1. This is a CLOSED book examination. Personal copies of textbook and class notes cannot be used. However, each student may bring one 8.5 inch by 11 inch double-sided sheet of summary notes, which cannot be shared this sheet with other students. Laptop computers, smart phones and calculators cannot be used.

2. There are 3 questions.

3. Put name, student id, email address, team group number, course id, course name, semester, and year on the cover page.

4. There should be at least one empty seat between adjacent students.

<table>
<thead>
<tr>
<th>Name:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID:</td>
<td></td>
</tr>
<tr>
<td>Email Address:</td>
<td></td>
</tr>
<tr>
<td>Group ID:</td>
<td></td>
</tr>
<tr>
<td>Course ID:</td>
<td>CSCI 4707</td>
</tr>
<tr>
<td>Course Name:</td>
<td>Practice of Database Systems</td>
</tr>
<tr>
<td>Semester:</td>
<td>Spring</td>
</tr>
<tr>
<td>Year:</td>
<td>2016</td>
</tr>
</tbody>
</table>
Q1 (25 pts)

Q1 Part A (12 pts) Consider the following relational tables A and B and answer the questions below:

A:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>John</td>
<td>Mpls</td>
</tr>
<tr>
<td>674</td>
<td>Doe</td>
<td>St Paul</td>
</tr>
<tr>
<td>444</td>
<td>Jane</td>
<td>Mpls</td>
</tr>
</tbody>
</table>

B:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>555</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>123</td>
<td>John</td>
<td>Mpls</td>
</tr>
<tr>
<td>398</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
</tbody>
</table>

Q1A1. Complete the table below to show the result of A PRODUCT B. The first row should contain the column names that would appear in the result. The first 2 columns in this row have been provided as an example.

Note: The result may not fill all rows.

**A PRODUCT B:**

<table>
<thead>
<tr>
<th>A.ID</th>
<th>B.ID</th>
<th>A.Name</th>
<th>A.City</th>
<th>B.Name</th>
<th>B.City</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>555</td>
<td>John</td>
<td>Mpls</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>123</td>
<td>123</td>
<td>John</td>
<td>Mpls</td>
<td>John</td>
<td>Mpls</td>
</tr>
<tr>
<td>123</td>
<td>398</td>
<td>John</td>
<td>Mpls</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
<tr>
<td>674</td>
<td>555</td>
<td>Doe</td>
<td>St Paul</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>674</td>
<td>123</td>
<td>Doe</td>
<td>St Paul</td>
<td>John</td>
<td>Mpls</td>
</tr>
<tr>
<td>674</td>
<td>398</td>
<td>Doe</td>
<td>St Paul</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
<tr>
<td>444</td>
<td>555</td>
<td>Jane</td>
<td>Mpls</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>444</td>
<td>123</td>
<td>Jane</td>
<td>Mpls</td>
<td>John</td>
<td>Mpls</td>
</tr>
<tr>
<td>444</td>
<td>398</td>
<td>Jane</td>
<td>Mpls</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
</tbody>
</table>

Q1A2. Write the Cardinality of the results of following relational expressions involving the above tables A and B.

i) Restriction on A where Name like ‘J%’ - 2

ii) Projection of B on City - 2

iii) Left Outer Join of A and B on A.ID=B.ID - 3

iv) Inner Join of A and B on A.ID=B.ID - 1

v) Inner Join of A and B on A.City=B.City - 4

Commented [RS1]: 7 points if all rows/columns are correct.

Commented [RS2]: 1 point/question. Cardinality is number of rows in the result. Some of you have written down the answer in an AxB format to show both rows and columns. This is incorrect.
Q1: Part B: (8 pts) Using relations A and B from Q1 Part A, show the output of the relational operations below.

Q1B1. (A Difference B) Union (B Difference A):

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>674</td>
<td>Doe</td>
<td>St Paul</td>
</tr>
<tr>
<td>444</td>
<td>Jane</td>
<td>Mpls</td>
</tr>
<tr>
<td>555</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>398</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
</tbody>
</table>

Q1B2. (A Union B) Difference (A Intersect B):

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>674</td>
<td>Doe</td>
<td>St Paul</td>
</tr>
<tr>
<td>444</td>
<td>Jane</td>
<td>Mpls</td>
</tr>
<tr>
<td>555</td>
<td>Bob</td>
<td>Fargo</td>
</tr>
<tr>
<td>398</td>
<td>Jill</td>
<td>Mpls</td>
</tr>
</tbody>
</table>

Q1 Part C (5 pts) Consider the following schema of 3 tables consisting of soccer players from 2 countries, namely US and Canada, and the details of the venues they have scored goals at. Primary keys are underlined and foreign keys are italicized. The primary and foreign keys share the names. US_Player table has five columns, namely Player_No, Player_Name, Team_Name, No_of_Goals, and Venue_Name, where Player_No is the primary key and Venue_Name is a foreign key. Canadian_Player table has five columns, namely Player_No, Player_Name, Team_Name, No_of_Goals, and Venue_Name, where Player_No is the primary key and Venue_Name is a foreign key. Venue_Details table has two columns, namely Venue_Name, City, and State, where Venue_Name is the primary key.

US_Player (Player_No, Player_Name, Team_Name, No_of_Goals, Venue_Name)
Canadian_Player (Player_No, Player_Name, Team_Name, No_of_Goals, Venue_Name)
Venue_Details (Venue_Name, City, State)

Fill in closest relational operator, e.g., Union, Intersect, Difference, Relational Division, Left Outer Join, Right Outer Join, Natural Join), for various sentences in the following table. (You do NOT need to list relevant tables.)

<table>
<thead>
<tr>
<th>Relational Operator</th>
<th>English Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational Division</td>
<td>E1: List US players who have scored goals in all of the venues.</td>
</tr>
<tr>
<td>Difference</td>
<td>E2: List players who have played for US but not for Canada.</td>
</tr>
<tr>
<td>Natural Join</td>
<td>E3: For each Canadian player, list the cities where they scored a goal.</td>
</tr>
<tr>
<td>Intersect</td>
<td>E4: List players who have played for both US and Canada.</td>
</tr>
<tr>
<td>Right/Left Outer Join</td>
<td>E5: For each player, list number of goal, e.g., 0, 1,..., at every venue, including all the venues in the result.</td>
</tr>
</tbody>
</table>
Q2 (35 pts) Consider the following library database schema with 7 tables, namely BOOK, BOOK_COPY, BOOK_AUTHOR, AUTHOR, PUBLISHER, BOOK_LOAN and BORROWER. Primary keys are underlined and foreign keys are italicized. The primary and foreign keys share the names (e.g., “ISBN” stays the same in BOOK and BOOK_AUTHOR tables).

- BOOK table has three columns, namely ISBN, Title and Publisher_name, where ISBN is the primary key and Publisher_name is a foreign key;
- BOOK_COPY table has two columns, namely ISBN and Copy_no, where ISBN, Copy_no is the primary key and ISBN is a foreign key;
- BOOK_AUTHOR table has two columns, namely ISBN and Author_id, where ISBN, Author_id is the primary key and ISBN and Author_id are two foreign keys, respectively;
- AUTHOR table has three columns, namely Author_id, Author_name and Author_age, where Author_id is the primary key;
- PUBLISHER table has three columns, namely Publisher_name, Publisher_address and Phone, where Publisher_name is the primary key;
- BOOK_LOAN table has five columns, namely ISBN, Copy_no, Card_no, Date_out and Due_date, where ISBN, Copy_no and Card_no together form the primary key, and ISBN, Copy_no together is a foreign key of two columns and Card_no is another foreign key;
- Borrower table has four columns, namely Card_no, Name, Address and Phone, where Card_no is the primary key.

Table 2.1 Table schemas and constraints

<table>
<thead>
<tr>
<th>Table Schema</th>
<th>Foreign key constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOK (ISBN, Title, Publisher_name)</td>
<td></td>
</tr>
<tr>
<td>BOOK_COPY (ISBN, Copy_no)</td>
<td>RESTRICT on ISBN</td>
</tr>
<tr>
<td>BOOK_AUTHOR (ISBN, Author_id)</td>
<td>DEFAULT on ISBN, Author_id</td>
</tr>
<tr>
<td>AUTHOR (Author_id, Author_name, Author_age)</td>
<td></td>
</tr>
<tr>
<td>PUBLISHER (Publisher_name, Publisher_address, Phone)</td>
<td></td>
</tr>
<tr>
<td>BOOK_LOAN (ISBN, Copy_no, Card_no, Date_out, Due_date)</td>
<td>CASCADE on ISBN, Copy_no, RESTRICT on Card_no</td>
</tr>
<tr>
<td>BORROWER (Card_no, Name, Address, Phone)</td>
<td>-</td>
</tr>
</tbody>
</table>

Q2 Part A (18 pts): Complete the following SQL SELECT statement to list the information required by the query tasks. Each blank worth 2 points.

Query task 1: List the card numbers and phone numbers of borrowers.

SELECT _________ Card_no, Phone _________________
FROM _________ Borrower _________________

Query task 2: List the names of authors older than 40.

SELECT _________ Author_name _________________
FROM _________ Author _________________
WHERE _________ Author_age>40 _________________

Query task 3: List ISBNs of books with more than 5 copies.

SELECT _________ ISBN _________________
FROM _________ Book_copy _________________
GROUP BY _________ ISBN _________________
HAVING _________ count(Copy_no)>5 _________________

*Some of you understood the Copy_no as Number_of_Copies which is not the right interpretation for this table design. The “no” in Copy_no is the same as “no” in Card_no which is an identifier. This can be inferred from the BOOK_COPY table, where the primary key is a combination of ISBN and Copy_no. If Copy_no means the number of copies then it will not be part of the primary key and the table needs to be merged with BOOK. Similarly, in BOOK_LOAN table, Copy_no is also part of the primary key, together with Card_no and ISBN, which indicates the same thing.
Q2 Part B (12 pts): Complete the following SQL SELECT statement to list ISBN, title and author names of each book that is currently borrowed by ‘Aaron Brooks’. Answer the three query formulation questions (a), (b) and (c) first before completing the query. Cross out irrelevant clauses of the given template SELECT statement.

(a) Which tables are needed for this query?
   BOOK, BOOK_AUTHOR, AUTHOR, BOOK_LOAN and BORROWER
(b) What is the join path to connect the tables?
   BL.Card_no = BR.Card_no
(c) Is aggregation needed? No.

SELECT BK.ISBN, BK.Title, A.Author_name
FROM BOOK BK, BOOK_AUTHOR BA, AUTHOR A, BOOK_LOAN BL, BORROWER BR
   BR.Card_no AND BR.Name = 'Aaron Brooks';

Q2 Part C (5 pts): The following tables contain sample data for a few selected columns in BOOK, BOOK_COPY and BOOK_LOAN tables. Foreign keys are italicized. The action on delete constraints on foreign keys are shown in Table 2.1.

For the following two queries, determine the number of rows deleted in each table. One point each blank and all points will be taken if five or more of the answers are wrong.

Query 1:
DELETE FROM BOOK
WHERE ISBN >= '0003';
Number of rows deleted by Query 1:
    BOOK: ______ 0  BOOK_COPY: ______ 0  BOOK_LOAN: ______ 0
(No rows can be deleted due to RESTRICT on ISBN in BOOK_COPY, an error message will be returned)
* The first blank is trickier. If you answered with 1, no point is cut since ‘0004’ is not in BOOK_COPY so if we only consider that row there is no problem in deleting it. However, the database will just return an error since the other parts of the deletion violate the rule. This is not covered in the class so 1 was also treated as a correct answer.

Query 2:
DELETE FROM BOOK_COPY
WHERE ISBN < > '0002';
Number of rows deleted by Query 2:
    BOOK: ______ 0  BOOK_COPY: ______ 4  BOOK_LOAN: ______ 3
(Foreign key needs to reference the whole primary key so “ISBN, Copy_no” in BOOK_LOAN references those in BOOK_COPY. Since the deletion does not affect rows in BOOK, the RESTRICT on BOOK_COPY.ISBN does not matter and we can delete 4 rows and 3 rows in BOOK_COPY and BOOK_LOAN according to the CASCADE action.)
Q3 (40 pts)

Q3: Consider the following Entity-Relationship Diagram (ERD) about a university database.

Q3A1. Based on the ERD above, what is the total number of the tables generated by (ERD to relational tables) conversion rules? (2 points. If you merge tables Advisor and Graduate_Student together and answer 10 here, no point is taken.)

11

Q3A2. Convert the above ERD into relational tables by filling in the following table. Leave a field blank if not applicable. (14 points. 0.35 point/blank)

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Primary Key</th>
<th>Foreign Key</th>
<th>Other Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>SSN</td>
<td>SSN references Person.SSN</td>
<td>Class</td>
</tr>
<tr>
<td>Person</td>
<td>SSN</td>
<td></td>
<td>Name, Bdate, Sex</td>
</tr>
<tr>
<td>Faculty</td>
<td>SSN</td>
<td>SSN references Person.SSN</td>
<td>Rank, Foffice, Fphone, Salary</td>
</tr>
<tr>
<td>Graduate_Student</td>
<td>SSN</td>
<td>SSN references Student.SSN</td>
<td>College, Degree, Year</td>
</tr>
<tr>
<td>Undergrad_Student</td>
<td>SSN</td>
<td>SSN references Student.SSN</td>
<td></td>
</tr>
<tr>
<td>Advisor</td>
<td>F_SSN, GS_SSN</td>
<td>F_SSN references Faculty.SSN, GS_SSN references Graduate_Student.SSN</td>
<td></td>
</tr>
<tr>
<td>Department</td>
<td>Dname</td>
<td>Dname references Faculty.SSN</td>
<td>Office, Dphone</td>
</tr>
<tr>
<td>Belongs</td>
<td>Dname, SSN</td>
<td>Dname references Department.Dname</td>
<td></td>
</tr>
<tr>
<td>Course</td>
<td>C#, Dname</td>
<td>Dname references Department.Dname</td>
<td>Cname, Cdesc</td>
</tr>
<tr>
<td>Section</td>
<td>Sec#, C#, Dname</td>
<td>C#, Dname references Course.C#, Course.Dname, T_SSN references Faculty.SSN</td>
<td>Year, Qtr</td>
</tr>
<tr>
<td>Enrollment</td>
<td>Sec#, C#, Dname, SSN</td>
<td>Sec#, C#, Dname references Section.Sec#, Section.C#, Section.Dname, SSN references Student.SSN</td>
<td>EnrGrade</td>
</tr>
</tbody>
</table>

Commented [YL6]: Graduate_Student and Undergrad_Student are subtypes of Student.

Commented [YL7]: Optional 1-M relationship is converted to a table.

Commented [YL8]: Each department must have one chair.

Commented [YL9]: Course is a weak entity. It is identified by C# and its department’s primary key: Dname.

Commented [YL10]: Section is a weak entity. It is identified by Sec# and its course’s primary key: C# and Dname.

Commented [YL11]: Enrollment is a M-M relationship, which is converted to a table. It is identified by its enrolled student and its section.
Q3: Part B (16 pts). Consider the following database schema for a small private airport database that is used to keep track of airplanes, airport technicians and pilots. Primary keys are underlined and foreign keys are italic. The foreign keys and the primary keys they reference share names. For example, in Works_On table the primary key is SSN and Model, and SSN is one of the foreign keys which references Technician’s SSN.

Person (SSN, Name, Address, Phone)  
Technician (SSN, Salary, Shift)  
Pilot (SSN, Rest, Lic_num)  
Plane_Type (Model, Capacity, Weight)  
Airplane (Reg#, Hangar#, Model)  
Hangar (Hangar#, Capacity, Location)  
Service (Workcode, S_Date, Reg#, Hours)  
Works_On (SSN, Model)  
FLIES (SSN, Model)  
Maintain (SSN, Workcode, S_Date, Reg#)  

- Each airplane is of a particular plane type, and is stored in a particular hangar. If there is no airplane of a certain type, the plane_type record will be deleted from the database. Many airplanes can be of the same plane_type. A hangar can be empty and can store many airplanes.
- Works_On table records plane_types and technicians who have worked on them by maintaining services. Flies table records plane_types and the pilots who have flown them. There may be new technicians who have not worked on any plane_type, and new pilots who have not flown any plane_type. There may be new plane_types which have not been worked on or flown.
- The database also keeps track of the technicians who have maintained the airplanes through services. There are new technicians who have not maintained any services, while a service must be maintained by at least one technician.
- There are new airplanes which have not undergone any service. An airplane may undergo many services.
- Pilots and technicians are persons working at airports.

Q3B1. Draw an ERD based on the above database schema and information. (12 points)

- 1 point for Optional M-N relationship “FLIES”.
- 1 point for generalization hierarchy. (The foreign key is implicit.)
- 1 point for Optional M-N relationship “Maintain”.
- 1 point for “Be_of” relationship.
- 1 point for “Store” relationship.
- 1 point for weak entity “Service” (entity and the identifying relationship).
- Other points are taken for redundant foreign keys, missing attributes, etc.
Q3B2. Based on the above database schema and information, answer the following questions.
(a) Identify cycles with three or more entities. For each cycle list the name of entities in the cycle. (2 points)
   Technician, Works_on, Plane_Type, Airplane, Service, Maintain
(b) Does this cycle indicate redundancy (i.e. Can one of the relationships be inferred using the other relationship)? A. Yes; B. No. Briefly justify your answer. (2 points)
   Yes. The relationship “Works_on” can be inferred from technicians maintaining services of airplanes.

Q3: Part C. (8 pts). For each violation of consistency and completeness rules in the ERD, identify the violated rule and suggest a possible resolution based on the provided information. (results may not fill out all the empty rows)

<table>
<thead>
<tr>
<th>No.</th>
<th>Violated Rule</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project entity does not have a primary key.</td>
<td>Assign PId as the primary key of Project entity.</td>
</tr>
<tr>
<td>2</td>
<td>“Has” relationship has no cardinality on the direction to “Dependent”.</td>
<td>Assign the cardinality to be  , because “Each employee may have many dependents”. However, in this case a dependent cannot be identified only by its employee. According to the narrative “which have unique names in their families”, dependent should have an attribute named “name” as a part of the primary key.</td>
</tr>
<tr>
<td>3</td>
<td>“Manages” entity is linked to two identifying relationships, but it is not a weak entity.</td>
<td>Make “Manages” a weak entity.</td>
</tr>
<tr>
<td>4</td>
<td>The relationships linked to “Works_On” do not have names.</td>
<td>Name relationship between “Employee” and “Works_On” as “involve”. Name relationship between “Works_On” and “Project” as “On”.</td>
</tr>
<tr>
<td>5</td>
<td>1.33 points/blank.</td>
<td></td>
</tr>
</tbody>
</table>

Common mistakes:
1. Only change the cardinality of the “Has” relationship but not add a partial primary key to “Dependent”, which results in an incorrect ERD.
2. Miss the names of the relationships.