Faculty of Science
COMP-421 - Database Systems (Winter 2016)
Final Examination

April 20, 2016 Examiner: Bettina Kemme
18:00 - 21:00 Associate Examiner: Muthucumaru Maheswaran

Student Family Name:	**Student First Name:**
Student Number:	

Instructions:

- You should have received this exam paper, a scantron sheet, and an exam booklet for the answers of the open questions.
- **DO NOT TURN THIS PAGE UNTIL INSTRUCTED**
- **INDICATE THE VERSION-0 ON THE SCANTRON!!!**
- **WRITE YOUR NAME AND STUDENT ID ON THE SCANTRON.**
- **WRITE YOUR NAME AND STUDENT ID ON THIS FIRST PAGE OF THIS EXAM PAPER**
- **You may split apart this exam paper, for example, to make it easier to read the background information about the example application. But you MUST WRITE YOUR NAME AND STUDENT ID on each of the separated sheets.**
- You have to return the scantron, ALL pages of this exam paper as well as the exam booklet.
- This is a **closed book** examination; only eight letter-sized (8.5" by 11") **crib sheets** are permitted. This crib sheet can be single or double-sided; it can be handwritten or typed. Non-electronic translation dictionaries are permitted, but instructors and invigilators reserve the right to inspect them at any time during the examination.
- Additionally, only writing implements (pens, pencils, erasers, pencil sharpeners, etc.) and a simple calculator are allowed. The possession of any other tools or devices is prohibited.
- Answer **all** multiple choice questions on the scantron sheet.
- Answer open questions into the exam booklet.
- This exam paper has **17** pages including this cover page, and is printed on both sides of the paper.

- The Examination Security Monitor Program detects pairs of students with unusually similar answer patterns on mulitple-choice exams. Data generated by this program can be used as admissible evidence, either to initiate or corroborate an investigation or a charge of cheating under Section 16 of the Code of Student Conduct and Disciplinary Procedures.

Scoring

The exam is out of 137 points distributed as follows:

1. Section 1 (multiple choice with multiple answers): 8 questions; each 4 points for a total of 32 points
2. Section 2 (true/false questions): 10 questions; each 1.5 points for a total of 15 points
3. Section 3 (Modelling and Queries): 6 questions for a total of 35 points
4. Section 4 (Query Evaluation): 3 questions for a total of 35 points
5. Section 5 (Transactions): 4 questions for a total of 20 points

Version-0

This page is kept intentionally blank.

Application Description for this Exam

A good part of the questions on this exam is related to the following application.
As part of McGill's community outreach initiative, the School of Computer science is helping local *mom-N-pop* entrepreneurs to improve their business by empowering them with a modern technological platform. "Fleuriste Dora" is a local florist shop in the Montreal downtown region which will benefit of this program. The plan is to setup an online shop that can be used to place orders, thus increasing the customer base as well as providing convenience. As building this online platform requires familiarity in different software technologies, a COMP 421 team has volunteered with the data modelling and database setup for this project. The following provides a specification of the requirements. It is kept short and does not contain all the information depicted in the following E/R diagram. In particular, content covered in obvious attributes is omitted in this description.

- Dora's flower shop sells flowers only as Bouquets. Each bouquet is given a unique name (e.g. "Floral embrace", "Blooming love", etc.) by Dora herself. Every bouquet also has an associated price with it and a general description about it. Additionally the website team requires that the database maintains the name of an image file depicting the bouquet. They need it to find the image file when displaying the bouquet on the website. Each bouquet is made of specific set/number of flowers (e.g. a dozen red roses and four pink daisy poms); thus, Dora wants to keep track of the name and colours (e.g. "daisy pom", "pink") and quantities (e.g. 5) of flowers required for each bouquet. Bouquets can belong to optionally many categories (for example "Birthday", "Valentine's day", "Spring Blossoms" etc.)

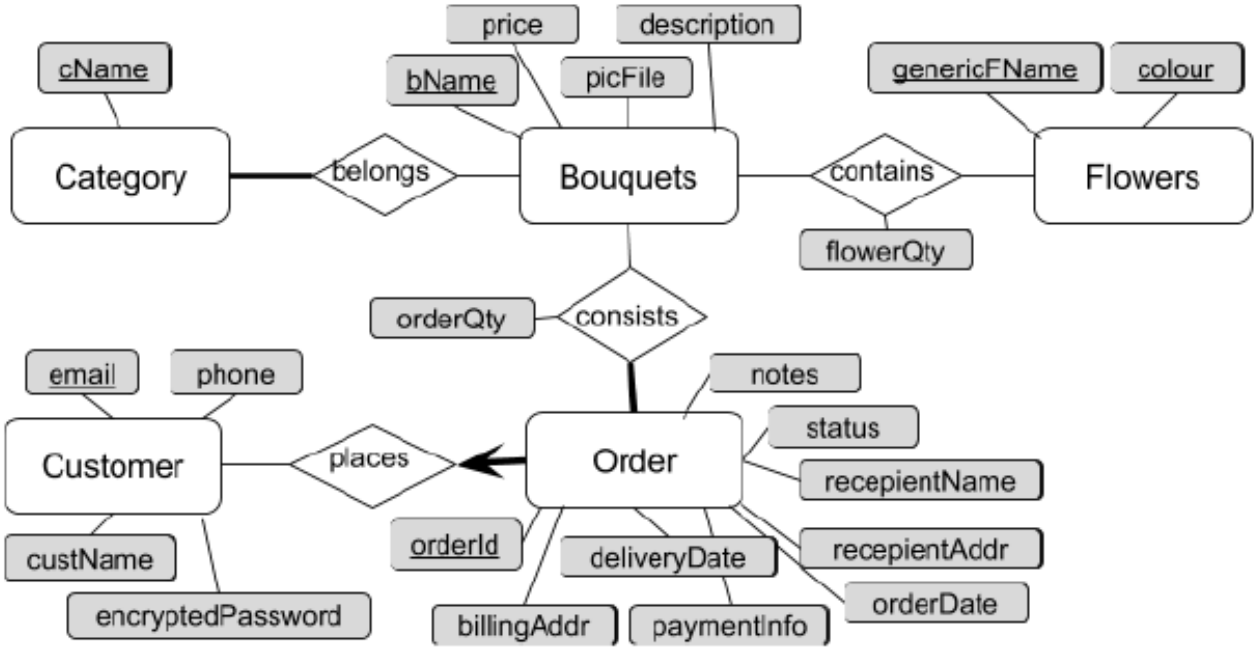
- Customers can register themselves online and place an order for one or more bouquets. For each customer, the typical information is kept (see attributes in the E/R diagram).

- For each order, the website only accepts credit card payment via an external portal, and we are to store only the last 4 digits of the credit card info, along with whether it was a "visa/mastercard /etc..." card, all in one attribute `paymentInfo`. Each order has a billing address, and since most of the bouquets are ordered as gifts, customers are also required to provide the name of the recipient and a shipping address, which can be different from the billing address. Customers also specify when (day) they want the bouquet to be delivered when they place an order. They can additionally leave a short note regarding the order (like "please call the phone number, doorbell is broken?).

- From receiving an order to the bouquets associated with the order being delivered, the order goes through different status values. An order is first "open". Once Dora has created the bouquet the status will be set to "prepared", followed by "out for delivery", when it is sent out for delivery on Dora's van. Once it is delivered, the status is set to "delivered". On rare occasions, an order might be "cancelled" and the customer is refunded. In some cases the delivery team may not be able to make a delivery, so the status might become "returned"" in which case the recipient has the option of picking it up from shop in person (if so, the status will become "delivered").

- The shop does not keep an inventory of its stock. Dora relies on the previous purchases to decide how much of flowers she needs to buy from wholesale distributors to keep up with the demand. This can sometimes lead to shortage/wastage of flowers depending on whether the stock of flowers she bought is in short/excess of the actual demands.

The following is the E/R diagram built by the comp 421 team to capture the functional and data requirements.



Further, the team has created the following relational schema for above ER schema:

`Customers(email, phone, custName, encryptedPassword`
`Orders(orderId, orderDate, email, billingAddr, paymentInfo, deliveryDate,`
`recepientAddr, receptientName, status, notes)`
`(staffId, staffName)`
⟶ `email` references `Customer(email)`
`Bouquets(bname, price, picFile, description)`
`Flowers(genericFName, colour)`
`bouquetContains(bName,genericFName, colour, flowerQty)`
⟶ `bName` references `Bouquets(bName)`
⟶ `genericFName,colour` references `Flowers(genericFName, colour)`
`orderDetails(orderId, bName, orderQty)`
⟶ `orderId` references `Orders(orderId)`
⟶ `bName` references `Flowers(bName)`
`Category(cname)`
`bouquetCategories(bName,cname)`
⟶ `bName` references `Bouquets(bName)`
⟶ `cname` references `Category(cname))`

This page is kept intentionally blank.

1 Multiple Answer Multiple Choice Question (32 points)

This section contains a set of multiple choice questions all referring to the example application given in the background information. Each question has **one or more correct answers**. **You have to select ALL correct answers**. If you miss a correct answer or you select a wrong answer, you will get 0 points for that question. For each question, we have indicated how many correct answers exists. Use this to guide your decision.
You can achieve a total of 32 points in this section (4 per each of the 8 questions).

1. (4 Points) For the E/R diagram depicted in the application description, which of the following statements holds true (2 correct answers).

   (A) The same order can be associated with multiple customers.

   (B) It is possible to have billing address same as recipient address.

   (C) One cannot create an empty order.

   (D) An order can have any number of a particular bouquet, but cannot have different bouquets in the same order.

   (E) The cost of the bouquet cannot be negative.

2. (4 Points) For the E/R diagram depicted in the application description, which of the following statements holds true (2 correct answers).

   (A) As per the E/R, it is possible to have a bouquet without any flowers in it.

   (B) *Order* should not have a participation constraint in the relationship with *Customer* as a customer might register, but not place an order.

   (C) *Flowers* is not a valid entity set in E/R terminology as it does not have any non-key attributes.

   (D) The total number of flowers used in a bouquet is not available since *flowerQty* is an attribute of the *Contains* relationship instead of being part of the *Bouquets* entity set.

   (E) For an order status that is "Delivered" we cannot tell conclusively what was its immediate previous status.

3. (4 Points) For the E/R diagram depicted in the application description, which of the following statements holds true (2 correct answers).

   (A) It is not possible to find the total cost associated with the order as this information is not captured in the *consists* relationship between *Order* and *Bouquets*.

   (B) All the bouquets in the same order will have to be delivered to the same address.

   (C) Since *consists* is not a ternary relationship, a customer will not be able to buy the same bouquet again as part of a different order.

   (D) The E/R captures information that the shop can use to find out which customers place orders frequently.

   (E) *flowerQty* in the *contains* relationship can represent the inventory stock of flowers available in the shop.

4. (4 Points) For the relational translation depicted in the application description, which of the following statements holds true (2 correct answers).

(A) Since the `Flowers` relation has no non-key attributes, we can remove the `bouquetContains` relation entirely by just moving the `flowerQty` attribute into the *Flowers* relation.

(B) The relation `Orders` should not have the attribute `email` as this belongs to the relation `Customers`. The attribute should be removed from `Orders` as it can cause redundancy and inconsistency.

(C) Instead of having `email` as an attribute in `Orders`, we could have an extra relation `OrderPlacement(email, orderId)`, `email` referencing `Customers` and `orderId` referencing `Orders`.

(D) With the current translation, if a customer changes their email, the orders that are already fulfilled (`status` set to "delivered"), do not need to change the `email` attribute accordingly.

(E) Categories cannot be nested

5. (4 Points) When creating the tables using SQL CREATE statements, which of the following statements holds true (2 correct answers).

(A) We can guarantee at table creation time through constraint definitions included in the SQL CREATE statements that there is always a customer associated with an order.

(B) We can guarantee at table creation time through constraint definitions included in the SQL CREATE statements that each order has at least one bouquet associated with it.

(C) `orderDetails` can be created before creating `Customers`.

(D) `Flowers` can possibly be the last table to be created.

(E) `bouquetContains` can possibly be the last table to be created.

6. (4 Points) Which of the following will be true for SQL queries written against these tables? (2 correct answers).

(A) It will involve a minimum of four tables in the join to find the names of bouquets that a given customer (given `email`), has ordered.

(B) There are four tables that can provide the names of all the bouquets offered by the shop (by reading column `bName`) without any joins.

(C) To get the number of bouquets that the shop has to make for a given delivery day (e.g., '2015-04-30'), it is enough to do a join over `orderDetails` and `Orders`.

(D) To compute the price of an order given its `orderId` we need to join `Orders`, `OrderDetails` and `Bouquets`.

(E) The number of bouquets under each category can be computed as a group by and aggregation over a single table.

7. (4 Points) Which of the following SQL statements will not produce duplicates in its output? (3 correct answers).

(A)
```
SELECT bname, genericFName FROM bouquetContains
```

(B)
```
SELECT price, COUNT(*) FROM Bouquets GROUP BY price
```

(C)
```
SELECT DISTINCT flowerQty from bouquetContains
```

(D)
```
SELECT B.bName, B.price
FROM Bouquets B, bouquetContains BC
WHERE B.bName = BC.bName
AND BC.flowerQty = (SELECT MAX(BC2.flowerQty) FROM bouquetContains BC2)
```

(E)
```
SELECT B.bName
FROM Bouquets
WHERE B.bName NOT IN (SELECT bName FROM bouquetContains)
```

8. (4 Points) This is a question regarding indexing. Note that this is not a straightforward question. There is one correct answer. Let us assume that the average length of a tuple in the `Orders` relation is 300 Bytes. The `Orders` relation is ordered by `email`. The page size is 4KB but you can assume that only 3KBytes are filled with records. On an average there are 100 orders to be delivered each day. A frequent query asks for all orders of a specific delivery date. Thus, you consider to create an index on `deliveryDate`. To simplify your calculations you start with no records in the `Orders` table on day 1 and each day you get only orders for delivery for the next day. After approximately how many days it is very likely that executing the query using the index will be better or equal than a simple scan of the `Orders` relation in terms of I/O.

   (A) 1

   (B) 10

   (C) 50

   (D) 100

   (E) 1000

2 TRUE/FALSE Multiple Choice Question (15 points)

This section contains 10 short TRUE/FALSE questions. Some of them relate to our example application, others are not related.

Each question is worth 1.5 points for a total of 15 points for this section.

9. Given the relational model of our application and the following two SQL queries.

```
SELECT F.genericFName, F.colour
FROM Flowers F, bouquetContains BC
WHERE F.genericFName = BC.genericFName AND F.colour = BC.colour

SELECT F.genericFname, F.colour
FROM Flowers F
WHERE F.genericFName, F.colour IN
              (SELECT BC.genericFName, BC.F.colour FROM bouquetContains)
```

Both queries return exactly the same result tables.

(A) TRUE

(B) FALSE

10. For our example application, a B+ Tree index is preferred over a Hash index on the `price` attribute of `Bouquets` if customers want to do a search on the website based on price ranges of bouquets.

(A) TRUE

(B) FALSE

11. For our example application, an index created on `picFile` will be useful to find the name of the image file associated with a bouquet (i.e., given a `bName`).

(A) TRUE

(B) FALSE

12. The smaller the reduction factor for a condition on any attribute `att` the more useful it is to use an index on `att` for the query.

(A) TRUE

(B) FALSE

13. Insertion order into a B+ Tree will effect the tree's end structure.

(A) TRUE

(B) FALSE

14. Consider two relations R(A, B, C) and S(A, D, E), sharing a common attribute A. Assuming that S.A is the primary key of relation S, R.A is a foreign key referencing S.A, and R.A may not be NULL, the estimated size of $|R \bowtie S|$ is $|R|$

(A) TRUE

(B) FALSE

15. It is possible to create two clustered indexes on two different attributes of a given relation, as long as one of them is on the primary key.

(A) TRUE

(B) FALSE

16. HBase uses horizontal partitioning but does not support vertical partitioning.

(A) TRUE

(B) FALSE

17. Neo4J has a query capability that is conceptually similar to path queries in XPath.

(A) TRUE

(B) FALSE

18. Using the 3-valued logic that was taught in class, and assuming the following values for A-E:

`A: TRUE, B:FALSE, C:TRUE, D:UNKNOWN, E:UNKOWN`

Is the following computation statement true or false?

`(A AND D) AND (B OR E)` evaluates to UNKNOWN

(A) TRUE

(B) FALSE

3 E/R, Relational Model and Queries: Open Questions (35 points)

This section contains open questions regarding the data model of and queries for our example application. For some of the questions, you have to rewrite or extend the existing E/R diagram. You ONLY need to draw the new entity sets, relationship sets and their respective attribute sets and the existing entity sets to which they connect. Furthermore, if you change an existing entity set or a relationship set, you should fully redraw it in the modified version.

1. (5 Points) *Extend the ER diagram so that a customer can keep a ranked list of his/her favorite bouquets. The information maintained should be able to capture that, e.g., customer with email* `cust@email.com` *has ranked the bouquet with* `bname='RoseExplosion'` *as his/her number 1, the bouquet with* `bname='AprilSurprise'` *as his/her number 2, etc.*

2. Dora would like to keep track of her flower stock, i.e., the number of flowers she has of each flower type.

   (A) (3 Points) *Extend the ER diagram so that this information can be maintained.*

   (B) (4 Points) Assume now a proper translation of this information into the relational model. Assume now that Dora buys 100 new 'pink' 'daisy pom'. *Indicate the SQL modification she has to perform to keep the information in the database up to date.*

   (C) (3 Points) *In which other situation(s) needs this information to be updated? You do not have to write any SQL statement but only provide a description of the situation(s).*

3. (5 Points) Dora wants to remove bouquets that have not attracted any customers lately. *Write a query (SQL or Relational Algebra) that will return the names of bouquets that have not been ordered this year (since January 2016).*

4. (4 Points) The web team is planning to enhance the page displayed for each bouquet by including at its bottom a list of bouquets that are usually purchased together with it (similar to how websites like amazon have at the bottom "customers also bought"). *Write an SQL query that returns the names of all bouquets that were purchased together with the bouquet named 'SpringMaiden' in the same order.*

   You get two bonus points if the output is sorted in descending order of popularity, i.e., the more often a bouquet was bought together with 'SpringMaiden', the earlier in the output should it appear. For instance, if 'SpringMaiden' was purchased only by orders $O1$ and $O2$, and the corresponding entries for these two orders in `orderDetails` are {(O1, 'SpringMaiden', 2), (O1, 'AprilSurprise', 2), (O1, 'RoseExplosion', 1), (O2, 'SpringMaiden', 1), (O2, 'RoseExplosion', 2)}, then the output should be first 'RoseExplosion' and then 'AprilSurprise', because the first was purchased at a quantity of 3 together with 'SpringMaiden' while 'AprilSurprise' only twice.

5. (6 Points) Dora wants to do some optimization in her purchases of flowers from wholesale distributors. *Write an SQL query that will return for each flower the generic name, the colour and the amount required to fulfill the bouquet orders for a particular delivery date (say 2016-04-31).*

6. (5 Points) *Write a PIG LATIN query for the same request: It should return for each flower the generic name, the colour and the amount required to fulfill the bouquet orders for a particular delivery date (say 2016-04-31).*

4 Query Evaluation (35 Points)

This section contains open questions regarding query evaluation. The questions are not related to our example application. Instead, we look at the following simple, Canadian-only social media webpage. It keeps track of persons. Furthermore, persons can exchange messages and all messages are logged. The schema is as follows

`Person (login:INT, name:VARCHAR(40), city:VARCHAR(30), ZIP:CHAR(3), phone1:CHAR(12), phone2:CHAR(12), zodiacal:CHAR(10), occupation:VARCHAR(40), birthyear:CHAR(4))`
e.g.: (555, 'Dora', 'Montreal', 'H3A', '514-111-2222', '514-111-2222', 'Capricorn ', 'Explorer', '2000')

`Messages(mid:INT, sender:INT, receiver:INT, when:DATE, msubject:VARCHAR(20))`
e.g.: (12346, 555, 666, '2016-04-20', 'RE: important notice')
$\longrightarrow$ `sender` references `Person(login))`
$\longrightarrow$ `receiver` references `Person(login)`

`MessageText(mid:INT, mtext:VARCHAR(3500))`
e.g., : (12346, 'Hi Sniper, thanks for writing me; that was a surprise !....')
$\longrightarrow$ `mid` references `Messages(mid)`

`Person` has 100,000 tuples. The average size of a tuple is 100 Bytes. There are approx. 3000 data pages for this relation. You can assume that the values in VARCHAR attributes have on average half of the maximum size. The ZIP only keeps track of the first 3 letters/digits of the ZIP code (e.g., H3A) and is never NULL. There are around 2000 different such ZIP codes. There are around 200 different occupations to choose from (not a free text input field).

`Messages` has 2,000,000 tuples. The average size of a tuple is 35 Bytes. There are approx. 20,000 data pages for this relation.

You can make the following assumptions. There are indexes on all primary keys. You might suggest any other index for a question if you think it might be useful. Root and intermediate pages of B+-tree indices are always in main memory. No other page is assumed to be in main memory at the begin of a query execution. There are around 100 buffer frames available.

If you think that the description misses values and sizes of attributes etc. that you think are necessary for your calculations make reasonable assumptions and indicate them together with your answer.

1. (10 Points) Given the query

```
SELECT m.sender, p.name, COUNT(*), COUNT(DISTINCT receiver)
FROM Person p, Messages m
WHERE p.login = m.sender
AND occupation = 'Explorer'
GROUP BY m.sender, p.name
```

Give the best execution plan (indicating the indexes and join types used and any other optimization you might do – discussing that things might fit into memory, or other tricks that you might perform) and indicate the costs of I/O. If you have a correct but not the optimal solution, you will get partial points.

Comment: I suggest you to make a good guess in regard to strategy, and do the calculations for that strategy. If you finish early with the exam, you might want to come back to this question and try out other strategies to see whether they work better.

2. (7 Points) Now the query is slightly changed only affecting the WHERE clause

```
SELECT m.sender, p.name, COUNT(*), COUNT(DISTINCT receiver)
FROM Person p, Messages m
WHERE p.login = m.sender
AND p.birthyear > 1975
GROUP BY m.sender, p.name
```

How would now your best strategy look like?

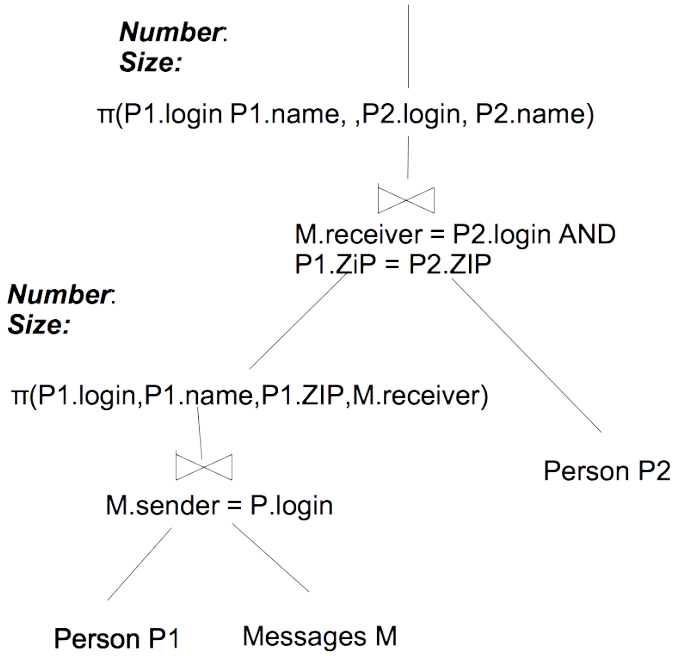
3. (8 Points) Let's now have a look at how to use the map-reduce framework.

Assume that the designers realize it would be beneficial to have an additional attribute `length` in `Messages` that indicates the size of the actual text (which is stored in `MessageText`).

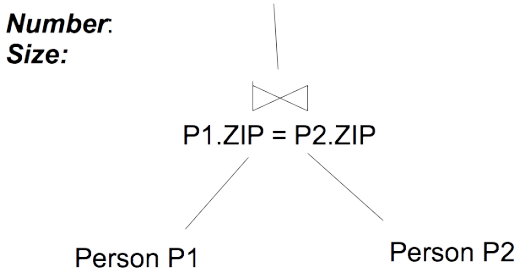
Outline the implementation of a map-reduce process that takes as input `Messages` and `MessageText` (with the primary key as key, and all other attributes as values), and outputs a relation similar to `Messages` but with an additional attribute `length` in the value. In particular, outline mapper and reducer functions, indicating the format of their input and their output, and how they process each of their input records. You can write pseudo-code or provide a description in half-formal English similar to how the implementation of the basic relational operators using map-reduce was discussed in class.

4. (10 Points) Consider the following two execution trees. The first one finds all pairs of persons living in the same ZIP area and having exchanged at least one message. The second returns all pairs of persons living in the same ZIP area (note that the second one also returns each person paired with him/herself).

   For both execution trees, indicate the number of tuples and the size of each tuple that flow from one operator to the next for the positions indicated in the figures with bold and italic "number" and "size" fields.



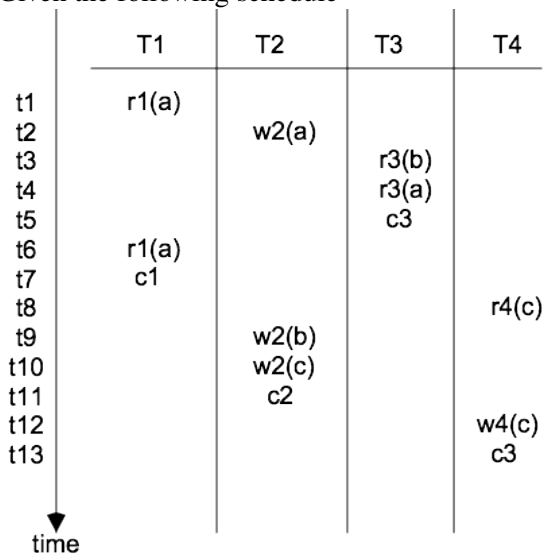
(a) persons at the same ZIP having exchanged a message



(b) persons at the same ZIP

5 Concurrency Control (20 Points)

Given the following schedule



1. (3 Points) *Show the serialization graph for this schedule. Indicate whether the schedule is serializable or not. If it is serializable, provide an equivalent serial schedule.*

2. (4 Points) For each of the anomalies indicated below, indicate whether it occurs (YES or NO). If YES, indicate the time(s) of the operation(s) (t1-t13) that are responsible for this anomaly.

   (A) Lost Update

   (B) Dirty Read

   (C) Unrepeatable Read

3. (8 Points) Assume now that the execution above does not depict the final schedule but the sequence of operations submitted to a DBMS that uses strict 2PL.

   (A) Describe how strict 2PL handles each of the sequences above. Add lock $S_i(a)$ when T_i requests shared lock on a, $X_i(a)$ when T_i requests exclusive lock on object a, and unlock request $U_i(a)$ when T_i releases any lock on object a to the above sequence[1].

   Give the execution for the following type of lock manager:

   Upon a shared lock request $S_i(a)$ of transaction T_i, if there are only shared locks active on a AND no lock is waiting on a, then grant $S_i(a)$, else $S_i(a)$ must wait:

   (B) Provide a conflict-equivalent serial schedule.

4. (5 Points) Now assume that the system uses a variation of the locking protocol using **short** shared locks and standard exclusive locks: (i) Before each read operation on data item a, a shared lock on a is acquired. This lock is released immediately after the operation finishes (and not only at end of

[1]The DBMS processes actions in the order shown. If a transaction is blocked, indicate this in the schedule; assume that all of its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed sequence) of an unblocked transaction. If there is a deadlock, one of the transactions is aborted (undoing first all update operations and then releasing the locks).

transactions). (ii) Before each write operations on data item a an exclusive lock on a is acquired. Such an exclusive lock is only released after the transaction commits.

Show how this adjusted locking protocol handles the sequence above. Add lock $S_i(a)/X_i(a)$ *and* $U_i(a)$ *at the appropriated steps.*