Midterm 2017, questions and answers

Database Systems (Mcgill University)

Faculty of Science
COMP-421 - Database Systems (Winter 2017)
Midterm Examination

March 08, 2017	Examiner:	Joseph Dsilva
18:00 - 19:00		Associate Examiner:

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 three 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 **10** 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 65 points distributed as follows:

1. Section 1 (multiple choice with multiple answers): 4 questions; each 5 points for a total of 20 points
2. Section 2 (Open Questions): 5 questions for a total of 45 points

Version-0

This page is kept intentionally blank.

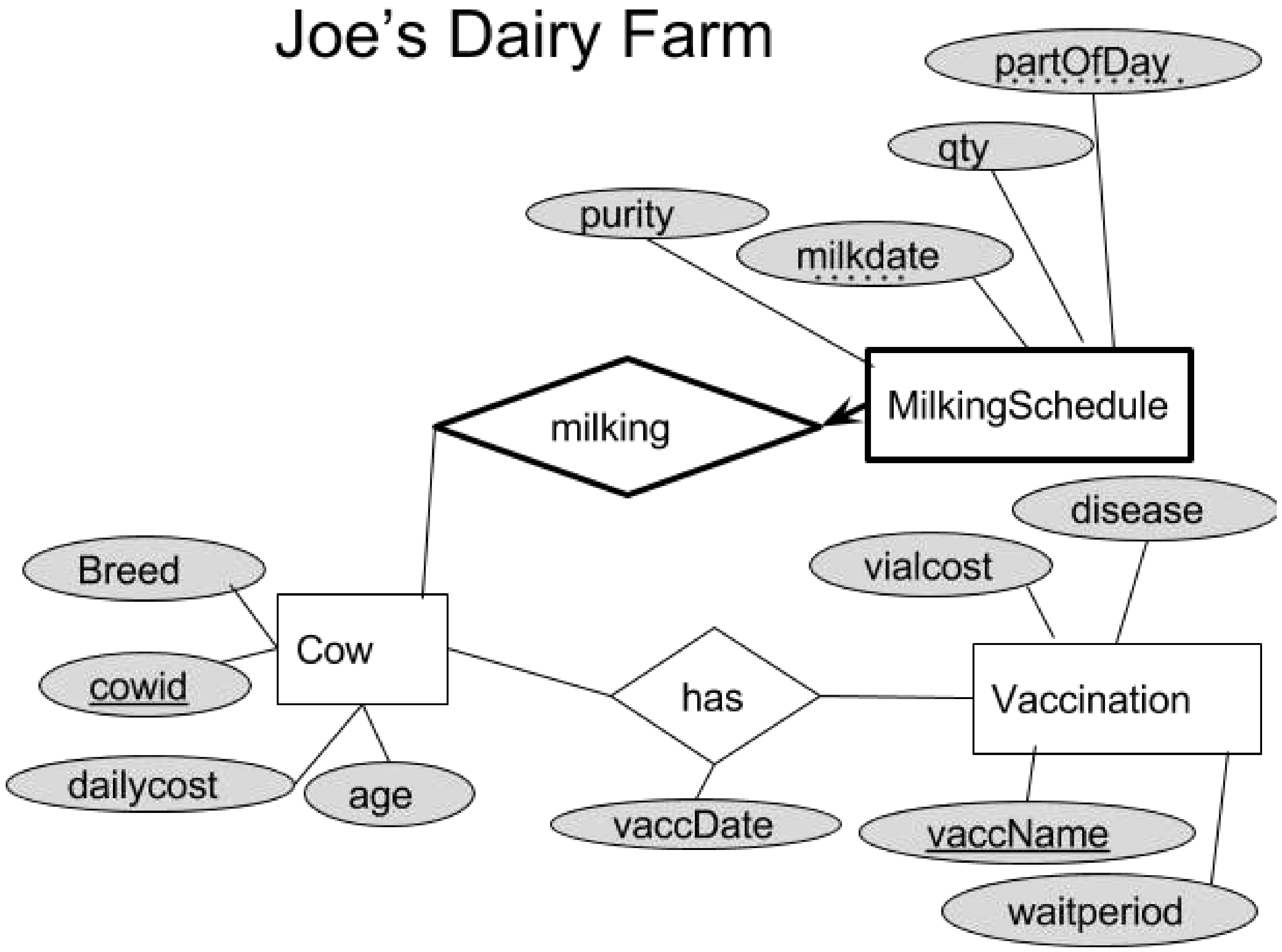
Background Information for this Exam

All questions on this exam is related to the following application.

Farmer Joe is a dairy farmer in Quebec, owning a farm run by his family. He raises cattle for milk production and sells milk to local stores. His daughter Debbie Kiwi, a former comp 421 student, has decided to help her dad with his business by 'IT-enabling' it.
The following are the requirements that Debbie has managed to gather.

- Cows are tagged with ids, so each cow in the farm can be identified and associated with the id.
- A cow will be of one of many breeds (i.e., *Jersey , Holstein, Shorthorn etc* ...). We will not worry about hybrids.
- We also record how old a cow is by recording its age (yes, this is a bad approach instead of using birth date of the cow, but we will keep it simple)
- We are also keeping store the average daily cost (a fixed estimate) towards each cow (incurred from feed, employee wages of the farm, farm maintenance, vaccinations, etc)
- Cows may be milked in the *morning* and/or *evening* every day (it is possible they are not milked on some days). We need to keep track of the quantity of milk produced by the cows on each of these milkings. A lactometer test is also performed and a percentage of purity is also calculated for each of this milking and recorded along with the quantity of milk.
- There are different vaccines (identified uniquely by vaccine name) for different diseases. (A vaccine can be good only for one particular disease). A vaccine also has a cost per "shot" associated with it. One shot will vaccinate a cow against a particular disease.
- Cows will be vaccinated and we need to keep track of the date on which they were vaccinated using a particular vaccine. A particular vaccine is applied to the cow only once as a single shot.
- There is a "wait period" in days for each vaccine, during which the cows should not be milked to avoid getting the medicines into the milk. For example if a cow is vaccinated on 2017-02-12 and the vaccine's wait period is 2, it can be milked only on 2017-02-14.

Debbie has done a good job in producing an E/R diagram to capture this information, which is given below. Note that there might be some minor issues in Debbie's model. She may have also made some reasonable assumptions, where the requirements are not explicitly recorded.



She has also created a relational model from the ER, as given below.

```
cow(cowid, breed, dailycost, age)
vaccination(vaccName, vialcost, disease, waitperiod)
milkingSchedule(cowid, milkDate, partOfDay, qty, purity)
⟶ cowid references cow
vaccinationSchedule(cowid, vaccName, vaccDate)
⟶ cowid references cow
⟶ vaccName references vaccination
```

This page is kept intentionally blank.

Multiple Answer Multiple Choice Question (20 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 20 points in this section (5 per each of the 4 questions).

1. (5 Points) Based on the ER translation of the requirements, which of the conclusions are correct ? (choose 2 correct answers).

   (A) Total cost of vaccines used for an individual cow cannot be determined.

   (B) A particular cow may receive multiple vaccinations on the same day.

   (C) `vaccdate` could be an attribute of the cow instead of being an attribute of the relationship and still capture the vaccination information in requirements.

   (D) A disease can possibly be treated by more than one vaccine.

   (E) It is not always possible to find out when a particular cow was vaccinated using a particular vaccine as only the last vaccination date is available in `vaccdate`.

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

   (A) `milkingSchedule` should not have a participation constraint as some cows may never have been milked for the first time yet.

   (B) We can find the total amount of milk produced by the cows in the farm, but not the total milk quantity for each breed in the farm.

   (C) It is possible to tell whether a cow's milk production is decreasing based on the data captured by the ER.

   (D) `purity` and `qty` should not be an attribute of `milkingSchedule` as it is a weak entity. Instead they should be part of the milking relationship.

   (E) If Joe decides to milk some cows at *midnight* and keep track of it, it will not require changes to the ER.

3. (5 Points) Which of the following relational algebra queries will help Joe determine the distinct names of vaccines ever given to *Holstein* breed of cows but never to *Jersey* breed of cows ? (choose 2 correct answers)

   (A) $\Pi_{vaccName}(\sigma_{breed='Holstein'}(cow \bowtie vaccinationSchedule \bowtie vaccination)) -$
   $\Pi_{vaccName}(\sigma_{breed='Jersey'}(cow \bowtie vaccinationSchedule \bowtie vaccination))$

   (B) $\sigma_{breed='Holstein'}(\Pi_{vaccName}(cow \bowtie vaccinationSchedule \bowtie vaccination)) \cap$
   $\sigma_{breed='Jersey'}(\Pi_{vaccName}(cow \bowtie vaccinationSchedule \bowtie vaccination))$

   (C) $\Pi_{vaccName}(\sigma_{breed='Holstein'}(cow \bowtie vaccinationSchedule)) \cap$
   $\Pi_{vaccName}(\sigma_{breed='Jersey'}(cow \bowtie vaccinationSchedule))$

   (D) $\sigma_{breed='Holstein'}(cow \bowtie \Pi_{vaccName}(vaccinationSchedule) \bowtie vaccination) -$
   $\sigma_{breed='Jersey'}(cow \bowtie \Pi_{vaccName}(vaccinationSchedule) \bowtie vaccination)$

(E) $\Pi_{vaccName}(\sigma_{breed='Holstein'}(cow \bowtie) \bowtie vaccinationSchedule) -$
$\Pi_{vaccName}(\sigma_{breed='Jersey'}(cow \bowtie) \bowtie vaccinationSchedule)$

4. (5 Points) For the table `milkingSchedule`, assume that `cowid`, `qty`, `purity` fields are stored using 16 bit integers, `milkDate` occupies 4 bytes and `partOfDay` has an average length of 7 bytes. The row header is another 3 bytes. (This information gives you the average size of a record).
What is the expected maximum number of records that could be fit into a page of size 8000 bytes, accounting for the slot directory overhead ? Size of each slot is 32 bits, including the slots used to store the total number of slots and pointer to start of free space. (Choose 1 correct answer)

(A) 400

(B) 442

(C) 333

(D) 292

(E) 425

Open Questions (45 points)

This section contains open questions based on our example application.

5. (5 Points) There has been reports of *Botulism* among cattle in the region and Joe wants to take a precautionary measure and vaccinate his cattle. Write a relational algebra query to help Joe find a vaccine(s) with the least waiting period that can treat *Botulism*.
   You will receive partial credit (3 points) if instead you chose to write the SQL query.

6. (6 Points) Write a SQL to find the `cowids` of cows that have received the vaccination for *Botulism* but not the shots for *Mastitis* . Use a co-related subquery to solve this problem. You will get 2 points less if you use a different approach.

7. (8 Points) Write a SQL to find the `breed` and `cowid` of the cow(s) that produces the highest average quantity of milk in a day amongst *Shorthorn* and *Holstein* cattle breeds. (To keep it simple, you need to count only the days in which the cow gets milked to compute its average milk output).
   If you are not able to solve this query, for 4 points, write a query that will give you the `cowid`, `breed` and average quantity of milk produced in a day amongst *Shorthorn* and *Holstein* cattle breeds where the average milk produced per day is more than 20 liters (you can assume the `qty` field is stored in liters).

8. (11 Points) Assume a simple scenario where Joe has 100 cows, and each cow is milked twice a day and that every cow is milked everyday, starting from the very first day.
   `milkingSchedule` has an average record size of 20 bytes. Data/Index page size is 4000 bytes and is on an average around 75% full.
   Further, we have created an unclustered index on `milkDate` that follows the type II indexing scheme of using rid-list . `milkDate` is 4 bytes and rids are 5 bytes long.

   Note:- If you write down steps, you might get partial marks for them even if your numerical calculation is wrong.

   (a) (2 Points) On an average, how many records are in a data page for `milkingSchedule` ?

   (b) (2 Points) How many rids are there on an average for a data entry for the index ?

   (c) (2 Points) On an average approximately how many data entries are in an index leaf page ?

   (d) Approximately how many days of milking should pass before a query on `milkingSchedule` for a specific `milkDate` value starts definitely benefiting by using the index ? (i.e. results in less IO compared to scanning the entire table). Assume independent uniform distribution of data records across all the pages. (i.e a given record can be in any page irrespective of any of its attribute values). You can also assume that all Index root and intermediate pages are in memory and IO cost for index is only due to access to leaf pages. (5 points)

9. (15 Points) Joe has decided to team up with his fellow dairy farmers and form a co-op “cowbec co-op”. This will help them collectively keep track of their cattle herds. As part of this, the database has to be enhanced. Here are the additional data requirements.

   - Each farmer will be given a unique farmerid by the co-op. In addition to this, we need to keep track of their name and address. You need to have a farm to be part of the co-op.
   - Some farmers also own multiple dairy farms, though each farm is only (and must definitely) owned by a single farmer. Additionally, a farm can be of type either *grassfed* or *regular* (i.e possibly a corn fed-factory style setting). This needs to be kept track as well. Farms also have a farm name associated with them, which thankfully is unique to each farm. A newly opened farm may not have any cows yet.
   - Most importantly, each farmer has their own tagging mechanisms, while each cow still gets tagged with an id, its id is unique only with respect to a given farm. We need to be able to still identify to which farm a cow belongs.

   (a) (10 Points) Debbie is unfortunately away, and you have to help Joe. Draw a modified ER diagram to capture the additional information.
   You **ONLY** need to draw the new entity sets, new relationships and their respective attributes and the existing entity sets (without their attributes) 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 with all its attributes.**
   Do not add any attributes other than what is given in the (old or new) data requirements. You may pick a reasonable name for your attributes.
   Try to find a way of making the ER changes without introducing any new artificial keys of your own.You will lose 3 points if you do not follow this instruction.

   Note:- Where required, make sure your “thick lines” in the diagram are evidently thicker than your thin lines. You may draw double lines if it is easier to do so.

   (b) (5 Points) Write the relational model for any new/modified relations resulting from your ER changes. Remember to also review existing relational model to see if you need to make any changes to it. If a relation changes, you need to write it down completely along with any foreign keys etc that it has (new and old). Do not forget to underline primary keys and write out the foreign key references.

Q1 B,D
Q2 C,E
Q3 A,E
Q4 C

Q5

(5 Points)

$$\rho(v1, \sigma_{disease='Botulism'}(vaccination))$$
$$\rho(v2, v1)$$
$$\Pi_{vaccName}(v1) - \Pi_{v1.vaccName}(v1 \bowtie_{v1.waitperiod>v2.waitperiod} v2)$$

OR (3 Points)

```
SELECT vaccName
FROM Vaccination
WHERE disease = ‘Botulism’
  AND waitperiod <= ALL ( select waitperiod
                       FROM Vaccination
                       WHERE disease = ‘Botulism’ )
------------------------------------------------------

SELECT vaccName
FROM Vaccination
WHERE disease = ‘Botulism’
  AND waitperiod <= ( select min(waitperiod)
                       FROM Vaccination
                       WHERE disease = ‘Botulism’ )
```

Q6 (6 Points)

```
SELECT s.cowid
FROM vaccinationSchedule s, vaccination v
WHERE s.vaccName = v.vaccName
AND v.disease = 'Botulism'
AND NOT EXISTS
(
  SELECT s2.cowid
  FROM vaccinationSchedule s2, vaccination v2
  WHERE s2.vaccName = v2.vaccName
    AND v2.disease = 'Mastitis'
    AND s2.cowid = s.cowid
)
```

OR (4 Points)

```
SELECT s.cowid
FROM vaccinationSchedule s, vaccination v
WHERE s.vaccName = v.vaccName
AND v.disease = 'Botulism'
AND s.cowid NOT IN
(
  SELECT s2.cowid
  FROM vaccinationSchedule s2, vaccination v2
  WHERE s2.vaccName = v2.vaccName
    AND v2.disease = 'Mastitis'
)
```

Q7 (8 Points)

```
SELECT c.cowid, c.breed
FROM cow c, (SELECT cowid, sum(qty)dqty
             FROM MilkingSchedule
             GROUP BY cowid, milkdate
             )  m
WHERE c.cowid = m.cowid
  AND c.breed IN (‘Shorthorn’, ‘Holstein’)
GROUP BY cowid, breed
HAVING avg(dqty) >= ALL
(
  SELECT avg(dqty)
  FROM Cow C,
      (SELECT cowid, sum(qty)dqty
       FROM MilkingSchedule
       GROUP BY cowid, milkdate
      ) M
  WHERE C.cowid = M.cid
    AND c.breed IN (‘Shorthorn’, ‘Holstein’)
  GROUP BY cowid
)
```

OR (4 Points)

```
SELECT cowid, breed, avg(m.dqty)
FROM cow c, (SELECT cowid, sum(qty)dqty
             FROM MilkingSchedule
             GROUP BY cowid, milkdate
             ) m
WHERE c.cowid = m.cowid
  AND c.breed IN (‘Shorthorn’, ‘Holstein’)
GROUP BY cowid, breed
HAVING avg(dqty) >= 20
```

Q8 (11 Points)

(1) 4000 / 20 x 0.75 = 150 records per page.

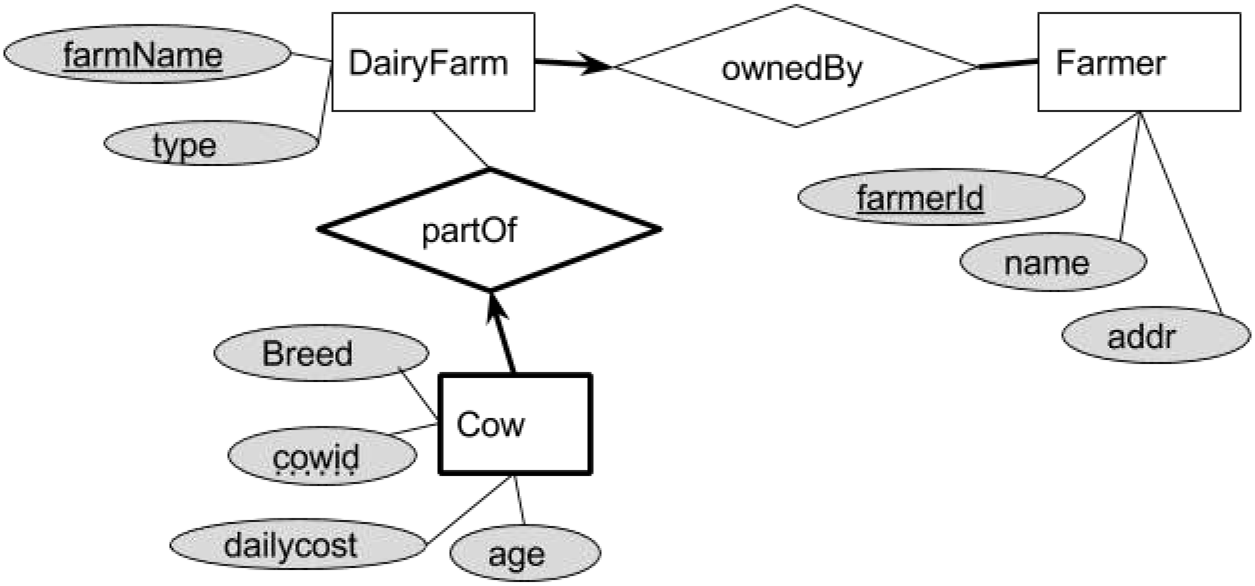
(2) 100 cows x 2 times a day = 200 per **milkDate** (i.e per data entry)

(3) Index entry size = 4+5x200 = 1004
4000/1004 x 0.75 = approx 3 data entries per index leaf page

(4) Number of records in a day inserted into **milkingSchedule** (from ANS(2)) = 200
Therefore a query for a specific **milkDate** may have to access 200 different data pages on a worst case.
From ANS(1), each day's milking activity requires 200/150 = 4/3 pages.
Number of days to get to 200 pages = 200 / (4/3) = 150 days. (or 151 days accounting for the IO of the index leaf page itself)

Q9

Joe’s Dairy Farm



Cow(cowid,farmName, dailycost, Breed, age) farmName Ref DairyFarm

DairyFarm(farmName, type, farmerId) farmerId Ref Farmer

Farmer(farmerId, name, addr)

milkingSchedule(cowid,farmName, qty, purity, milkDate, partOfDay) cowid,farmName Ref Cow

vaccinationSchedule(cowid,farmName, vaccName, vaccDate) cowid,farmName Ref Cow, vaccName Ref vaccination