McGill

April 2017
Final Examination

VERSION 0

Database Systems

COMP-421-001 Winter 2017

April 24th 2017 14:00 – 17:00

EXAMINER: Joseph Dsilva ASSOC. EXAMINER: Bettina Kemme

STUDENT NAME:		**McGILL ID:**									

INSTRUCTIONS:

EXAM:	CLOSED BOOK ☒ OPEN BOOK ☐
	SINGLE-SIDED ☐ PRINTED ON BOTH SIDES OF THE PAGE ☒
	MULTIPLE CHOICE ☒ NOTE: The Examination Security Monitor Program detects pairs of students with unusually similar answer patterns on multiple-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. ANSWER IN BOOKLET ☒ EXTRA BOOKLETS PERMITTED: YES ☒ NO ☐ ANSWER ON EXAM ☐
	SHOULD THE EXAM BE: RETURNED ☒ KEPT BY STUDENT ☐
CRIB SHEETS:	NOT PERMITTED ☐ PERMITTED ☒ Specifications: Eight 8 1/2X11 handwritten or typed double-sided sheets
DICTIONARIES:	TRANSLATION ONLY ☒ REGULAR ☐ NONE ☐
CALCULATORS:	NOT PERMITTED ☐ PERMITTED (Non-Programmable) ☒
ANY SPECIAL INSTRUCTIONS:	WRITE YOUR NAME AND STUDENT ID ON THE SCANTRON AND EXAM BOOKLET ALSO. 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. Answer all multiple choice and TRUE/FALSE questions on the scantron sheet and open questions in the answer booklet.

This page is kept intentionally blank.

Scoring

The exam is out of 140 points distributed as follows:

1. Section 1 (Multiple Choice with Multiple Answers): 8 questions; each 5 points for a total of 40 points
2. Section 2 (TRUE/FALSE Questions): 10 questions; each 1.5 points for a total of 15 points
3. Section 3 (SQL Queries): 3 questions for a total of 16 points
4. Section 4 (Query Evaluation): 3 questions for a total of 21 points
5. Section 5 (Transactions & Concurrency Control): 3 questions for a total of 15 points
6. Section 6 (Large Scale Data Processing): 2 questions for a total of 17 points
7. Section 7 (Graph Databases and Cypher Query Language): 3 questions for a total of 16 points

Some Useful Suggestions

- Multiple Choice as well as TRUE/FALSE questions in general help you familiarize with the background of the application.
- Try to attempt the questions in the sections that you are more comfortable first.
- If a question seems hard, leave it to the end to come back and try again.
- Hard questions are marked with an asterik (*).
- If you are not able to find the best solution for a question (if that has been required), write your best guess, you might be able to tune it to a better solution later or get partial marks for your attempt.

Version-0

Background Information for this Exam

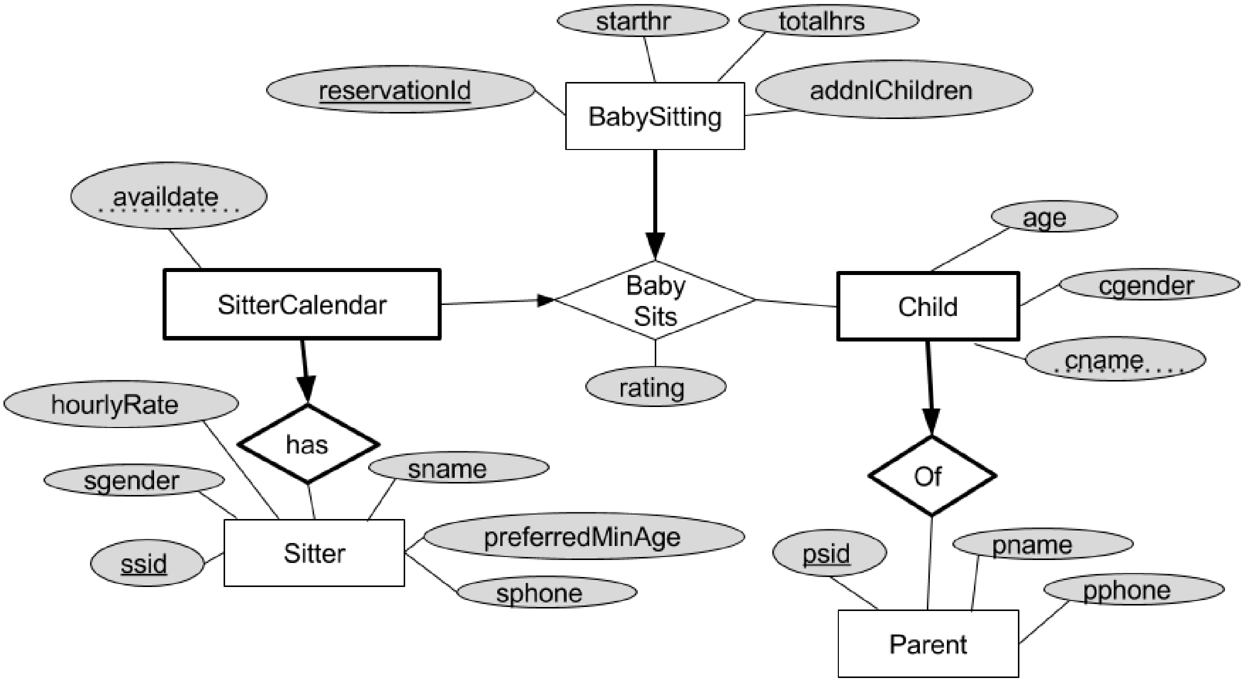
A good part of the questions on this exam is related to the following application.

Several students are also parents, and babysitting is an essential service for them. In order to facilitate student parents, the nursing students have come up with a babysitting service system. A comp 421 project group has been enlisted to build a database to capture all the information that is required to be stored and facilitated by the system.

Below are the requirements they have gathered.

- Students can sign up as baby sitters. Their name, phone, student id, hourly rate and the minimum age of the children they are willing to babysit should be captured.
- Additionally, some parents prefer a babysitter of the same gender as that of their child. In order to facilitate this, the gender of the child as well as that of the sitter should be captured.
- Age (yes bad practice, but let us keep it simple) and name of the child should be captured. It is assumed that a parent will not have two children with the same name.
- The name, student id and phone number of the parents also needs to be captured.
- Sitters generally let the office know of their availability (we will assume an entire day not part of it) in advance, which is recorded in a calendar for babysitters.
- For parents to make a reservation with a particular sitter, The system first checks that person's availability calendar for the requested date and next it checks if a reservation for this sitter already exists for the said date.
- When the parents make a baby sitting reservation with a particular sitter, a reservation id is assigned to it, and additionally the hour of the day the assignment starts (assume 0 - 23) and the total hours required (assume again whole integers) need to be captured. We will assume that the job does not "cross over" calendar dates. We will also assume that a baby sitter will have only one booking a given day.
- Parents can rate (1 - 10) a particular babysitting session (reservation) but are not mandatory to do so. It is left unpopulated (NULL) by default.
- When a parent needs more than one child to be taken care of, an additional children attribute needs to be captured for each reservation. Each additional child will incur a 50% extra on the hourly rate of the babysitter. By default, this attribute will be 0.

Below is the ER Diagram created by the comp 421 team.
Note that there might be some minor issues in the model. They may have also made some reasonable assumptions, where the requirements are not explicitly recorded.
Additionally, there may be some requirements that may not have been possible to capture in the ER.



They have also created the relational model from ER, as given below.

```
sitter(ssid ,sname ,sphone ,sgender ,hourlyRate ,preferredMinAge)
parent(psid ,pname ,pphone)
sitterCalendar(ssid ,availDate)
⟶ ssid references sitter
child(psid ,cname ,cgender ,age)
⟶ psid references parent
babySitting(reservationId, ssid ,availDate, psid ,cname ,starthr, totalhrs,
addnlChildren, rating)
⟶ (ssid ,availdate) references sitterCalendar
⟶ (psid ,cname) references child
```

This page is kept intentionally blank.

1 Multiple Answer Multiple Choice Questions (40 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 40 points in this section (5 per each of the 8 questions).

1. (5 Points) Based on the ER translation of the requirements, which of the conclusions are correct ? (choose 3 correct answers)

   (A) A child's information can be associated only with one parent

   (B) Given a child, we can find the siblings of that child ONLY if they are all registered with the same parent.

   (C) There cannot be a parent without any children associated with them.

   (D) If child entity set is given an artificial key, then we can model the ER to associate more than one parent with the same child.

   (E) Two children of the same parent can have the same name as long as they have different gender.

2. (5 Points) Based on the ER translation of the requirements, which of the conclusions are correct ? (choose 2 correct answers)

   (A) A child could be babysat at the same day and/or time by more than one sitters.

   (B) A baby sitter cannot have more than one babysitting assignments on the same day.

   (C) The ER has a defect because of the key constraint from `sitterCalendar` to `babySits`. This will result in a baby sitter being able to have only a single babysitting assignnent ever and will not be able to babysit another/same child for a different date.

   (D) An alternate ER model could have a relationship (with key constraint) from `sitter` to `babySits` directly, without requiring the `sitterCalendar` and make `availDate` as an attribute of the `babySits` relationship. This will be conceptually identical to the given ER model.

   (E) There should be a participation constraint from `sitterCalendar` to `babySits` as (per the given requirements) the assigned babysitter is known when the reservation is made.

3. (5 Points) Based on the ER translation of the requirements, which of the conclusions are correct ? (choose 3 correct answers)

   (A) ER does not enforce that the child's gender should match that of the baby sitter.

   (B) In an alternate ER model, `rating` can be an attribute of `babySitting` instead of `babySits`. This will be conceptually identical to the given ER model.

   (C) ER cannot enforce that a babysitter is not booked twice for the same date.

   (D) The total charge incurred towards a babysitting reservation can be computed without any additional attributes to this ER.

   (E) `child` should have a participation constraint to `babySits` .

4. (5 Points) For the relational translation depicted in the application description, which of the following statements holds true ? (choose 1 correct answer)

(A) The key constraint from `sitterCalendar` to `babySits` is not enforced in the relational model.

(B) `babySits` should be a relation of its own as it is a ternary relationship.

(C) `rating` should not be part of `babySits` relation as it is from a relationship set.

(D) `ssid` in `babySitting` could refer to `sitter` instead of `sitterCalendar` and it will conceptually be the same.

(E) If we add a foreign key reference from `sgender` of `sitter` to `cgender` of `child`, we can enforce that the child and babysitter should have the same gender.

5. (5 Points) When creating the tables for the given relational model using a set of SQL `CREATE TABLE ...` statements that contain all the constraints we plan to impose, which of the following statements hold true ? (choose 2 correct answers)

(A) `babySitting` can be created before `sitter`.

(B) The first two tables to be created can only be `sitter` and `parent`.

(C) `child` can possibly be the second table to be created.

(D) `babySitting` need not be the last table to be created.

(E) `parent` will have to be created before `babySitting`.

6. (5 Points) Which of the following will be true for SQL queries written on tables created from the given relational model ? (choose 1 correct answer)

(A) It will take a minimum of 3 tables to find out the distinct number of parents who have actually made a babysitting reservation using the system.

(B) It will take a minimum of 2 tables to find out if a particular babysitter (given `ssid`) is available for babysitting for a particular date (has no other reservations as well).

(C) Given a sitter (`ssid`), the total amount of money the person made from babysitting in a given month can be computed by a SQL query on just one table.

(D) We have to join at least 3 tables to find the names of registered babysitters who have never been booked.

(E) Given a specific child `(psid, cname)` and a babySitter `(ssid)` it will take a 4 table join to verify if they are of the same gender or not.

7. (5 Points) Review the possible values of the attribute `rating` of the relation `babySitting` as given in the data requirements. Find two SQL queries below such that their ouput records are ALWAYS identical to each other. (choose 2 correct answers)

(A)
```
SELECT ssid, AVG(rating)
FROM babySitting
GROUP BY ssid
```

(B)
```
SELECT ssid, AVG(rating)
FROM babySitting
WHERE rating IS NOT NULL
GROUP BY ssid
```

(C)
```
SELECT ssid, COUNT(DISTINCT rating)
FROM babySitting
GROUP BY ssid
```

(D)

```
SELECT ssid, COUNT(rating)
FROM babySitting
GROUP BY ssid
```

(E)

```
SELECT ssid, COUNT(*)
FROM babySitting
GROUP BY ssid
```

8. (5 Points) Consider the database statistics that is given below for some of the columns of the table `sitter` (assume it has 100 records for now to make your computation simple).

		hourlyRate			preferredMinAge	
Range	→	13-15	16-19	20	0-4	5-12
Number of records	→	30	60	10	20	80

You may assume that the records are uniformly distributed within any given range and that the distribution of attributes are independent of each other. Using the information given in this question, compute the reduction factor for the following SQL query.

```
SELECT * FROM sitter
WHERE hourlyRate BETWEEN 14 AND 16 AND preferredMinAge < 5
```

(choose 1 correct answer)

(A) 0.35

(B) 0.55

(C) 0.70

(D) 0.07

(E) 0.01

2 TRUE/FALSE Choice Questions (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. Both the SQL queries given below will output the same records

```
SELECT DISTINCT ssid FROM babySitting

SELECT ssid FROM sitter WHERE ssid IN (SELECT ssid FROM babySitting)
```

   (A) TRUE
   (B) FALSE

10. We cannot enforce the constraint that `rating` can only be NULL or between 1 and 10 as it is updated only later.

   (A) TRUE
   (B) FALSE

11. If we create an index on `psid` of parent, a type II indirect indexing will take less space compared to a type I indirect indexing.

   (A) TRUE
   (B) FALSE

12. A common way to find a suitable babysitter is to first start searching the `sitter` table by a range on the `hourlyRate` and a greater than or equal condition on min `minPreferredAge`. This can benefit from clustering the table on `minPreferredAge` compared to an unclustered table.

   (A) TRUE
   (B) FALSE

13. Using the relational model of our background application and assuming we have 20 data pages for the `parent` relation, a query that searches parent information in the `parent` table with a search condition on a phone number (say `pphone` =*'514-555-8888'*) will result in, on an average 10 IOs

   (A) TRUE
   (B) FALSE

14. The efficiency of a page nested loop join algorithm cannot be increased by adding more buffer frames (assuming that the total memory buffer frames are still less than the total number of pages that either relation has.)

   (A) TRUE
   (B) FALSE

15. The height of a B+-Tree index is not dependent on the size of the data entries.

(A) TRUE

(B) FALSE

16. Adding an extra attribute that is not part of the search key of a index will reduce the number of page pointer entries that can be fit into an intermediate node of an index structure.

    (A) TRUE

    (B) FALSE

17. In MapReduce, the *Combine* function is executed at the *Reduce* task.

    (A) TRUE

    (B) FALSE

18. In a graph database like Neo4j, a node/vertex in the graph is similar to the concept of an entity in the ER model.

    (A) TRUE

    (B) FALSE

3 SQL Queries: Open Questions (16 points)

1. (4 Points) SSMU would like to reach out to the babysitters that will be around during the summer and offer them opportunities to take part in a free training program. For this purpose they need to know the name and phone number of babysitters who have an entry in the `sitterCalendar` for the months of *May* through *August* of *2017*.

   Write a SQL or Relational Algebra Query to accomplish this. Ensure that there are no duplicates in the output. You may assume that phone number is unique to a person for the results produced.

2. (5 Points) SSMU is planning to hire some of the babysitters to work in their daycare facility. Write a SQL query that will give the `ssid`, `sname` and `sphone` of the babysitters who has at least 5 `rating`s above 7 for their babysitting assignments.

3.* (7 Points) Find the psid of parents who have more than one child, of which at least one of them is a *female* child.

4 Query Evaluation (21 Points)

Note:-***If you find query evaluation section to be hard in general, I recommend that you attempt it in the end.***

```
sitter(ssid:INT ,sname:VARCHAR(40) ,sphone:CHAR(12) ,sgender:CHAR(6)
,hourlyRate:INT ,preferredMinAge:INT)
parent((psid):INT, pname:VARCHAR(40), pphone:CHAR(12))
babySitting(reservationId:INT, ssid:INT ,availDate:DATE, psid:INT
,cname:VARCHAR(40) ,starthr:INT, totalhrs:INT, addnlChildren:INT, rating:INT)
⟶ (ssid ,availdate) references sitterCalendar
⟶ (psid ,cname) references child
```

INT takes 4 Bytes, FLOAT takes 8 bytes, all VARCHAR fields take half of the size specified. DATE field is 10 Bytes
All pages are 4K size (you can use 4000 for calculations)

`sitter` has 1,000 records spread across 20 data pages. 20% of the babysitters are *male*
`parent` has 20,000 records.
`babySitting` has 1,000,000 records spread across 20,000 pages. `availDate` has 10,000 distinct values. `totalhrs` have values in the range 1-20.

You can make the following assumptions. There are indexes on all primary keys (unclustered). 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 10 buffer frames available. (you can assume another couple of frames are available if it helps your computation to be easier)

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, which computes the total amount of money earned by each babysitter in the first 5 days of March 2017,

```
SELECT s.ssid, s.sname,
       SUM(s.hourlyRate*b.totalhrs*(1+0.50*addnlChildren))
FROM sitter s, babySitting b
WHERE s.ssid = b.ssid
  AND b.availDate BETWEEN ’2017-03-01’ AND ’2017-03-05’
GROUP BY s.ssid, s.sname
```

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.

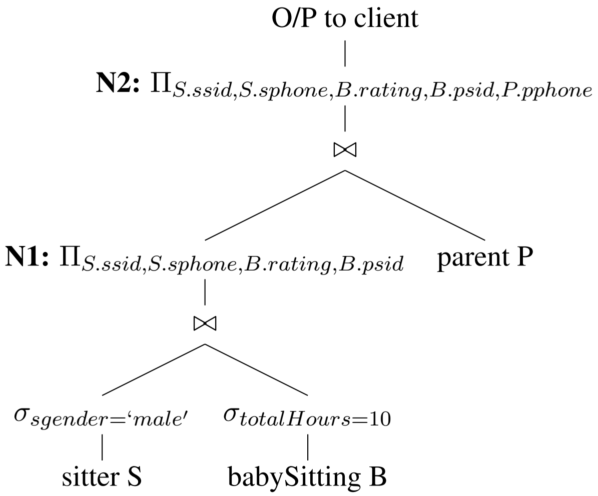
You can, but need not draw an execution tree. In any case you should provide explanations of your strategies and your calculations.

Note:- *I suggest you to make a good guess in regards to the 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. (5 Points) Suggest any indexes that you can add/modify in the tables to decrease the I/O cost of the above query. Describe the new execution plan and compute the costs.

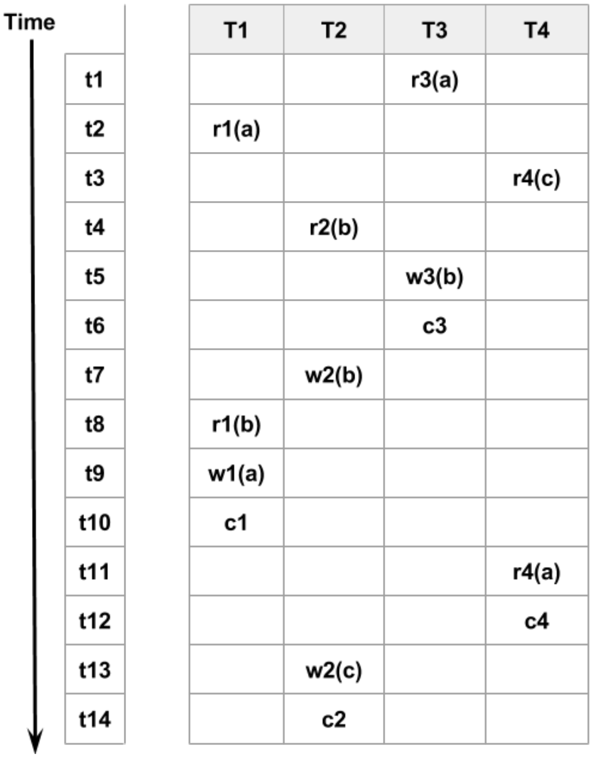
3. (6 Points) Consider the following execution tree. It finds the information about *male* babysitters who have had a 10 hour booking and the parent's phone.

   *We have marked two of the operators with labels **N1** and **N2**. For each of these operators write out the size and number of tuples entering the operator as well as leaving the operator. Use the labels given to indicate in your answersheet which operator the answer is meant for.*


O/P to client
N2: $\Pi_{S.ssid,S.sphone,B.rating,B.psid,P.pphone}$
$\bowtie$
N1: $\Pi_{S.ssid,S.sphone,B.rating,B.psid}$
parent P
$\bowtie$
$\sigma_{sgender=\text{‘}male'}$
$\sigma_{totalHours=10}$
sitter S
babySitting B

5 Transactions & Concurrency Control (15 Points)

Given the following schedule



Time	T1	T2	T3	T4
t1			r3(a)	
t2	r1(a)			
t3				r4(c)
t4		r2(b)		
t5			w3(b)	
t6			c3	
t7		w2(b)		
t8	r1(b)			
t9	w1(a)			
t10	c1			
t11				r4(a)
t12				c4
t13		w2(c)		
t14		c2		

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. (8 Points) Assume now that the execution above does not depict the final schedule but the sequence of operations submitted to a DBMS. In class, we discussed strict 2PL, but here we look at a different kind of locking that is less restrictive but may allow some anomalies to happen. In particular, the system uses **short** shared locks and standard exclusive locks: (i) Before a transaction performs a read operation on data item **a**, it has to acquire a shared lock on **a**. This lock is released immediately after the operation finishes (and not only at end of transactions). (ii) Before a transaction performs a write operation on data item **a**, it has to acquire an exclusive lock on **a**. Such an exclusive lock is only released after the transaction commits.

   Describe how this locking system 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]

3. (4 Points) For each of the anomalies indicated below, indicate (YES/NO) whether it occurs in the execution you have just sketched for question 5. 2 . If YES, indicate in the execution where this anomaly occurred.

   (A) Lost Update (B) Dirty Read (C) Unrepeatable Read

[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).

6 Large Scale Data Processing (17 Points)

1. (5 Points) Given a parent's `psid` (say *123456789*) write a Pig Latin script that will give the ssid, name and phone number of all the *'female'* babysitter(s) who have been booked by this parent at least 10 times. No rows to be returned if none exists.You may assume that phone number is unique to a person. Order the output by the name of the babysitter.

   Note:- You do not have to do the "load" operator, assume that the data is already available in relation names identical to those given in the relational model and their attribute names.
   The output can be either stored in the filesystem or displayed to the screen.

2. (12 Points) Computing the rank of a web page is a very important part of search engine algorithms. A very primitive way of doing this will be to count the number of webpages that link into (i.e., points/references to) a particular webpage. To facilitate this, at first we need to count the number of webpages (URLs) that has a reference to each webpage (URL).
   so for example if,
   URL1 has references to URL2 and URL3
   URL2 has reference to URL1
   URL3 has reference to URL1
   Then the number of URLs referencing URL1 is 2, the number of URLs refrencing URL2 is 1 and the number of URLs refrencing URL3 is also 1.

   (A) (5 Points) Using pseudo code and simple english description, write a Mapper function and Reduce function logic that will take as input the key value pairs in the form (referencingURL, referencedURL_List) as shown below
   (URL1, (URL2,URL3))
   (URL2, (URL1))
   (URL3, (URL1))
   And output the url and the count of webpages referencing them as below.
   (URL1, 2)
   (URL2, 1)
   (URL3, 1)

   Your logic can ignore the webpages that are not referenced by any other webpage.
   You have to account for the following situations:

   - Do not include in the count if a URL references itself.
   - If a URL (say URL1) references another URL (say URL2) more than once, count it only once.

   (B) (7 Points) However such reference counts are often not true indication of how important a webpage is, hence it is often enriched with some additional information, such as the popularity of the webpage, which can be computed by accounting for the number of visits the webpage receives by processing the webserver logs.
   Given a popularity matrix `popularity(URL, popValue)` that contains values of the form
   (URL1, 12.2)
   (URL2, 10.5)
   (URL3, 6.2)

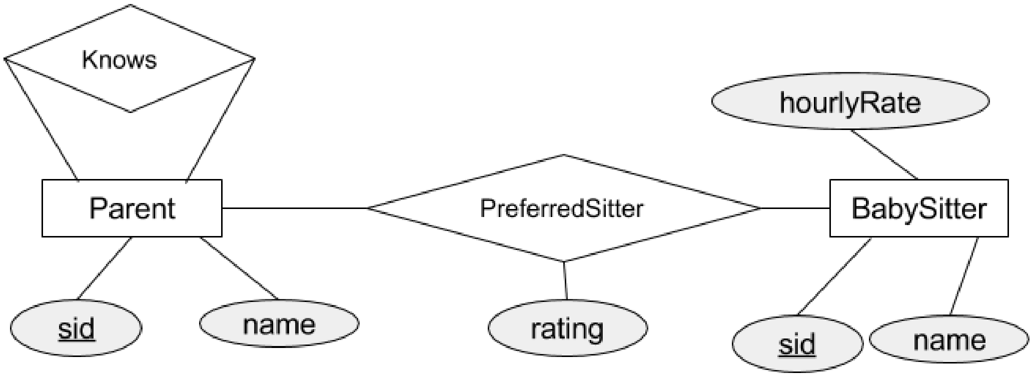
Write the logic for a MapReduce job, that uses the output of the previous MapReduce job along with with the `popularity` information given, to produce a weighted index output such that for each URL, we get in the output the URL and its weightage index (computed as the refernce count of the URL multiplied by the `popValue`).
For our example data set, this should produce in the output,

(URL1, 24.4)
(URL2, 10.5)
(URL3, 6.2)

If a URL does not have a record in the popularity matrix, its weighted index value should be the same as its reference count.

7 Graph Databases and Cypher Query Language (16 Points)

Consider the following simple ER, which we are using to keep track of each parent's preferred babysitters. Additionally some parents may know each other (you can assume that if a person X knows Y then Y also knows X).



Further, we have given an example instance of a relational database based on this ER.

Parent

sid	name
101	Jack
230	Moe
140	Kevin

BabySitter

sid	name	hourlyRate
202	Sheryl	12
190	Katie	14
679	Betty	11

PreferredSitters

pSid	sSid	rating
230	190	7
101	202	8
140	679	6

Knows

pSid	knowsPSid
230	101
101	230

1. (8 Points) Model and draw a graph database instance that can capture the data provided in the example relational instance and the relationships depicted in it.
   Remember that your approach should work even if more data is added into it.
   You can assume that the babysitters and parents are two distinct sets of students.
   Note:- *I recommend to use a fresh page and space out the nodes sufficiently from each other to have enough space to write properties and such.*

2. (4 Points) Two student parents *(799,Melanie)* and *(800,Susie)* just moved to Montreal. *Melanie* and *Susie* know each other, further *Susie* and *Moe* know each other.
   Write a Cypher Query that will help *Melanie* find all the preferred babysitters of people in her network (including parents she knows indirectly).
   OR instead, for 2 points, write a Cypher Query that will return the preferred babysitters of any parent that *Susie* knows directly.

3. (4 Points) A social networking company, `BigGlueCon` has acquired the babysitting data. They have realized that they can enrich the information that is already present. For example if `B` is the preferred babysitter for parent `P` then both `B` and `P` should also *know* each other.
   Write a cypher query that will read your current graph database and create these relationships in it.

This page is kept intentionally blank and is the last page of your exam booklet.