NAME: ____________________ ID: __________

CSC 343 H5F – SOLUTIONS

Day Class 01 Instructor: Dr. Michael Liut

DURATION: 2 hours

University of Toronto Midterm Examination October 30, 2019

Read all instructions before starting the exam

This test paper includes 19 pages and a total of 20 questions; 10 true-or-false questions, 6 fill-in-the-blank question, and 4 questions requiring a written answer. You are responsible for ensuring that your copy of the paper is complete. Bring any discrepancy to the attention of your invigilator.

question(s)	mark	out of
1 – 10		15
11 – 16		6
17		6
18		6
19		15
20		6
misc		n/a
total		54

Special Instructions :

1. The answers to the written questions (Questions 17-20) must be answered in the space provided (i.e. immediately following the question). Written answers are to be clear and concise; illegible solutions will not be marked.
2. Read all questions before starting the exam – some are easier than others.
3. The True-or-False section requires you to illustrate the validity by an example, or disprove the validity by a counter-example. If your justification is proof by definition, this is perfectly valid!
4. **This is a closed book exam.** No memory aids, notes, calculators, or, textbooks of any kind are allowed during the test. The use of any electronic device is strictly prohibited. Failure to comply will result in your immediate dismissal from the examination and a grade of 0.
5. Students are not allowed to be involved in any communication of any kind. Questions concerning this exam must be brought to the attention of the invigilator(s) or the instructor. Any attempt of alternative communication will be considered a case of academic dishonesty.
6. No unauthorized scrap paper is allowed to be used. The invigilator(s) will supply every student with needed scrap paper when asked. An additional mark will be awarded to students who circle this line; do not look around, just do it.
7. Documents to be returned: this questionnaire and all scrap paper if used. All of these documents must bear the student's name and number. Only the face page of the questionnaire needs to bear the student's name and number.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Question 1 – 10 are True-or-False questions. You are required to illustrate the validity by an example, or disprove the validity by a counter-example.

For each question only select one answer (i.e. True or False). To select an answer, you must circle the word "True" or the word "False". The negative marking scheme is not used. Incorrect or missing answers earn a mark of 0. Therefore, do not leave any questions blank. For full marks you must justify (prove/disprove) your answer.

Question 1 [1.5 marks]

Let's assume that we are working with subset of a banking schema.[1] Given two of the entity-sets: `Customer` and `Loan`, and their relationship `Borrower`, it is fair to assume that there is a partial participation between `Customer` ········ `Borrower` and clear to assume a total participation between `Borrower` —— `Loan`.[2]

⟹A. True.

B. False.

TRUE. Not every bank customer will have a loan, however, every Loan will have an associated bank customer (i.e. "borrower").

Marking Scheme:

[0.5 marks for identifying this statement is TRUE]
[1 mark for justifying that the statement is TRUE]
[mark of 0 if student identifies the answer as being FALSE]

Question 2 [1.5 marks]

Let's assume that `Borrower`, from Question 1, has a relationship `Interest` with a weak entity-set called `Rate`. In this case `Interest` is the identifying relationship and `Rate` is the amount of yearly interest applied to a specific loan (in `Borrower`). Given this logical construct, when converting from the ERD to DDL it would collapse `Rate` and `Interest` into one table with `Borrower`.

A. True.

⟹B. False.

[1] a banking schema is comprised of several divisions of banking, e.g., personal, loans, investments, etc.

[2] definition: ········ indicates partial participation and —— indicates total participation

FALSE. The collapse would only combine `Rate` and `Interest`, not all three. ALSO, `Interest` cannot be the relationship of another relatioship. EITHER ARE OK JUSTIFICATIONS.

Marking Scheme:
[0.5 marks for identifying this statement is FALSE]
[1 mark for justifying that the statement is FALSE]
[mark of 0 if student identifies the answer as being TRUE]

Question 3 [1.5 marks]
Given two relations R and S, the Cartesian Product is denoted $R \times S$, where R and S are sets: $R \times S = \{(r, s) : (r \in R) \text{ and } (s \in S)\}$. Thus, in general, $(R \times S) = (S \times R)$ (i.e. the Cartesian Product operation is commutative).

A. True.
⟹B. False.

Answer:

$$R \times S = (r, s) : (r \in R) and (s \in S)$$

Note: The Cartesian Product of two sets is a set, and the elements of that set are ordered pairs. In each ordered pair, the first component is an element of `R`, and the second component is an element of `S`.

For example, let:
$R = 1, 2, 3$ and $S = (w, x), (y, z)$,
$R \times S = (1, 2, (w, x)), (1, 2, (y, z)), (3, (w, x)), (3, (y, z))$.

If `R` and `S` are both finite sets, then $|R \times S| = |R| \cdot |S|$ as there are $|R|$ choices for the first component of each ordered pair and also $|S|$ choices for the second component of the ordered pair. Therefore, Cartesian Product is **not commutative** for the sets `R` and `S`. This is justified in the example above proving that $(R \times S)! = (S \times R)$.

FALSE.
Marking Scheme:
[0.5 marks for identifying this statement is FALSE]
[1 mark for justifying, proof above, that the statement is FALSE]
[mark of 0 if student identifies the answer as being TRUE]

Question 4 [1.5 marks]
Physical data independence separates the physical (refers to the hardware-level/system design) and conceptual schemas. Logical data independence separates the conceptual schema from the external schema, which depends on relations in the conceptual schema.

⟹A. True.
B. False.

TRUE. Proof by definition.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE]
[1 mark for justifying that the statement is TRUE]
[mark of 0 if student identifies the answer as being FALSE]

Question 5 [1.5 marks]
By definition, the *primary key* of a strong entity-set must be both a *candidate key* and a *superkey* of that relation. Further, given all *superkeys* for that same relation, it is possible that more than one key be uniquely minimal.

⟹A. True.
B. False.

TRUE. The first sentence is true by definition. The second sentence is true as you can have multiple candidate keys (e.g. student number, student email, and UtorID) that all uniquely (and minimally) identify the tuples.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE]
[1 mark for justifying that the statement is TRUE]
[mark of 0 if student identifies the answer as being FALSE]

Question 6 [1.5 marks]
Integrity constraints ensure that changes made to the database by authorized users do not result in a loss of data consistency. The following are three examples[3] of such constraints:

[3] built off of our University schema discussed in lecture

1. An instructor name cannot be `NULL`.
2. No two instructors can have the same instructorID.
3. The budget of a department must be greater than $0.00.

⟹A. True.

B. False.

TRUE. By definition. All of these are valid integrity constraints.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE]
[1 mark for justifying that the statement is TRUE]
[mark of 0 if student identifies the answer as being FALSE]

Question 7 [1.5 marks]
The database term `search` is a precise means of acquiring accurate information utilizing SQL, while a `query` is a less precise abstraction of SQL. Overall, a `query` is a pseudo-subset[4], of a `search`, that more often than not generates less accurate information.

A. True.

⟹B. False.

FALSE. Totally and utterly false. A search is based on keyword matching and often its results are ranked based on: popularity, reputation, and paid advertisements. A query is a request of information from a database utilizing SQL; generally designed for a more specific result than those in a search.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE/FALSE]
[1 mark for justifying that the statement is TRUE/FALSE]
[mark of 0 if student identifies the answer as being TRUE/FALSE]

[4] a resembling or imitating portion/part of a larger group

<u>**General Scenario**</u>

Given three entity-sets E_1, E_2, and E_3 with one ternary connecting relationship R_1:

> E_1 and E_2 are said to be strong as they both have *primary keys*: $E_1 : \{\text{aid}\}$ and $E_2 : \{\text{bid}\}$. E_3 is said to be weak as it does not have a means to uniquely identify itself, thus, utilizing R_1 to assist in identifying its tuples. Let's say that E_3's *primary key* is $\{E_1.\text{aid}, E_2.\text{bid}, \text{attr}\}$, where `attr` represents an arbitrary attribute belonging to E_3.

Question 8 [1.5 marks]

Given the general scenario, above:

> During the conversion from ERD to DDL the following relations will be created: E_1, E_2, E_3, and R_1. All tuples in all relations are uniquely accessible.

A. True.

$\Longrightarrow$B. False.

FALSE. During the conversion from ERD to DDL E_3 will collapse into a combined relation with R_1 and have all its tuples uniquely accessible. E_1 and E_2 will have their own relations, also with uniquely accessible tuples.

Marking Scheme:

[0.5 marks for identifying this statement is FALSE]

[1 mark for justifying that the statement is FALSE]

[mark of 0 if student identifies the answer as being TRUE]

Question 9 [1.5 marks]

In addition to the general scenario, above, let's assume we have a second weak entity-set E_4 that is connected to E_3 by a relationship R_2.

> E_3 can exist under this schema but must have total participation with R_1 to uniquely identify its tuples. E_4 can exist under this schema but must have total participation with E_3 and R_2 to uniquely identify its tuples.

A. True.

$\Longrightarrow$B. False.

FALSE. E_3 must have total participation to uniquely ID its tuples, as it is using the PKs of E_1 and E_2 as FKs to do so. Where as E_4's tuples will be inaccessible as a weak entity-set must rely on the identifying relationship between itself and a strong entity-set to uniquely ID its tuples (and naturally exist). Furthermore, E_4 cannot exist under this schema.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE/FALSE]
[1 mark for justifying that the statement is TRUE/FALSE]
[mark of 0 if student identifies the answer as being TRUE/FALSE]

Question 10 [1.5 marks]
The main construct of representing data in the relational model is a *relation*. A relation consists of: a *relational instance*, which is the table; and a *relational schema*, which describes the column headers of a given table.

$\Longrightarrow$A. True.
B. False.

TRUE by definition.

Marking Scheme:
[0.5 marks for identifying this statement is TRUE]
[1 mark for justifying that the statement is TRUE]
[mark of 0 if student identifies the answer as being FALSE]

Question 11 – 16 are fill-in-the-blank questions.

For each blank only provide one answer. Incorrect or missing answers earn a mark of 0. Therefore, do not leave any questions blank.

Question 11 [**1 mark**]

The number of rows in a relation can be defined by its ______________________ and the number of columns can be defined by its ______________________.

Answer: cardinality and degree/arity
[0.5 marks for each correct blank]

Question 12 [**1 mark**]

______________________ time defines the time period during which a row is committed to or recorded in the database, while ______________________ time defines the time period during which a row is regarded as correctly reflecting reality by the user of the database.

HELP! Here is a mini-word bank for question 12: {atomic, civil, cyclical, eastern-standard, linear, poisson, proportional, temporal, transaction, thanos, valid, zeus}.

Answer: Transaction and Valid. [0.5 marks for each correct blank]

NOTE: if they select "temporal" a partial grade will be awarded.
[0.25 marks for each partially-correct blank]

Question 13 [**1 mark**]

Given two relations `R` and `S`, when joined there exists a tuple of `R` which has no natural match in `S`. That tuple is said to be ______________________. Further, that tuple may be joined, but will be ______________________ by `NULL`.

Answer: Dangling and Padded.
[1 mark for the correct blank]

Question 14 [**1 mark**]

A ______________________-level trigger will be activated one time (even if no rows are updated).
A ______________________-level trigger will be activated millions of times (dependent on the number of iterations).

Answer: statement-level and row-level
[0.5 marks for each correct blank]

Question 15 [1 mark]

Given two relations `R` and `S`; `R` has n rows and c columns, and `S` has m rows and k columns. On output this means that `R` $\times$ `S` will produce $n \times m$ ________________ and $c + k$ ________________.
Answer: Rows, Columns.

[0.5 marks for each correct blank]

Question 16 [1 mark]

A database ______________________ is the skeleton structure that represents the logical view of the entire database, where as ______________________ is a term used to describe one of the fundamental uses of DBMSs. It also allows us to treat a relationship set as an entity set for the purposes of participation in (other) relationships.

Answer: Schema and Aggregation.
[0.5 marks for the correct blank]

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR SCRAP

Questions 17 – 20 are questions that require a written answer. The answers are to be written in the space provided below each question.

Question 17 [**6 marks**] In reference to our in-class running example of `Student`, `Enrolled`, and `Course`, let's consider the following `Enrolled` relation:

studentID	courseID	grade	attendance
101	CSC343	63	66
102	CSC309	79	88
102	CSC343	91	97
103	CSC324	69	70
103	STA302	90	82
104	CSC411	91	65
105	CSC343	77	70
106	CSC343	80	83

Primary Key: {`studentID, courseID`}

Given `Enrolled`, above, write two independent SQL statements[5] (on the following page, where indicated) that produces the following tables:

Table 1:

Student Number	Course Code	Final Grade
101	CSC343	63
103	CSC324	69
104	CSC411	91

Table 2:

Student Number	Course Code	Final Grade
102	CSC343	91
103	STA302	90

[5] this must be a dynamic statement, not specific to this dataset.

THIS PAGE IS INTENTIONALLY LEFT BLANK FOR Q17 ANSWER

Answers:

Query 1

```
SELECT studentID AS ‘Student Number’,
  courseID AS ‘Course Code’,
  grade AS ‘Final Grade’ FROM Grades
WHERE attendance <=70 AND (grade>90 OR grade<70);
```

Query 2

```
SELECT g.studentID AS ‘Student Number’,
  g.courseID AS ‘Course Code’,
  g.grade AS ‘Final Grade’
FROM (SELECT * FROM Grades GROUP BY studentID HAVING COUNT(studentID)>1) g
WHERE g.grade >= 90;
```

Question 18 [6 marks]

Consider the relations `Students`, `Faculty`, `Courses`, `Rooms`, `Enrolled`, `Teaches`, and `MeetsIn`:

`Students`(sid: string, name: string, login: string, age: integer, gpa: real)
`Faculty`(fid: string, fname: string, sal: real)
`Courses`(cid: string, cname: string, credits: integer)
`Rooms`(rno: integer, address: string, capacity: integer)
`Enrolled`(sid: string, cid: string, grade: string)
`Teaches`(fid: string, cid: string)
`MeetsIn`(cid: string, rno: integer, time: string)

(a) [3 marks] List all the foreign key constraints among these relations (where the foreign keys are and what relation they actually belong to).

(b) [1 mark] Circle all the relations that are entity-sets above.

(c) [1 mark] Put an asterisks (i.e. *) beside the relations that are relationships above.

(d) [1 marks] Give an example of a (plausible) constraint involving one or more of these relations. This constraint cannot be: a primary key constraint, a foreign key constraint, nor any type of domain-level constraint.

Answer:

(a) 1 mark for identifying that: `Enrolled`, `Teaches`, and `MeetsIn` all of FKs that belong to other tables.
`Enrolled`: sid belongs to `Students` and cid belongs to `Courses`.
`Teaches`: fid belongs to `Faculty` and cid belongs to `Courses`.
`MeetsIn`: cid belongs to `Courses` and rno belongs to `Rooms`.

(b) 1 mark for correctly identifying all of them, 0.5 marks for only identifying 3/4, 0 marks otherwise. The entity-sets are: `Students`, `Faculty`, `Courses`, and `Rooms`.

(c) 1 mark for correctly identifying all of them, 0.5 marks for only identifying 2/3, 0 marks otherwise. Relationships are: `Teaches`, `Enrolled`, and `MeetsIn`.

(d) 1 mark for giving an plausible constraint. NOTE THAT THIS CANNOT BE: a primary key constraint, a foreign key constraint, nor any type of domain-level constraint.

Question 19 [15 marks]

You have been hired as the database architect to design the `UTM Bank`'s entity-relationship diagram (ERD), whereby your superiors have given you extremely stringent[6] requirements[7]. The first of requirement being that this ERD <u>must be</u> constructed in Chen's Notation, so a reference sheet on page 17 has been attached for your convenience.

The UTM Bank has both `Customers` and `Employees`, both of which are naturally independent of one another. A Customer has a unique customer identifier, in addition to: a name, a date of birth, and assets (i.e. a dollar figure of the total assets they own). Every Customer is assigned exactly one personalized Employee to act as their banker. This banker is designated a type. An Employee has a unique identifier, a name, a start date (i.e. when they started working at the bank), and a set of dependents (i.e. children) which they may or may not have. Further, some employees are just workers, working for the manager (of which there is exactly one).

Customers can perform two actions: the first is to borrow money in the form of a `Loan`, and the second is to be a depositor of a certain `Account`. Accounts all have an account number and a balance. Accounts are broken into two disjoint types: a saving account and a chequing account. What distinguishes them is that a chequing account has an overdraft limit, while a savings account has an interest rate associated to it. Every account belongs to a specific `Branch`, which is what makes them uniquely identifiable. Branches are uniquely identified by their name, but also have a location and amount of assets (i.e. accumulated dollar figure to the total amount in assets they hold). Loans also belong to a specific branch, however, a loan can be uniquely identified by a loan number. In addition, every loan has an amount which can be paid. A `Payment` must have a payment number (unique per loan, but different loans could have the same payment number – e.g. Joe's first loan payment on his loan would be 0001 and Jane's first loan payment on her loan would be 0001), a date, and an amount.

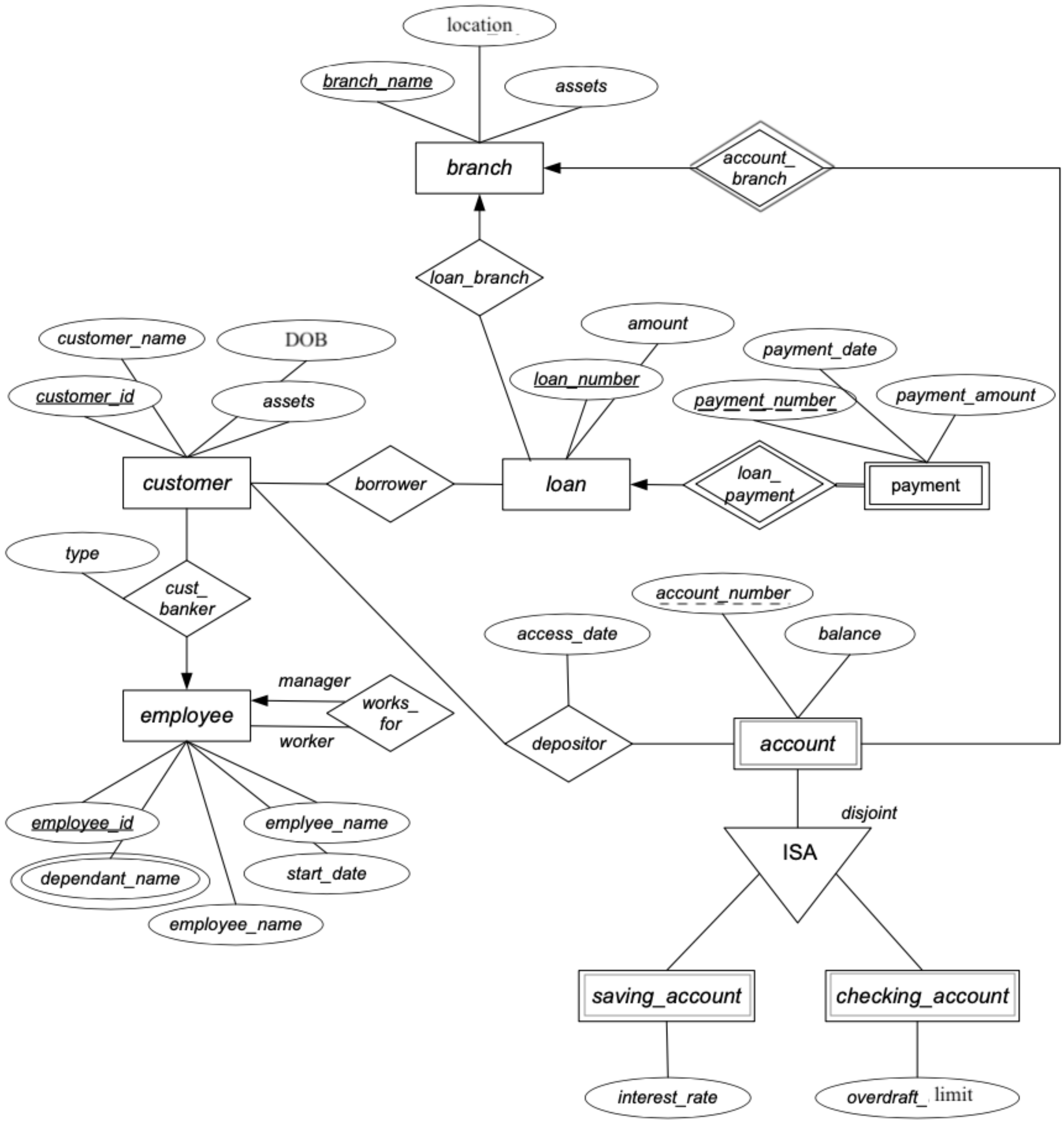
Some additional information: branches all start with a minimum of one account and can have more than one account, there cannot be joint accounts, and there cannot be joint loans.

---

[6]definition: (of regulations, requirements, or conditions) strict, precise, and exacting.

[7]you cannot deviate from the requirements and cannot add your own attributes, entity-sets, relationships, etc.

Answer:


location
branch_name
assets
branch
account_
branch
loan_branch
customer_name
DOB
amount
payment_date
customer_id
assets
loan_number
payment_number
payment_amount
customer
borrower
loan
loan_
payment
payment
type
cust_
banker
account_number
access_date
balance
manager
works_
for
employee
worker
depositor
account
disjoint
ISA
employee_id
emplyee_name
dependant_name
start_date
employee_name
saving_account
checking_account
interest_rate
overdraft_ limit

Grading Scheme:

4 marks for having the correct entity-sets and attributes.

2 marks for having the correct relationships.

2 marks for correctly identifying `Payment` and `Account` as weak entity-sets and for denoting their identifying relationship.

1 mark for correctly identifying the ISA hierarchy of Account.

3 marks for correct multiplicity.

1 mark for correctly identifying the primary keys.

1 mark for correctly identifying that `Employee`'s dependent is a multi-set attribute.

1 mark for identifying that `Works_For` is a relationship connected to itself.

Question 20 [6 marks]
[3 marks] From your ERD in Question 19, write the DDL statement for the `Account` relation. You may assume all other relations have been created.

Answer:

```
CREATE TABLE Account (
  account_number INT NOT NULL,
  balance DECIMAL(10,2) NOT NULL,
  branch_name VARCHAR(255) NOT NULL,
  FOREIGN KEY (branch_name) REFERENCES Branch(branch_name)
  PRIMARY KEY (account_number, branch_name)
);
```

1 mark for having all three attributes (syntax penalties come from this mark)

1 mark for properly completing the FK constraint

1 mark for properly including the PK (syntax penalties come from this mark)

[3 marks] Create a trigger track_assets that updates the Branch's assets each time a new Account is created. You may assume that Branch and its connecting relationship account_branch to Account have already been created. Further, your trigger may omit its functionality to update changes in balances (i.e. we only care about INSERT statements).

Answer:

```
delimiter $$
  CREATE TRIGGER track_assets AFTER INSERT ON Account
    FOR EACH ROW
    BEGIN
      DECLARE currentSum FLOAT;

      SELECT assets FROM Branch WHERE branch_name = NEW.branch_name INTO currentSum;

      SET currentSum = currentSum + NEW.balance;

      UPDATE Branch SET assets = currentSum WHERE branch_name = NEW.branch_name;
    END;
$$
delimiter ;
```

0.5 marks for having the delimiter.

0.5 marks for having For Each Row/BEGIN/END.

0.5 marks for correct declaration of counter/accumulator.

0.5 marks for selecting the value from Branch and inserting into accumulator.

0.5 marks for incrementing the counter/accumulator.

0.5 marks for updating the Branch asset.