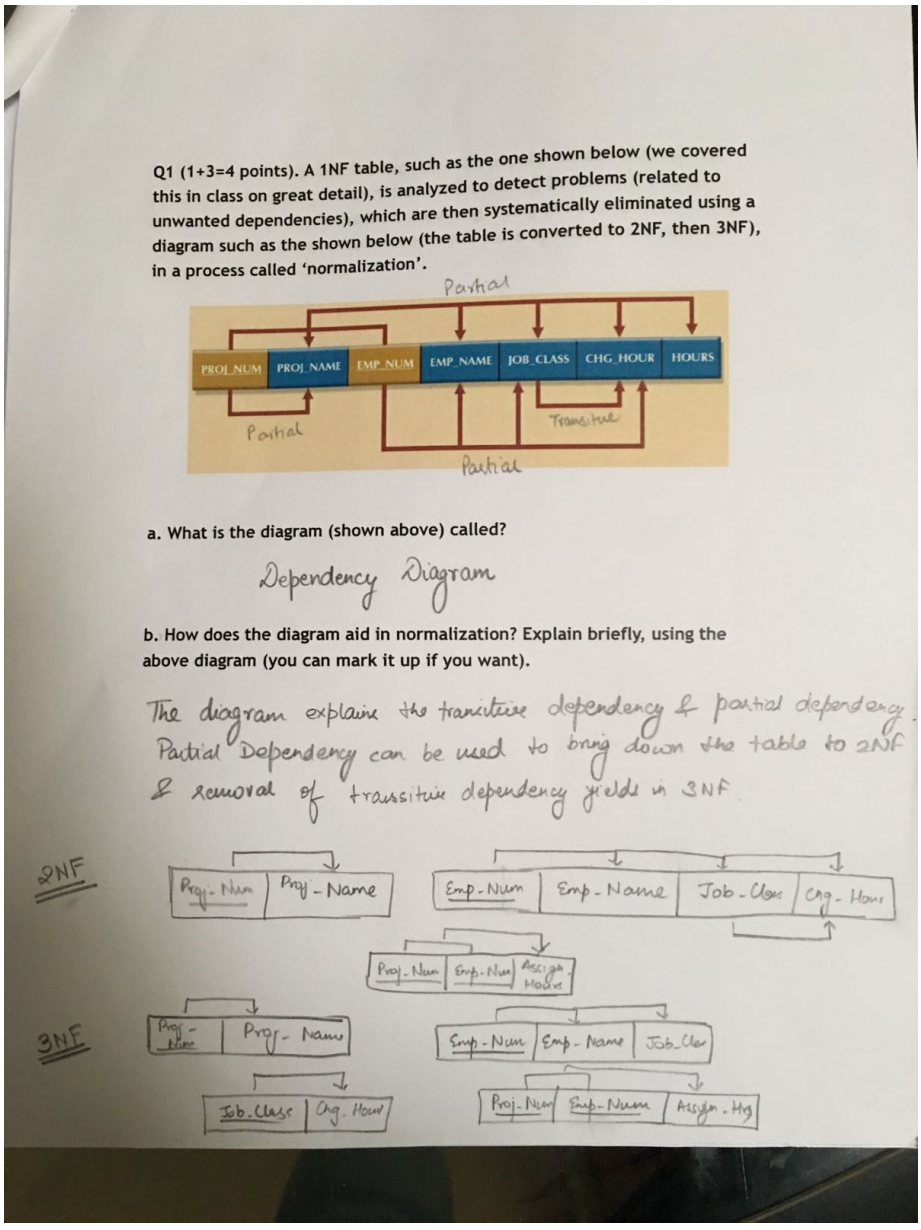
How would you add 4 matches to the above square (using matches identical in size to the ones above), that result in four triangles and two squares? No need to bend/break.. any match.

Question 1:

Part a)
Dependency Diagram

Part b)
Show partial and transitive dependency. Removal of partial dependency is done in 2NF and removal of transitive dependency is done in 3NF.

- 1 : Showing Partial and Transitive dependency
- 1 : Explaining the normalization process
- 1 : Explaining that reducing to 2NF involves eliminating partial dependencies and reducing to 3NF involves transitive dependency.



Q1 (1+3=4 points). A 1NF table, such as the one shown below (we covered this in class on great detail), is analyzed to detect problems (related to unwanted dependencies), which are then systematically eliminated using a diagram such as the shown below (the table is converted to 2NF, then 3NF), in a process called 'normalization'.

a. What is the diagram (shown above) called?

Dependency Diagram

b. How does the diagram aid in normalization? Explain briefly, using the above diagram (you can mark it up if you want).

The diagram explains the transitive dependency & partial dependency. Partial Dependency can be used to bring down the table to 2NF & removal of transitive dependency yields in 3NF

Question 2: Points are given as follows:

- 1 : For defining the steps in transaction T1
- 1 : For defining the steps in transaction T2
- 2 : For explaining the deadlock situation

Solution: as below from lecture slides

An important and unfortunate property of 2PL schedulers is that they are subject to *deadlocks*. For example, suppose a 2PL scheduler is processing transactions T_1 and T_3

T_1: $r_1[x] \rightarrow w_1[y] \rightarrow c_1$ T_3: $w_3[y] \rightarrow w_3[x] \rightarrow c_3$

and consider the following sequence of events:

1. Initially, neither transaction holds any locks.
2. The scheduler receives $r_1[x]$ from the TM. It sets $rl_1[x]$ and submits $r_1[x]$ to the DM.
3. The scheduler receives $w_3[y]$ from the TM. It sets $wl_3[y]$ and submits $w_3[y]$ to the DM.
4. The scheduler receives $w_3[x]$ from the TM. The scheduler does not set $wl_3[x]$ because it conflicts with $rl_1[x]$ which is already set. Thus $w_3[x]$ is delayed.
5. The scheduler receives $w_1[y]$ from the TM. As in (4), $w_1[y]$ must be delayed.

Question 3:

Points are given as follows :
The breakthrough was Relational Database and the operations associated with them.

- 4 marks - If mentioned about the relational database and explained properly.
- 3 marks - If explanation is not satisfactory
- 2 marks if only mentioned about set operations and not relational database/no explanation
- No marks if the student has written about Performance Tuning, SQL,Distributed Databases,File Systems, Use of Databases or any other irrelevant matter,

Question 4:

Part a)
The correct order is Partition Tolerance, Availability, Consistency (P,A,C). However, the order (A,P,C) is also accepted.
Points are given as follows:

- 0.5 : For correct order.
- 0.5 : For correct acronym expansion.
- 1 : For correct reasons. 0.5 each for justification of the most and the least important concern. NOTE: If your least or the most important concern is incorrect, then no points will be given for justification. For example, if your ordering is A,C,P, then 0.5 will be deducted, as the justification of the least important concern is incorrect. Similarly, if you have used incorrect acronym, for example, A for Accuracy, then your justification for A will be incorrect.

Part b)
Solution: Nodes can be located on other planets, satellites or different places on earth. Disaster scenario may meteoroid strike, problems in satellite etc.

Answer should include correct location information along with two disaster scenarios.
Points are given as follows:

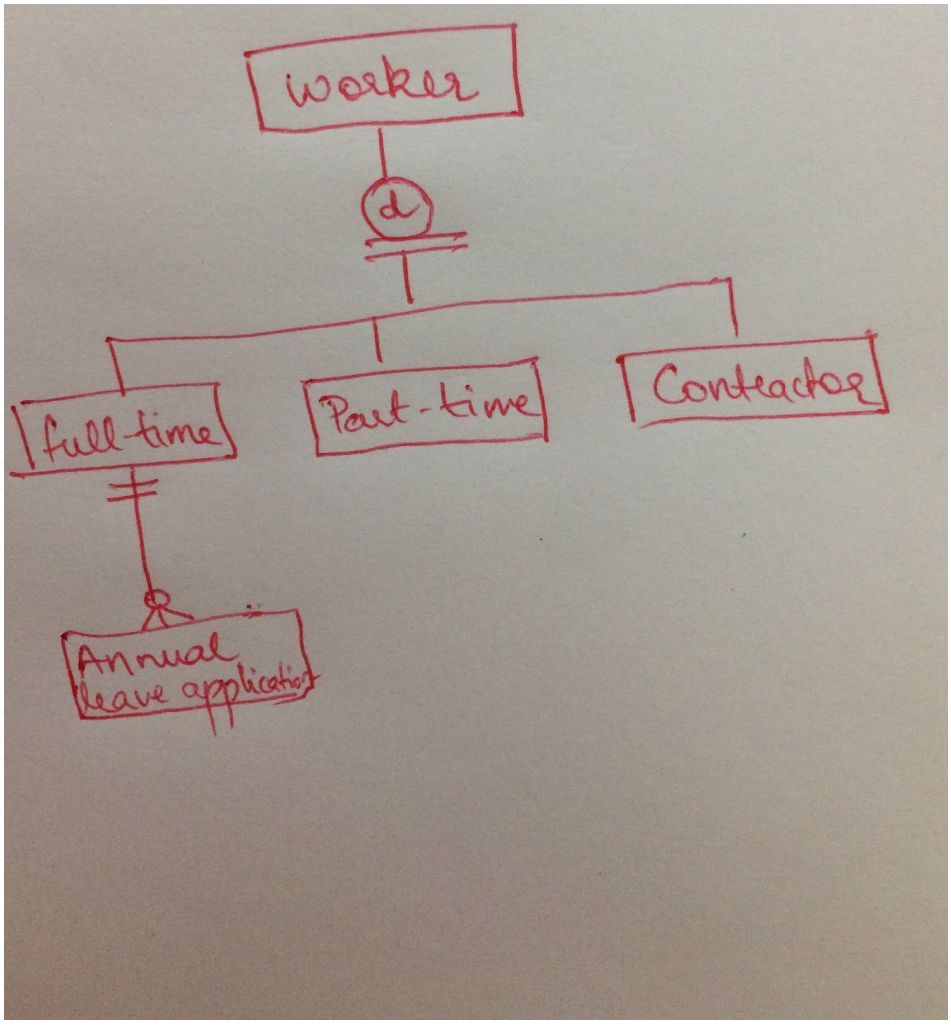
- 1 : Correct location information.
- 0.5 each for the disaster scenario . At least two disaster scenarios are required.

Question 5

Common mistakes for Q-5,

- Contractor is not shown as a separate sub-class
- Disjoint hierarchy is not used
- Annual leave application should be associated with only the full-time sub-class, not the worker super-class
- Diagram is required, only the explanation is not given grade

Solution for question 5,



Question 6:
This is the rubric i followed while awarding points for the question 6

1.5 points for correct explanation
0.5 points for final query result

Answer:
1) Expected Explanation: (1.5 points)
The query returns workorder_id of those projects whose '0'th step has a ‘completed’ or ‘C’ status and all other steps are in "awaiting" or ‘A’ status.
2) Query Result: Workorder_id: (0.5 points)
0100

Common Mistake:
Most students have given the correct explanation but have not provided the final result and hence lost -0.5 points for the same.

If the explanation is correct but the final answer is incorrect, then i have deducted 1 point.

Some students said the query will return Null as they have not understood that the query is correlated or made a mistake interpreting the inner subquery.

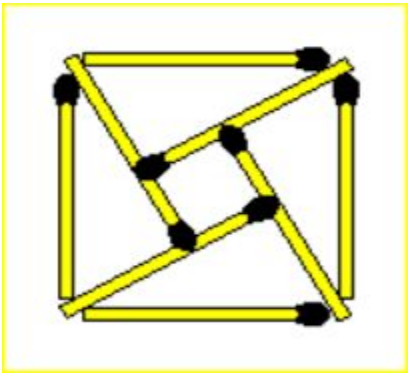
Question 7:

The common mistakes done by students were:

- They could only identify two of the queries to be similar.
- Another mistake was to mark Two choices as correct and specially the case was when marking all the three identical as well as just 2 queries identical to be the answer. In that case I had to give them 0.

Question 8 (BONUS)

ONLY ONE ANSWER (Kindly do not argue over your solution. Following is the only solution to this problem.)



CSCI 585: Database Systems

Spring 2016 - Midterm Exam

3/4/16, 6:00-7:00 PM

Name: _____

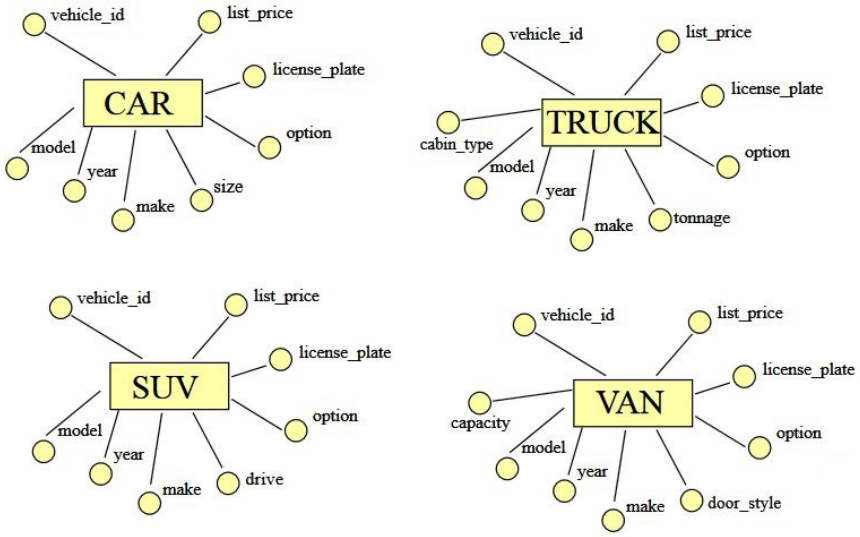
Student ID: _____

Question	Your score	Max score
1		3
2		6
3		4
4		3
5		4
6		2
7		5+1
8		3
Bonus		1
Total		32

HAVE FUN!!

Question 1 [3 points]

Consider the four entities shown below:



Draw a single E-R diagram that depicts the info contained, as a subclass/superclass hierarchy - use whatever notation you are familiar with. Be sure to indicate a suitable primary key.

The expected answer is this: a 'vehicle' (or transportation device etc) superclass, with all the common properties (including vehicle_id, list_price etc), with CAR, TRUCK, SUV, VAN being subclasses (each with its specific attributes, eg. CAR would have 'size'). vehicle_id is the PK.

Question 2 [6 points]

The following table is in 1NF:

```
CREATE TABLE Classes
(course CHAR(7) NOT NULL,
 section CHAR(1) NOT NULL,
 time INTEGER NOT NULL,
 room INTEGER NOT NULL,
 roomsize INTEGER NOT NULL,
 professor CHAR(25) NOT NULL,
 student CHAR(25) NOT NULL,
 major CHAR(10) NOT NULL,
 grade CHAR(1) NOT NULL);
```

Knowing the student and course is enough to determine section and grade. Also, a student can have just a single major. Knowing these facts, convert the above table to 2NF (where there are no partial dependencies); be sure to include primary keys.

2NF:

```
CREATE TABLE Classes (course, section, room, roomsize, time, professor, PRI

CREATE TABLE Enrollment (student, course, section, grade, PRIMARY KEY(stude

CREATE TABLE Students (student, major), PRIMARY KEY(student));
```

Further, 'roomsize' needs to depend (just) on 'room'. Knowing this, create a 3NF version (where there are no transitive dependencies); again, include suitable PKs.

3NF:

```
CREATE TABLE Classes (course, section, room, time, professor, PRIMARY KEY(c

CREATE TABLE Rooms (room, roomsize, PRIMARY KEY(room));

CREATE TABLE Enrollment (student, course, section, grade, PRIMARY KEY(stude

CREATE TABLE Students (student, major), PRIMARY KEY(student));
```

Question 3 [4 points]

Here is a SQL query:

```
SELECT sno, sname
FROM Suppliers
WHERE 100 > (SELECT SUM(quantity)
             FROM Shipments
             WHERE Shipments.sno = Suppliers.sno);
```

In the above, sno stands for 'supplier number', and sname, for 'supplier name'. Describe using a sentence or two, what the query does.

Lists suppliers who have shipped less than 100 shipments.

Here is another query that has to do with product managers (a product manager is responsible for selling a product - product managers get products out of warehouses and into stores, where they sell them). The Personnel table lists product managers and their products, while the Warehouses and Stores tables list products and their quantities. What does the query produce?

```
SELECT manager, product
FROM Personnel as P1
WHERE (SELECT SUM(qty)
       FROM Warehouses AS W1
       WHERE P1.product = W1.Product)
    < (SELECT SUM(qty)
       FROM Stores AS S1
       WHERE P1.product = S1.Product);
```

Lists product managers who have more product in stores than in warehouses.

Question 4 [3 points]

Imagine there are two items X and Y in a database, and two transactions T1 and T2 that operate (read, write) on them.

T1 reads X and Y, and modifies X: T1:R(X), T1:R(Y), T1:W(X), c1 [c1 is 'commit 1', ie. T1's commit].

T2 reads X and Y, and modifies both X and Y: T2:R(X), T2:R(Y), T2:W(X), T2:W(Y), c2.

When T1 and T2 interleave their operations shown above, the following problems can happen (unless we carefully avoid them, eg. using two-phase locking):

- write-write conflict, aka "lost update": two transactions overwrite an object - the second (latter) transaction's update makes the first transaction's update to become 'lost'
- write-read conflict, aka "dirty read": one transaction reads a value, after it has written over by another transaction which not yet committed
- read-write conflict, aka "unrepeatable read": one transaction reads the value of an object twice, and another transaction overwrites that value in-between the two reads of the first transaction

Examine the following transaction histories, and indicate which of the if any of the above three problems could occur, and if so, indicate which problem:

- T2:R(X), T2:R(Y), T2:W(X), T1:R(X) ... : write-read conflict
- T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:R(X) ... : read-write conflict
- T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:W(X) ... : write-write conflict

Question 5 [4 points]

Rewrite the following queries (eg. for optimization purposes):

```
SELECT product_id, product_name
FROM product
WHERE unit_price BETWEEN MAX(unit_price) and MIN(unit_price)
```

Answer:

```
SELECT product_id, product_name
FROM product
WHERE unit_price >= MAX(unit_price)
and unit_price <= MIN(unit_price)

<=MAX and >=MIN is also acceptable, because of the way I stated the query
```

```
SELECT *
FROM product p
WHERE product_id IN
(SELECT product_id
 FROM order_items)
```

Answer:

```
SELECT *
from product p
where EXISTS (SELECT * from order_items o
              where o.product_id = p.product_id)
```

Question 6 [2 points]

How would you categorize the following transaction scheme (T is a transaction, X and Y are distributed objects)?

- T starts execution
- T reads X at initiating site
- T writes Y at initiating site
- updated value of Y is sent to all nodes that have duplicates of Y, along with a timestamp of when the update occured
- if the other nodes can all update their Y values without conflict, the transaction goes through, ie is committed; otherwise the transaction is discarded (and restarted)

Optimistic locking (or optimistic concurrency control).

Question 7 [5+1 points]

USC has been around since 1880. There have been hundreds of thousands of undergrads who have been graduating since then, and we have pretty detailed info on their academic performance (via transcripts). Imagine we would like to do multidimensional data analysis, specifically on the cumulative GPAs of all the (undergrad) students who have graduated out of 'SC. The 'fact table' might consist of entries that include GPA, student name, major, etc.

We need to build a data warehouse to perform the analysis, and choose to use ROLAP to do so.
For such a scenario, draw a star schema that depicts the fact table along with several dimension tables. Indicate relevant attributes inside each table. In other words, what factors can we consider, to mine this rich data?

The fact table would consist of entries that include:

```
Cumulative_GPA Gender School_Within_University Major Minor City State Country Year_Graduated A
```

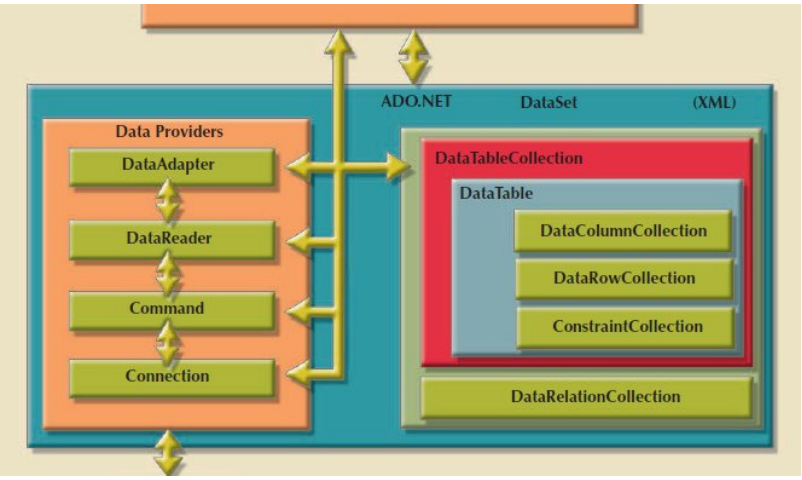
The corresponding dimension tables would be School, Major, Minor, Gender, Location, Year, Age.

Bonus (1 point): create a 'snowflake' schema instead.

A dimension table such as Location could further be split into City, State, Country; Major could be subdivided into Social_Science, Physical_Science, Engineering, etc.

Question 8 [3 points]

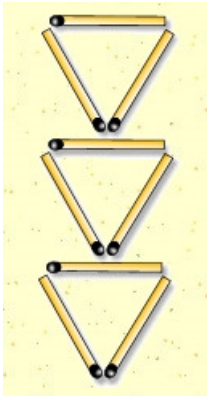
Shown below is the centerpiece of Microsoft's ADO.NET architecture. What does it enable (what capability does it provide)?



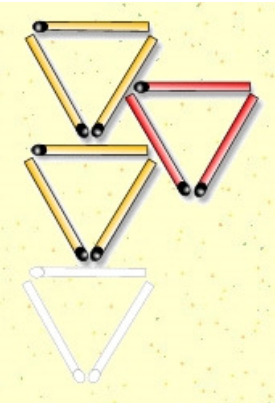
ADO.NET is used to create an XML-based, in-memory view of an underlying database.

Bonus (non-DB-related) [1 point]

Create four equilateral triangles by moving just three matches - no overlapping or breaking allowed.



Answer:



Other solutions are OK too - moving just two matches, or creating a tetrahedron.

CSCI585 Midterm exam

2017-03-03

Duration: 1 hour

Last Name: _____

First Name: _____

Student ID: _____

Email: _____

Hi there! There are 9 questions below (8 plus a bonus), one question per page. Please read each question carefully before answering. There's need to elaborate on anything, so you shouldn't need extra sheets.

The exam is **CLOSED** book/notes/devices/neighbors(!) but 'open mind' :) If you are observed cheating, or later discovered to have cheated in any manner, you will get a 0 on the test and also be reported to SJACS - so please don't! **DO YOUR OWN WORK**.

When we announce that the time is up, you NEED to stop writing immediately, and turn in what you have; if you continue working on the exam, we will not grade it (ie. you will get a 0). So please stick to the limit of one hour, use time wisely!

Have fun, and good luck - hope you do well!

Saty

Q1 (4 points).
Suppose an online vendor maintains its customer list like so:

firstName	lastName	address	city	state	ZIP	phoneNumber	SkypeID	emailAddress
A	B	123 Main St	Los Angeles	CA	90089	213-543-6543		AB@mail.com
Fam	Act	222 Burton Way	Beverly Hills	CA	90210		RichNFamous	RNF@imdb.com
MoreFam	Act	108 Roxbury St	Beverly Hills	CA	90210	323-654-1002		TheBest@BevHills.us
Grad	Student	154 Adams St	Los Angeles	CA	90089			DontBugMe@usc.edu

What two problems do you see with the above scheme, and how would you fix them? Your answer can be in the form of E-R (using any notation), or in table format (like above) or even SQL. And, feel free to create any new attributes that might be necessary.

Repetition of data, with city and state names (so make a separate table of these, with ZIP as the PK); missing (NULL) values for contact info (so, make a separate ContactInfo table with (ContactID, ContactType, ContactValue) rows and move the contact data there (no NULLS will exist because we will have a new row for each contact type a person has).

Q2 (4 points). Parents in a wealthy family want to create a DB of all their assets. For each asset, they would like to name benefactors - some or all of their five children who would get the asset. Each asset has a financial value associated with it, and a maturity date (when the kid(s) can cash in). They'd like to track the following diverse set of assets they own: bank accounts, real estate, stocks, jewelry, life insurance. **What would be a good design (using an ER diagram) for this?** You can make any assumptions you want about the assets, create whatever descriptors (columns) you need, etc.

Make a superclass entity Assets, and a Benefactors one, link them as 1:M. Under Assets, create BankAccounts etc. as subclass entities.

Q3 (3 points). A reality company keeps track of its home sales like so:

Seller	Buyer	LendingBank
S1	B1	BofA
S2	B1	Chase
S1	B2	Chase

Things seem fine (redundancy and all), until they hire you to 'clean up' their table. After analysis, you come up with these three separate tables [all linked properly with FK/PK], which makes for good design:

Table 'SellerBuyer', with rows such as (S1,B1).

Table 'BuyerBank', with (B2,Chase) as a sample row.

Table 'SellerBank', eg. with (S2,Chase) as a row.

You write the following three-way 'join' query just for fun, to see if you can recreate the original triplets (eg. S1,B1,BofA):

```
SELECT SB.Buyer, SN.Seller, BN.LendingBank
FROM SellerBuyer as SB, SellerBank as SN, BuyerBank as BN
WHERE BN.Buyer=SB.Buyer
AND BN.LendingBank=SN.LendingBank
AND SN.Seller=SB.Seller
```

Question: what, if any, is the problem with the above query?

The query will result in correct triples such as (S1,B1,BofA) etc, but ALSO wrong ones such as (S1,B1,Chase) [because it will multiply all three tables].

Q4 (1+1=2 points). You pull out your smartphone, log on to your banking app, and proceed to transfer $7200 (to pay for a 4-unit 'SC course!) from your savings account into your checking account. Prior to the transfer, you had $20,000 in savings and $800 in checking. While you are in the middle of doing this, due to poor DB design, a report generator (that would produce a monthly statement to email you) runs on the bank's server. **What could go wrong, and what is such a scenario called?**

If the report generator grabs the 'After' value of saving ($12,800) and 'Before' value of checking ($800), it will show our balance incorrectly as $13,600 [instead of $20,800]. This is an 'Inconsistent Retrieval'.

Q5 (2+2=4 points). How would you optimize (by rewriting) the following two queries?

a. `SELECT * FROM TBL WHERE substr(STATE,1,1)='C'`
[we want to select all rows containing states CA, CO, or CT;
substr(<string>,1,1) returns the first character of a string]

WHERE STATE IN (CA,CO,CT) [or, can use OR]

b. `SELECT * FROM TBL WHERE AGE>21`
[the AGE column stores ages as 0..99 integers; assume it has been indexed]

WHERE AGE >=22 [the = will result in the index being used to fetch all entrie s that are >=22, no row-by-row comparison in the main table needed!]

Q6 (4 points). In the world of relational DBs, the 'ACID' properties ensure that a DB always preserves data integrity. In the newer world of Internet-enabled, distributed DBs, there is instead 'BASE'. **What two essential features of a DB are traded off, in BASE?** Explain using an example (or two).

Consistency (all copies of a fragment need to contain identical data), and Availability (a transaction should always be achievable, without 'downtime').

Q7 (4 points). What operation does the following SQL query implement?

```
SELECT DISTINCT c
FROM TABLE_A as t1
WHERE EXISTS (SELECT *
              FROM TABLE_B as t2
              WHERE t1.c = t2.c);
```

Finds the INTERSECTION of t1 and t2.

Q8 (5 points).

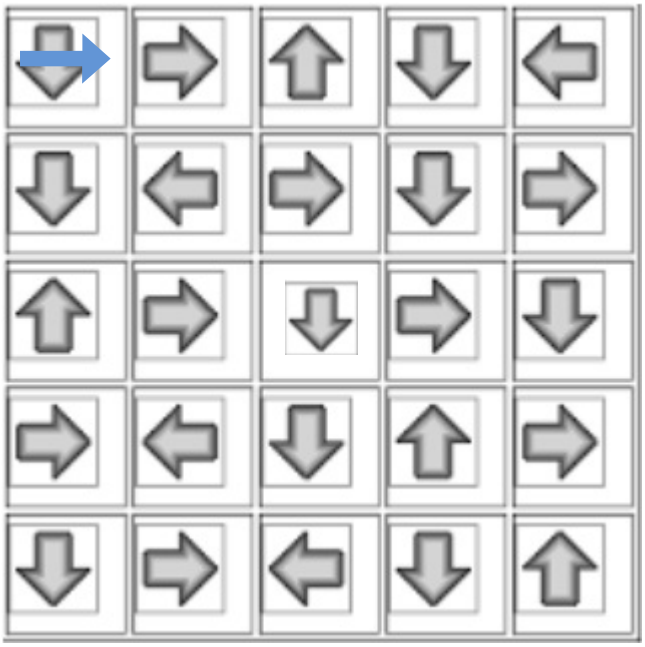
What does the following query do (:Name is simply a local variable)**?**

```
SELECT :Name, MAX(P1.reviewDate), P2.reviewDate
FROM EmpDB as P1, EmpDB as P2
WHERE P1.reviewDate<P2.reviewDate
      AND P1.EmpName=:Name
      AND P2.reviewDate = (SELECT(MAX(reviewDate) FROM
EmpDB)
GROUP BY P2.reviewDate;
```

Finds the 2 latest reviews for an employee.

Bonus question (1 point).

Complete the puzzle below..



Trace a clockwise spiral from the top-left, observe the sequence: down,right,up,down,left,right.. Repeat the sequence along our spiral path :) That makes the central square have a 'down' arrow.

CSCI585 Midterm exam

June 15th, 2017

Last Name: _____

First Name: _____

Student ID: _____

Email: _____

Signature: _____

Duration: 2 hours

CLOSED book and notes. No electronic devices.

DO YOUR OWN WORK.

If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS.

If you continue working on the exam after time is up you will get a 0.

Problem Set	Number of Points	Your Score
Q1	1+4=**5**	
Q2	1+1+1+2=**5**	
Q3	2+2+1=**5**	
Q4	1+2=**3**	
Q5	1+2=**3**	
Q6	2+1+1=**4**	
Total	**25**	

Q1. (5 points total) ER Modeling

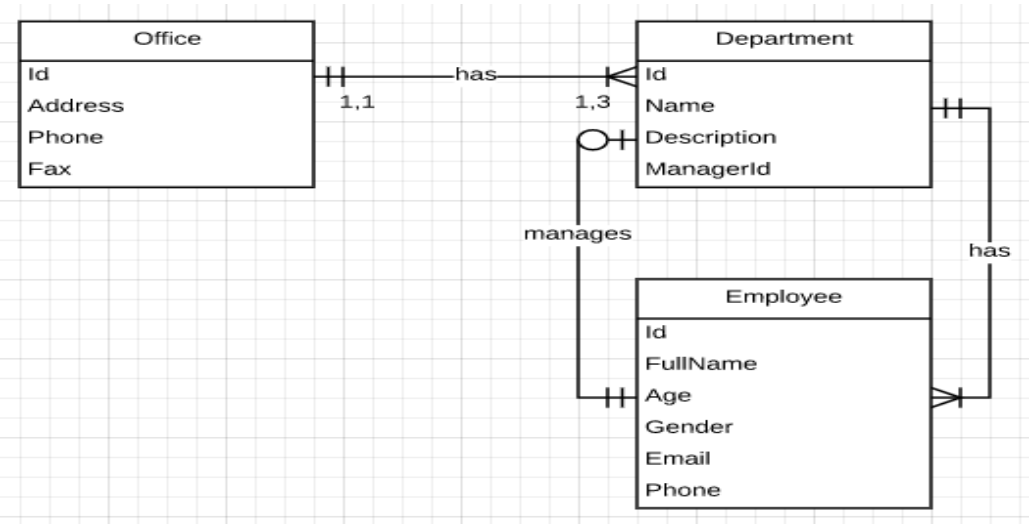
a. (1 point) What is weak entity? What is weak relationship? Give short example for each.

Answer: A weak entity is an entity that displays existence dependence and inherits the primary key of its parent entity. *Example: An EMPLOYEE might or might not have a DEPENDENT, but a DEPENDENT cannot exist without an EMPLOYEE.*

A weak relationship is a relationship in which the primary key of the related entity does not contain a primary key component of the parent entity. *Example: A course may have zero or many classes. A CLASS may have COURSE_ID as just a foreign key).*

b. (4 points) Draw ER Diagram based on the following description:

A company has multiple departments. Each department has multiple employees. An employee could not be in two departments at the same time. In each department, one employee is the manager that manages other employees. Each department is located at a specific office of the company. An office may have multiple departments, varying from one to three. **Answer:**



Q2. (5 points total) SQL

a. (1 point) What's the difference between inner and outer joins? Explain using example.

Answer: Inner join is a join operation in which only rows that meet selected criterion are selected. In outer join, all unmatched pairs are retained. Unmatched values in the related table are left null. Example: if joining product and vendor, outer join will also include products that aren't sold by vendors and vendors that aren't selling any products.

b. (1 point) Consider the following two query results:

SELECT count(*) AS total FROM books;

Total
100

SELECT count(*) AS author1_total FROM books WHERE authorId = '1';

author1_total
15

Given the above query results, what will be the result of the query below? Circle below.

```
SELECT count(*) AS author_not_1_total
FROM orders
WHERE authorId <> '1'
```

A) 50 B) 85 C) 15 D) Insufficient information

Answer: It will not necessarily be 85, because we do not know if there are any authorId's that are NULL. If there are 2 NULL authorId's, then the answer would be 83. Hence the answer is (d).

c. (1 point) Which relational operation does the following SQL query implement?

```
SELECT name
FROM driver
WHERE vehicle IN (SELECT vehicle FROM vehicles)
GROUP BY name
HAVING COUNT(*) = ( SELECT COUNT (*) FROM vehicles);
```

Answer: Division

d. (2 points) Given the following enrollment table, write a SQL query to list number of students enrolled in each course (ClassID).

StudentID	ClassID	Grade
321	CSCI495	B
564	CSCI110	C
321	CSCI585	A
564	CSCI495	A
789	EE101	F
321	EE101	B

Answer:

```
SELECT ClassID, COUNT(*) AS Total
FROM enrollment
GROUP BY ClassID;
```

Q3. (5 points total) NORMALIZATION

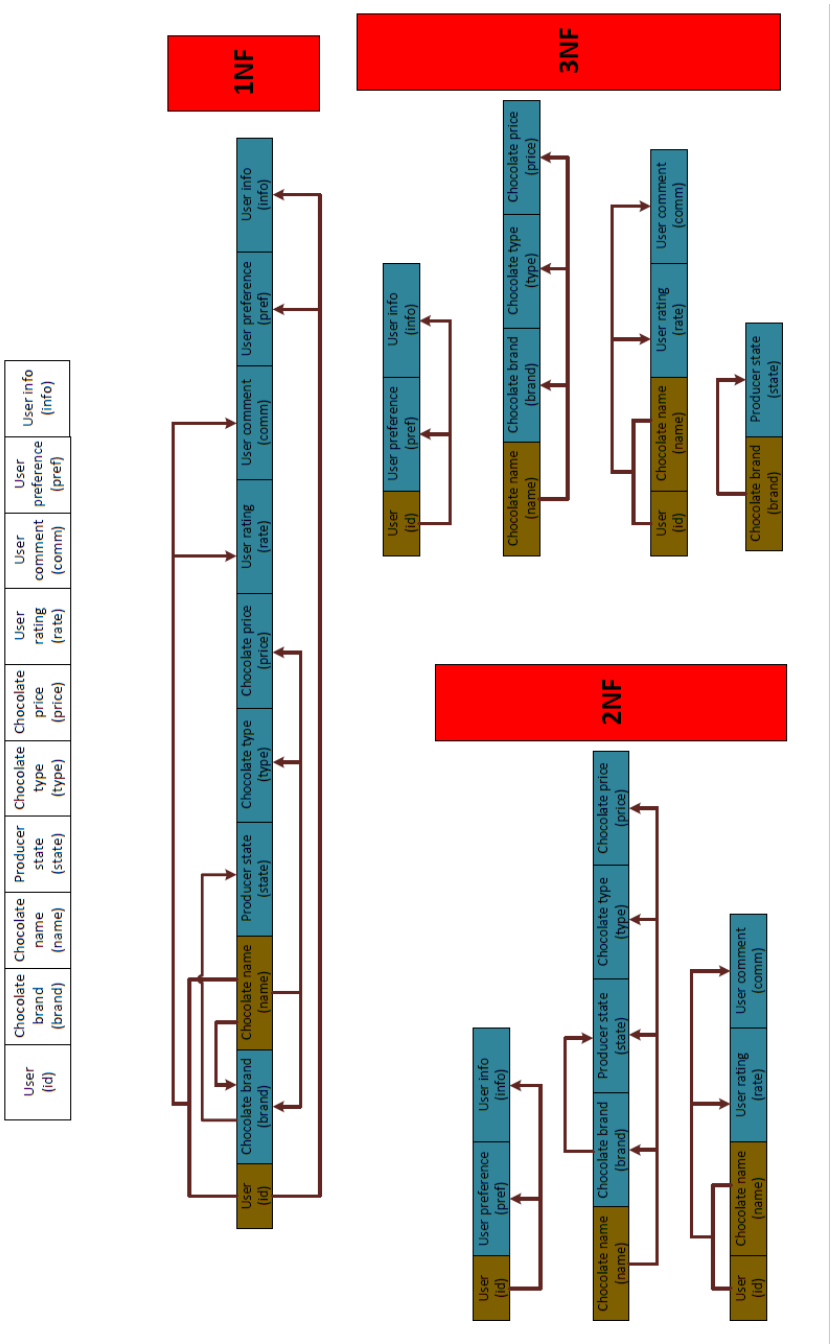
We are starting to design a new chocolate rating platform.

We have asked experts to taste different sweet products (sweets) and express their overall rating (0-5) and their professional comments. Each expert taster (ID) may try different products from different companies. We are only focused on US based companies and record the state of the headquarters of the company in our reports. Each product has a specific type, e.g. confection, candy, bar, etc. and also a unique price. Even though each taster can rate different types of products, we only record their top preference type for further reference. Moreover, basic personal information of each taster is available in our records.

Our sample data is in the following table:

ID	Brand	Name	State	Type	Price	Rate	Comm	Pref	Info
29001	Hershey	Reese's	PA	Confection	1.29	3	Good peanut butter!	Bar	Michael Mast
29001	Hershey	NutRageous	PA	Bar	1.39	4	Packaging isn't great	Bar	Michael Mast
29001	Hershey	Pieces	PA	Candy	0.99	2	Worst candy ever	Bar	Michael Mast
29001	Hershey	York	PA	Confection	0.99	3	Too sweet	Bar	Michael Mast
29001	Mars	Snickers	WA	Bar	1.45	3	Not a fan of almonds	Bar	Michael Mast
29001	Mars	Twix	WA	Bar	1.45	5	Best product ever	Bar	Michael Mast
29001	Mars	M&M	WA	Candy	1.79	3	The coating is too hard	Bar	Michael Mast
1202	Hershey	York	PA	Confection	0.99	3	Just a failed cop of PBCs	Candy	Paul Bulcke
1202	Hershey	Snack Barz	PA	Bar	0.89	3	Too hard	Candy	Paul Bulcke
1202	Mars	3Musketeers	WA	Bar	1.69	2	Best quality	Candy	Paul Bulcke
1202	Mars	Bounty	WA	Bar	1.55	3	Too much coconut	Candy	Paul Bulcke
1202	Mars	Twix	WA	Bar	1.45	3	Too much nuts	Candy	Paul Bulcke
1202	Mars	M&M	WA	Candy	1.79	5	The best taste ever	Candy	Paul Bulcke
1202	Hershey	Kisses	PA	Chips	0.23	2	Poor quality chocolate	Candy	Paul Bulcke

a. (2 points) Identify dependencies and draw dependency diagram.
b. (2 points) Normalize our record table. Show resulting tables in 3NF.



c. (1 point) What's the purpose of DEnormalization? Give example when completely normalized table could be undesirable.

Answer: A process by which a table is changed from a higher-level normal form to a lower level normal form, usually to increase processing speed.

Q4. (3 points) TRANSACTION MANAGEMENT

a. (1 point) What are the names of the four ACID properties?

Answer*:* Atomicity, Consistency, Isolation and Durability.

b. (2 points) Given below is the transaction log.

TRL_ID	TRX_NUM	Table	ROW ID	Attribute	BEFORE VALUE	AFTER VALUE
341	101	****Start Transaction				
352	101	PRODUCT	1558-QW1	PROD_QOH	25	23
363	101	CUSTOMER	10001	CUST_BALANCE	525.75	615.73
365	101	****End Transaction				

Assuming the system crashed somewhere between TRL_ID 353 – 364 (i.e, the last two rows are not in the logs), what should be the values of the two fields that were updated in that transaction after recovery?

A (25; 525.75)

B (23; 525.75)

C (23; 615.73)

D (25; 525.75)

Briefly explain your answer.

Answer: D

The transaction hadn't committed before the database crashed. After recovery, none of its changes should be presented. Therefore, the answer is (25; 525.75)

Q5. (3 points) OPTIMIZATION

a. (1 point) Given below are two queries that perform the same function. Which one do you think would be more efficient, and why?

```
SELECT id, name
FROM viterbi_Students
WHERE branch = 'Computer Science' AND courseTaken = 'CS 585';

SELECT id, name
FROM viterbi_Students
WHERE courseTaken = 'CS 585' AND branch = 'Computer Science';
```

Answer: The second one is more optimized as in AND, we should write the condition that is more likely to be false first. Clearly, the number of students taking CS 585 would be much lesser than the number of students who are enrolled in Computer Science.

b. (2 points)
In a smartphone store, where roughly 20000 smartphones are added to the inventory every week, what recommendation would you give the designer about the use of derived attributes? Write an improved query based on your assumption.

```
SELECT model.modelCompany, AVG(model.modelPrice *
company.ratingCompany)
FROM model INNER JOIN company ON model.modelCompany = company.name
WHERE model.modelPrice > 400
GROUP BY model.modelCompany;
```

Answer:
Consider having an attribute in model which stores model.modelPrice * company.ratingCompany.

```
SELECT modelCompany, AVG(priceBase)
FROM model
WHERE modelPrice > 400
GROUP BY modelCompany;
```

Q6. (4 points) DISTRIBUTED DATABASES

Every year a large venture capital company needs to invest many projects. They use a table PROJECT to keep track of each project. However, at the end of each year, their financial department needs to check how much money they have invested. Recently, they are considering to change to a distributed database to store the data.

PROJECT

Project_id	PName	Budget	Location	Manager

a. (2 points) Which data fragmentation technique should they use to meet the requirement of the financial department? Show new design.

Answer: Vertical fragmentation.

Project_id	Budget

Project_id	PName	Location	Manager

b. (1 point) If budget department were to have multiple branches in different states to manage their local projects, which data fragmentation technique should they use?

Answer: Horizontal fragmentation. Each sub-table stores its state's project records (rows).

c. (1 point) Which protocol is used to control the distributed concurrency (by DDBMS)?

Answer: **DO-UNDO-REDO protocol**: Roll transactions back and forward with the help of the system's transaction log entries.

CSCI585 Fall '18 Midterm Exam

October 19th, 2018

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 1 hour. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0. This document contains 12 pages including this one.

Signature: _____

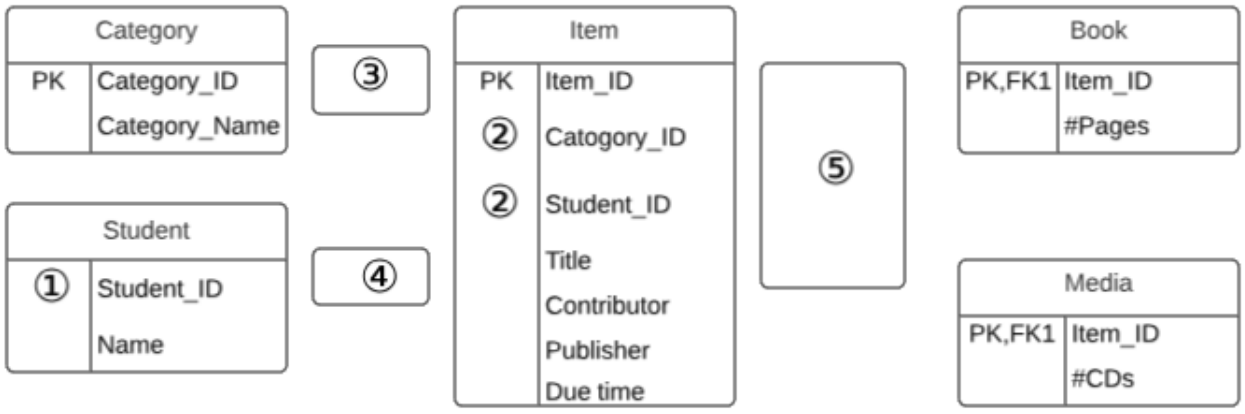
Problem Set	Number of Points
Q1	5
Q2	5
Q3	6
Q4	7
Q5	7
Q6	4
Q7	1
Total	**35**

Q1. (5 points total) ER MODELING

You are required to fill in the five blanks in the ER Diagram of a library database so it meets the following requirements. For blanks 1 and 2, please write the key type. For blanks 3, 4 and 5, please draw an edge to represent the relationship between its entities. Feel free to draw edges on the diagram, but please copy them on the blanks as well (to be graded).

The library has two types of items to check out, books and media. For each item, the database needs to record its unique Item_ID, title, contributor and publisher. Each item is also assigned one category like Science, Art, History and so on and each category is assigned to one or more items. Each item can be checked out by at most one student and the database should record who borrowed the item and due date for return. For a book, the database should record its number of pages. For a media item, the database should record number of CDs contained in it. All items should be in the database regardless whether they are available or have been already checked out.

Students can borrow zero, one or more items from the library. Each student has a unique Student_ID. The database should record all students' Student_IDs and names.

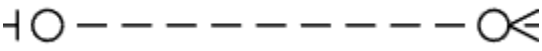


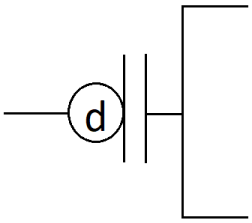
Solution

①PK

② FK1, FK2 (or just FK)

③ 

④ 



⑤

Q2. (5 points total) SQL

A. (2 points) Write a brief description of what the following query does. The semantics should be straightforward, but you can make any reasonable assumptions (ie: Viterbi is a school within USC, etc.)

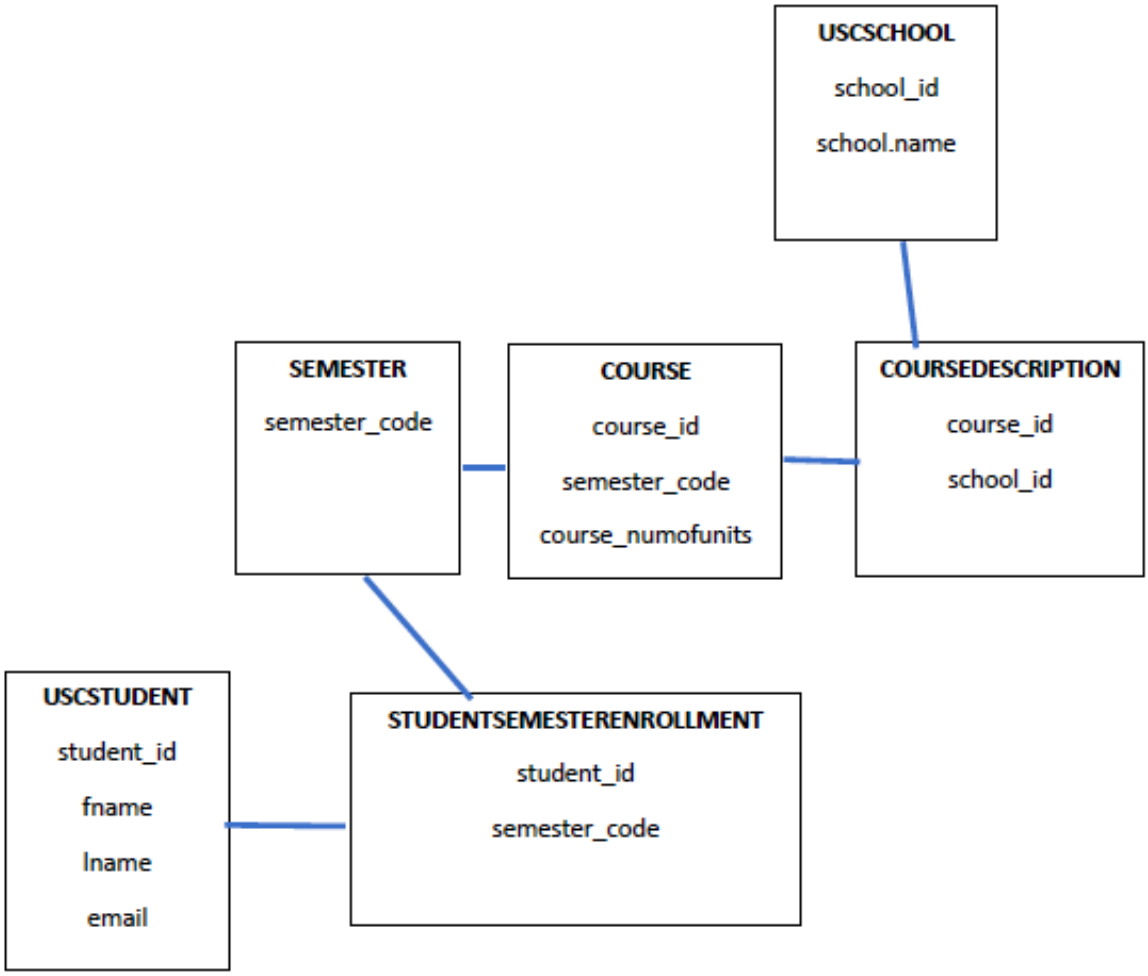
B. (3 points) Sketch the basic ER diagram/schema, show entities, attributes, and connections between them (relationships). Table names are: uscstudent, course, coursedescription, uscschool, semester, and studentsemesterenrollment.

```
SELECT stu.student_id, stu_fname, stu_lname, stu_email, totalunits
FROM   uscstudent stu
JOIN (
  SELECT uscstudent.student_id, Sum(course.course_numofunits) AS totalunits
  FROM (
    SELECT *
    FROM   studentsemesterenrollment sse
    JOIN uscstudent scs ON ( sse.student_id = scs.student_id )
    JOIN semester sem ON ( sse.semester_id = sem.course_id )
  ) sem
  JOIN course c ON sem.semester_code = c.semester_code
  JOIN coursedescription cd ON c.course_id = cd.course_id
  JOIN uscschool sch ON sch.school_id = cd.school_id
  WHERE uscschool.school_name = 'VITERBI'
  AND semester.semester_date BETWEEN '01-JAN-18' AND '31-DEC-18'
  GROUP BY uscstudent.student_id
) tommy ON stu.student_id = tommy.student_id
WHERE totalunits = (
  SELECT Max(totalunits)
  FROM (
    SELECT uscstudent.student_id,Sum(course.course_numofunits) AS totalunits
    FROM (
      SELECT *
      FROM   studentsemesterenrollment sse
      JOIN uscstudent scs ON ( sse.student_id = scs.student_id )
      JOIN semester sem ON ( sse.semester_id = sem.course_id )
    ) sem
    JOIN course c ON sem.semester_code = c.semester_code
    JOIN coursedescription cd ON c.course_id = cd.course_id
    JOIN uscschool sch ON sch.school_id = cd.school_id
    WHERE uscschool.school_name = 'VITERBI'
    AND semester.semester_date BETWEEN '01-JAN-18' AND '31-DEC-18'
    GROUP BY uscstudent.student_id
  )
);
```

Q2. Solution

This is a query to display the student id, student first name, student last name, e-mail, and total course units taken for the student who took the most Viterbi school classes between January 1, 2018, and December 31, 2018.

The following sub query:
FROM (SELECT * FROM studentsemesterenrollment sse JOIN uscstudent scs ON (sse.student_id = scs.student_id) JOIN semester sem ON (sse.semester_id = sem.course_id)) sem
Is a bridge table that links the uscstudent and semester tables with M:N relationship. The rest should be clear with the following diagram:



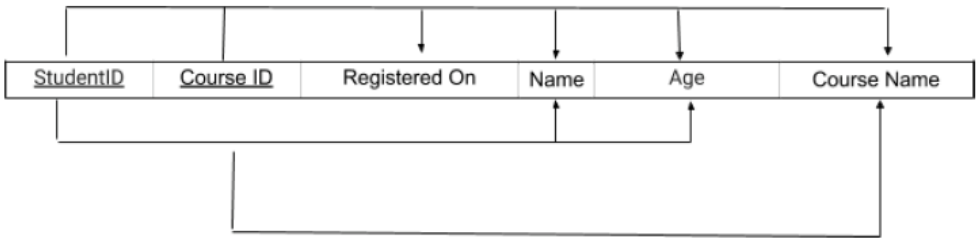
Q3. (6 points total) NORMALIZATION

Show dependency diagram and normalize the following table in 3 NF.

StudentID	Name	Age	Course ID	Course Name	Registered On
12	Alex	19	CSCI 511	C++	08/11/2018
			CSCI 510	Java	08/12/2018
123	Bin	20	CSCI 511	C++	08/05/2018
			CSCI 670	Algorithms	08/05/2018
32	Young	18	CSCI 550	Data Structures	08/15/2018
			CSCI 511	C++	08/11/2018
			CSCI 585	Database Systems	08/11/2018
133	Tracy	20	CSCI 520	Math	08/09/2018
			CSCI 510	Java	08/09/2018

Solution

Dependency Diagram



(StudentID, Course ID) → (Registered On, Name, Age, Course Name)
StudentID → (Name, Age)
CourseID → Course Name

3NF transformation

StudentID	Name	Age
12	Alex	19
123	Bin	20

32	Young	18
133	Tracy	20

Course ID	Course Name
CSCI 511	C++
CSCI 510	Java
CSCI 670	Algorithms
CSCI 585	Database Systems
CSCI 520	Math

StudentID	Course ID	Registered On
12	CSCI 511	08/11/2018
12	CSCI 510	08/12/2018
123	CSCI 511	08/05/2018
123	CSCI 670	08/05/2018
32	CSCI 550	08/15/2018
32	CSCI 511	08/11/2018
32	CSCI 585	08/11/2018
133	CSCI 520	08/09/2018
133	CSCI 510	08/09/2018

Q4. (7 points) TRANSACTION MANAGEMENT

You are given the example tables that represent information of a factory, a retailer, and a customer. Each table has information of products and their counts. Also provided is a transaction log (on the next page), which contains 2 transactions: one represents production of 100 products from factory to retailer, the other represents a purchase of 150 products by a customer.

(1) Consider the case that locking is not properly implemented in the DBMS. Discuss whether the results of the two transactions are deterministic. (No need to consider other external transaction, but failure or roll back can happen).

(2) Consider that the DBMS in use is implementing a locking mechanism. Is two-phase locking required to ensure correctness of the two transactions? State your reasons. (No need to consider other external transactions, but failure or roll back can happen).

(3) Consider that pessimistic locking is implemented with two-phase locking protocol. Create a chronological list of locking, unlocking, and data manipulation activity that would occur during the completion of the two given transactions. (No step fails and no rollback happens).

Example tables:

FACTORY

PRODUCT_ID	PRODUCT_COUNT
42	1000

RETAILER

PRODUCT_ID	PRODUCT_COUNT
42	58

CUSTOMER

CUSTOMER_ID	PRODUCT_ID	PRODUCT_COUNT
1007	42	3

Q4. (Continued)

Transaction log

TRL_ID	TRX_NUM	PREV PTR	NEXT PTR	OPERATION DESCRIPTION
214	101	Null		****Start Transaction
216	101	214	225	Update "RETAILER" table on the row with PRODUCT_ID = 42 and add 100 to PRODUCT_COUNT
225	101	216	233	Update "FACTORY" table on the row with PRODUCT_ID = 42 and subtract 100 from PRODUCT_COUNT
233	101	225	Null	****End of Transaction
220	105	Null		****Start Transaction
227	105	220	239	Check that PRODUCT_ID = 42 in RETAILER table has PRODUCT_COUNT > 150 and wait until the condition is met.
239	105	227	243	Update "RETAILER" table on the row with PRODUCT_ID = 42 and subtract 150 to PRODUCT_COUNT
243	105	239	252	Update "CUSTOMER" table on the row with PRODUCT_ID = 42 and CUSTOMER_ID = 1007 and then add 100 to PRODUCT_COUNT
252	105	243	Null	****End of Transaction

Solution

(1) The result will be non-deterministic if no locking is implemented. Even if transaction 105 checks that PRODUCT_ID 42 should have at least 150 items before proceeding which seems to suggest that transaction 105 will not proceed before transaction 101 is done, it is still possible that one of the transaction is aborted that may cause inconsistencies, for example, consider the following events:

* TRX 101 starts
* TRX 101 updates RETAILER table, now RETAILER.PRODUCT_COUNT = 158 for RETAILER.PRODUCT_ID = 42
* TRX 105 starts
* TRX 105 checks RETAILER table, find the RETAILER.PRODUCT_COUNT > 150 for RETAILER.PRODUCT_ID = 42, and proceed to the next step
* TRX 105 updates RETAILER table by subtracting 150 for RETAILER.PRODUCT_ID = 42, now RETAILER.PRODUCT_COUNT = 8
* TRX 101 failed to update FACTORY table in its next step, and the whole TRX 101 is reverted, now RETAILER.PRODUCT_COUNT = 58 again.
* TRX 105 updates CUSTOMER table, now that CUSTOMER.PRODUCT_COUNT = 153 for

CUSTOMER.PRODUCT_ID = 42 and CUSTOMER.CUSTOMER_ID = 1007

The result of this example shows the customer successfully bought 150 items while the other tables are not updated properly. Hence, a proper locking mechanism is required.

(2) Two-phase locking protocol is required, because in TRX 101, updating of RETAILER table happens before updating FACTORY table, which has the possibility that the latter may fail and rollback the transaction (like the example given in the above answer). Without two-phase locking protocol, only locking one table may not ensure correctness once errors occur.

(3) Example of chronological events

Time	TRX_NUM	Event
1	101	Lock table RETAILER
2	101	Lock table FACTORY
3	101	Update table RETAILER by adding 100 to PRODUCT_COUNT of PRODUCT_ID = 42
4	101	Update table FACTORY by subtracting 100 to PRODUCT_COUNT of PRODUCT_ID = 42
5	101	Unlock table FACTORY
6	101	Unlock table RETAILER
7	105	Lock table RETAILER
8	105	Lock table CUSTOMER
9	105	Check table RETAILER of PRODUCT_ID = 42 that PRODUCT_COUNT > 150
10	105	Update table RETAILER by subtracting 150 to PRODUCT_COUNT of PRODUCT_ID = 42
11	105	Update table CUSTOMER by adding 150 to PRODUCT_COUNT with PRODUCT_ID = 42 and CUSTOMER_ID = 1007
12	105	Unlock table CUSTOMER
13	105	Unlock table RETAILER

Q5. (7 points) OPTIMIZATION

Consider the three following tables for an airport database and all attributes are neither indexed nor sorted.

- AIRPLANES (aid, brand, size), aid is the primary key.
- PILOTS (pid, name, age), pid is the primary key.
- LastFlight (aid, pid, date), aid and pid are a composite primary key.

And we want to execute the following SQL query:

```
SELECT P.name
FROM AIRPLANES A, PILOTS P, LastFlight L
WHERE A.aid = L.aid AND P.pid = L.pid
AND P.age < 35 AND A.brand = 'Boeing 737';
```

Assuming:

- There are 1,000 rows in AIRPLANES, 1,000 rows in PILOTS and 1,000,000 rows in LastFlight.
- PILOTS.age ranges from [30 to 49] (both inclusive) equally distributed in PILOTS.
- AIRPLANES.brand has 100 distinct values equally distributed in AIRPLANES.
- LastFlight has every combination of aid and pid.

Suppose the cost of running a SELECT operation is the number of rows in the source table and the cost of running a JOIN operation (Cartesian product) is the total rows of the two source tables. If we execute the query with following access plan, the cost will be 1,001,001,002,000.

Step	Operation	Cost	Estimated result rows
A1	Cartesian product (A, L)	1,001,000	1,000,000,000
A2	Cartesian product (A1, P)	1,000,001,000	1,000,000,000,000
A3	Select rows in A2 with all conditions	1,000,000,000,000	2,500*

* Here is how the number of resulting rows were estimated:
- The possibility of A.aid = L.aid is 1/1,000 for there are 1,000 different aid.
- The possibility of P.pid = L.pid is 1/1000 for there are 1,000 different pid.
- The possibility of an airplane brand = 'Boeing 737' is 1/100 for there are 100 different brands.
- The possibility of P.age < 35 is 5/20.
- Since all conditions are independent, the number of resulting rows in A3 is about:

1,000,000,000,000 * (1/1,000) * (1/1,000) * (1/100) * (5/20) = 2,500.

Do you have a better access plan to execute the query with a lower total cost?
Please fill the following form about your access plan.

- You don't have to fill all rows depending on how many steps are in your access plan.
- There should be enough room in each cell for you to answer and make corrections.

- Q5. Solution

Best answer:

Step	Operation	Cost	Estimated result rows
B1	Select rows in P with ages < 35	1,000	250
B2	Select rows in A with brand = 'Boeing 737'	1,000	10
B3	Cartesian product (L, B2)	1,000,010	10,000,000
B4	select rows in B3 with A.aid = L.aid	10,000,000	10,000
B5	Cartesian product (B1, B4)	10,250	2,500,000
B6	select rows in B5 with P.pid = L.pid	2,500,000	2,500

Total cost: 13,512,260 (not required to answer)

-1 Point

Step	Operation	Cost	Estimated result rows
B1	Select rows in P with ages < 35	1,000	250
B2	Select rows in A with brand = 'Boeing 737'	1,000	10
B3	Cartesian product (L, B1)	1,000,250	250,000,000
B4	select rows in B3 with P.pid = L.pid	250,000,000	250,000
B5	Cartesian product (B2, B4)	250,010	2,500,000
B6	select rows in B5 with A.aid = L.aid	2,500,000	2,500

Total cost: 253,752,260 (not required to answer)

-2 Points

Step	Operation	Cost	Estimated result rows
B1	Select rows in P with ages < 35	1,000	250
B2	Select rows in A with brand = 'Boeing 737'	1,000	10
B3	Cartesian product (L, B2)	1,000,010	10,000,000
B4	Cartesian product (B1, B3)	10,000,250	2,500,000,000
B5	select rows in B4 with P.pid = L.pid and A.aid = L.aid	2,500,000,000	2,500

Total cost: 2,511,002,260 (not required to answer)

Q6 (4 points) DISTRIBUTED DATABASES

List and explain characteristics of distributed databases (provide clear explanation and/or examples).

Solution

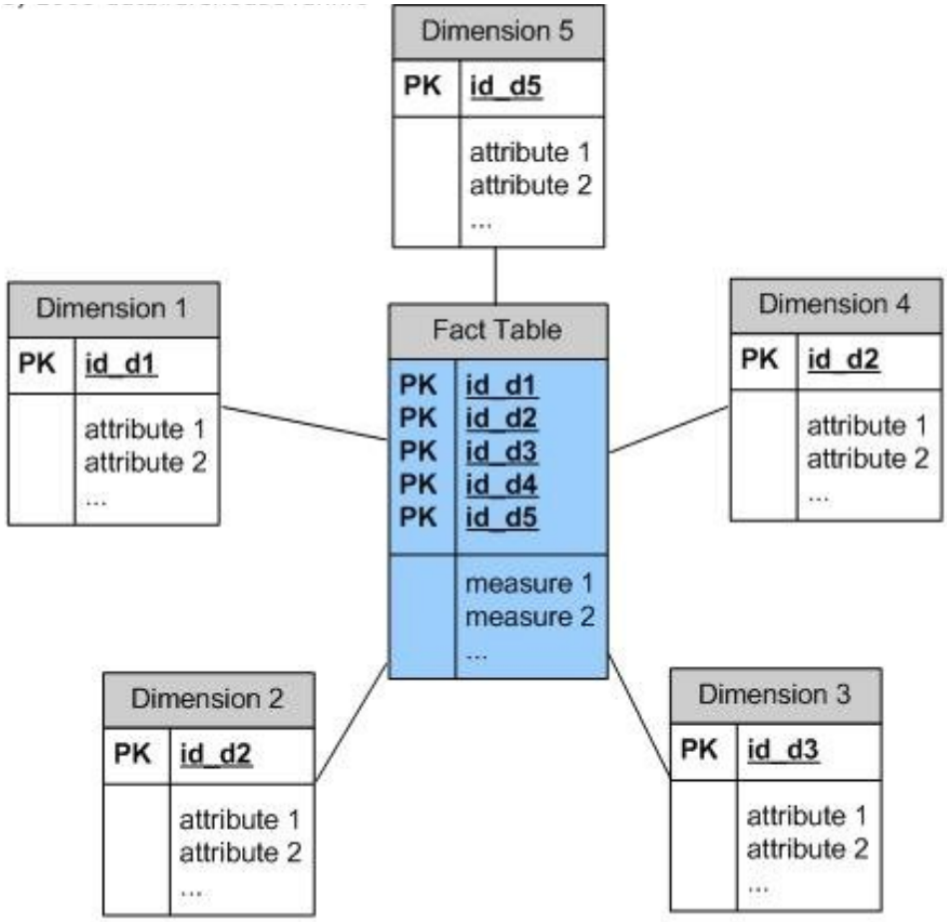
This question was designed to test student's understanding of distributed database systems.

One potential answer is to list and explain several DDBMS functions. For sample answer, please refer to chapter 12-4 on page 559 of class textbook.

The alternate answer was to list and explain the distributed database transparency features: distribution, transaction, failure, performance, and heterogeneity transparencies. For sample answers, please refer to chapter 12-7 on page 564 of class textbook.

Q7. (1 point) BUSINESS INTELLIGENCE

What kind of schema does the ER diagram demonstrate?



Solution

Star schema

BONUS!!! (1 point)

What was your favorite part of Science documentary shown in class? If you have seen the entire movie, feel free to reference the part not displayed in class.

Solution
Your mileage may vary ☺

CSCI585 Spring '18 Midterm Exam & Solutions

March 9th, 2018

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 1 hour. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0.

Signature: _____

Problem Set	Number of Points
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	5
Total	**35**

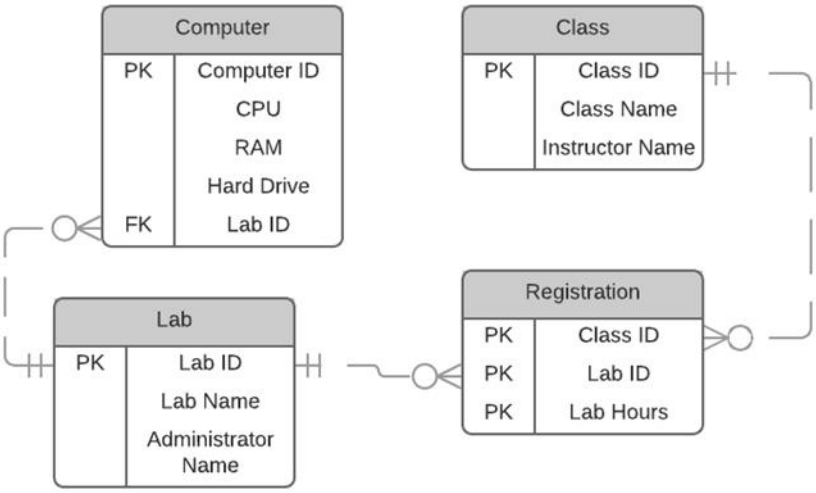
Q1. (5 points total) ER MODELING

Design ERD using Crow's foot notation for the following problem:

Computer Science department needs to design a database to manage computer labs using the following information:

- Each lab has one unique identifier, name, administrator name, and many computers.
- Each computer has a unique identifier, configuration information (CPU, RAM, hard drive) and location (in one of the labs).
- Each class has a unique identifier, class name, and instructor's name.
- Each class can have lab hours in multiple labs and one lab can be registered for multiple classes. A timestamp is stored to indicate a class is registered for a lab session.

Answer:



Q2. (5 points total) SQL
After the Oscars award ceremony last Sunday, you have been contacted by the organizers to write some queries. Their database consists of the following tables:
MEMBERS (MEMBER_ID, NAME).
MOVIES (MOVIE_ID, RELEASE_YEAR, TITLE, DIRECTOR).
REVIEWS (REVIEW_ID, *MEMBER_ID*, *MOVIE_ID*, TEXT, REVIEW_DATE, RATE).
ACTORS (NAME, *MOVIE_ID*).
Primary keys of every table are underlined while foreign keys are italic. The RELEASE_YEAR attribute of a movie is a number, such as 2018.

A (2 points) Display unique member IDs of all the members who reviewed at least one of the
movies reviewed by user with member ID "M1". The list of member IDs must exclude "M1".
Answer:

```
select distinct r1.MEMBER_ID
from REVIEWS r1
where r1. MEMBER_ID  != 'M1' and r1. MOVIE_ID in (
select r2. MOVIE_ID
from REVIEWS r2
where r2. MEMBER_ID = 'M1');
```

B (1 point) Delete all reviews that have the term "horrible" in their text. If the text contains "XhorribleX" where X refers to any character(s), its review must be deleted as well.
Answer:

```
delete from REVIEWS where TEXT like '%horrible%';
```

C (2 points) Display the actors' names and average rating for the movies with the highest average rating.
Answer:

```
select NAME, avg(RATE) from ACTORS a, REVIEWS r where r. MOVIE_ID = a.
MOVIE_ID
group by NAME
having avg(RATE) = (select max(avg(RATE)) from REVIEWS group by MOVIE_ID);
```

Q3. (5 points total) NORMALIZATION

Convert the following table into:

a. The 1NF. (1 point)
b. The 2NF. (2 points)
c. The 3NF. (2 points)

Show the dependency diagram for each form and identify the primary key for each table.

Parent_ID	Parent_Name	Home_Address	Children_Names	Enrollment	Start_Hour	End_Hour	Daycare_ID	Daycare_Location
1	Alice	627 Green St., LA	Mike, Sara	Full	7am	5pm	324	1214 Hover St., LA
2	Brad	93 27th St., LA	Liam	Morning	7am	12pm	324	1214 Hover St., LA
2	Brad	93 27th St., LA	Nina	Full	7am	5pm	324	1214 Hover St., LA
3	Claire	45 Pico Blvd., LA	Luke	Full	7am	5pm	324	1214 Hover St., LA
4	Tom	1308 55th Pl., SD	Sara	Afternoon	1pm	5pm	564	453 5th Ave., SD
5	Alice	433 Maple St., SD	Tony, Yara	Full	7am	5pm	564	453 5th Ave., SD

a. 1NF:

Dependency diagram:

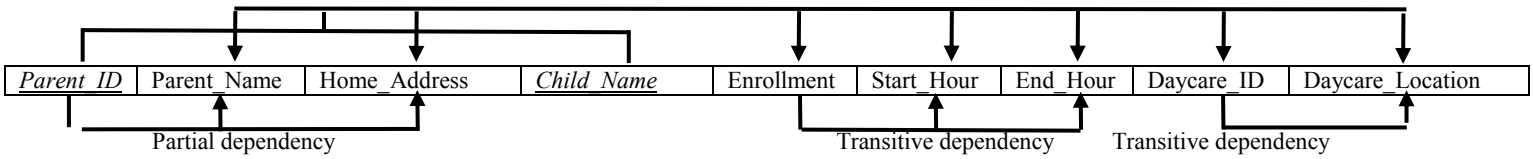
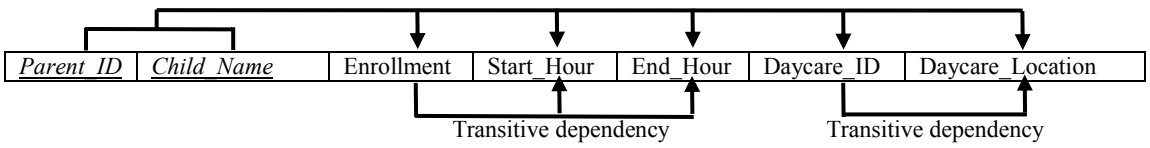


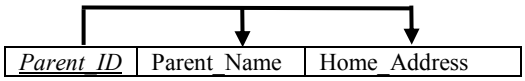
Table in 1NF:

Parent_ID	Parent_Name	Home_Address	Child_Name	Enrollment	Start_Hour	End_Hour	Daycare_ID	Daycare_Location
1	Alice	627 Green St., LA	Mike	Full	7am	5pm	324	1214 Hover St., LA
1	Alice	627 Green St., LA	Sara	Full	7am	5pm	324	1214 Hover St., LA
2	Brad	93 27th St., LA	Liam	Morning	7am	12pm	324	1214 Hover St., LA
2	Brad	93 27th St., LA	Nina	Full	7am	5pm	324	1214 Hover St., LA
3	Claire	45 Pico Blvd., LA	Luke	Full	7am	5pm	324	1214 Hover St., LA
4	Tom	1308 55th Pl., SD	Sara	Afternoon	1pm	5pm	564	453 5th Ave., SD
5	Alice	433 Maple St., SD	Tony	Full	7am	5pm	564	453 5th Ave., SD
5	Alice	433 Maple St., SD	Yara	Full	7am	5pm	564	453 5th Ave., SD

b. 2NF:

Dependency diagrams:





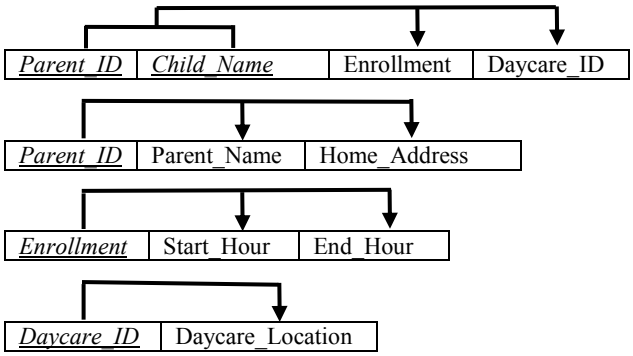
Tables in 2NF:

Parent_ID	*Child_Name*	Enrollment	Start_Hour	End_Hour	Daycare_ID	Daycare_Location
1	Mike	Full	7am	5pm	324	1214 Hover St., LA
1	Sara	Full	7am	5pm	324	1214 Hover St., LA
2	Liam	Morning	7am	12pm	324	1214 Hover St., LA
2	Nina	Full	7am	5pm	324	1214 Hover St., LA
3	Luke	Full	7am	5pm	324	1214 Hover St., LA
4	Sara	Afternoon	1pm	5pm	564	453 5th Ave., SD
5	Tony	Full	7am	5pm	564	453 5th Ave., SD
5	Yara	Full	7am	5pm	564	453 5th Ave., SD

Parent_ID	Parent_Name	Home_Address
1	Alice	627 Green St., LA
2	Brad	93 27th St., LA
3	Claire	45 Pico Blvd., LA
4	Tom	1308 55th Pl., SD
5	Alice	433 Maple St., SD

c. 3NF:

Dependency diagrams:



Tables in 3NF:

Parent_ID	*Child_Name*	Enrollment	Daycare_ID
1	Mike	Full	324
1	Sara	Full	324
2	Liam	Morning	324
2	Nina	Full	324
3	Luke	Full	324
4	Sara	Afternoon	564
5	Tony	Full	564
5	Yara	Full	564

Parent_ID	Parent_Name	Home_Address
1	Alice	627 Green St., LA
2	Brad	93 27th St., LA
3	Claire	45 Pico Blvd., LA
4	Tom	1308 55th Pl., SD
5	Alice	433 Maple St., SD

Enrollment	Start_Hour	End_Hour
Full	7am	5pm
Morning	7am	12pm
Afternoon	1pm	5pm

Daycare_ID	Daycare_Location
324	1214 Hover St., LA
564	453 5th Ave., SD

Q4. (5 points) TRANSACTION MANAGEMENT

A. (3 points) What does ACID in ACID properties stand for? Give an example of a scenario where atomicity is violated.

Answer:

ACID stands for Atomicity, Consistency, Isolation and Durability.
A transaction is atomic if either all or none is executed. Users cannot observe a state that is mid-fly.

An example of violating atomicity: Assume Alice's initial bank account balance is $100, while Bob's is $50. There are two transactions:

T1- Alice transfers $20 to Bob, which is executed in two steps:
+ Subtract $20 from Alice's balance. Alice's new balance becomes $80.
+ Add $20 to Bob's balance. Bob's new balance becomes $70.
T2- Administrator queries for the sum of Alice and Bob's balance.

With atomicity, T2 should always observe value $150. If T2 at some point observes the mid-fly state of executing transaction T1 (i.e., between step 1 and step 2) which results in the sum of Alice and Bob's balance is $130, then atomicity is violated.

B. (2 points) What is two-phase locking (2PL)? Give an example to illustrate how deadlock may happen with two phase locking.

Answer:

Two-phase locking is a locking mechanism used in database systems, which consists of two phases:
1: Growing Phase (Acquire locks)
2: Shrinking Phase (Release locks)
A scenario where dead-lock may happen with two-phase locking:
Consider two transactions:
T1- Update X=X+1, Y=5
T2- Update Y=2*Y, X=7
The execution flow below causes dead-lock. T1 waits for T2 to release lock on Y, while T2 waits for T1 to release lock on X.

T1	T2
Lock(X)	
Lock(Y)	
X = X+1	Y = 2*Y
Lock(Y)	
Lock(X)	

Q5. (5 points) QUERY OPTIMIZATION

Consider the three following tables for an online-sale database and all attributes are neither indexed nor sorted.

1. CUSTOMER (cid, name, age), cid is the primary key.
2. PRODUCT(pid, seller), pid is the primary key.
3. TRANSACTION(tid, cid, pid), tid is the primary key.

And we want to execute the following SQL query:

```
SELECT T.tid, C.name
FROM TRANSACTION T, CUSTOMER C, PRODUCT P
WHERE C.cid = T.cid
AND P.pid = T.pid
AND seller = 'Olivera'
AND C.age >= 25
AND C.age <= 34
```

Assuming:

- There are 100 rows in CUSTOMER, 5,000 rows in PRODUCT and 10,000 rows in TRANSACTION.
- There are 100 different sellers equally distributed in PRODUCT.
- Customers's ages range from 20 to 44 (both inclusive) equally distributed in CUSTOMER.
- cid and pid are independently equally distributed in TRANSACTION.

Now our task is to optimize the query with a Cost-based optimizer. **Suppose the cost of running a SELECT operation is the number of rows in the source table** and **the cost of running a JOIN operation is the total rows of the two source tables**. If we execute the query with following access plan, the cost will be 5,050,015,100.

STEP	OPERATION	COST	ESTIMATED RESULT ROWS
A1	Join T and C	15,000	50 milliion
A2	Join A1 and P	50,000,100	5 billion
A3	Select rows in A2 with all conditions	5 billion	40 (Explained below)

The possibility of C.cid = T.cid is 1/100 for there are 100 different cid. The possibility of P.pid = T.pid is 1/5000 for there are 5000 different pid. The possiblity of seller = 'Olivera' is 1/100 for there are 100 different sellers. The posibility of C.age >=25 and C.age <= 34 is 10/25. Since all conditions are independent, the number of result rows in A3 is about 5 billion/100/5000/100*(10/25)=40.

T, C and P are abbreviations for TRANSACTION, CUSTOMER and PRODUCT, respectively.

Do you have a better access plan to execute the query with a lower total cost? Please fill the following form (on the next page!) about your access plan with STEP 1 given.

- You don't have to fill all rows depending on how many steps in your access plan.
- Try not to ruin this form. There should be enough room in each cell for you to answer and make corrections.

  Answers on the following page.

Best answer:

STEP	OPERATION	COST	RESULT ROWS
B1	Select rows in C with ages between 25 and 34	100	40
B2	Select rows in P with seller = 'Olivera'	5,000	50
B3	JOIN B2 and T	10,050	500,000
B4	select rows in B3 with P.pid = T.pid	500,000	100
B5	Join B1 and B4	140	4,000
B6	Select rows in B5 with C.cid = T.cid	4,000	40

Total cost: 519,290 (not required to answer)

STEP	OPERATION	COST	RESULT ROWS
B1	Select rows in C with ages between 25 and 34	100	40
B2	Select rows in P with seller = 'Olivera'	5,000	50
B3	JOIN B1 and T	10,040	400,000
B4	select rows in B3 with C.cid = T.cid	400,000	4,000
B5	Join B2 and B4	4,050	200,000
B6	Select rows in B5 with P.pid = T.pid	200,000	40

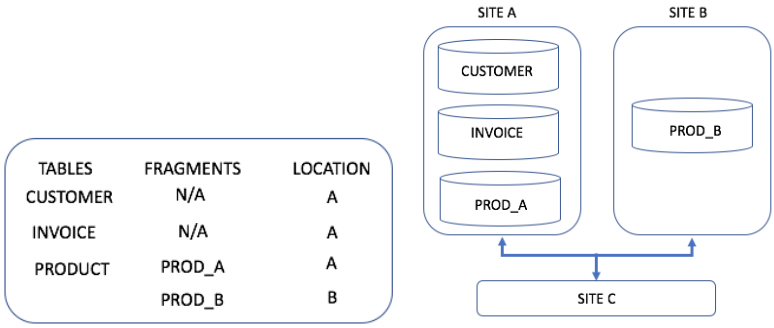
Total cost: 619,190 (not required to answer)

Partial correct answers

STEP	OPERATION	COST	RESULT ROWS
B1	Select rows in C with ages between 25 and 34	100	40
B2	Select rows in P with seller = 'Olivera'	5,000	50
B3	JOIN B1 and B2	90	2,000
B4	JOIN B3 and T	12,000	20,000,000
B5	select rows in B4 with C.cid=T.cid AND P.pid = T.pid	20,000,000	40

Total cost: 20,017,190 (not required to answer)

Q6. (5 points) DISTRIBUTED DATABASES



TABLES	FRAGMENTS	LOCATION
CUSTOMER	N/A	A
INVOICE	N/A	A
PRODUCT	PROD_A	A
	PROD_B	B

For the DDBMS above, specify the type of operation the database must support (remote request, remote transaction, distributed transaction or distributed request) to perform each of the following operations at SITE C:

a. SELECT *
FROM PRODUCT
WHERE PROD_QOH > 20;

Answer: Distributed request

b. SELECT CUS_NAME, INV_TOTAL
FROM CUSTOMER, INVOICE
WHERE CUSTOMER.CUS_NUM = INVOICE.CUS_NUM;

Answer: Remote request

c. BEGIN WORK;
UPDATE PRODUCT
SET PROD_QOH = PROD_QOH + 5
WHERE PROD_NUM = '123';
INSERT INTO CUSTOMER(CUS_NUM, CUS_NAME, CUS_STATE)
VALUES('111', 'Tommy Trojan', 'CA');
COMMIT WORK;

Answer: Distributed transaction

Q7. (5 points) DB SECURITY, WEB TECHNOLOGIES, BUSINESS INTELLIGENCE

A (1 point) Contrasting between activities of a "database administrator" (DBA) and a "data administrator" (DA), who sets policies and standards?

Answer: Data administrator (DA)

B Which Web technology has a class named DataSet?

Answer: ADO.NET

C (1 point) Name the components of Star schema.

Answer: 1. Facts , 2.Dimensions , 3.Attributes , 4. Attribute hierarchies

D. (1 point) Is snowflake schema normalized or denormalized?

Answer: Normalized

E. (1 point) Name the two extensions SQL offers for OLAP.

Answer: 1. ROLLUP , 2.CUBE

CSCI585 Summer '18 Midterm Exam

June 11th, 2018

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 2 hours. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0.

Solutions are displayed in red font!

Signature: _____

Problem Set	Number of Points
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	5
Total	**35**

Q1. (5 points total) ER MODELING
A. (2 points) Explain the difference between weak and strong entity and provide example.

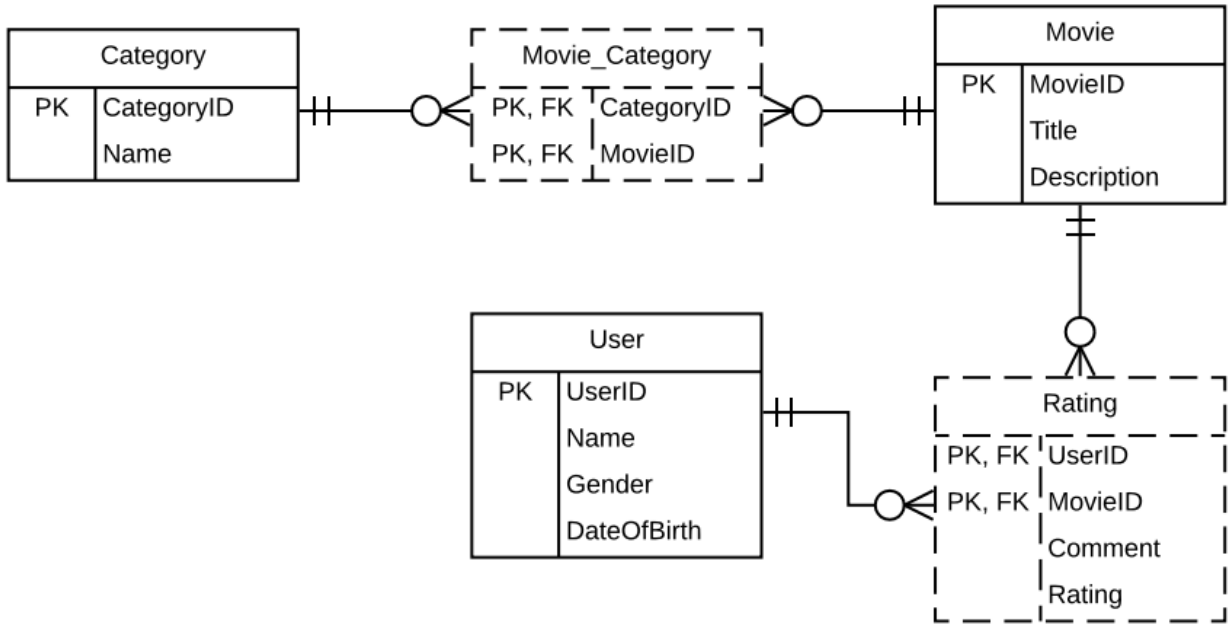
Weak entity: Entity that depends on another entity to exist. Its primary key contains the primary key of another entity.
Strong entity: Entity that **may** exist independently with other entities.

Example:

Department(DeptID, Name, Address, Phone)
Employee(DeptID,EmplID, Name, Age, Gender)

Department is a strong entity, while Employee is a weak entity.

B. (3 points) Design ERD using Crow's foot notation for the following description:
A movie-rating site like the Internet Movie Database (IMDB) has multiple movie categories. Each category has zero, one or more movies. One movie may belong to multiple categories. A person must create an account to rate a movie. To rate a movie, he/she provides a comment along with a rating star ranging from 1-5. A person can rate any numbers of movies he/she likes, but cannot have multiple ratings for the same movie (hence, he/she may provide either no rating or one rating per movie).



Q2. (5 points total) SQL

Write the following queries for an online store that sells books. Below are the tables for the store, primary keys are underlined, foreign keys are *italic*. AvailableCount attribute in Book table records the number of copies of a book that are available for sale. Quantity attribute in Order table records the number of copies of a book that is ordered by a customer. For simplicity, we assume a customer can only order a book in one order (he can order more copies of it, however).

Category(CategoryID, Name)
Book(ISBN, *CategoryID*, Title, Author, Description, PublishDate, AvailableCount, Price)
Customer(CustomerID, Name, Age, Gender, Balance)
Order(OrderID, *CustomerID*, *ISBN*, Quantity, Total, OrderDate)

A. (1 point) List up to 10 books in categories “Science fiction” or “Romance”. For each book, show all its attributes.

SELECT ISBN, BOOK.CAT_ID, TITLE, AUTHOR, DESCRIPTION, PUBLISH_DATE, AVAILABLE_COUNT, PRICE FROM CATEGORY, BOOK WHERE BOOK.CAT_ID = CATEGORY.CAT_ID AND CATEGORY.NAME IN ('SCIENCE FICTION','ROMANCE') LIMIT 10;

B. (2 points) List all books customer id 5 ordered. For each book, show the ISBN, title, the number of copies he/she bought and the total amount he/she spent on that book.

SELECT BOOK.ISBN, TITLE, SUM(QUANTITY), SUM(TOTAL) FROM BOOK, BOOK_ORDER WHERE BOOK.ISBN = BOOK_ORDER.ISBN AND CUSTOMER_ID = 5 GROUP BY ISBN;

C. (2 points) List up to five customers that ordered “Horror” books the most (order counts).

SELECT CUSTOMER_ID FROM CATEGORY, BOOK, BOOK_ORDER WHERE BOOK.CAT_ID = CATEGORY.CAT_ID AND CATEGORY.NAME = 'HORROR' GROUP BY CUSTOMER_ID ORDER BY COUNT(ORDER_ID) DESC LIMIT 5;

Q3. (5 points total) NORMALIZATION

A. (1 point) Explain the difference between 2NF and 3NF by providing example of 2NF but not 3NF.

While 2NF is converted from 1NF by removing the partial dependencies, 3NF is converted from 2NF by removing the transitive dependencies.
Example:
(Department, Employee, JobType, Salary)
There is no partial dependency, so the table is 2NF.
However, there is transitive dependency: (JobType → Salary), to the table is not 3NF.

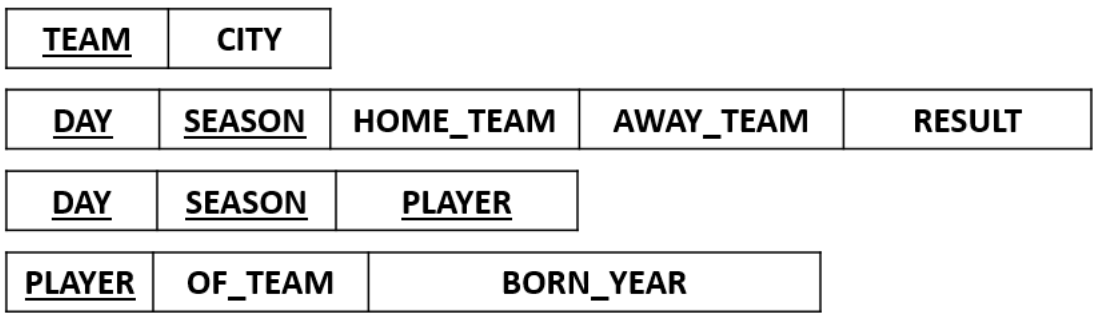
B. (4 points) Convert the following table to 3NF.
Show dependency diagram for each form and identify the primary key for each table.

Season	Day	Home Team	Home Team City	Away Team	Away Team City	Result	Scored By	Of Team	Born Year
2016	June 7	MU	Manchester	Chelsea	London	2-1	A. Herrera	Manchester	1989
2016	June 7	MU	Manchester	Chelsea	London	2-1	Rashford	Manchester	1997
2016	June 7	MU	Manchester	Chelsea	London	2-1	Hazard	Chelsea	1991
2016	Oct 8	Arsenal	London	Liverpool	Liverpool	1-0	O. Giroud	Arsenal	1986
2015	Apr 15	Everton	Liverpool	MU	Manchester	1-1	Pogba	Manchester	1993
2015	Apr 15	Everton	Liverpool	MU	Manchester	1-1	W. Rooney	Everton	1985

1NF(Season, Day, HomeTeam, HomeTeamCity, AwayTeam, AwayTeamCity, Result, ScoredBy, OfTeam, BornYear)

Partial Dependencies: (Season, Day → HomeTeam, AwayTeam, Result)
Transitive Dependencies:
(HomeTeam → HomeTeamCity)
(AwayTeam → AwayTeamCity)
(ScoredBy → OfTeam, BornYear)

3NF:



Q4. (5 points) TRANSACTION MANAGEMENT

A. (4 points) Assume PRODUCT table has a record for a notebook, whose product ID is 996 and its quantity on hand is 10. A developer executes a SQL statement "UPDATE product SET quantity = 6 WHERE id = 996" through the program. However, after the operation, the developer runs SELECT query and finds out that this notebook's quantity is still set to 10. The developer makes sure the update statement was executed and he retrieves the data from the table directly. What could be two possible reasons for this situation?

1. That statement is rollbacked after the execution
2. Lost updates: some other transaction may update the notebook's quantity at the same time.

B. (1 point) Locking is widely used in concurrency control in databases, but we need to make sure there are no deadlocks between transactions. Briefly explain (or give example) how deadlock occurs in transaction management.

A deadlock occurs when two transactions wait indefinitely for each other to unlock data.

Q5. (5 points) QUERY OPTIMIZATION
A. (2 points) The following two queries are performing the same function. Which one do you think is more efficient? Why? (2 points)

SELECT id FROM users WHERE DATEDIFF(MONTH, registerDate, '2015-04-28') < 0;

SELECT id FROM users WHERE registerDate > '2015-04-28';

The second one is more efficient. The first one uses a runtime function, and the database has to visit all rows to retrieve the required data.

B. (2 points) Consider the query discussed during class and pertaining to the schema discussed in class. List at least two ways to optimize this query.

```
SELECT    CUS_CODE, MAX(LINE_UNITS*LINE_PRICE)
FROM      CUSTOMER NATURAL JOIN INVOICE NATURAL JOIN LINE
WHERE     CUS_AREACODE = '615'
GROUP BY  CUS_CODE;
```

1. Filter by area code 615 before joining with customer
2. Store line_units*line_price in a column called line_total instead of deriving it.

C. (1 point) In order to retrieve the orders that are placed by the residents of cities whose name starts with "Cha", the developer writes a query:
SELECT * FROM orders WHERE city LIKE '%Cha%'.
Is there a problem with this query? If yes, write down your optimized version.

Yes, the query will also pull unexpected results, cities that contain "Cha" but not starts with "Cha".

The optimized query is SELECT * FROM orders WHERE city LIKE 'Cha%'

Q6. (5 points) DISTRIBUTED DATABASES

A nationwide commercial bank has many branches in each state. At the end of each month, the bank's audit department wants to know the deposits and loans for each state. All the data is stored in a distributed database.
A. (2 points) In order to meet the requirement of the audit department, which data fragmentation technique should be used, and how to do that?

Horizontal fragmentation. Each sub-table stores its state's records (rows).

B.(2 points) For each branch, the headquarters will establish a monthly goal according to the operational data (ie. deposits and loans). Assume that there is a table Branch which contains the following attributes:
branchId, address, manager, deposit_amount, loan_amount, audit_Time

If we only consider meeting the requirements of the headquarters, which data fragmentation technique should be used, and how?

Vertical fragmentation.

One fragmented table contains branchId, deposit_amount, loan_amount, audit_Time.

BranchId, address, manager are stored in another fragmented table.

C. (1 point) What are the two phases in two-phase commit protocol (2PC)?

1. Preparation
2. The final COMMIT.

Q7. (5 points) INTRODUCTION AND DATA MODELING

A (1 point) Explain how division (one of relational algebra operations) works. Feel free to use example.

It answers queries about one set of data being associated with all values of data in another set (looks for commonality).

B (1 point) What does DDL stand for?

Data Definition Language. The language that allows a database administrator to define the database structure, schema, and subschema.

C (1 point) Provide an example of discipline specific database.

Any database that contains data focused on specific subject areas (answers will vary).

D. (1 point) In which database (operational database or data warehouse) data doesn't get modified?

Data warehouse (analytical database).

E. (1 point) Explain one of the many functions that DBMS must support.

Answers will vary.

BONUS!!! (1 point) What was your favorite part of Science documentary shown in class? If you have seen the entire movie, feel free to reference the part not displayed in class.
Your mileage may vary. ☺

CS585 Midterm

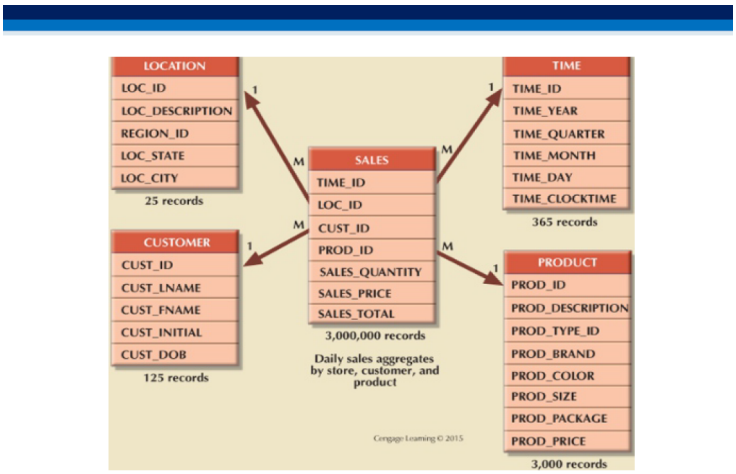
Spring term, 10/11/19
Duration: 1 hour

Instructions/notes

- the exam is closed books/notes/devices/neighbors, and open mind :)
- there are 6 questions, a 'non-data-related' bonus, for a total of **35** points
- there are no 'trick' questions, or ones with long calculations or formulae
- you can write on the two blank sheets (that are at the end) if you like
- **please DO NOT CHEAT; you will get a 0 if you are found to have cheated**
- **when time is up, please STOP WRITING**; **you will get a 0 if you continue**

Q	Your score	Max possible score
1		6
2		6
3		6
4		5
5		6
6		6
Bonus		1
Total		**35 (NOT 36)**

Q1 (4+2=6 points).
In a snowflake schema, the fact table (at the center) is related to several dimension tables (on the periphery, ie. 'outside'):

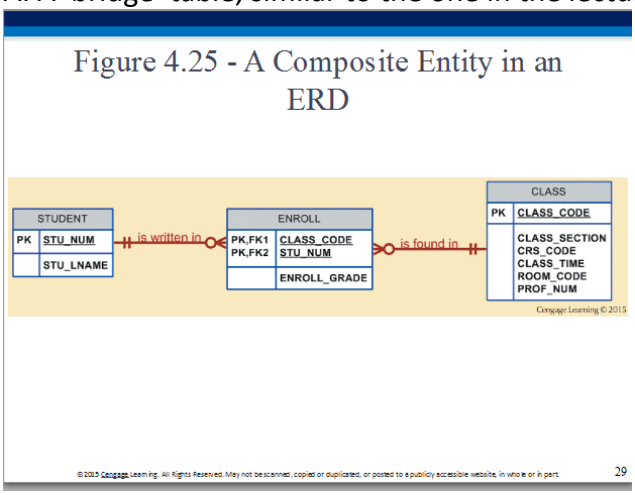


Often, a dimension table contains columns that form a hierarchy (eg. TIME and LOCATION, above) - **what is the purpose of this?**

A. To be able to drill down or roll up (zoom out) along that dimension.

Also, you have come across the equivalent of a fact table before - what is it? Explain briefly, using a diagram.

A. A 'bridge' table, similar to the one in the lecture:

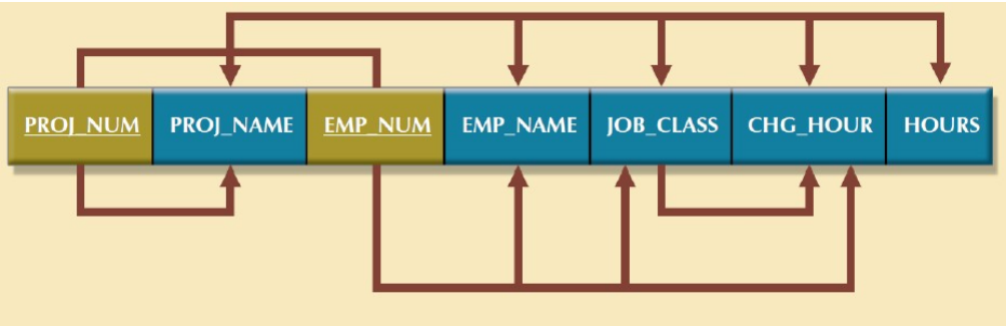


Q2 (4*1.5=6 points). For your HW1, you were asked to design an E-R diagram for a 'STEM' organization. The next step would be, to use the diagram to create tables, connect them appropriately, and deploy the resulting database. **What are important principles and practices that would result in a 'good' relational database?** Think 'across' all the relevant material you learned. Describe each item (principle or practice) using a sentence or two. Provide at least 4 items.

A.

* choose a 'blind' (non-intelligent) (and numeric) primary key
* create normalized entities
* choose 'good' names for entities and attributes
* create indices for non-PK columns frequently used in queries
* create a bridge entity for M:N related entities - limit redundancies to these
* ...

Q3 (2+4=6 points). A 1NF table, such as the one shown below (we covered this in class on great detail), is analyzed to detect problems (related to unwanted dependencies), which are then systematically eliminated (the table is converted to 2NF, then 3NF).



a. **What is the diagram (shown above) called?**

A. Dependency diagram.

b. **How does the diagram aid in normalization?** Explain briefly, using the above diagram (you can mark it up (draw on it) if you want).

A. It helps identify partial and transitive dependencies, thereby allowing us to create 2NF, then 3NF normal forms that systematically eliminate such unwanted dependencies.

Q4 (3+(2*1)=5 points).
a. In the context of database performance tuning, what is an 'access plan'?

A. An access plan is a sequence of optimized I/O operations for data fetching and storage, that result from parsing and optimizing a SQL query - loosely, it is an ‘assembly language’ version of higher-level code statements, after ‘compilation’.

b. What are a couple of ways using which a SQL programmer can enhance her queries (make them be executed efficiently)?

A.
* in expressions, use literals where possible
* in an OR compound expression, place the subexpression most likely to succeed first
* ...

Q5 (2+4=6 points).
Given an EMP table of the form (EMP_ID,EMP_NAME,EMP_DEPT,EMP_SALARY,EMP_MGR), where the column names have 'usual' meanings, **what would the following SQL query output? You need to explain your answer** (ie. how the query produces the result).

```
SELECT DISTINCT salary
FROM EMP E1
WHERE 2 = (SELECT COUNT(DISTINCT EMP_SALARY)
           FROM EMP E2
           WHERE E1.EMP_SALARY <= E2.EMP_SALARY);
```

A. The query will produce the second largest salary value in the EMP table.

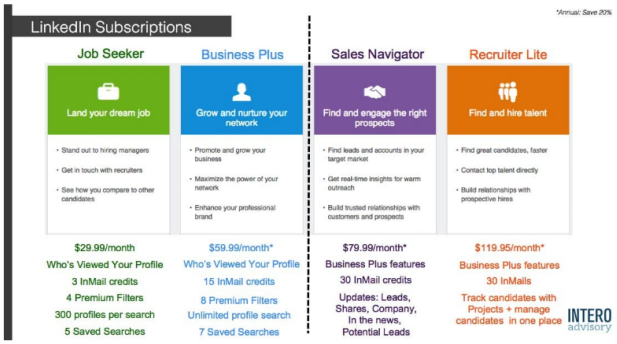
Because EMP_SALARY is used in the sub query, to compare E1's value with E2, it is a correlated subquery - for each row's value of E1.EMP_SALARY, we count how many distinct values in all of E2 are greater than or equal to it. **When we get to an E1 row that contains the second highest salary**, our count will be 2, because in E2, that E1 value and a higher value (the overall highest salary value) are the only 2 values that will satisfy the <= condition.

Q6 (6 points). The diagram below (made by 'INTERO advisory'), lists the 5 types of LinkedIn subscriptions a user can sign up for. **Represent them using a small EER diagram using appropriate notations; for the entity supertype and each entity subtype, list a few relevant attributes.**

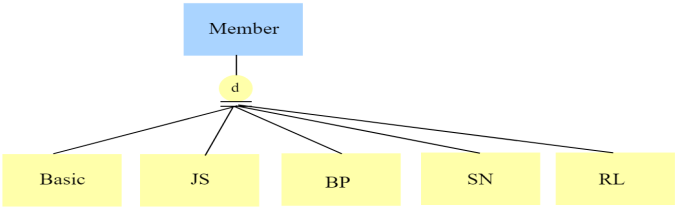
There are 5 types of LinkedIn subscriptions:

1. Basic (free)
2. Job Seeker
3. Business Plus
4. Sales Navigator
5. Recruiter Lite

I put this image together to help easily distinguish the differences between the paid LinkedIn subscription levels:



A.



Member:
* ID
* subscription type (subtype discriminator)
* name
* number of contacts
* date joined
* number of endorsements
* ...

Basic:
* ID

JobSeeker:
* ID
* number of profile viewers
* InMail credits used
* Premium Filters used
* Saved Searches used

BusinessPlus:
* ID
* number of profile viewers
* InMail credits used
* Premium Filters used
* Saved Searches used

SalesNavigator:
* ID
* number of leads used

RecruiterLite
* ID
* number of candidates tracked

Other possibilities for the diagram (two level hierarchies, with disjoint and total constraints at both levels):

* a 2-level hierarchy, with Basic and Paid(JS,BP,SN,RL):

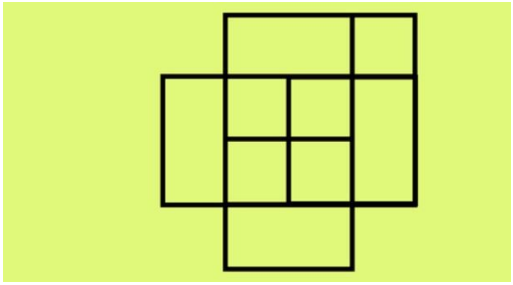
```
Member
  Basic
  Paid
    JS BP SN RL
```

* or a 2-level hierarchy with (Basic,JS,BP) and (SN,RL):

```
Member
  JSBP
    JS BP
  SNRL
    SN RL
```

Bonus (1 point).

How many squares are in the figure below?



10.

CS585 Midterm

Spring term, 2/22/19
Duration: 1 hour

Instructions/notes

- the exam is closed books/notes/devices/neighbors, and open mind :)
- there are 8 questions, a 'non-data-related' bonus, total = 35+1 points
- there are no 'trick' questions, or ones with long calculations or formulae
- **please do NOT cheat;** you get a 0 if you are found to have cheated
- **when time is up, stop your work**; you get a 0 if you continue

Q	Your score	Max possible score
1		4
2		5
3		4
4		3
5		4
6		5
7		5
8		5
Bonus		1
Total		**36**

Q1 (4 points).
Suppose an online vendor maintains its customer list like so:

firstName	lastName	address	city	state	ZIP	phoneNumber	SkypeID	emailAddress
A	B	123 Main St	Los Angeles	CA	90089	213-543-6543		AB@mail.com
Fam	Act	222 Burton Way	Beverly Hills	CA	90210		RichNFamous	RNF@imdb.com
MoreFam	Act	108 Roxbury St	Beverly Hills	CA	90210	323-654-1002		TheBest@BevHills.us
Grad	Student	154 Adams St	Los Angeles	CA	90089			DontBugMe@usc.edu

What two problems do you see with the above scheme, and how would you fix them?
Your answer can be in the form of E-R (using any notation), or in table format (like above) or even SQL. And, feel free to create any new attributes that might be necessary.

Q2 (5 points). Parents in a wealthy family want to create a DB of all their assets. For each asset, they would like to name benefactors - some or all of their five children who would get the asset. Each asset has a financial value associated with it, and a maturity date (when the kid(s) can cash in). They'd like to track the following diverse set of assets they own: bank accounts, real estate, stocks, jewelry, life insurance. **What would be a good design (using an ER diagram) for this?** You can make any assumptions you want about the assets, create whatever descriptors (columns) you need, etc.

Q3 (4 points). A reality company keeps track of its home sales like so:

Seller	Buyer	LendingBank
S1	B1	BofA
S2	B1	Chase
S1	B2	Chase

Things seem fine (redundancy and all), until they hire you to 'clean up' their table. After analysis, you come up with these three separate tables [all linked properly with FK/PK], which makes for good design:
Table 'SellerBuyer', with rows such as (S1,B1).
Table 'BuyerBank', with (B2,Chase) as a sample row.
Table 'SellerBank', eg. with (S2,Chase) as a row.

You write the following three-way 'join' query just for fun, to see if you can recreate the original triplets (eg. S1,B1,BofA):

```
SELECT SB.Buyer, SN.Seller, BN.LendingBank
FROM SellerBuyer as SB, SellerBank as SN, BuyerBank as BN
WHERE BN.Buyer=SB.Buyer
AND BN.LendingBank=SN.LendingBank
AND SN.Seller=SB.Seller
```

Question: what, if any, is the problem with the above query?

Q4 (2+1=3 points). You pull out your smartphone, log on to your banking app, and proceed to transfer $7200 (to pay for a 4-unit 'SC course!) from your savings account into your checking account. Prior to the transfer, you had $20,000 in savings and $800 in checking. While you are in the middle of doing this, due to poor DB design, a report generator (that would produce a monthly statement to email you) starts to run on the bank's server. **What could go wrong, and what is such a scenario called?**

Q5 (2+2=4 points). How would you optimize (by rewriting) the following two queries?

a. `SELECT * FROM TBL WHERE substr(STATE,1,1)='C'`
[we want to select all rows containing US states CA, CO, or CT; substr(<string>,1,1) returns just the first character of a string]

b. `SELECT * FROM TBL WHERE AGE>21`
[the AGE column stores ages as 0..99 integers; assume it has been indexed]

Q6 (3+2 = 5 points). For a while now, NASA has been conceptualizing a network called the Interplanetary Internet, which could come in handy 'someday' when we colonize Mars [when pigs fly out of our butts :)]. If that were to come to fruition, Eric Brewer's 'CAP theorem' would be highly relevant and applicable to such a distributed system of nodes. As per the CAP theorem, 'you can't always get what you want' (at least not C,A,P all at once, all equally guaranteed).

In an Interplanetary Internet, how would you rank C,A,P in terms of concerns? In other words, which would we worry about most, and relatively which, the least? You need to state why (justify your ordering).

Where might nodes be located, for an Interplanetary Internet? And, what disaster scenarios can you envision (that affect the network)?

Q7 (5 points). What operation does the following SQL query implement?

```
SELECT DISTINCT c
FROM A as tA
WHERE EXISTS (SELECT *
              FROM B as tB
              WHERE tA.c = tB.c);
```

Q8 (5 points). Here are a pair of tables – a PRODUCTS table that lists products a company sells, and SALES, which records sales of the products (each unit of a product that is sold, gets a separate row in SALES):

```
PRODUCTS(PRODUCT_ID, PRODUCT_NAME);
SALES(SALE_ID, YEAR, PRODUCT_ID, PRICE);
```

Consider the following three queries, we're calling them Q1, Q2, Q3. In Q2, fyi, 'SELECT 1' returns a 1, which we can ignore (it is not essential to our query).

```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
 JOIN
 PRODUCTS P
 ON (S.PRODUCT_ID = P.PRODUCT_ID)
GROUP BY S.PRODUCT_ID;
```

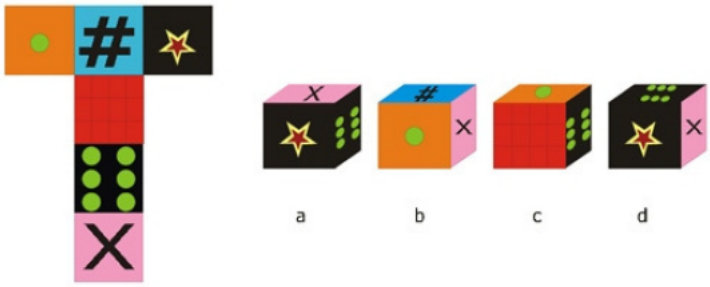
```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
WHERE EXISTS
(
 SELECT  1
 FROM PRODUCTS P
 WHERE P.PRODUCT_ID = S.PRODUCT_ID
)
GROUP BY S.PRODUCT_ID;
```

```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
WHERE S.PRODUCT_ID IN
(
 SELECT  PRODUCT_ID
 FROM PRODUCTS P
)
GROUP BY S.PRODUCT_ID;
```

Circle the correct choice below:

a. Q1, Q2, Q3 are all different (they produce different results)
b. Q1, Q2, Q3 are all identical
c. Q1 and Q2 are identical
d. Q1 and Q3 are identical
e. Q2 and Q3 are identical

Bonus (1 point). Look at the flattened cube below on the left, and four cubes on the right - which of the four would produce the flattening?



Q1 - 4 **points**
Q2 - 5 **points**
Q3 - 4 **points**
Q4 - 3 **points**
Q5 - 4 **points**

Q6 **(3+2=5 points). For a while now, NASA has been conceptualizing a network called the Interplanetary Internet, which could come in handy 'someday' when we colonize Mars [when pigs fly out of our butts :)]. If that were to come to fruition, Eric Brewer's 'CAP theorem' would be highly relevant and applicable to such a distributed system of nodes. As per the CAP theorem, 'you can't always get what you want' (at least not C,A,P all at once, all equally guaranteed).**
In an Interplanetary Internet, how would you rank C,A,P in terms of concerns? In other words, which would we worry about most, and relatively which, the least? You need to state why (justify your ordering).

The most ideal order is Partition Tolerance (highest concern/priority), Availability, Consistency (lowest concern/priority) or (P,A,C). The order (A,P,C) is also acceptable.
Points are given as follows:
- 0.5 : For incorrect order.
- 0.5 : For incorrect acronym expansion for any/all of C, A, P.
- 2 : For incorrect reasons. 1 for justification of each - **the most** and **the least** important concern.
Also, -0.5 if the order in incorrect, the justification wouldn't make sense for either one of the most important concern or the least important concern. Similarly, if the full-form for any of C, A, P is incorrect.

Where might nodes be located, for an Interplanetary Internet? And, what disaster scenarios can you envision (that affect the network)?
Nodes can be located on planets, satellites or different places on earth. Also, locations of the nodes should be explicitly mentioned.

Disaster scenario should be practical and making sense in the given scenario like meteoroid strike, technical problems in satellite, node breakdown on earth due to natural calamity, etc. Replies like "End of universe", "Big Bang", "Black Hole", etc. should not be considered towards a valid answer.
Points are given as follows:
- 1 : incorrect location information.
- 0.5 for each impractical/incorrect disaster scenario. At least two disaster scenarios are required.

Q7 - 5 **points**

Q8 (5 points): Here are a pair of tables – a PRODUCTS table that lists products a company sells, and SALES, which records sales of the products 9each unit of a product that is sold, gets a separate row in SALSES):

PRODUCTS (PRODUCT_ID, PRODUCT_NAME);

SALES (SALE_ID, YEAR, PRODUCT_ID, PRICE);

Consider the following three queries, we're calling them Q1, Q2, Q3. In Q2, fyi, 'SELECT 1' returns a 1, which we can ignore (it is not essential to our query).

Q1:

```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
        JOIN
        PRODUCTS P
        ON (S. PRODUCT_ID = P.PRODUCT_ID)
GROUP BY S.PRODUCT_ID;
```

Q2:

```
SELECT S.PRODUCT_ID, SUM(PRICE)
FROM SALES S
WHERE EXISTS
(
        SELECT 1
        FROM PRODUCTS P
        WHERE P.PRODUCT_ID = S.PRODUCT_ID
)
GROUP BY S.PRODUCT_ID;
```

Q3:

```
SELECT S.PRODUCT_ID,SUM(PRICE)
FROM SALES S
WHERE S.PRODUCT_ID IN
(
        SELECT PRODUCT_ID
        FROM PRODUCTS P
)
GROUP BY S.PRODUCT_ID;
```

Circle the correct choice below:

a. Q1, Q2, Q3 are all different (they produce different results)

b. Q1, Q2, Q3 are all identical

c. Q1 and Q2 are identical

d. Q1 and Q3 are identical

e. Q2 and Q3 are identical

Choice “b” is the correct solution. The selection is performed using PRODUCT_ID key from both P and S tables and grouped together.

No partial marks.

-5: for incorrect answer

Bonus - 1 point

Answer: **‘a’** (+1 point)

-1 wrong answer, No fractional points

CSCI585 Summer '19 Midterm Exam

June 17th, 2019

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 2 hours. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0.

Solutions are in red font.

Signature: _____

Problem Set	Number of Points
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	5
Total	**35**

Q1. (5 points total) INTRODUCTION AND DATA MODELING

Using your school's student information system, print your class schedule. The schedule probably would contain the student identification number, student name, class code, class name, class credit hours, class instructor name, the class meeting days and times, and the class room number.

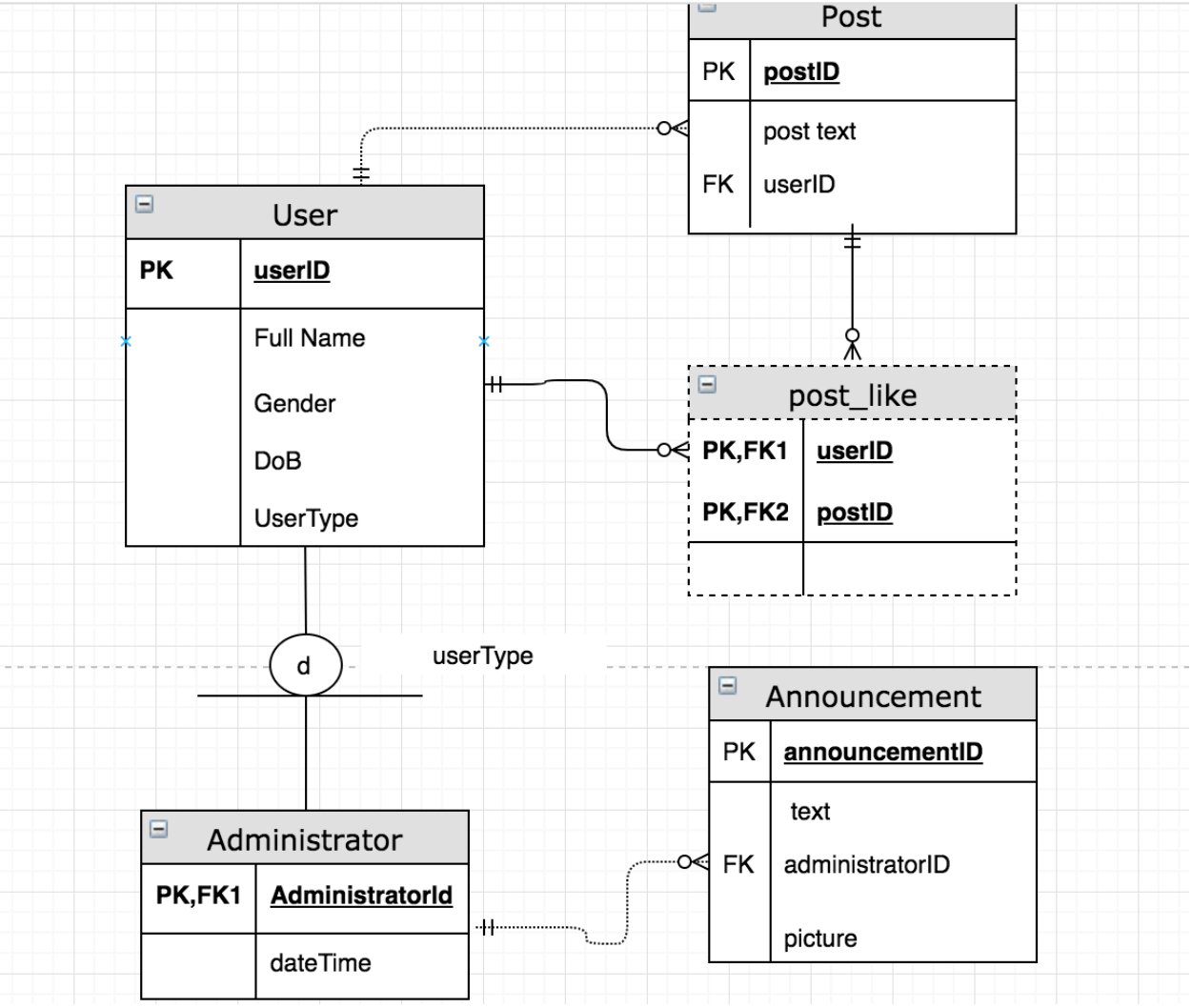
a. Create a spreadsheet using the table shown below and enter your current class schedule.
b. Enter the class schedule of two of your classmates into the same spreadsheet.
c. Discuss the redundancies and anomalies caused by this design.

STU_ID	STU_NAME	CLASS_CODE	CLASS_NAME	CRED_HRS	INSTR_NAME	CLASS_DAYS	CLASS_TIMES	ROOM

Students are likely to identify the redundancies around the class information since all three schedules (the student's own schedule plus the schedules of the two classmates) will have at least the database class in common. (1 point)
This leads to discussions of separating the data into at least two tables in a database. (2 points)
However, that still leaves the redundancies of redundant student data with each class that they are taking. Students might realize that a table for student data, a table for class data, and a table to relate the students and classes is appropriate. (2 points)

Q2 (5 points total) ER MODELING

Design ERD using Crow's foot notation for the following description:
A Forum is a website for users to discuss and exchange ideas. Each user has one unique profile including name, gender, and date of birth. A user can be either a normal user or an administrator user. If a normal user become an administrator user, the most recent date of promotion should be recorded.
In the forum, a normal user can make a post with text content. Besides posts that a normal user can make, an administrator can also make an announcement, with both text content and one picture. Only post could be liked by multiple users, and the system would keep track of all the users who like a post.



(1pt) Reasonable entities with PKs defined (e.g. User, Post, Announcement)
(1pt) User-Administrator hierarchical specialization (or similar solution to address data redundancy)

(1pt) Bridge entity to address many to many relationship between user and post
(1pt) correct weak relationships between entities (with FK properly defined)
(1pt) correct strong relationships between entities (with FK properly defined)

Q3. (5 points total) SQL

Write the following queries for an Employee database. Below are the tables for the same, primary keys are underlined, foreign keys are italic. Min_Salary and Max_Salary represent the minimum and maximum salary given to a type of job.

Employees(Employee_Id, First_Name, Last_Name, Hire_Date, *Job_Id*, Salary, *Department_Id*)
Departments (Department_Id,Department_Name)
Jobs(Job_Id, Job_Name, Min_Salary, Max_Salary)
Job_History(*Employee_Id*,Start_Date,End_Date,*Job_Id*,*Department_Id*)

a.(3 points) Write a query to display the job_name, First_Name and Department_Name of all employees who started their jobs before 8 August, 2015.

```
SELECT Job_Name, Department_Name, First_Name , Start_Date
FROM Job_History
WHERE jobs.Job.Id=Job_History.Job_Id AND
Departments.Department_Id=Job_History.Department_Id AND
Employees.Employee_Id=Job_History.Employee_Id AND
start_date<'2015-08-08';
```

b.(2 points) Write a query to display Job_Name , first and last name of employees who make at least $10000 less than maximum salary at their current job.

```
SELECT Job_Name, First_Name, Last_Name
FROM Employees, Jobs
WHERE Employees.Job_Id= Jobs. Job_Id AND
Jobs.max_salary-Employees.salary >10000;
```

Q4. (5 points total) NORMALIZATION

a. (1 point) Write down the highest Normal Form for the following table and explain why.

s_id	Course	hobby
1	Science	Football
1	Math	Tennis
2	Physics	Hockey
2	Php	Baseball

It satisfies 3NF, since the non prime keys (i.e. course and hobby) does not have transitive dependency. (1 point)
Optional explanation: It has multivalued dependency (two records associated with s_id = 1), and therefore does not meet 4NF.

b. (4 points) Convert the following table to 3NF, show or explain the dependency diagram and the primary key of the table(s).

emp_id	emp_name	zipcode	state	city	district
1001	John	282005	UP	Arga	Dayal Bagh
1002	Joseph	222008	TN	Chennai	M-City
1003	Lily	282007	TN	Chennai	Urrapakkam
1004	Steve	292008	UK	Pauri	Bhagwan

Transitive Dependencies:
zipcode -> state, city, district

3NF tables:
(**emp_id**, emp_name, zipcode)
(**zipcode**, state, city, district)

(1 point was deducted for those who created a new table for emp_id, zipcode)

Another Accepted solution:
(**emp_id**, emp_name, zipcode)
(**zipcode**, district)
(**district**,city)
(**city**,state)

Q4. (5 points total) NORMALIZATION – CONTINUED

Please use this as extra space for solving this (normalization) question.

Q5. (5 points total) TRANSACTION MANAGEMENT

Based on a simple relation TA(SID, CID, Stipend) stores TA assignments and stipends. TA's are identified by their student ID's (SID's). Consider the following, if transaction executed by 2 different clients at approximately the same time. Before either transaction starts, there is a tuple (987, 'CS585', 1900) in TA.

T1	T2
Step1: UPDATE TA SET Stipend= Stipend + 200 where CID= 'CS585' ; Step2: UPDATE TA SET Stipend= Stipend + 200 where CID= 'CS585' ; COMMIT	UPDATE TA SET Stipend= 2000 where Stipend > 2000 ; COMMIT

a. (2 points) Suppose T1 and T2 executes with the possibility of interleaving of operations of the two. What could be the possible final stipend values for TA987?

(2 points)
1) 2000 is the result of the execution sequence: T_1.step1, T_1.step2, T_2 (0.5 point)
2) 2200 is the result of the execution sequence: T_1.step1, T_2, T_1.step2 (1 point)
3) 2300 is the result of the execution sequence: T_2, T_1.step1, T_1.step2 (0.5 point)

b. (1 point) Now, suppose instead T1 and T2 executes with isolation level guaranteed. What could be the possible final stipend values for TA987?

(1 point) 2000 and 2300 are still possible, but 2200 is not, because T2 cannot get in between T_1.step1 and T_1.step2 to read the uncommitted value written by T_1.step1. (0.5 point each value)

Q5. (5 points total) TRANSACTION MANAGEMENT - CONTINUED
Consider the following schedule:

T1	T2
	R(A) A=A+10 R(B) A=B+10 W(A)
R(B) B=B+10 R(A) B=A+10 W(B) Commit	
	Commit

c. (1 point) Which of the transaction problems is present in the given schedule and why?

(1 point) Dirty Read/Uncommitted data as transaction T1 is reading the uncommitted data modified by T2.
No points for deadlock/Inconsistent retrieval/ lost update(as both are writing different data values one is writing A and another B)

d. (1 point) Name a technique that can help us avoid the above problem using an example.

(1 point) 2PL Locking. It does that by ensuring serializability of the transactions.

(0.5 point for technique,
0.5 point for explanation/example)

Q6. (5 points total) QUERY OPTIMIZATION

a. (1 point) What are some of the reasons for poor performance of a query?

(1 point) if at least one reasonable answer is listed (answers might vary).
Some of possible answers:
1) No indexes
2) Excess recompilations of stored procedures.
3) Procedures and triggers without SET NOCOUNT ON.
4) Poorly written query with unnecessarily complicated joins
5) Highly normalized database design.
6) Excess usage of cursors and temporary tables.

b. (4 points) Optimize the following two queries and provide explanations for your choice of optimization techniques used

```
SELECT *
FROM employee
WHERE salary != 98000
```

```
SELECT *
FROM employee
WHERE salary > 98000 or salary <98000
```

(1 point for optimized query)

SQL engine prefers '=' instead of '!=' when it comes to query optimization. In order for this optimization to work, It is assumed that there's an index (sorted) on salary column.
(1 point for explanation of issue/solution - a point will be given when issue is correctly identified, even if student couldn't think of a correct solution)

```
SELECT id, name, salary
FROM employee
WHERE salary + 10000 < 35000;
```

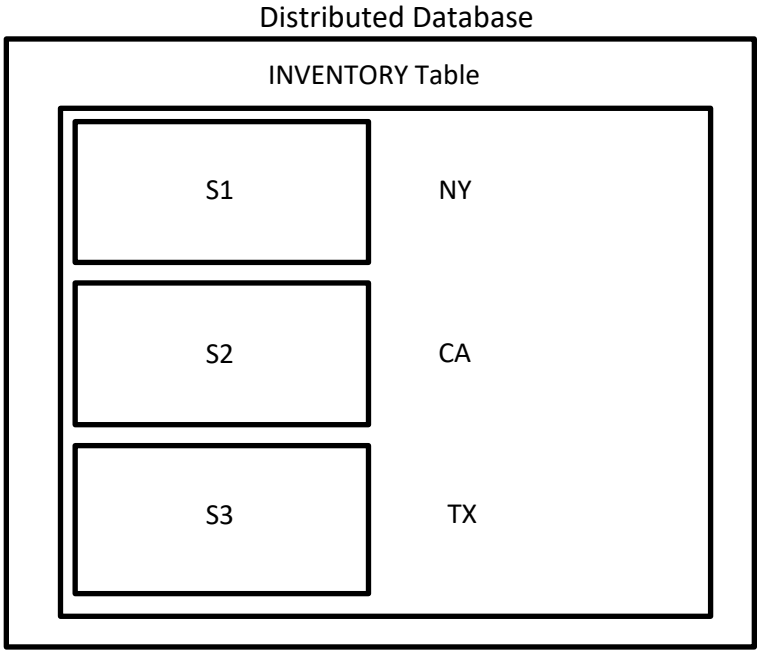
```
SELECT id, name, salary
FROM employee
WHERE salary < 25000;
```

(1 point for optimized query)

This query is performing (35000-10000=25000) calculation for every row (as many times as there are rows in the salary table). The solution is to calculate that value only once.
(1 point for explanation of issue/solution - a point will be given when issue is correctly identified, even if student couldn't think of a correct solution)

Q7. (5 points total) DISTRIBUTED DATABASES

Refer to the diagram below and explain what kind of distribution transparency is supported by the database when considering each query. Please write your answer next to each query.



The INVENTORY table is fragmented into 3 parts. Each fragment is stored at one of the nodes residing at 3 locations- NY, CA, TX.

Queries to list all the details of the books whose Quantity On Hand(QOH) is less than 50.

a. (1 point) Fragmentation transparency

```
SELECT *
FROM INVENTORY
WHERE QOH < 50;
```

b. (1 point) (Local) Mapping Transparency

```
SELECT *
FROM S1 NODE NY
WHERE QOH < 50
UNION
SELECT *
FROM S2 NODE CA
WHERE QOH < 50
UNION
SELECT *
FROM S3 NODE TX
WHERE QOH < 50
```

Q7. (5 points total) DISTRIBUTED DATABASES - CONTINUED

c. (1 point) Location Transparency

```
SELECT *
FROM S1
WHERE QOH < 50
UNION
SELECT *
FROM S2
WHERE QOH < 50
UNION
SELECT *
FROM S3
WHERE QOH < 50;
```

Consider the following inventory table of an electronics manufacturing company. Looking at each of the scenarios, suggest a fragmentation strategy to be used.

ID	Product	Cost(USD)	QOH	State
21345	Laptop	3000	10	CA
26312	Mobile	800	5	CA

15263	TabletPC	1500	5	TX
17854	Notebook	1900	12	AZ
28896	Laptop	3500	19	TX
95645	Mobile	4000	25	TX
78451	Laptop	700	25	AZ
95512	TabletPC	1800	14	CA

d. (1 point) The marketing team of this company needs to analyse and compare the cost of each of its products with the competitor's products.
(1 point) Vertical Fragmentation.
The attributes needed by the marketing team are ID, Product and Cost(USD).

e. (1 point) The offices at CA, AZ and TX need need to calculate the local sales of each product.
(1 point) Horizontal Fragmentation.
The rows can be divided as per the partition key- State.

SCRATCH PAPER PAGE - Please use this as your scratch paper (not graded).

If you'd like your work on this page graded, please make a note in the question that you're continuing to answer on this page.

CSCI 585 Midterm Exam Fall 2020

Sample Solutions
&
Rubrics

Q0 [0 points]. DO turn this in - DO NOT omit doing so [you will LOSE 5 points if you skip this].

Please write the following line, and sign it - it is your acknowledgment of having read USC's policies on academic misconduct (https://policy.usc.edu/scampus-part-b/ (https://policy.usc.edu/scampuspart-b/), 11.11-11.14) and agreement to honor them.

I have read USC's standards on academic integrity, and agree to abide by them.

Q1 [2.5*2 = 5 points].

Modern applications that run on the Internet are able to connect to multiple databases, via microservices. In the early days of databases, applications were monolithic and centralized, in 'SDSP' fashion. Trace the evolution, from the old to the new - in other words, what were the intermediate steps? Discuss at least 2 steps.

A. A variety of intermediate steps exist(ed) that can be named+discussed: ODBC, JDBC, COM, COM+, CORBA, cgi-bin (server-side scripts), client-side support (eg. using ActiveX, Node...), dedicated web-to-db middleware.

Rubrics:

2 or more technologies mentioned and described from above list fetch 2.5 * 2 = 5 points total.

1 point for mentioning technology name

1.5 points for appropriate technology description

Q2 [2+4 = 6 points].

A typical DB transaction consists of table reads and writes in a speci c order - the order matters, because earlier operations typically a ect later ones (eg we read from a cell value, multiply it by 1.1, write it into the cell). In an interleaved transaction schedule, multiple transactions' reads and writes are included, preserving each transaction's order of reads and writes. The DBMS allows the schedule to proceed, as long as the schedule is 'serializable', ie. as long as it is equivalent to the transactions running serially (in sequence, one a er another). But if the schedule is not serializable, one or more transactions would need to be aborted and restarted.

2a. Why not make all schedules always run serially?

A. Doing so will make the throughput (processing rate) su er, ie. users would face longer wait times compared to running transactions interleaved.

To check if a schedule is serializable (and therefore executable by the DBMS), we can SWAP two non-con icting operations at a time, multiple times, till we transform it to a serial schedule (note that two

operations con ict if they belong to di erent transactions, operate on the same cell value, and one or both of them write to the cell).

2b. Is the following schedule, serializable or not, and why (show your steps)? In the schedule shown, A and B are the data items (cells), R and W are read and write operations, transactions are numbered 1 and 2.

```
R1(A) R2(A) R2(B) W2(B) R1(B) W1(A)
```

A. We can analyze the schedule in one of two ways - by swapping noncon icting pairs of operations like mentioned above, or, by 'sliding' them visually along the timeline, to group together steps belonging to each transaction [which amounts to the same thing as the rst technique, but the sliding needs to respect con icting ops, ie. should produce the SAME serial result].

Swapping:

```

```

```
R2(A) R1(A) R2(B) W2(B) R1(B) W1(A)
[swap R1(A),R2(A)]
```

```
R2(A) R2(B) R1(A) W2(B) R1(B) W1(A)
[swap R1(A),R2(B)]
R2(A) R2(B) W2(B) R1(A) R1(B) W1(A)
[swap R1(A),W2(B)]
```

Since swapping yielded transaction 2 followed by transaction 1, we conclude that the original schedule is indeed serializable.

On a timeline [where time increases downwards]:

```
     R2(A)
     R2(B)
     W2(B)
R1(A)
R1(B)
W1(A)
```

RUBRICS:

2a.

+2 For mentioning throughput/processing rate/any other synonym

+1 for any other partially correct reason (ANY reason related to efficiency, less delay etc, is fine)

0 for incorrect reason

2b.

+2 just for mentioning the schedule is serializable

+2 for explaining either swapping/sliding method

+1 for partial/incomplete explanations

0 for mentioning the schedule is non-serializable, irrespective of the explanation

Q3 [2+4 = 6 points].

What is the advantage of managing distributed transactions using the 2PC protocol?

A. A distributed transaction has a much smaller chance of failing, one account of one or more nodes in the cluster not being able to commit their transactions - this is because the coordinator polls all the nodes, and asks them to commit only if all are able to.

What are a couple of ways by which (even) 2PC can fail?

A. The coordinator node can fail. Or, network partitioning can occur during phase 2, when all the nodes are committing their transactions (meaning, a commit could fail after commencing).

Rubrics:

1. +2: Mentioning correct advantage of using 2PC protocol - close to the given solution. If solution is not clear and requires more explanation, deduct - 1.

2. +4: Minimum 2 points needed - blocking problem, network partitioning during phase 2, etc.

   If two similar points have been written using interchangeable words, consider it as one point only (but consider giving 3 points, instead of 2, or instead of 4)

Q4 [2 + 2*2 = 6 points].

What problem is the 2PL scheme designed to prevent?

A. The problem of lock starvation, ie. deadlock - on account of two or more in-progress transactions need to wait for each others' currently held locks to get released ('circular wait').

In the version of 2PL we looked at (called Conservative 2PL), there is a lock acquisition phase, operation phase, and a lock release phase (where the locks are being released before the transaction has been committed). With Conservative 2PL, what problem can occur during lock acquisition, and during lock release?

A. During lock acquisition, deadlocks can occur (but can be resolved, by rolling back o ending transactions). During release, if the uncommitted operation needs to be aborted, this will result in cell values possibly being reverted, which in turn could result in other transactions (which were granted locks to these cells a er our transaction began releasing them) having read incorrect (pre-rollback) data, which means THEY too need to be aborted (ie. it's the problem of 'cascaded aborts').

Rubric:

1. 2PL Scheme designed to prevent:
    a. Any answer explaining deadlock/ lock starvation/circular wait fetches 2 marks.
    b. If there is no near mention of deadlock -1 marks.
2. Conservative 2PL
    a. Lock Acquisition Phase
        i. Problem: Mention Deadlocks or any definition of the same. (2 marks)
    b. Lock Release Phase
        i. Problem: Explain process of cascading aborts. (2 marks)
        ii. If uncommitted data needs to be aborted, it may lead to other transactions getting aborted.(Another way of saying the same!(2 marks)
3. In 2nd part, If only one of the two processes [of lock acquisition and release phase] is explained (2/4 marks)

Q5 [1.5*4 = 6 points].

SQL is a set-oriented/declarative language, designed to operate on data held in tables. However it is not a complete programming language (such as C++, C#, JavaScript, Python, etc). What four (or more) features would you add to SQL, to make it be a more powerful/complete language?

A. Variables, flow control (branching, looping!), higher level types such as list and dictionary, import/include/source facility for reading in other scripts, static functions that operate on all rows of a table, functional constructs such as map() and filter(), etc…

Rubrics:

+ 1.5 for each of the above mentioned features.

If there are 4 valid features in the answer then the student gets 6 points.

Q6 [1.5*2 + 3 = 6 points].

A university chooses to represent courses and pre-reqs this way (CourseNumber is the PK):

```
[CourseNumber, Prereq1, Prereq2,
additional course-related columns...]
```

What are a couple of problems with representing the data like shown above?

A. One problem is wasted space - not all courses will have two prereqs; the other problem is the opposite - if a course has >2 prereqs, there is no way to list them.

What is a better design?

A. A separate 'prereqs' table, with just a pair of columns: [CourseNumber, Prereq]. Each prereq for a course, whether it is 1,2 or >2 for a given course, will get its own row. We could enhance the functionality with extra columns, eg. 'Mandatory' [a Boolean - if a prereq is not mandatory, it could be waived by a student's advisor or the course instructor].

Rubrics:

Problems: +1.5 for one reason (wasted space/ more than 2 prereq)

+1.5 for the other reason

Better design:

+1 for one suggestion for better design each upto a total of 3 (Separate 'prereq' table, Separate row for each prereq, extra column about mandatory or not) .

Any valid reason close to the one in the rubrics can also get +1.

Q7 [6 points].

The process of data modeling is one of abstraction, where, as a first step, relevant data-describing features ('columns', in a relational table) for an application need to be identified. Describe this process. Specifically, how does feature selection occur (where do features "live")?

A. Features are part of entities - and entities are gathered from business rules, policies, operating manuals, interviewing stakeholders (employees,

managers, other users of the DB), etc. Once entities are identified, "relevant" features for those entities need to be listed - these again come from business rules etc. In general, **features "live" in the minds of the users and designers** - they are NOT inherent to the world - we create features out of necessity - they help model whatever function/process/application we want to describe.

Rubrics:

No description of how to identify features. -2 marks

No mention of “entities” or similar. -1 mark

No mention of where features live. -2 marks

No mention of “minds of users and designers, or business rules” or similar. -1 mark

CS585 Midterm

Spring term, 2/28/2020
Duration: 1 hour and 15min

Instructions/notes

- the exam is closed books/notes/devices/neighbors, and open mind :)
- there are 6 questions, and a non-data-related bonus, for a total of **35** points
- there are no 'trick' questions, or ones with long calculations or formulae
- you can write on the two blank sheets (that are at the end) if you like
- **please DO NOT CHEAT; you will get a 0 if you are found to have cheated**
- **when time is up, please STOP WRITING**; **you will get a 0 if you continue**

Q	Your score	Max possible score
1		6
2		5
3		6
4		6
5		6
6		6
Bonus		1
Total		**35 (NOT 36)**

Q1 (3+3 = 6 points)

Consider a table, with several (2 or more) columns that contain repeating data values (within each column).

a. when is this acceptable? Explain, with an example.

A. It is acceptable when it is a bridge table, where redundancies cannot be avoided (eg. an 'Enrollment' bridge entity with a (StudentID,CourseID) PK where values can repeat in each column.

b. when is this not acceptable? Explain, with an example.

A. Not acceptable when it is in the form of a 1NF table - here we have NEEDLESS repetition, which calls for the table to be segmented using normalization principles (eg. the 'Project' example from the lecture).

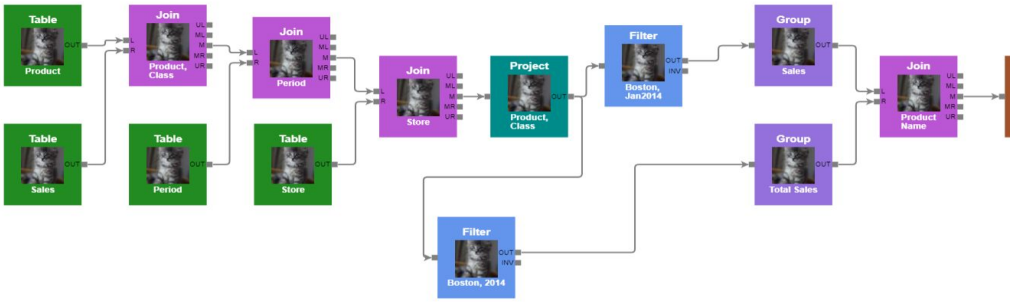
Q2 (5 points)

A ‘shared lock’ is granted to a set of transactions that all want to read data (unlike an ‘exclusive lock’ that is granted to a single transaction that wants to write data). Why is this even a thing, ie. why does reading (which causes no change to contents being read) require locking at all?

A. Because while the readings are occurring other transactions might write to the cells that being read, causing non-repeatable reads.

Q3 (3+3 = 6 points)

The diagram below, shows a SQL dataflow graph. Users can construct non-trivial queries by dragging and dropping 'ops' (operators/nodes) on to a palette (graph area), and wiring them up by connecting inputs and outputs as shown - in other words, queries can be built up by chaining nodes, as shown (eg. in the graph shown, we are reading four tables using a 'Table' op, and performing joins using 'Join', projecting using 'Project', etc.).



a. what is the advantage of such visual construction, over coding queries by hand using SQL commands (like you did for HW2)?

A. The advantage is that non-programmers (casual users, business analysts...) can create SQL queries without having to write code.

b. conversely, what are the advantages of being able to hand-code queries?

A. Hand-coding offers flexibility, control and power - queries that can't be visually constructed (eg ones with two or three levels of subquery nesting, correlated subqueries etc) can always be explicitly coded.

Q4 (3+3 = 6 points)

As you know, in 2PL, a transaction acquires locks during the locking phase, carries out the transaction, then releases locks during the unlocking phase (there is no interleaving of locking and unlocking).

a. what issue can arise, during the locking phase, how is it resolved?

A. Deadlocks can arise, on account of two or more transactions in the process of lock acquisition being unable to complete the step on account of mutual dependencies. Deadlock detection and mitigation strategies will help resolve the issue (the dependency cycles need to be broken).

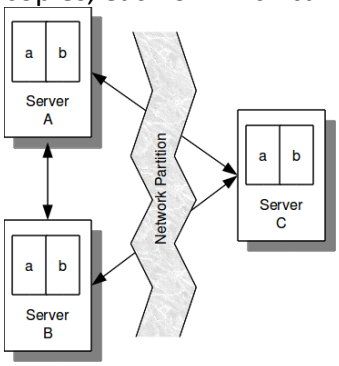
b. what issue can arise during the unlocking phase, how is that resolved?

A. During sequential unlocking, the transaction doing the unlocking might need to abort, in which case other transactions that acquired the released locks and started their own transaction operations will also need to be aborted ('cascading aborts'). To resolve (prevent) this, we hold on to the locks until our transaction is commited (or aborted), and release the locks after (using a protocol called 'Strict 2PL').

Q5 (3+2+1 = 6 points)

During a network partition (failure), a portion of a distributed DB (with replicated data stored in multiple nodes, and the cluster operating as a single unit) gets separated from the rest (eg. on account of a network switch failure).

In the figure below, a partition separates nodes A and B (which remain connected to each other), from node C - this will lead to AB, and C, operating as two independent DB copies, each of which can receive read/write requests.



Consider a write request arrives at node C.

a. what are the two options that C would choose from?

A. C can accept (perform) the write operation, or it can refuse the request.

b. what are the implications of choosing either option?

A. If C carries out the write, we will have an inconsistency (between C, and AB) - we are prioritizing availability; if we refuse, the transaction requesting the write would need to be aborted and possibly retried - we are prioritizing consistency.

c. what distributed DB principle/idea/'law'/theorem are we referring to?

A. The CAP Theorem.

Q6 (6 points)

Consider the following table (PrinterControl), which is used to assign specific printers to named users as well as guests, in a workgroup (the first three entries are named users):

```
PrinterControl
user_id_start    user_id_finish   printer_name
printer_description
============================================================
'chacha'   'chacha'        'LPT1'   'First floor's printer'
'lee'      'lee'           'LPT2'   'Second floor's printer'
'thomas'   'thomas'        'LPT3'   'Third floor's printer'
'aaaaaaaa' 'mzzzzzzzz'     'LPT4'   'Common printer #1 '
'naaaaaaa' 'zzzzzzzz'      'LPT5'   'Common printer #2'
```

Explain what the following query does, when the :my_id variable can contain a variety of userIDs, of named users and guests (eg. it could contain 'lee' or 'archit' or 'yiming') - the BETWEEN keyword returns a boolean if a given string lexicographically (alphabetically) lies between two others. Be very specific in your explanation (eg. be sure to explain why we use MIN()).

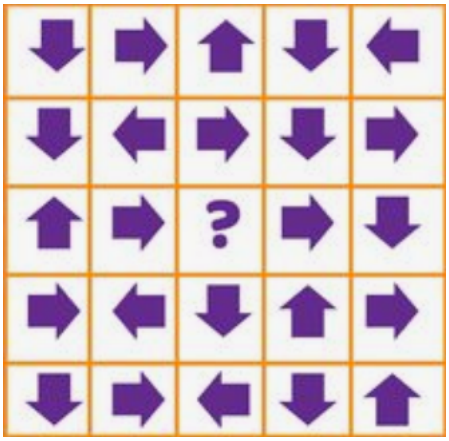
```
SELECT MIN(printer_name)
  FROM PrinterControl
 WHERE :my_id BETWEEN user_id_start AND user_id_finish;
```

A. The query assigns (selects) a printer, based on incoming username - if the user is chacha or lee or thomas, the assigned printer is LPT1, LPT2 or LPT3, respectively; for all other users, LPT4 is assigned if their username lies in a..m, or LPT5 otherwise (n..z). The MIN operation selects LPT1 instead of LPT4 for chacha (and likewise for the other two named users).

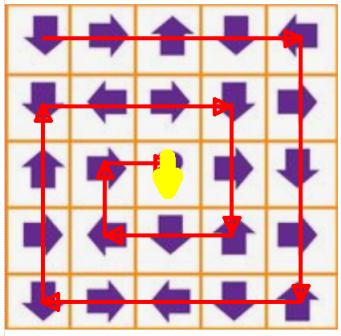
Bonus (1 point)

Note - this bonus is optional - if you do get it right, you get 1 point, which is counted only when you don't have 35/35 already :) In other words, the max you can get for the whole test is 35, not 36.

In which direction should the missing arrow point (there is only ONE right answer :))?



A. Down - starting from top-left, follow this spiralling path where 'Down-Right-Up-Down-Left-Right' repeats:



CSCI 585, midterm exam, 6/11/20

Please read the following carefully, before starting the test.

• the exam is open books/notes/devices - feel free to look up whatever you want!

• there are 7 questions plus a 'non-data-related' bonus, for a max of 35 points

• there are no 'trick' questions, or ones with long calculations or formulae

• please do NOT cheat - this means NOT communicating with anyone via any device/medium/channel - you will get a 0, and be reported to SJACS, if you are found to have cheated; ANY attempt to get help from others in any form is a VIOLATION, as per https://policy.usc.edu/scampus-part-b/, sections 11.11 through 11.14 [read it, if you are not familiar with it]

• when the time is up (75 minutes), stop your work, then spend the rest of time (30 minutes) on submission

Good luck!

Q0 [0 points]. DO turn this in - DO NOT omit doing so.

Please write the following line, and sign it - it is your acknowledgment of having read USC's policies on academic misconduct (https://policy.usc.edu/scampus-part-b/, 11.11-11.14) and agreement to honor them.

I have read USC's standards on academic integrity, and agree to abide by them.

Q1 [2*2 = 4 points].

'Loose coupling/loose dependence is preferable to tight coupling/tight dependence.'

What are TWO different cases where the above is true, in the world of data-handling (ie. in what we've covered in the course)?

Answer:

1. File system DBs vs other kinds, eg. relational DBs. In a DB, we want the DB engine for adding, modifying, deleting and querying data, to be de-coupled from the internals of how the data is stored, ie we want structural independence.

2. With DB connectivity, we'd like loose coupling between data consumers (eg client apps that use web services), and data producers (the systems that supply data, in response to service/microservice requests). In other words, an answer could be services/microservices, where requests and responses are decoupled.

For each case:
+2 for correct answer which is very similar to this
~~-1 if decoupling between system and data is not mentioned~~
[the correct answer would need to involve an example where decoupling is desirable, like in the two cases I've shown].
+1.5 partial credit if some other case is written which is very similar to the correct answer

Q2 [3*2 = 6 points].

We saw examples of where a SQL query was intermixed with C#, Java, Python... **What are practical reasons** (at least two) for needing to do so?

Answer:

In real-life, most data is accessed via apps and other client software, that are written in host languages such as C++, Java, Python, JS etc, that contain the UI code, presentation logic, etc. So there needs to be way for these apps to communicate with a DB engine that expects SQL queries - so there is intermingling, typically in the form of SQL queries being handled as strings in the host language and handed off to the SQL engine.

Non-SQL languages have powerful data structures and associated methods, and language-level features (eg functional programming, iterators etc) - the programmer is able to leverage these, by mixing them with SQL (eg. resultsets returned from the DB engine can be traversed using methods such as .next()).

(+3 for each correct answer which is very similar to this)
(+2 partial credit if some other reason is written which is very similar to the correct answer)
(0 if the reason is not practical and/or doesn't make sense)

Q3 [5 points].

An investment company has invested its members' wealth, in a variety of holdings: stocks, real-estate, gold and other forms of jewelry, antiques, famous paintings. You are asked to help catalog the assets. How would you represent the wealth being invested, via a simple EER diagram? You can assume whatever you need (in regards to representing the various types of holdings).

Answer:

Multiple answer variations are possible - overall, a simple two level design will do.

Superclass entity: Asset, with these columns: AssetID (PK), MemberID, PurchaseDate, PurchaseAmt, TodaysWorth, AssetType [subtype discrim column, MUST be disjoint]

Member (MemberID, Name, Address, Phone, Email, MemberSince, MemberClass, etc): links to Asset as 1:M

Subclass for each type of asset, eg. Stock, RealEstate, Valuables, Antiques, Painting [each will have specific columns, eg. for Painting: Medium, Size, Painter, YearPainted...].

Rubrics:
-0.5: If incorrect relationship between member and asset.
-0.5: If incorrect relationship between superclass and subclass (disjoint)
-1: If member entity not mentioned
-1: If asset is not the Superclass entity
-1: If only some of the subclass are mentioned
-2: If none of the subclass are properly mentioned
-1: If all attributes are not mentioned
-0.5: If some attributes are mentioned

Q4 [3+2 = 5 points].

What is the problem with the 'original' CAP Theorem? Explain in your own words, in a few sentences. **What is the** modern, preferable alternative formulation? Do not simply state the CAP Theorem - answer the questions asked!

Answer:

In the original CAP theorem, partition tolerance ('P') was on equal footing with C and A, making it seem like there was a choice between CA, AP and PC that a DB designer could provide end-users; but in reality, CA is never a choice (dropping P is not an option).

In the modern interpretation, we consider the theorem as providing a choice between availability vs consistency, when a partition failure occurs - whether we keep operating (A) while letting data become inconsistent, or prioritize consistency at the expense of lowering system availability.

Rubrics:
Part 1 (3 points): Include CA is not realistic / low P is not a option, and give correct explanation gets full marks
-1: if specify CA not realistic but the explanation is wrong.
-2: if only specify we can choose at most 2 out of 3 [CA, PA, PC] or cannot achieve CAP all at once
-3: if totally wrong or unrelated

Part 2 (2 points): Include choice between A and C gets full marks, else include BASE or PACELC also gets full marks, or if they state the principle of BASE of "sacrifice consistency in favor of availability" .
-2: if not related to above

Q5 [2+3 = 5 points].

What does the following query do?

```
SELECT x.Name,
       x.City,
       (SELECT CompanyName FROM Company WHERE CompanyID =
x.CompanyID) AS CompanyName
FROM Customer x
```

The query is better expressed as follows. **Why?**

```
SELECT r.Name,
       r.City,
       c.CompanyName
FROM Customer r
    LEFT JOIN Company c
         ON r.CompanyID = c.CompanyID
```

Answer:

Given a Customer and Company table, the query lists the name, city, company (including null) for each customer.

The join query is better, because the joining is a one-time operation that produces the same result - but in the previous case, the correlated query required the Company table to be scanned repeatedly, for each customer in the Customer table (highly inefficient!).

Rubrics:
Part 1:
2points, if the answer is written correctly or as close to one above .
Either full or zero

Part 2:
2 points for specifying why the join query is better
1 point for specifying previous query as correlated query(and its inefficiency).

Q6 [1*6 = 6 points].

How does normalizing tables, help or hinder the following?

• data integrity

• querying

• creating and using indexes

• data updates

• concurrency

• DB design

Answer:

Integrity: Goes up, because data is kept in just a single location, and referenced from it elsewhere.

Querying: can be easy if searching on one or a small # of tables, but with a large number of participating tables, join queries are more verbose, and are slower (inefficient).

Indexes: Easier to create, and more importantly, be used by the query engine. It helps in faster data retrieval.

Updates: Easy and fast, since we only update a portion of our entire DB [including the fact that we only need to update fewer indexes]

Concurrency: Also goes up, since locking is restricted to just the tables/rows where there is contention (multiple requests).

DB design: Helps create a clean design with no partial or transitive dependencies. There must be balance in normalization and performance. Overly normalized DB will not perform optimally

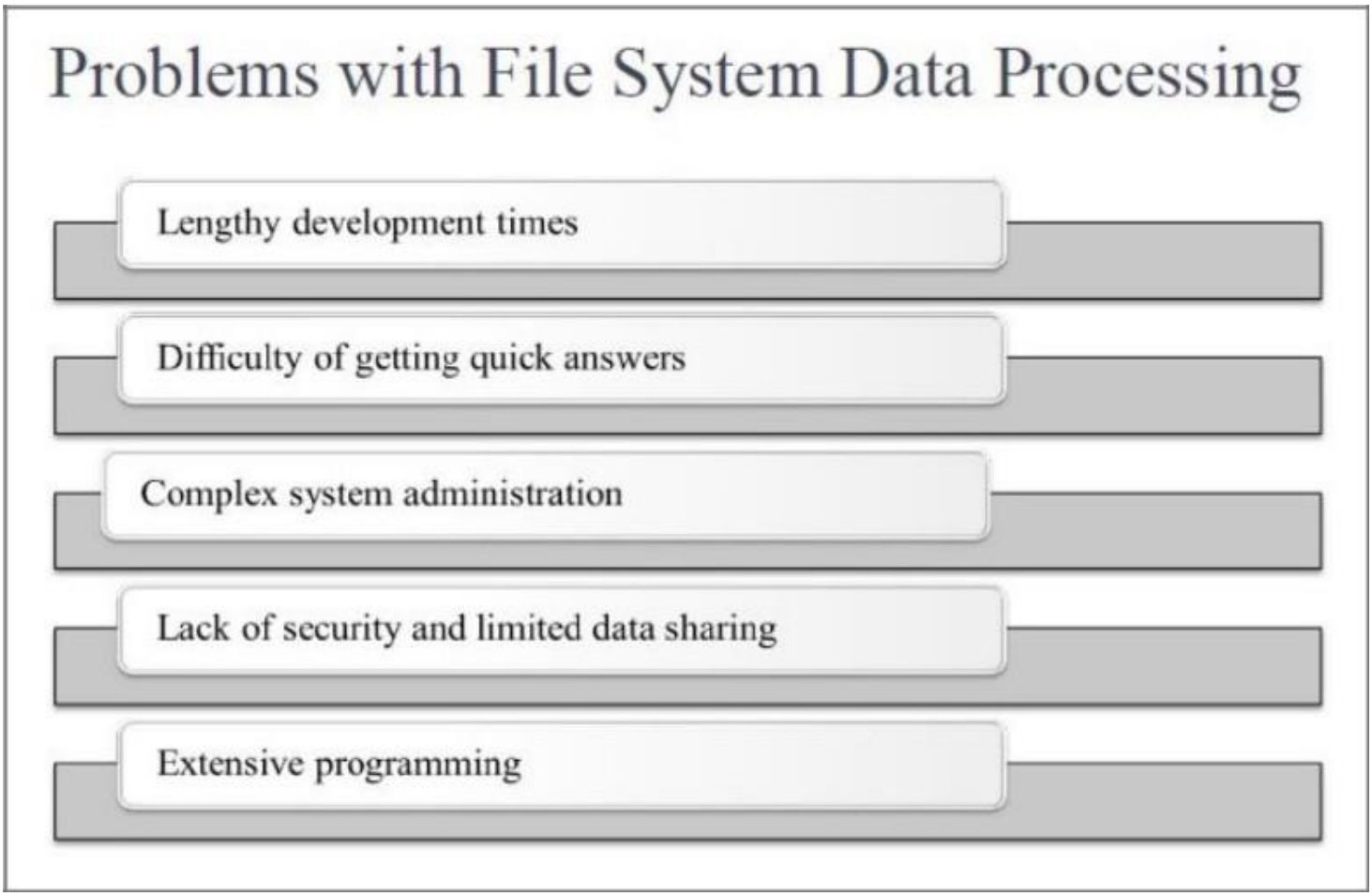
Rubrics:

1 point for each of the above points if mentioned correctly about how normalization helps/prevents in providing integrity,querying etc

-0.5 if a point doesn't specify correctly how does it assists/prevents
-0.5 if a student writes normalization will always help in DB design. Overly normalized DB will not perform optimally

Q7 [2*2 = 4 points].

We discussed the following problems, when it comes to using a simple file system as a database:



The above problems are all due to a single core reason: lack of a

standard way to query the data. **What are additional problems** (at least two) with using loose files to store and query data?

Answer:

1. Lack of security and access restrictions (harder to enforce).

2. No way to enforce transaction management (eg two-phase locking) for concurrent transactions.

3. Difficulty in updation because of structural dependence.

4. Data Redundancy and the associated problems like data inconsistency.

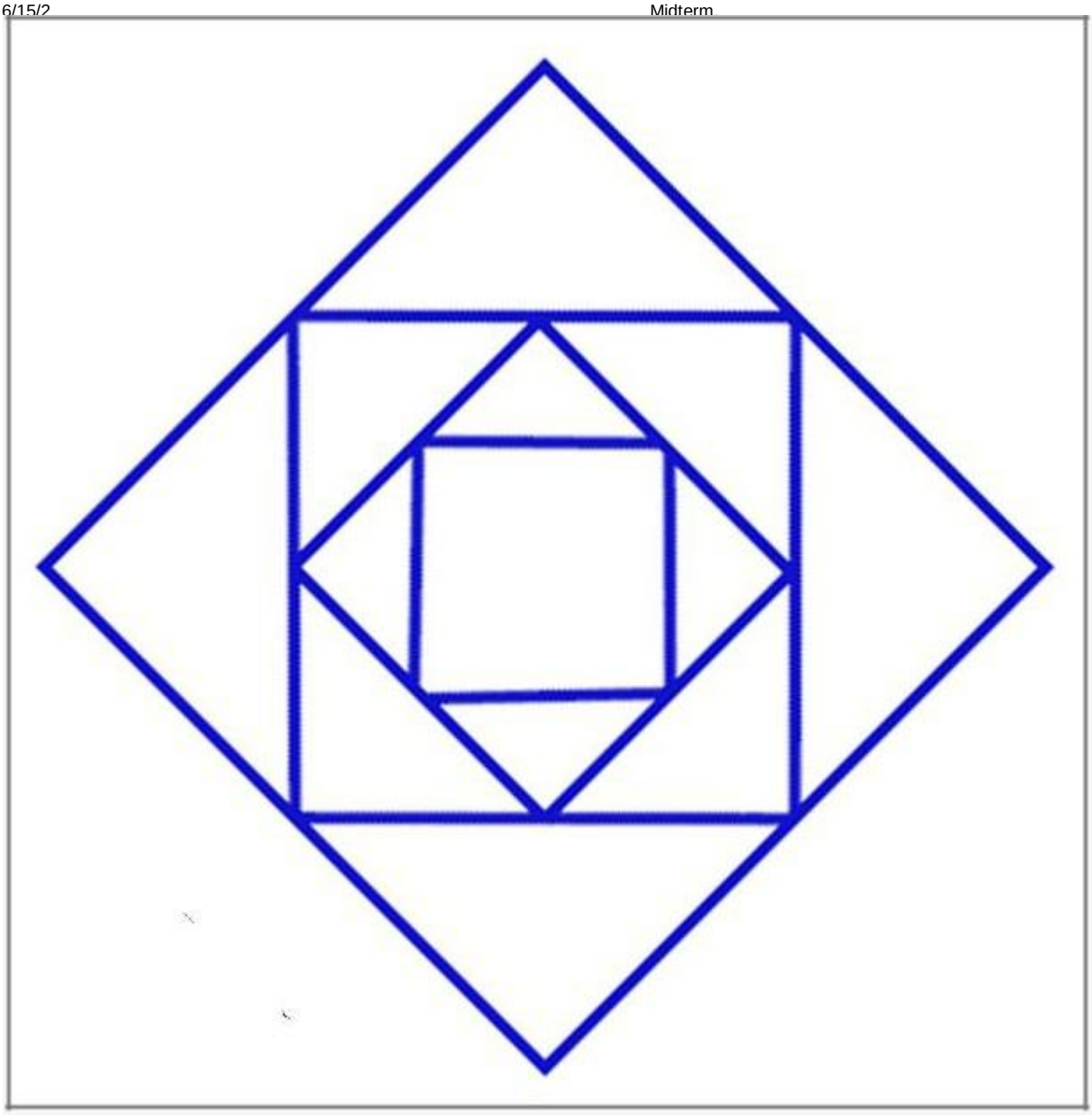
Rubric:

2 points each for any of the two points mentioned above or Any answer that points to RDBMS features (that are absent in file system DBs) is acceptable.

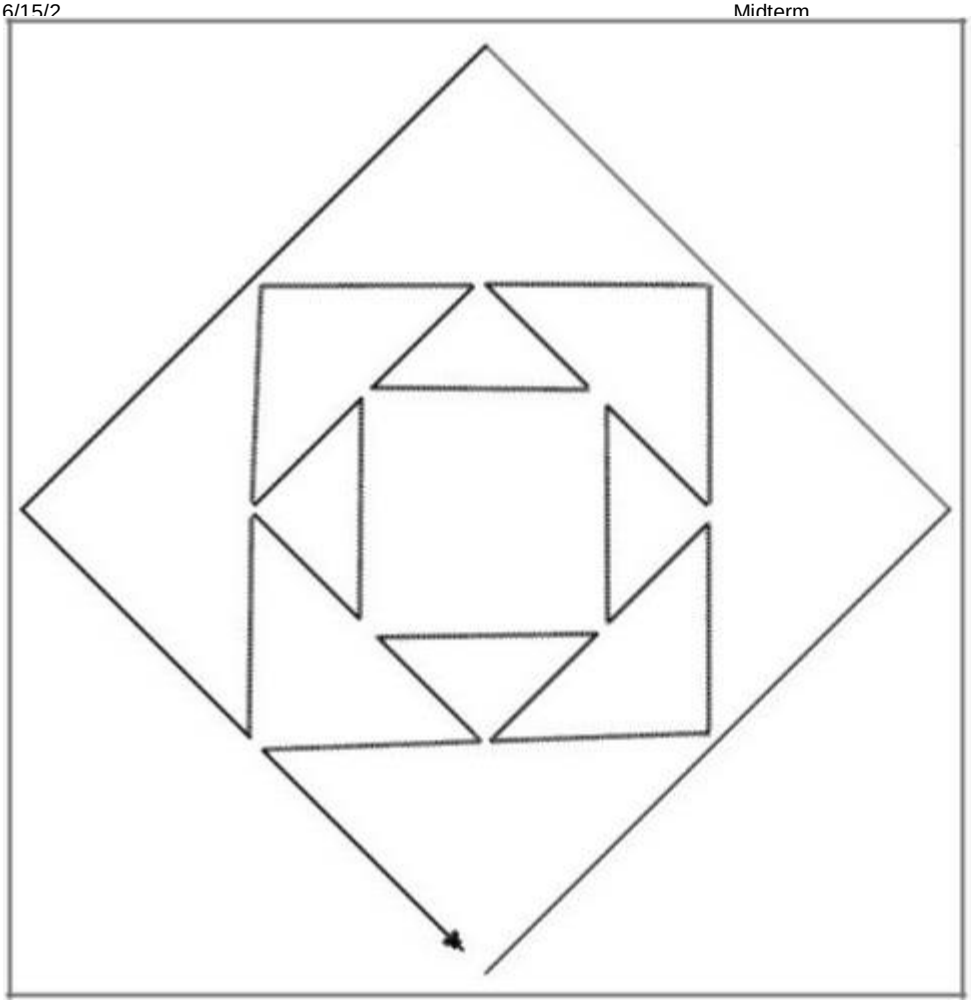
-2 if one of the problem doesn't point out to RDBMS features -1
if a problem partially mentions about the problems
0 if both (or more problems) don't mention the RDBMS features.

Bonus [1 point].

How would you draw the following blue figure using just a single, unbroken line where you don't draw over what you already drew? In other words, you can't lift the pen while drawing, and, you can't draw over even a part of an existing line.



Answer:



Rubric:
Full 1 mark or 0 for drawing the above figure correctly.

Midterm Rubrics:-

Q1 -

Q1 (5 points)

Q1 [2 + 3*1 = 5 points].

In the Powerball jackpot lottery, the winning ticket [where the winning amount can be quite high if there are no prior winners, eg. more than a half million dollars!] must match 6 numbers. There can be more than one winner [when different people end up having guessed the 6 winning numbers], in which case the prize money is split between them.

Using an analogy with DB keys, how would you characterize the numbers that people pick (ie the millions of lottery tickets sold), and, the winning number?

Pick three (online) sites you use, where (primary) keys are used – name the site, and indicate how it uses keys (ie for what purpose).

Answer part 1: any reasonable answer can be correct, we show an example interpretation as below

(1 point)
The number people pick will be stored in 6 different columns [**secondary keys - user table]** to query 6 **primary keys [lottery table]** to form 6 matched numbers as columns [**secondary keys**] in an information/winning table.
E.g. the tb_usr_input table as below (no need for students to write a table).

(1 point)
The winning number can be characterized as 6 values queried by 6 positional keys.
E.g. the value in the lottery number column of the tb_lottery table (no need for students to write a table).

Example: assume the lottery number is '037625'
Table 1: tb_usr_input

UID	Digit_1	Digit_2	...	Digit_6
6789	9	0	...	6
56830	2	9	...	5

Table 2: tb_lottery

Nth_Digit	Lottery_Number
0	0
1	3
2	7
3	6
4	2
5	5

Queried results:

PlayerID	Match1	Match2	Match3	Match4	Match5	Match6	Total	Match_Winner
6789	1	0	0	0	1	1	2	no
56830	1	1	1	1	1	1	6	yes

……

[Saty] This simpler answer is fine, too:
All the numbers form a set of **candidate keys**, loosely speaking (even if multiple people pick a number!). The winning number (even if multiple people pick it) is eqvt to a **primary key** - because it would be in the list of candidate keys if someone or a group of people picked it.

Answer part 2: Any reasonable answers work.
For example:
(1 point)
Amazon eCommerce website: primary keys are used to manage the account, create orders, track orders
(1 point)
Facebook: primary keys are used to define the user feed based on likes and dislikes, build a friend network - mutual friends
(1 point)
GSuite account: primary keys are used to identify the user and allow them to use/manage Google services like Gmail, Google Drive, Google Photos

[Saty] More examples: eBay auctions, LinkedIn profiles, bit.ly etc URL shorteners, YouTube video IDs, Medium.com article IDs, etc.

Q2:-

Q2 (5 points)

Q2 [5 points].

In SQL, in what sense are these similar: a natural join, and the 'EXISTS' command? Discuss, with examples.

1.(2 points) use English sentence to describe the similarity
Both natural join and “EXISTS” operator filter the common attributes on two tables.

2 (2 points) provide an example of tables,
Consider, the two tables Customer and Agent

Customer Table

CUS_CODE	CUS_NAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1231242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	37134	231

Agent Table

AGENT_CODE	AGENT_PHONE

125	615223211
167	615223299
231	615223200
333	615223288

3 (1 point) write correct SQL to show the similarity

Using natural JOIN,
SELECT * FROM Customer NATURAL JOIN Agent

Using EXISTS
SELECT * FROM Customer
WHERE EXISTS (SELECT * FROM Agent WHERE Customer.AGENT_CODE = Agent.AGENT_CODE)

Outcome: Both the queries will produce the same result.

Q3:-

Q3 (5 points)

Q3 [5 points].

Between two entities, a **strong** relationship, paradoxically, can lead to a **weak** entity :) Explain the meaning of 'strong' and 'weak' in this context.

Between two entities, a strong relationship can lead to a weak entity.

An entity is considered weak when its existence is dependent on another entity. In other words, it is weak when its primary key is completely or partially derived from another entity's primary key.

A strong relationship implies that the child entity has a primary key involving the parent's primary key.

Hence, Strong relationship between two entities would lead to a weak entity.

Rubric:

- -3 If only one part of the answer is correct.

- -5 if the explanation of both Strong relationship and weak entity is wrong.

Q4: -

Q4 (5 points)

Q4 [5 points].

When we 'mine' data for insights, we are looking for something new (the 'gold') the data can provide us. How could 'GROUP BY'(the SQL command) help with this? You can provide a general/oveview answer, NO need for code or an algorithm. Simply give it some thought, and write them down.

The GROUP BY Statement in SQL is used to aggregate identical data into groups and calculate simple descriptive statistics. It provides a way for classification of data based on particular attributes. It is very good at summarising, transforming, filtering, and a few other very essential data analysis tasks.

Also include any real world example that helps reaffirm the idea.

Rubric:

- -5 if GROUP BY description is wrong
- -3 if the relationship between GROUP BY and data mining is missing.
- -2 if only an example is described instead of general answer

[Saty] Mentioning that GROUP BY is in essence, ‘itemizing’ will also be a correct answer - by itemizing, we are able to ‘drill down’ into data.

Q5:-

Q5 (5 points)

Q5 [2+3 = 5 points].

Here is a (SQL) query ['sno' and 'sname' are number and name]:

```
SELECT sno, sname
FROM Suppliers
WHERE 100 > (SELECT SUM(quantity)
 FROM Shipments
 WHERE Shipments.sno = Suppliers.sno);
```

What is such a query called?

What does the above, do? Be specific, and, explain your answer.

1. Query with subquery [Saty: even just mentioning 'subquery' or 'inner query' is fine]
2. - Get the sum of quantity we call it as "Sum" under the condition that the number in shipments and in suppliers are equal.
   - Then we get the number and name from the Supplier table under the condition that "Sum" < 100 from shipments table.
   [Saty: in other words, the query finds the suppliers from whom we have ordered 100 units or less]

Q6:-

Q6 (5 points)

Q6 [1 + 2 + 2*1= 5 points].

What is the logic behind 2PC?

Explain how 2PC works, in your own words, and using a diagram.

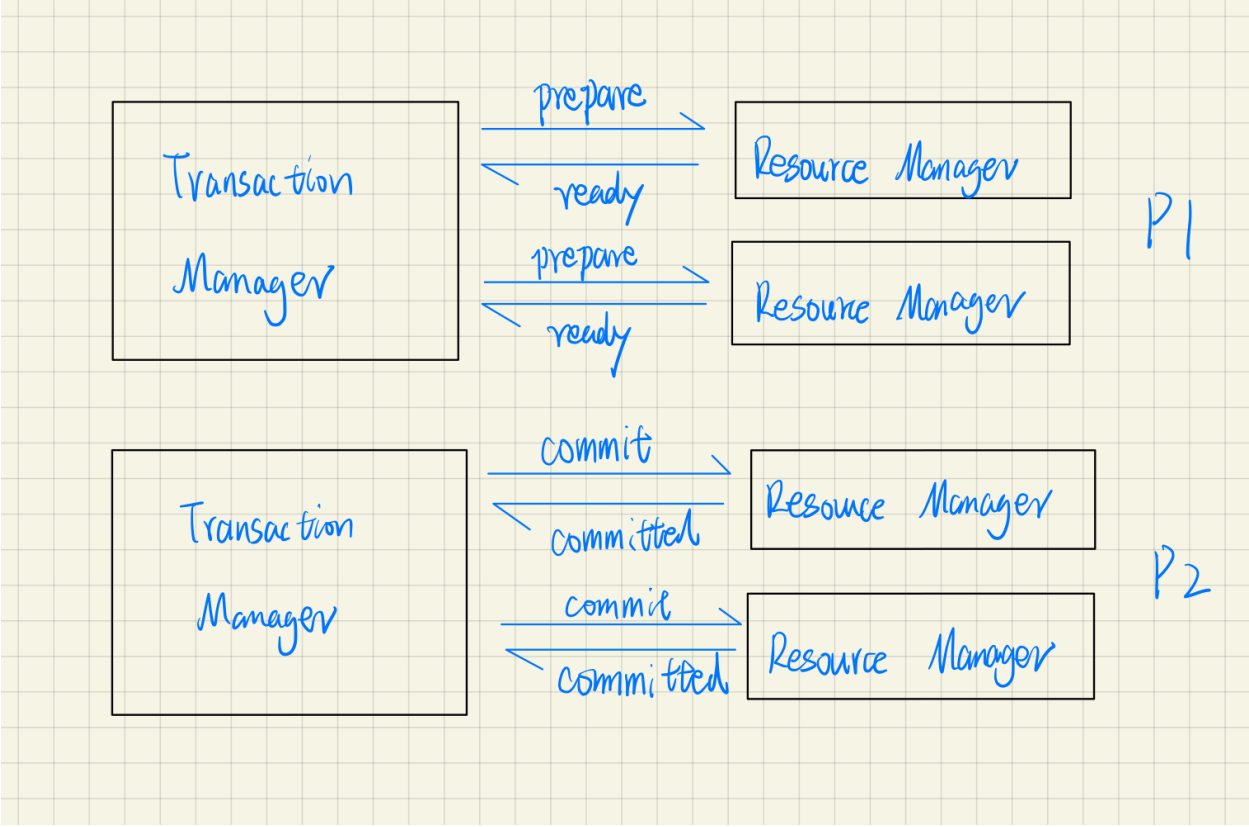
What two things can possibly go wrong (even) in 2PC?

The logic is to make the commit as 2 phases:

1. Preparation stage(voting)

2. Commit stage(execution)

A request comes through the coordinator, sends a ‘Prepare’ message to each participant, executes the local script but does not commit the transaction. If the coordinator receives the participant’s failure message or timeout, it directly sends a rollback message to each participant; otherwise, sends a commit message; the participant performs the commit or rollback operation according to the coordinator’s instructions, releasing all the resources occupied in the process of transaction processing. 2PC has achieved all operations either successfully or completely failed.



Shortcomings:

- The greatest disadvantage of the two-phase commit protocol is that it is a blocking protocol. If the coordinator fails permanently, some participants will never resolve their transaction. After a participant has sent an agreement message to the coordinator, it will block until a commit or rollback is received.
- There is inconsistency among data when there is network delay(jitter)

[Saty] The logic is that we minimize transaction failure by making sure first that a (distributed) transaction can indeed be carried out for real.

As for failures - the coordinator can fail, after sending out the first set of broadcast messages (query phase) but before, or while, sending out the commit messages. Also, a node can fail after responding with a ‘yes’, but before actually committing.

Q7:-

Q7 (5 points)

Q7 [5 points].

In the following three scenarios, there are two transactions T1 and T2, sequentially doing reading (R) and writing (W), on cells X and Y.

```
T2:R(X), T2:R(Y), T2:W(X), T1:R(X) ...
T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:R(X) ...
T2:R(X), T2:R(Y), T1:R(X), T1:R(Y), T1:W(X), T2:W(X) ...
```

In each, indicate what is problematic, in terms of what we covered in class.

How would the problems be fixed? Explain briefly.

---

7)

1st Scenario (write read conflict) - 1 pt
Dirty Read – If T2 for any reason fails to commit & is rolled back, T1 that has already read the value of X will have inconsistent data.

2nd Scenario (read write conflict) – 1 pt
Uncommitted Data – After T1 writes X, T2 has read the updated value. If T1 fails to commit and gets rolled back, T2 has read the wrong value and would be in an inconsistent state.

3rd Scenario (write write conflict) – 1 pt

Lost Update – Both T1 and T2 are trying to write a value of X, it is possible that one transaction overwrites the value of other as a result of which the updated value is lost.

Problems can be fixed by locking mechanisms such as 2PL locking protocol, exclusive locks, shared locks and acquiring all the locks before writing the data. – 2pt

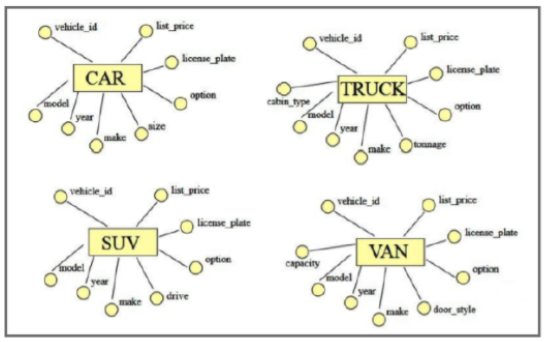
Q8:-

Q8 (5 points)

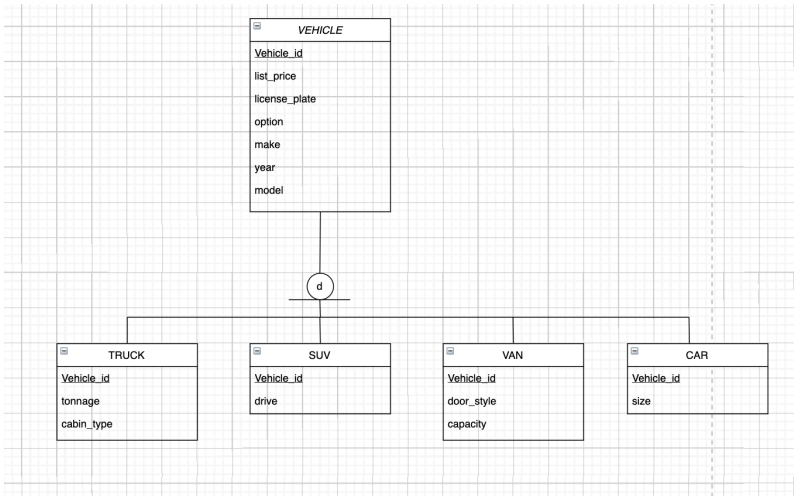
Q8 [4+1 = 5 points].

Shown below, in non-standard notation, are 4 entities. Draw an EER diagram to represent them. 'option' btw indicates one of 'owned' or 'leased'.

The concept of EER is analogous to what, in software development? Explain in a sentence or two.



A 'VEHICLE' (or transportation device etc) superclass, with all the common properties, with CAR, TRUCK, SUV, VAN being subclasses, each with its specific attributes. Vehicle_id is the PK.



EER diagrams are basically an expanded upon version of ER diagrams. EER models are helpful tools for designing databases with hierarchy relationships.

Rubrics: -1 if superclass is not correct.

-1 if missing subclass tables

-1 if attributes in tables are not correct.

-1 if no mention of PK.

-1 if the EER concept part is wrong.

Q9:-

Q9 (5 points)

Q9 [5 points].

Compare SQL with your favorite programming language (eg JS, Python, C++ etc), along 5 'dimensions' (aspects). You can point out similarities, as well as differences. Eg here is one: C++ has classes, SQL does not.

- 1 point for each similarity/difference in 5 unique dimensions/aspects
- No point for any similarity/difference that is essentially the same aspect (e.g. the following are all related to language elements, so 1 point should be awarded, not 3)
    - (Java has classes, SQL does not; Java has interfaces, SQL does not; Java has annotations; SQL does not)

- Use same language when comparing with SQL for all 5 points (if compared with different languages, take the maximum points for one language)
- No point if any similarity/difference is inaccurate in terms of SQL features e.g. SQL has threads.
- SQL features across flavours (e.g. MySQL, PostgreSQL) are OK.
- 0.5 points for PL/SQL specific features (e.g. Variables, For loops, Triggers) rather than standard SQL.

Possible similarities/differences:

1. Language Type
    - Lang is imperative, SQL is declarative / Lang is procedural, SQL is non-procedural
2. Language Features
    - Lang has operators, SQL has operators
3. Language Domain
    - Lang is generic domain, SQL is domain specific
4. Language Compilation
    - Lang is compiled, SQL is interpreted
5. Language Execution
    - Lang is executed independently, SQL is run on DB server

CSCI 585 Spring 2021 Midterm Rubrics

Q1. In joining four tables A, B, C, D (where we want to list all the columns from all the tables), **what** can you say, about the join order (ie order matters, etc)? Explain, with simple illustrations.
A1. Join order will NOT matter, for **inner** joins - the operations are commutative, associative.

Join order DOES matter, for **left and right** outer joins (not commutative); order does NOT matter, for **full** outer joins!

No need to mention commutation/association. But, need to mention the above three cases.

+2 Correct answer for inner join case (+1 if partially correct and/or is close to correct answer)
+1 Correct answer for left join case (+0.5 if partially correct and/or is close to correct answer)
+1 Correct answer for right join case (+0.5 if partially correct and/or is close to correct answer)
+2 Correct answer for full outer join case (+1 if partially correct and/or is close to correct answer)

Q2. **What** is a new table operation can you think of, that won't break closure?

What about a new operation that does break closure?

Your operations don't need to be useful in practice, but, they could be :)

Note - changing column names/types, adding or deleting columns or rows are not considered table operations, as you know [so those can't be your answers :)] - other than that, you can come up with ANY operations!
A2.
Examples of a new operation that will not break closure: uppercasing all strings, filling in null values with defaults, removing (filtering) rows with ‘too many’ nulls, filtering out rows that look ‘too similar’ to other rows..

Examples of an operation that will break closure: std_deviation() of a column, or variance(), finding geom mean, harmonic mean…

The above are not the only answers, more might be possible.

Note: Operation preserving closure is the one, which when performed gives the same type of dataset as the one it is acting upon allowing to chain multiple operations together, one over another.
+3 for table operation that doesn’t break closure
+3 for table operation that does break closure

Q3. In the 2PL algorithm we considered, a transaction can't start until it has acquired all the locks it needs (we call this, Conservative 2PL).

There is a different scheme possible, where a transaction does NOT need to wait for all its locks - it can start its transactions before all the locks have been acquired.

What would be good about such a scheme, and, **what would be bad**? Do feel free to illustrate with a diagram.
A3.
A transaction that can start even while locks are being acquired, will lead to **higher throughput** (less waiting). On the flip side, the problem is, such a transaction, after starting, **could hang - on account of deadlock** when waiting for the remainder of its locks - so it would need to be aborted and restarted.

+3 correct answer for good point about scheme (+1.5 if partially correct and/or is close to correct answer)
+3 correct answer for bad point about scheme (+1.5 if partially correct and/or is close to correct answer)

Q4. When a webserver is set up to serve data (in addition to documents and hypermedia, ie ‘usual’ HTTP content), the stereotypical way it does so, is by fetching data from a DB (eg. via SQL, from a relational DB).

What are two other ways (sources) using which the webserver can send data to its clients?

Discuss each, using a sentence or two - be sure to maintain their utility (ie. why they would be useful).
A4.
The webserver can **compute data** to send - random number distributions, synthetic data for ML augmentation, geometric models using submitted/default parameters, etc. The webserver can also **measure or collect data** to send, eg. temperature, a picture of the night sky via a telescope… Or a webserver can use services to **request data** from other servers. All these are non-DB lookups, there might be more answers.

Different data sources must be mentioned NOT other protocols to send data like FTP, SMTP etc.
+3 for mentioning two different data sources (1.5 x 2)
+3 for explanation for each of the above two data sources (1.5 x 2)

Q5. A teacher would like to track the status of various tests (eg. in English, Math, Physics, Design...) each student in her class needs to complete, over a period of a month. A test has multiple steps, the number of steps can be different for different subjects (eg English has 3 steps, Design has 5). A student can do the steps in any order, eg. 1,3,2 for the English test. When the student does a step, the teacher records the subject, step#, completion date, in a table like so (a separate table exists for each student):

![p1.png](https://usercontent.crowdmark.com/fd3f8a55-25c8-4525-8d6b-0b95834537c1.png)

At the end of the testing period, the teacher runs this query (for each student):

![p2.png](https://usercontent.crowdmark.com/bf7fce4b-442c-43c4-ad81-787e7ed77a3f.png)

What does the query produce? In a few sentences, explain what it does (ie. how it does what it does).
A5.
The query outputs **completed tests** - for which all steps have been done by the student and therefore entered into the table by the teacher.

It works by **outputting the complement** (via NOT EXISTS) of entries where there is a NULL completion date [the triplets stored being subject,step,date-completed], ie, all fully completed tests. And, it **eliminates duplicates** (which would exist because of the multiple steps), via SELECT DISTINCT.

+2 for correct answer of outputting completed tests
+2 for explaining the logic of NOT EXISTS
+2 for explaining duplicate elimination with SELECT DISTINCT

Q6. In addition to the fact that a spreadsheet being used as a ‘database’ looks plain UGLY (!) on account of repeating groups [blocks of empty cells], what are **five** ‘real’ problems/issues with this?
A6.
Insertion anomaly - a new entry might need to be made, just to put in partial data (eg a new project that hasn’t formally started yet)

Deletion anomaly - deleting a row would delete valuable data not stored elsewhere.

Modification (update) anomaly - possible to introduce a typo, or omit to update an entry while updating all others…

Cumbersome to extend, in the future.

Unnecessary disk usage.

Unnecessary memory usage, bandwidth usage.

…

Total 5 points with breakdown as follows:
+3 for correct answer of 3 anomalies (insert, delete, update) (need not specify exact term, similar explanations also fetch full marks)
+2 for two more problems/issues that are logically valid and make sense

CSCI 585 midterm exam: 6/9/22
Duration: 90+30 minutes

Please read the following items carefully, before starting the test:

• the exam is **open** books/notes/devices - feel free to look up whatever you want, from wherever! But don't look around 'too much', you'll lose time!

• from Q1 through Q10 (worth 5 points each), you can **choose any 7**, for a total of 35; if you want you can do **8, 9, or even, all 10** - we will ADD up ALL your scores (even partial scores) from **all** the questions you answer, then **CAP it at 35** - sick, right?!

• there are no 'trick' questions, or ones with long calculations or formulae or needing lengthy explanations, and there's certainly nothing to memorize [remember - it's all OPEN :)] BUT - this doesn't at all mean that the questions are trivial!

• please do answer carefully: in other words, **answer only WHAT IS ASKED**, otherwise you won't get points (eg. if a question is -ABOUT- 2PL, don't DESCRIBE/DEFINE 2PL!) - it's the quality (of your answer) that counts, not quantity (verbosity)... You can't demand points later, for providing non answers

• **please do **NOT** cheat**- this means NOT communicating with anyone via any device/medium/channel - you will get a 0, and be reported to SJACS, if you are found to have cheated; ANY attempt to get help from others in any form is a VIOLATION, as per , sections 11.11
https://policy.usc.edu/scampus-part-b/
through 11.14 [read it, if you are not familiar with it]

• when the time is up (90 minutes), stop your work, then spend the rest of time (30 minutes) on submission [students with DSP accommodations - your exam duration will be as per DSP determination] - **submitting pastthe deadline comes with a penalty**, because it is not fair to others if you go over when they don't

• note that you need to **submit each answer separately** (not all of them as a single PDF) - it's a Crowdmark req., it enables questions to be graded in parallel

Good luck! Hope you enjoy answering the questions, hope you find them to be easy, fun, thought-provoking.

Q1 [1*5 = 5 points].

Consider a typical web page, to be eqvt to a 'table row'. Such a table will have hundreds of billions of rows, given there are hundreds of billions of pages [this is just an observation - it's not a bad thing].

a. What would be the natural PK for such a table?

A. The URL.

b. What would be non-PK?

A. Page contents.

c. What would be FK?

A. Links!

d. How could referential integrity violation happen?

A. When pages' links point to a non-existent page (leading to '404 Not Found'!).

e. What might be a better PK?

A. A hashed (eg. using bit.ly) version of the URL, or a bookmark with manually shortened titles, etc.

Q2 [2+2+1 = 5 points].

a. Explain 'self join', using an example that we did not discuss in class.

A. Countries and neighbors (or states and neighbors, or counties and neighbors...), friends, book references...

b. What would be an example of a table that would have MULTIPLE (ie more than one) types of self-join? Note - this is NOT asking about recursive self joins!

A. People (employees) in a company - there can be FK columns for 'manager', 'spouse', 'neighbor', 'work friend', 'project partner', 'workout partner', etc.

c. Recursive self join is equivalent to what data structure?

A. Tree.

Q3 [1*5 = 5 points].

Let's play a game called,"I load data into a site (ie. into a DB connected to a site), you query it".

Eg. I (job seekers) load resumes into LinkedIn, you (employers) query it; I (stock market) load stock prices into stock sites, you (investors) query it.

Provide FIVE more such examples - look at your apps, sites you frequent, USC (!), etc. for tips/hints :) Express each in the same 'I..., you...' format like the above two samples.

A:

Ralph's product manager loads grocery item data into their DB, shoppers query them

Lyft/Uber riders load data into Lyft/Uber, drivers query it

eBay auctionable items get loaded onto eBay.com, bidders query them

USC Registrar and depts load course catalog into USC SOC, students query them

Twitter users load data (tweets) onto twitter.com, APIs/other users query them

... MANY more such examples are allowable answers

Q4 [1+1+1+1+1 = 5 points].

Below is a Scottish tartan pattern (called 'Cameron of Erracht', if you're curious!):

a. How does it 'relate' to data?

A. The pattern resembles a table, where each row/column intersection (cell) is a 'relation'.

b. You would get the same overall tartan pattern (roughly speaking) whether you weave the horizontal ("weft") threads over the vertical ("warp") threads, or vice versa. How does this equivalence (that the order doesn't matter) compare, when we restrict rows and columns in a table?In other words, what happens if we swap the order?
A. Whether we PROJECT some columns first, then SELECT from

them, or SELECT rows first and then PROJECT the columns we want, we would get the same result.

c. When order doesn't matter in a binary operation, what is that property called?

A. Commutative Property (or 'Commutation').

d. What is an example type where order does matter? A. Matrix

(for mult), vector (for cross product), number division...

e. Even there are two different restriction operators in relational algebra, there is only one in SQL, that does both. Why?

A. Because one is enough! The 'SELECT' keyword accepts column names (to PROJECT), and the WHERE condition (to SELECT).

Q5 [2+2+1 = 5 points].

a. What is a real-life ('RL') example of 2PL? Explain briefly (ie don't just name it).

A. For a project where resources are shared by multiple people (eg scissors, printers, flash drives, etc), oftentimes good wisdom dictates that we first gather all of what we want, THEN start the project, and proceed without getting stopped mid-way.

b. What is a(n) RL example of 2PC?

A. A 'coordinator' - eg. a purchasing coordinator, or a mob boss (!), or a dept. manager, etc.

c. In 2PC, the coordinator simultaneously (ie in parallel) polls the worker nodes for yes/no (ready to commit, or not). In a situation where node transaction failures are common, what might be the

advantage of polling them one by one, instead?

A. Once any node in the list replies with a 'no', we don't need to poll the rest - we can skip to Phase 2, and broadcast an 'ABORT' message to all the nodes.

Q6 [2+2+1= 5 points].

What is a design activity that is similar in principle, to table normalization? What factor do we consider, in doing this (ie. what is the basis)?

A. Class design (eg. in C++, Java, etc.). We use 'separation of concerns, 'or tight cohesion', as the guide - related members and methods would be housed in a single class.

How about EER - what other design activity resembles this? Again, what is the basis?

A. Class hierarchy (ie superclass/subclass) design. The basis is this: common members and methods would be moved 'up' to a common superclass, while each subclass will retain items (members, methods) that are specific to it.

Too much 'clean design' might not always be a good thing, with tables. How so?

Because it would lead to table fragmentations, hence, slow query execution that involves joining the numerous tables.

Q7 [1+1+3*1 = 5 points].

a. In our course context, what is 'connectivity'?

A. Connectivity is about connecting a 'backend' (DB or a data

generation 'service' in general) to a web server.
b. Why are there, SO MANY 'connectivity' technologies?

A. Because data is CENTRAL to so many operations that are carried out by lay people, businesses, companies, gov't organizations, educational institutions... Connecting data services to a web server, and from there to clients, is the most effective way to provide access to data.

c. Briefly explain THREE connectivity solutions, in a sentence or two for each: be sure to pick an 'older' one, a later (middle years, eg. late 90s) one, and a current one.

A. OBDC or JDBC..., CFML or cg-bin etc, microservices/Flask/Node... etc.

Q8 [1+1+1+2 = 5 points].

a. What is the following (called)?

A. Dataflow!

b. What is its alternative?

A. A script, where the ops are traditional function calls. c.

Why is this (what's shown) preferable to the alternative?

A. Because the graph can be selectively updated, saving vast amounts of computation (by not recomputing what can be reused).

d. EXPLAIN how it works (ie. what makes this better), using a small diagrammed example of your own.

A. A graph 'manager' tracks what nodes have completed running, what needs to run next, and runs them accordingly. That way, a change made in one node would lead the manager to mark affected/downstream nodes as 'dirty', and run just those (recursively). Your own diagram can be related to simple math operations, or image processing, or traffic analysis (ie TensorFlow), or a traditional data pipeline, etc, etc.

Q9 [1*5 = 5 points].

Compare SQL with C++, in terms of the following five programming language aspects, using a sentence or two for each: datatypes, operators, variables, function, class.

A.

Datatypes: C++ has int, short, byte, char, double, float, string... SQL has number, date, boolean, string (varchar2)

Operators: C++ has the usual ones (+ etc) plus &&, += etc; SQL has just + - * / ^ for operators.

Vars: C++ has them based on types (eg int i;); in SQL, columns can be considered vars, as can tables (because they can be named views)

Function: Both languages allow user-created ones; both come with built-in ones too (via 'libraries')

Class: in C++, we have the 'class' keyword; in SQL, the CREATE TABLE command serves as 'class definition'

Q10 [1+2*2 = 5 points].

a. What is Docker, **in your own words** (not a Googled result)?

A. Docker is a VM-lite of sorts - a few processes (a process tree) that serve as a contained OS ('containers'), to run applications ('images')

b. What are two advantages of using Docker (ie. why is it popular)? Again, explain each advantage in a sentence or two, in your own words.

A. Applications can be made to be self-contained and deployed as images, which are guaranteed to run on any host platform - there is no dependency on host services (eg the host does not need to have requisite libraries installed). Also, container instances can be trivially scaled, ie. multiple ones can be run in parallel.

CSCI-585 Spring 2023 Midterm Exam Rubrics

Q1 [3+2 = 5 points].
a. In an EER diagram, we could have overlapping subtypes, as you know. Assuming there are 3 subtypes A, B, C, what are **three** different ways of modeling such a situation? Illustrate by drawing tables.

Assume the following attributes for each entity:
i. Supertype S: S_ID (Primary key), S1, S2
ii. Subtype A: S_ID (Primary key, Foreign key), A1, A2
iii. Subtype B: S_ID (Primary key, Foreign key), B1, B2
iv. Subtype C: S_ID (Primary key, Foreign key), C1, C2

Solutions:
1. Universal table with mapping table

S_ID	S1	S2	Subtype_ID	A1	A2	B1	B2	C1	C2

Subtype_ID	Subtype_type
000	Unused
001	A only
010	B only
011	C only
100	A and B
101	B and C
110	A and C
111	A, B and C

There is a single table containing all attributes of all entities. A subtype column contains a binary string/integer referenced from a mapping table. Each tuple contains values only for the attributes that are common (S1, S2) and for the attributes that are specific to its subtype(s).
Pros: simple; easy to maintain if the number of subtypes is fixed and known beforehand.

Cons: highly inflexible, wastes storage due to null values.

2. Universal table only

S_ID	S1	S2	Subtype_A	Subtype_B	Subtype_C	A1	A2	B1	B2	C1	C2

Instead of a mapping table and the Subtype_ID column, there is a boolean attribute for each subtype. A value of 1 indicates that the tuple belongs to that subtype.
Pros: same as #1.
Cons: even more inflexible that #1.

3. Supertype table with separate subtype tables

S_ID	S1	S2	Subtype_ID

S_ID	A1	A2

S_ID	B1	B2

S_ID	C1	C2

Pros: highly flexible, modular, loosely coupled.
Cons: to view all the data for a given S_ID, you have to first retrieve its Subtype_ID and then query the concerned tables.

4. Subtype tables only

S_ID	S1	S2	A1	A2

S_ID	S1	S2	B1	B2

S_ID	S1	S2	C1	C2

Pros: one fewer table to maintain.
Cons: since there is no way to know the subtype, you have to query each table for a given S_ID. Redundant storage of common attributes.

Grading:
+ 1 pt for each unique method with tables
- 0.5 pt per method if only explanation provided without table
- 0.5 pt overall if any notation other than tables used
Students may have other solutions.
Explanation provided here is for understanding only.

b. When we talk about entity supertypes and subtypes in EER, we are making an analogy with a class hierarchy e.g. a C++ or Java one. But the analogy between a table and a typical class isn't quite accurate. Why not? And what would make them equivalent, conceptually speaking?

The analogy isn't accurate because:
i. There is no inheritance between supertypes and subtypes. It is simply a visualization of the human understanding of relationships between entities. Eventually, all entities are implemented as tables with perhaps a primary key-foreign key relationship between them.
ii. As a result of i, subtypes-supertypes are more tightly coupled than superclasses-subclasses. A subclass can be instantiated independent of the superclass. On the other hand, if a primary key-foreign key relationship exists between a supertype and a subtype, you cannot add a row in the latter without first adding one in the former.
iii. Class inheritance is a mechanism for code reuse and polymorphism, whereas supertype-subtype is a mechanism for modeling complex entities with shared characteristics. Inheritance allows a class to inherit the behavior of its parent class and also to define its own behavior. In contrast, supertype-subtype is used to model complex entities with shared attributes and relationships, where the subtype inherits the attributes and relationships of the supertype and may also have its own unique attributes and relationships.

iv. **A table can hold only data whereas a class may have methods/functions. Hence, you cannot implement function overloading and overriding in tables.**

Just mentioning iv. is sufficient :) - Saty

How to make them equivalent:
i. Let the subtype implementation inherit the supertype's attributes and thus allow the subtype to be an actual extension of the supertype.
ii. Add behavior to entities using methods/functions and allow these to be inherited as well.

Grading:
+ 0.5 pt for listing at least 1 difference
+ 0.5 pt for listing at least 1 way to overcome the difference
Let students discuss the difference and the solution in their own words as long as the answer is coherent and their understanding is correct.

Q2 [1+4 = 5 points]
a. What is the benefit of normalization, what is its drawback?
Normalization aims to improve the database structure in order to create an appropriate database design. The main goal of database normalization is to minimize data redundancies. By reducing redundancies in the database, data anomalies (e.g. insert/update anomalies) will be reduced as well.

Benefit & Drawbacks [Total 1 point]

The **benefits** of normalization include:

- Reducing data redundancy and inconsistency, which can improve data integrity and accuracy.
- Improving data consistency and maintainability, which can simplify database design and management.
- Reducing data storage requirements and improving query performance, which can save disk space and processing time.

The **drawbacks** of normalization include::

- Increasing the complexity of database design and management, which can make it harder to understand and maintain the database.
- **Reducing query performance for complex queries, which can slow down the response time of the database.**
- Limiting the flexibility of the database schema, which can make it harder to adapt to changing requirements.

Grading:
+ 0.5 pt for listing at least 1 benefit
+ 0.5 pt for listing at least 1 drawback

Please note it is an open ended question, so any of the benefits/drawbacks mentioned above and/or other are correct, if justified

*****

b. In a class of students, each student has an ID and a name. Each student is assigned (given) a book by a popular author to read; many students could be assigned the same book (e.g. many might be assigned to read 'The adventures of Tom Sawyer', by Mark Twain). The class teacher uses the spreadsheet to keep the track of the # of hours a student puts in, towards the reading her/his book.

Show using a table, how the teacher would store data incorrectly. Show how you would help fix the table. To save time when you answer, you can use 'simple' values A, B, C.. for your data [they don't need to be 'real']

How the teacher would store data incorrectly [Total 2 points]

Each row represents a student and their assigned book, along with the number of hours they have put in towards reading it. The columns represent the different attributes of the data, such as ID, Name, Book, Author, and Hours

Example 1 :

ID	Name	Book	Author	Hours
1	A	Book1	Author1	10
2	B	Book2	Author2	12
3	C	Book3	Author3	8
4	D	Book1	Author1	15
5	E	Book2	Author2	10

Example 2 :
This table is not normalized because it contains multiple attributes in repetition

Table : Books

ID	Book	Author
1	Book1	Author1
2	Book2	Author2
3	Book3	Author3
4	Book1	Author1
5	Book2	Author2

Table : Student and Books

ID	Name	Book
1	A	Book1
2	B	Book2
3	C	Book3
4	D	Book1
5	E	Book2

Table : Student and hours

ID	Name	Hours
1	A	10
2	B	12
3	C	8
4	D	15
5	E	10

Grading:

[Full points] +2 point for representing data in any of the above forms /denormalized form/ forms showing redundancy

[Partial points] +1 point for any valid explanation and justification of why they find data representation to be incorrect by the teacher.

-0.5 point for missing the concept of normalization/denormalization to represent data

-1 point for missing any fields or their explanation or incorrect justification

Show how you would help fix the table [2 points]

Example 1:

Table 1 represents the students in the class, with each row representing a different student and their ID (SID) and name. Table 2 represents the books assigned to the students, with each row representing a different book and its author. Table 3 represents the assignments of books to students, with each row representing a different student and book combination. Table 4 represents the number of hours each student has put in towards reading their assigned book, with each row representing a different student and the number of hours they have read.

Table 1: Students

SID	Name
1	A
2	B
3	C
4	D
5	E

Table 2: Books

Book	Author
Book1	Author1
Book2	Author2
Book3	Author3

Table 3:Assignments

SID	Book
1	Book1
2	Book2
3	Book3
4	Book1
5	Book2

Table4: Hours

SID	Hours
1	10
2	12
3	8
4	15
5	10

Example 2:

Table 1: Students

SID	Name
1	A
2	B
3	C
4	D
5	E

Table 2: Books and student

Book	Author	Name
Book1	Author1	A,D
Book2	Author2	B, E
Book3	Author3	C

Table4: Hours

Name	Hours
A	10
B	12
C	8
D	15
E	10

TO FIX IT, ALL WE NEED TO DO, IS STORE (SID, Book, Hours) - this is similar to the (EmployeeID, ProjectID, Hours) in the lecture slides. - Saty

Grading:

[Full points] +2 point for representing data in the above form/ normalized form

[Partial points]

+1.5 points for representing data in any other normalized form partially

+1 point for any valid explanation and justification of why they find data representation to be correct by the teacher.

+0.5 points for representing at least 1 table correctly

-1 point for missing any fields or their explanation or incorrect justification
-1 point for redundant data representation as it misses the entire point of normalization

Note : all other normalized forms can be given partial credits based on how close they are to the correct representations

*****

Q3: [1+2+2] = 5 points

> Q3 [1+2+2 = 5 points].
>
> a. What is an example of data that is suitable for a single-user DB? What is another example, for a special-purpose DB?
>
> b. Why is structural dependence a bad thing, when it comes to storing data? Illustrate structural dependence using a small example of your own [with some sample data].
>
> c. On the flip side (of structural dependence), we have layered data abstraction - what benefit does layering provide? Explain in two or three sentences (NOT more!).

a. Example of data for single-user DBs (0.5 pts):
Data related to confidential information; personal information and so on. **(also, contact list on our phones)**

Example of data for special-purpose DBs (0.5 pts):
Any data that have specific use cases: students' midterm grades, stock price, weather record of a local area, **and, molecule data, building info systems ("BIM"), CAD drawings' 'bill of materials' (BOM), etc.**

Grading:

1. One type of data for single-user DBs +0.5pts
2. One type of data for special purpose DBs +0.5pts
3. If only answer the examples of DBs, one can only get 0.5pts for this question.

*** pitfall: what is an example of **DATA** that is suitable for single-user/special-purpose DB.

b. Why is structural dependence a bad thing when it comes to storing data (1pt)?
When data is stored with structural dependence, the drawbacks are (at least) twofold: i) it may cause trouble for applications that want to add/delete/modify/query the data but do not know the structure of the data. ii) when the data structure changed, all the applications that used to customize to the old structures need to be modified.

Illustrate structural dependence using a small example of your own(1pt):
A list of data containing the name and the name and DOB of students. When we want to add another attribute of students to the list, e.g. address, all the applications that are

used to query the data list using the old order should be changed or they are not able to query the data correctly.

Alex
05/02
Bill
06/13
Chris
01/31

Grading:

1. Only one drawback +0.5pt; at least two drawbacks +1pt
2. Examples related to data structures +0.5 pt; explanations +0.5 pt

c. What benefit does layering provide (one point: 1pts; >= three points: 2pts)?

Development efficiency, flexibility; Data reusability; Database design flexibility.

Grading:

1. The first mentioned benefit +1pts
2. Each one more benefit +0.5pts.

Q4 (5 points)

Q4 [1+2+2 = 5 points].

a. In the 'story' of connectivity, about how it all started, leading to where we are today, what were key stops along the way, ie. what were milestones? You can simply list them, no need to elaborate.

b. How does data connectivity occur today, ie what is the dominant architecture? Explain in your own words, using your own diagram.

c. Briefly discuss two ways via which the web server was (is) augmented to serve data to the client.

a. Story of connectivity (1point)

Grading:

1. Provides a clear and concise list of key milestones in the history of connectivity, ranging from the invention of the telegraph to the impact of the COVID-19 pandemic

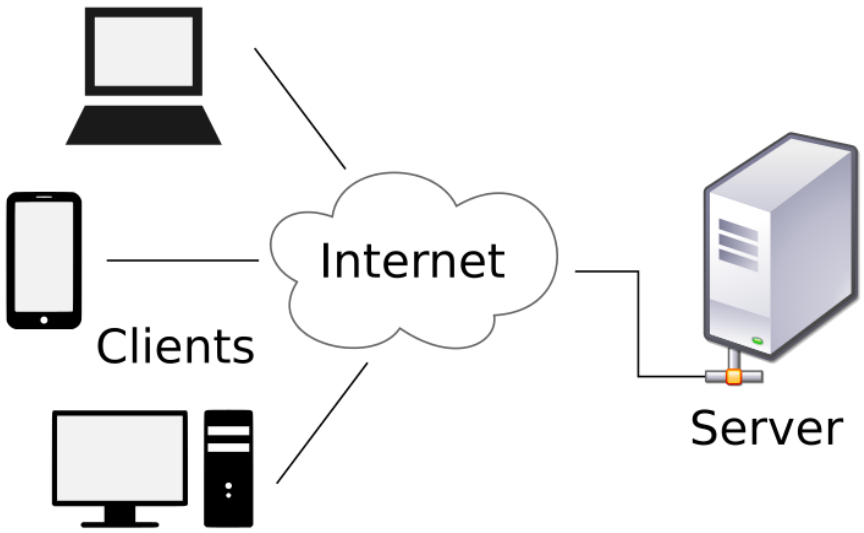
on remote communication technologies in 2020 (This is just an example, just the story line should match the history). **[+1point]**
Partial grading points: 0.5 pts

The story can start with MS and Sun providing ODBC and JDBC respectively, then continuing with server-side scripting (eg. 'cgi', CFML etc), with microservices+container+cloud being today's tech.

b. Architecture (2 point)

Grading:

1. Provides a clear and accurate description of the dominant architecture for data connectivity today, explaining how client devices communicate with servers over the internet to request and receive data. **[+1point]**
2. Includes a diagram that effectively illustrates the client-server architecture, demonstrating an understanding of how data connectivity occurs in modern applications and services. **[+1point]** (Attaching a sample architecture for reference, it might differ from the student's diagram)



If no diagram, give partial points for description [+0.5 points]
If no description, give partial points for diagram [+0.5 points]

The dominant architecture would be REST or GraphQL APIs, or even 'MCC' - microservices, container, cloud (where REST/GraphQL runs in containers on a cloud server).

c. Briefly discuss the two ways via which server serves data to client

The server can either send raw data (XML or JSON or plaintext etc) to the client, or, add markup to data and send HTML.

Grading:

1. Provides a clear and accurate description of two common methods for augmenting web servers to serve data to clients, (Example: Caching, and Compression). **[+1point]**
2. A Brief explanation of 1st method **[+0.5 points]**
3. A brief explanation of 2nd method **[+0.5 points]**
4. **Partial Points [+1 points]**: if no methods are specified but has explained the working

Q5 [4+1 = 5 points]
a. Pick any two apps/sites on your phone/tablet/laptop using which you access data, describe how your UI actions (eg. searching, or doing data filtering) might result in SQL, using one example for each app/site (so two examples total).

Example 1: Doordash
When a user searches for restaurants in a specific area, Doordash's UI would trigger an SQL query that filters all restaurants based on location from the database. For example, if a user searches for "Mexican food" in "San Francisco", the UI would trigger an SQL query such as:

SELECT * FROM restaurants WHERE cuisine = "Mexican" AND city = "San Francisco"

Example 2: Amazon
When a user filters a search result by a specific category or price range, Amazon's UI would trigger an SQL query that retrieves data from the database based on the filter criteria. For example, if a user searches for "smartphones" and filters by "under $500", the UI would trigger an SQL query such as:

SELECT * FROM products WHERE category = "smartphones" AND price < 500

Grading:
Example 1: (0.5 mark for stating the name of the app + 1.5 mark for description)
Example 2: (0.5 mark for stating the name of the app + 1.5 mark for description)

Please note that writing a SQL query is optional. Since this is an open-ended question, there could be many valid different apps that students may write (for example. Lyft, Uber, Instagram, Spotify, Airbnb, etc*).* ***Please consider all valid answers.***

b. Assuming (like in 'a' above) that your app-driven-querying does turn into SQL, where would such conversion (ie. transformation from UI-based query to SQL) occurs?

The conversion of a UI-based query to SQL occurs on the server-side of the app/site with the help of a web-to-database middleware. Once the user initiates a query through the UI, the request is sent to the middleware from the server, which processes the request and translates it into SQL. The middleware then sends the SQL query to the database, retrieves the relevant data, and then formats the data and sends it back to the server, which finally sends it to the UI for display. The web-to-database middleware acts as an intermediary between the server and the database, handling the translation of requests and queries between the two systems.

Grading:
+1 for any valid explanation. (Student must write either “backend server” or “web-to-database middleware” in the answer, -0.5 if not written)

Q6. [2+2+1 = 5 points]

a. In 2PL for data access during transactions, what is the most important phase? Explain.

   Sample Answer 1 based on Conservative/Static 2PL:
   Growing phase. A transaction acquires all the locks it needs to read and modify data items in the growing phase. Once a lock is acquired, it cannot be released until the transaction commits or aborts. This makes sure that the same data cannot be updated by another transaction while it is being read or modified by the current transaction.

   Sample Answer 2 based on 2PL:Locking phase.
   A transaction acquires all the locks on the data items it needs to access before making any modifications or performing any other operations on them in the locking phase. This makes sure that the transaction has exclusive access to the data items and prohibits concurrent modification of them by other transactions, which can result in data inconsistency and other problems.

   Grading:
   +1: For mentioning Locking phase/Growing phase/Phase 1
   +1: Valid explanation for the phase mentioned

*****

Alternately, a student might consider the EXECUTION phase to be the most important (the "pyramid plateau" I discussed in class, where we can think of 2PL as 3PL in fact: lock acquisition, EXECUTION, lock release) - this is ok, too.

b. In 2PL, when we release locks, if we release locks prematurely, what issue might that cause? How would we fix it?

Sample Answer:
Issues: Data Inconsistency, Incorrect Reads, Lost Update, Rollbacks, Inconsistent retrievals

Prevention: Release locks only after transaction fully finishes.

Grading:
+1: Identifying at least one correct issue as mentioned
+1: Valid technique for prevention

*****

c. What issue might arise, when we do TM without locks? How would we fix it?

Sample Answer:
Issue: Dirty Read or Data Inconsistencies

Prevention:

- Use Concurrency Control Techniques.
- Use locking mechanism: Shared and Exclusive Locks
- Use Recovery Managers

A different answer: there might be data corruption on account of different transactions overwriting cell values; we would fix it by analyzing the log files that record transactions, identify errors and manually rolling back the offending transactions. (Background, for the graders: T.M without locks is only rarely practiced, when we are sure that such collisions will be extremely rare).

Grading:
+0.5: Identifying at least one correct issue as mentioned
+0.5: Valid technique for prevention

*****

Q7(1+4 = 5 points)

a.Codd's relational operators for data processing, lead to 'closure'. Why is this advantageous? Illustrate.

Sample Answer:

The advantage of closure is that it allows for efficient and effective data processing.
In addition, closure ensures that the data stored in the database is consistent and free of redundancy. Since the relational operators are based on set theory, they ensure that each relation contains only unique and relevant data. This eliminates the need for duplicate data storage and reduces the risk of inconsistencies and errors in the data.
Overall, the use of relational algebra and closure leads to more efficient and effective data processing, improved query performance, and better data consistency and integrity. It is therefore a fundamental concept in the design and implementation of relational databases.

Better answer: closure permits CHAINING of operations, ie. creating a tree (dataflow) of SQL operations.

Grading:

Keywords included + 1 (other similar keywords may also be acceptable)
Keywords: **Efficient, flexible**
Integrity and **Consistency** (offers a standard way).
Simplify the design and maintenance of the database.
Mentioning operation 'chaining' (or equivalent) is sufficient.

b. For these four data types, list an operation that does preserve closure, and one that does not: vector (e.g. with components x,y,z), matrix, complex number, color (with RGB components).

Sample Answer:

(1 point) 1. Vector:
(0.5point) Preserve: Vector addition, vector cross product keeps closure R^3 -> R^3
(0.5point) Not preserve: Dot product, doesn't keep closure R^3->R^1
(1 point) 2. Matrix:
(0.5point) Preserve: matrix-scalar multiplication, matrix addition etc.
(0.5point) Not preserve:
e1.matrix-matrix multiplication, it changes the matrix's dimension (only square matrix is an exception.)
e2. MaxPooling, or finding the determinant (which produces a scalar)
(1 point) 3. Complex Number:
(0.5point) Preserve: operations like conjugation, addition etc.
(0.5point) Not Preserve:
e1. Modulus
e2. Defined operation x->(x, x*2) from C-> C^2
(1 point) 4. Color:
(0.5point) Preserve:
brightness change, contrast change, color<->gray

Operations remaining in the same color channel domain(0,255) are also acceptable. E.g. color value/255, color normalization. From 0,255->0,1

(0.5point) Not Preserve:

Computing luminance [0.7*R+0.2*G+0.1*B] returns a float

Grading:

Other operations are acceptable. Please note operation is some form of computation; it can be an operator symbol or can be a function. 'Closure' means that the operation will output the same type as input.

Thus, any operation that makes results remain in the same domain as input could be regarded as a 'closure' operation and vice-versa.

Q8 [2+2+1 = 5 points]

1. During 2PC in distributed transactions, the transaction coordinator might fail. How would we fix that?
2. During 2PC, a non-coordinator site might fail partway (between phase 1 and phase 2) - how would we fix the problem (ie prevent bad transactions)?
3. During 2PC, a non-coordinator site might fail at the start (before phase 1) - how would we deal with that?

Sample Answer:

1. We can set up a **backup coordinator** or order a non-coordinator site (one that connects to every site) to work as a backup coordinator when the original one fails. The backup coordinator should have **access to the same transaction logs** and other relevant information as the original coordinator, so that it can continue the 2PC process from where the failed coordinator left off. If the coordinator falls between phase 1 and phase 2, none of the non-coordinator machines will get a COMMIT or ABORT message. After a **timeout period**, the non-coordinators will notify the designated backup, which will take over the reminder of the task
2. We can use a **timeout mechanism**: after a certain time not hearing back from a site, the timeout mechanism is called and the coordinator auto marks the transaction as failed and **reschedules** the tasks so this failed site will not be used in the next time.
3. Before the start of any transaction, we can set a protocol that for the coordinator to **send a message** to all the non-coordinator sites, simply ask a message back to the coordinator. If any of the sites is not responsive after a certain amount of time, the coordinator can mark that site as failed and schedule the tasks so this failed site will not be used in the next time.

Grading:

1. (2 points total) **+1:** if mension mechanism of "Backup coordinator" or "a non-coordinator site function as the coordinator"; **+1:** for the detail of the mechanism, including "back-up

coordinator should have the access to the log file", or "if fail down happened between phase 1 and phase 2, a timeout mechanism will be used", or other reasonable detail.
2. (2 points total) **+1:** if mension mechanism of "timeout"; **+1:** for the detail and explanation of how the mechanism works.
3. (1 point total) **+1** if mension sends a message to all sites before any transaction starts.

Other reasonable answers are okay, 1 and 2 require a brief explanation to get full points.

Q9. [1+2+2=5 points]

a) What are a couple of uses for 'computed columns' ? [1+2+2=5 points]

Computed columns are virtual columns in a database table that derive their values from expressions or functions that operate on other columns in the same table. Here are a few uses of computed columns:

1. Calculation of values: Computed columns can be used to calculate values that are based on other columns in the table. For example, you could use a computed column to calculate the total price of an order by multiplying the quantity ordered by the unit price.

2. Data transformation: Computed columns can also be used to transform data from one format to another. For example, you could use a computed column to format a date column in a specific way, or to concatenate columns into a single string.

3. Data validation: Computed columns can be used to enforce data validation rules. For example, you could use a computed column to ensure that the values in a column meet certain criteria, such as being within a certain range.

4. Indexing: Computed columns can be used as part of an index to improve query performance. For example, you could create an index on a computed column that combines the values of several columns, so that queries that use those columns will be more efficient.

Overall, computed columns can help simplify data management and improve data integrity by reducing the need for manual data manipulation and improving data consistency.

Grading:
+1: If at least one among the following are mentioned correctly.

b) Given a table with columns of sines and cosines (for 0 to 360 degrees, in increments of 1 degree), eg called COS and SIN, how would you verify the following formula/ identity?

To verify sin^2 Q+ cos^2Q=1 form the given table with attributes SIN and COS,

- Perform (SIN*SIN) +(COS*COS) and add the value to new column say result. If the value is 1 then return True else return False.
- Sample code: ALTER TABLE myTable ADD result AS (sin* sin + cos * cos)
- This will add a new computed column named `result` to the table `myTable`. The computed column will contain the result of the calculation sin^2 + cos^2 for each row in the table.
- You can then query the `result` column to retrieve the calculated values: SELECT result FROM myTable.

Grading:
+2: If similar steps are written. (Either theoretically or using SQL command).

c) Given a table with a pair of columns called X and Y, containing (x,y) values from a scatterplot for example, how would you calculate the (Pearson) correlation coefficient? Again, just describe the steps [no need for SQL].

(+1 point)

Here's a step-by-step explanation of how to calculate the Pearson correlation coefficient between two columns X and Y:

1. Calculate the mean of each column: Compute the mean (average) of each column by adding up the values in each column and dividing by the number of values (n).

Mean of X: $\mu_x = \Sigma(x_i) / n$ Mean of Y: $\mu_y = \Sigma(y_i) / n$

2. Compute the deviations from the mean: For each value in the columns X and Y, subtract the mean of the respective column to get the deviation.

Deviation of X: $x_i - \mu_x$ Deviation of Y: $y_i - \mu_y$

3. Calculate the product of the deviations: Multiply the deviations for each corresponding pair of values from columns X and Y.

Product of deviations: $(x_i - \mu_x)(y_i - \mu_y)$

4. Sum the products of the deviations: Add up the products of the deviations calculated in the previous step.

$\Sigma[(x_i - \mu_x)(y_i - \mu_y)]$

5. Calculate the sum of squared deviations for each column: Square the deviations for each value in columns X and Y, and then sum them up.

$\Sigma(x_i - \mu_x)^2$ and $\Sigma(y_i - \mu_y)^2$

6. Calculate the Pearson correlation coefficient: Divide the sum of the products of the deviations (step 4) by the square root of the product of the sum of squared deviations for X and Y (step 5).

(+1 point)

$\text{Pearson's } r = \Sigma[(x_i - \mu_x)(y_i - \mu_y)] / \sqrt{[\Sigma(x_i - \mu_x)^2 * \Sigma(y_i - \mu_y)^2]}$

That's it! The result, Pearson's r, will be a value between -1 and 1, indicating the strength and direction of the linear relationship between the variables in columns X and Y.

Grading:
+1: If the parameters required to measure correlation coefficient are similarly represented.
+1: If the Pearson's correlation coefficient is represented correctly.

CSCI-585 Summer 2023
Midterm 1 Rubrics

Q1.

- ➔ An entity declaration for storing data, is loosely like a class definition, in a coding language - in what sense?
- ➔ But the definition isn't quite analogous - what is missing?
- ➔ What would be the advantage in making entities, entirely analogous to classes?

Ans.

- Entities have data fields and classes also have data fields (1 pt)
- Classes have methods/operations associated with them. However, Data Entities do not. (1 pt)
- The advantage would be that we'd be able to define behaviors/operations on the stored data as well. This means we could call methods directly on data, eg. price.placeOnSale() (1 pt)

Q2.

- ➔ SQL resembles a traditional coding language such as C/C++, JS, Python. How?
- ➔ SQL is not a 'full-blown' language though. Why not?
- ➔ Given that so many types of modern data (eg. your Spotify playlist) isn't stored in tables, why does SQL continue to be relevant?

Ans.

- Syntactic and syntaxes define instructions (1 pt) There are commands, built-in functions, operators, expressions, etc.
- Does not support logical branching (1 pt) Also, no variables, looping, class definition, etc.
- The syntax is known by lots of developers, and is highly expressive/capable [comparable queries become more complex to express, in traditional coding languages] (1 pt)

Q3.

- ➔ In pseudocode form, how would you express SELECT ... FROM ... WHERE..., when it comes to analyzing data in a table?
- ➔ How would you express via pseudocode, the Cartesian product operation on two tables?
- ➔ What would be pseudocode for a classic JOIN operation that combines data from two tables?

Ans. We can accept any pseudocode, however loose, as long as it captures the logic behind the operations.

- Here is an example:

```
function selectFromWhere(table, columns, condition):
    result = empty table
    for each row in table:
        if condition(row):
            selectedRow = createRowWithSelectedColumns(row, columns)
            result.addRow(selectedRow)
    return result
```

(Grading: 0.5 given to the table traverse part, 0.5 given to the correctness of the rest)

- **Here is an example:**

```
function cartesianProduct(table1, table2):
    cartesianResult = empty table
    for each row1 in table1:
        for each row2 in table2:
            cartesianRow = concatenate(row1, row2)
            cartesianResult.addRow(cartesianRow)
    return cartesianResult
```

(Grading: 0.5 given to the row combination part, 0.5 given to the correctness of the rest)

- Here is an example:

```
function joinTables(table1, table2, joinColumn):
    result = empty table
    for each row1 in table1:
        for each row2 in table2:
            if row1[joinColumn] == row2[joinColumn]:
                joinedRow = concatenate(row1, row2)
                result.addRow(joinedRow)
    return result
```

(Grading: 0.5 given to the data combination part, 0.5 given to the correctness of the rest)

Q4.

➔ What do we gain, by carrying out normalization? The answer is NOT about describing 1NF, 2NF etc!!

➔ Normalization involves steps: 0NF -> 1NF, etc. What is the importance of (need for) these explicit steps (ie. why not skip them)?

➔ How does data normalization relate to classic software development [what similarities exist]?

Ans.

- **Any two (0.5 point each) of the following:**
  - Reduce redundancy
  - Reduce anomalies
  - Improve data consistency across the dataset
  - Improve storage efficiency
  - Improve query efficiency
  - Improve flexibility and scalability
- The systematic steps ensure that we don't accidentally overlook a partial or a transitive dependency.
- **Any one (1.0 point) of the following (it's about 'separation of concerns', ie abstraction/modularization, and loose coupling+tight binding; in other words, it's about minimizing duplication/redundancy):**
  - Both are systematic approaches
  - Both utilize modular design
  - Both utilize abstraction and encapsulation
  - Both require maintainability and extensibility
  - Both require data integrity and consistency
  - Both for performance optimization
  - Or any other explanation that make sense for both

Q5.

➔ What exactly is the problem, when we do distributed data processing with a coordinator?

➔ What alternate mechanism [other than the use of a coordinator] would you propose, for fixing the problem?

➔ What might be the problem with your alternative proposal?

Ans.

- **Any one (1.0 point) of the following:**
  - Single point of failure
  - Communication overhead
  - Coordination bottleneck
- The answer depends on the answer to the previous question. A sample answer: decentralized coordination or distributed consensus protocols. **(1.0 point for any correct solution) [eg. each node would wait to hear directly from all other nodes - this leads to excessive communication]**
- The answer depends on the answer to the previous question. A sample answer: complexity, excessive transmission/bandwidth utilization. **(1.0 point for any correct problem pointed out)**

Q6.

Windows (OS) is pretty widely deployed in the world.
Imagine a small company that has been MS Windows since the mid-1980s. It continues to use 80s era DBs such as Access and Paradox, to store valuable company data.

- ➔ What is the danger of doing so [continuing to use Access etc]?
- ➔ How would you help the company 'rescue' the data that resides in Access etc [so that Access and friends can be retired from continuing use by the company]? Be specific - describe your approach.

Ans.

- The answer doesn't need to be exactly the same, but should explain the principles in a meaningful and correct way. **(1.0 point)**: Here is a list of correct answers:
  - Security Vulnerabilities
  - Limited Scalability and Performance
  - Data Incompatibility and Integration Challenges
  - Etc.
  - The product might become discontinued (no more bug fixes, features, support); people who know how to use it would be hard to find
- The answer doesn't need to be exactly the same, but should give a correct answer based on the challenge the student pointed out above. **(1.0 point) The solution is to use ADO.NET to retrieve entire tables (with SELECT * FROM <table>), and save them as XML files on disk (text-based, easy to parse and transform to JSON or csv, ingest into modern DBs).**

Q7.

- ➔ What is data?

Ans.

- **Data refers to CHARACTERISTICS we select/define/specify, for an entity.** Answers such as 'raw facts', 'information' etc are useless (they don't explain anything) and are not acceptable.

Q8.

- ➔ What is data modeling?

Ans.

- From the slides: Iterative and progressive process of creating a specific data model for a determined problem domain (e.g., an application)
- **It is the process of selecting the appropriate STRUCTURE for the given data (eg. hierarchy/tree, network/graph, tables etc).**

Q9.

➔ What three core principles did you learn, related to database modeling/design, ie. when it comes to storing and analyzing data? Explain each in a few lines.

Ans.

- The answer doesn't need to be exactly the same, but should explain the principles in a meaningful and correct way. **(1.0 point each, 0.5 point for the principle, 0.5 for correct explanation)** Here is a list of potential answers:
  - Data Integrity
  - Normalization
  - Scalability
  - Consistency
  - Avoiding needless redundancy
  - Correct Abstraction/model (eg E-R)
  - Each "row" (data object, ie collection of columns and values) needs to be made unique, eg via a primary key or rowkey
  - Etc.

Q10.

In the course of DB design, we moved away from file systems, preferring to store data in tables, etc.

➔ But, if you were to make a case to use a text-based file system for a modern app you create (where you would store data as plaintext, ie ASCII), what three distinct advantages would you list? Explain each, in a line or two.

Ans.

- The answer doesn't need to be exactly the same, but should explain the advantages in a meaningful and correct way. **(1.0 point each, 0.5 point for the advantage, 0.5 for correct explanation)** Here is a list of potential answers:
  - Easier Version Control
  - Customized query formats
  - Easy Reusability/Portability
  - **Might be faster/easier to parse**
  - **Can store it on disk compactly, eg using a custom compression scheme**
  - **Easily editable to make small changes**

Q11.

➔ What mechanism exists in SQL, to create your own functions (commands)?

➔ Often we analyze two columns of data together, eg. 'weight' and 'blood pressure' of patients. What two functions would you consider creating, to do such analysis? Name them, and explain what they would do, and return.

Ans.

- User Defined Functions **(1.0 point)**
- **correlation(), regression(); correlation() would measure, on a -1 to 1 scale, the correlation between the columns, while regression() would fit an equation (eg a line, with a specific slope and intercept)**
  [the specific correct answer is above - vague descriptions are NOT acceptable!]

Q12.

Pick two apps you use on your smartphone - for each, describe in pseudocode-SQL, what data-oriented queries you generate as you go about using your app. For each, describe:

- ➔ what action you are carrying out (ie what you are 'making the app do')
- ➔ what SQL might result from your action [this does not need to be syntactically-valid, or complete, SQL code]

Ans.

- The answer depends on what students choose. The keywords don't need to be the same as SQL, but the queries should be correct **(0.5 point)** and unambiguous **(0.5 point)**. **In other words, the actions (eg. looking for someone on LinkedIn), and the corresponding pseudo-SQL, need to make sense (plausible, correct).**

Q13.

- ➔ Summarize in a few lines, the contents of the 'Data modeling' and 'E-E-R diagrams' lectures.

Ans.

- **Data modeling**: Data modeling is an iterative and progressive process of creating a specific data model, which is a simple representation of complex real-world data structures, for a determined problem domain. (Chapter: Data modeling) The answer should explain **abstraction of real world facts** in a meaningful way. **(1.0 point)**
- **EER**: The answer should explain the difference between EER and the original ER diagram in a meaningful way, especially about **specialization and constraints (NEED to mention disjoint vs overlapping constraints, and partial vs total completeness)**. **(1.0 point) [**

Q14.

- ➔ The homeworks in this course are specifically designed to provide hands-on practice with handling data. With that in mind, summarize the intent ("point") of HW1, HW2.

Ans.

- HW1 is about designing an ER diagram, which is the designing phase of a database. The answer should point out "**designing (relational) database**" or the equivalents. **(1.0 point) [it is about succinctly capturing the structure of a database in a single**

diagram that is comprised of entities/tables [collections of related columns] and the relationships (1:1, 1:M, M:N) between them].

- HW2 is about using queries to pull data from a database. The answer should point out **“using database/ query data from database”** or the equivalents. **(1.0 point) [it is about using SQL to query the data held in tables].**

Q15.

➔ The IMO specifies that ships transmit AIS data that contain several pieces ("columns"), eg. ship's identity. What two other pieces of data would you add to the existing list of data requirements?

Ans.

- Departure terminal
- SOS data
- Ship’s dimensions
- Carrying capacity [cargo, humans]
- …