ECE 356 F17 Midterm Examination

INSTRUCTOR: Prof. Wojciech Golab
DATE: October 27th, 2017 at 3:30pm

CLOSED BOOK, NO AIDS
TIME ALLOWED: 75

INSTRUCTIONS:

(1) **Print** your name and student number at the top of **all pages** and on the cover page **below**.

(2) The examination paper is **8** pages long (including this cover page). Check that you have all of them and let the proctors know **immediately** if you are missing any pages.

(3) Answer the questions in any order. Read an **entire** question before you begin to answer it.

(4) Write your answer in the space provided. If you need more space for rough work, you may use the empty pages at the end of the exam. **Write legibly and show all your work**.

(5) Please **do not separate** the pages of this exam.

(6) If required for a given question, or if in doubt, **clearly state any reasonable assumptions** and then proceed to complete the question.

(7) The number in square brackets [...] after each problem statement denotes the **weight** assigned to that question.

(8) Grading will take into account the correctness, clarity and simplicity of your solutions, as well as the soundness of any assumptions you make. Course staff will not comment on your solutions while the exam is in progress.

(9) Questions vary in difficulty and weight. Do not get stuck on one question. Good luck!

Student name (please print): ________________________________

Student number: ____

Student signature: ______________________________________

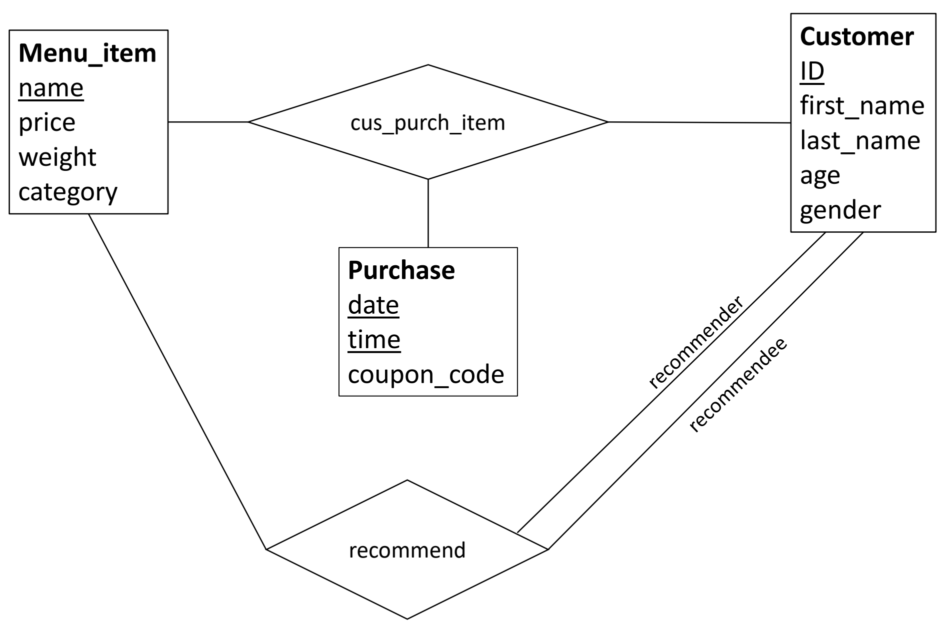
Q1	Q2	Q3	Q4	Q5	Q6		Total
5	5	12	5	5	4		36

Question 1: Entity-relationship model [5 points]

A restaurant would like to construct a database of their menu items customers. The following requirements have been identified:

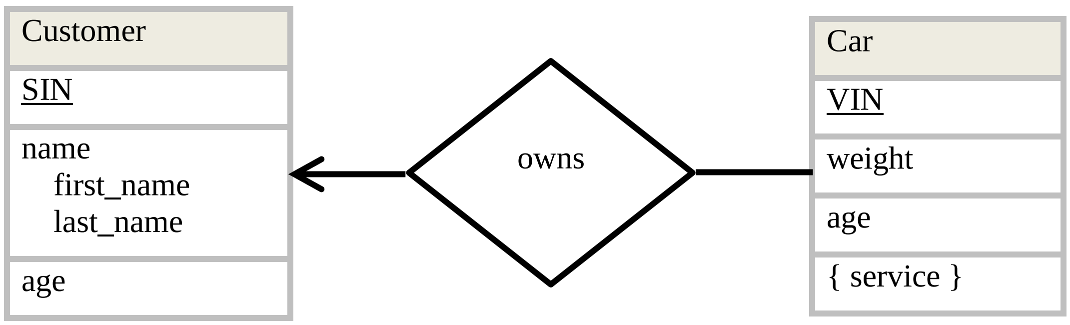
- Each menu item has a unique name, price, weight, and category (e.g., appetizer, entrée, dessert).
- Each customer has a unique ID, first name, last name, age, and gender.
- A customer may purchase a menu item, possible more than once on different dates or at different times.
- Each purchase is associated with a coupon code.
- A customer may recommend a menu item to another customer.

Design an entity-relationship (ER) diagram to model this data set. Indicate clearly the following details: entity sets, weak entities (where appropriate), relationship sets, primary keys, discriminators, cardinality constraints, and participation constraints.



Question 2: Relational model [5 points]

Using the procedure taught in lecture, translate the following ER diagram to a set of relations. For each relation state the schema (i.e., name the columns but ignore their domains), choose an appropriate primary key, and state any relevant foreign key constraints.



Relations:

1. Customer (SIN, first_name, last_name, age)
2. Car (VIN, weight, age)
3. CarService (VIN, service)
4. Owns (SIN, VIN)

Foreign keys:

CarService.VIN references Car.VIN

Owns.SIN references Customer.SIN

Owns.VIN references Customer.VIN

Question 3: Query languages [12 points]

Consider the following relational schemas used to describe the data maintained by a pet hospital.

Pet (<u>pid</u>, name, age, species, breed)
Owner (<u>oid</u>, first_name, last_name, age, address, phone, email)
Owns (<u>pid</u>, <u>oid</u>)
Owns.pid is a foreign key that references Pet.pid
Owns.oid is a foreign key that references Owner.oid

Answer the following questions using the indicated query languages. (SQL denotes Structured Query Language, and RA denotes Relational Algebra.) Let *pet*, *owner*, and *owns* denote instances of the relations Pet, Owner, and Owns, respectively.

Part (a) [2 points]: Using RA, find phone numbers of owners who do not own any cats.

$$\Pi_{Owner.phone}\Big(\big(\Pi_{oid}(Owner) - \Pi_{oid}(\sigma_{species = 'cat'}(Pet) \bowtie Owns)\big) \bowtie Owner \Big)$$

Part (b) [2 points]: Using SQL, find the distinct emails of owners who own exactly one pet.

```
SELECT DISTINCT email FROM Owner NATURAL JOIN
(SELECT oid, COUNT(*)
 FROM Owns
 GROUP BY oid
 HAVING COUNT(*) = 1)
```

Part (c) [2 points]: Using SQL, count the number of pets for each species represented in the database. Output the species and counts only, and sort in descending order by species.

```
SELECT species, COUNT(*)
FROM Pet
GROUP BY species
ORDER BY species DESC
```

Part (d) [2 points]: Using RA, find the name of every cat that has an owner who also owns a dog.

$temp1 \leftarrow \Pi_{oid} (Owns \bowtie (\sigma_{species = 'cat'} (Pet))) \cap \Pi_{oid} (Owns \bowtie (\sigma_{species = 'dog'} (Pet)))$
$\Pi_{name} (temp1 \bowtie Owns \bowtie \sigma_{species = 'cat'} (Pet))$

Note: Using set intersection

Part (e) [3 points]: Using SQL, find the distinct oids of owners who own the maximum number of pets. (E.g., if the maximum number of pets per owner is N, then find the oids of owners who own exactly N pets. Return all such owners if there is more than one.)

```
SELECT oid FROM Owner A
WHERE NOT EXISTS
    (SELECT oid FROM Owner B
     WHERE (SELECT COUNT(pid) FROM Owns WHERE Owns.oid = B.oid) >
           (SELECT COUNT(pid) FROM Owns WHERE Owns.oid = A.oid)
    )
```

Alternative solution using aggregation:

```
SELECT oid
FROM (
    SELECT oid, COUNT(*)
    FROM Owns
    GROUP BY oid
    HAVING COUNT(*) = (
        SELECT MAX(Cnt)
        FROM (
            SELECT oid, COUNT(*) AS Cnt FROM Owns GROUP BY oid
        )
    )
)
```

Part (f) [1 point]: What SQL referential action would ensure that an SQL DML statement cannot delete an Owner tuple if the database still contains at least one pet owned by that owner?

ON DELETE RESTRICT (or NO ACTION)

Question 4: Functional Dependencies [5 points]

Let R denote a relation schema with four attributes: (A,B,C,D). Consider the following set of functional dependencies over the attributes of R:

$F = \{ \quad AB \rightarrow D, \quad C \rightarrow B \quad \}$

Part (a) [1 point]: Compute the attribute closure $(CD)^+$ with respect to F:

$(CD)^+ = BCD$

Part (b) [2 points]: Prove using Armstrong's Axioms (and only those axioms) that F logically implies $ABC \rightarrow BD$. Show your work, and for each step state clearly which axiom you are applying. (Hint: Armstrong's Axioms are *reflexivity*, *augmentation*, and *transitivity*.)

(1)	$AB \rightarrow D$	[given in F]
(2)	$ABC \rightarrow CD$	[(1) augmented with C]
(3)	$C \rightarrow B$	[given in F]
(4)	$CD \rightarrow BD$	[(3) augmented with D]
(5)	$ABC \rightarrow BD$	[transitivity on (2) and (4)]

□

Part (c) [2 points]: Determine whether F is logically equivalent to $F' = \{ ABC \rightarrow D, \quad C \rightarrow B \}$.

1) Determine whether F logically implies F'.

   Using F: $(ABC)^+ = ABCD$, therefore F implies $ABC \rightarrow D$.

   F also implies $C \rightarrow B$ since F contains this functional dependency .

   Conclusion: yes, F logically implies F'.

2) Determine whether F' logically implies F.

   Using F': $(AB)^+ = AB$, therefore F' does not logically imply $AB \rightarrow D$

   Conclusion: no, F' does not logically imply F

Final answer: F is not logically equivalent to F'.

Question 5: Normalization [5 points]

Let R denote a relation schema with four attributes: (A,B,C,D). Assume that R is in 1NF. Consider the following set of functional dependencies over the attributes of R:

F = { $AB \rightarrow D$, $B \rightarrow C$ }

Part (a) [2 points]: Find all the candidate keys for R with respect to F. Show your work in detail and state your final answer clearly.

Observation: since A and B only appear on the left side of functional dependencies, they must both be included in every candidate key.

$(AB)^+$ = ABCD implies that AB is itself a candidate key.

<u>Conclusion: AB is the only candidate key.</u>

Part (b) [3 points]: Determine whether R is in 3NF. If not, then perform a 3NF decomposition using the algorithm taught in lecture. Show your work in detail.

Step 0: R is not in 3NF because of $B \rightarrow C$, where B is not a superkey and C is a non-prime attribute.

Step 1: Canonical cover. There are no extraneous attributes in the functional dependencies (FDs) in F. Furthermore, the union rule does not apply to any pair of FDs in F. Thus, F is its own canonical cover.

Step 2: decompose R into R_1(A, B, D) and R_2(B, C) using the canonical cover.

Step 3: R_1 contains a candidate key, and so no additional relations need to be added.

Step 4: remove redundant relations – nothing to do.

<u>Final answer:</u> R is decomposed into R_1(A, B, D), R_2(B, C)

Question 6: Storage and Indexing [4 marks]

Part (a) [2 marks]

A RAID system is being constructed using 10 identical 1TB hard disks. The raw storage capacity of this system is exactly 10TB. Each disk individually is able to perform 100 random I/O operations per second.

Estimate the maximum throughput (in terms of files read or written per second) of the RAID system in the following cases:

i. RAID 5 is used, the workload writes a large number of very small files
(write down the formula and plug in numbers but do not simplify)

(10 disks × 100 iops/disk) / (4 iops per file) [small write penalty!]

ii. RAID 10 is used, the workload reads a large number of very small files
(write down the formula and plug in numbers but do not simplify)

(10 disks × 100 iops/disk) / (1 iop per file)

Part (b) [2 marks]

This question refers to the schema from Question 3 (Pet/Owner/Owns).

What index or indexes would the MySQL InnoDB engine (same version as in the lab environment) create automatically for the Owner table? For each index state the search key.
[1 mark]

There will be an index on the primary key: Owner.oid

What index would be most effective in speeding up a query that finds all dogs whose age is two years or less? Answer by stating the search key, as well as the relation on which the index is defined. [1 mark]

An index on (Pet.species, Pet.age) would be most effective.