Beijing-Dublin International College

SEMESTER II EXAMINATION - 2022/2023

School of Computer Science

COMP2013J DATABASES AND INFORMATION SYSTEMS (SE)

Dr. Robert Ross
Associate Professor Neil Hurley
Dr. Ruihai Dong *

Time Allowed: 120 minutes

Instructions for Candidates

This paper consists of 4 questions. Answer all questions. All questions carry equal marks.

BJUT Student ID: _____ **UCD Student ID:** _____

I have read and clearly understand the Examination Rules of both Beijing University of Technology and University College Dublin. I am aware of the Punishment for Violating the Rules of Beijing University of Technology and/or University College Dublin. I hereby promise to abide by the relevant rules and regulations by not giving or receiving any help during the exam. If caught violating the rules, I accept the punishment thereof.

Honesty Pledge: 刘乐 _____ **(Signature)**

Instructions for Invigilators

Non-programmable calculators are permitted.

No rough-work paper is to be provided for candidates.

Obtained score

Question 1:

(a) For each of the following three relational concepts, explain the key ideas behind them, using suitable examples.

- Domain Integrity ALL columns in a relational database be declared upon a defined domain.
- Entity Integrity Every table should have a primary key and it be unique and not null
- Referential Integrity It is used to maintain the consistency among rows in the tables

[10 marks]

(b) Describe three phases of database design. ② Logical Design: Translation of the conceptual schema into logical schema. ③ physical Design: logical schema is compiled with ① Conceptual Design: Design an Entity-Relationship Model. the details of the physical implementation. **[6 marks]**

(c) In relational database theory, what is the closure property? Why is this important when performing operations on relations?

The result of any operation is another relation.

The closure property means it is possible to write nested expressions.

[4 marks]

(d) Show the Cartesian product of two relations R and S described as below. Assume that R has two attributes: **A, B,** and S has three attributes: **C, D, E.**

R

A	B
1	2
4	5

S

C	D	E
4	4	3
5	6	6
2	4	9

A	B	C	D	E
1	2	4	4	3
1	2	5	6	6
1	2	2	4	9
4	5	4	4	3
4	5	5	6	6
4	5	2	4	9

[5 marks]

[Total 25 marks]

Obtained score

Question 2:

(a) Write an SQL statement to create a table called "Students", with the following details:

Attributes:

- **stu_id**, which contains a student's ID number: a number that is 8 digits long.
- **first_name**, which is a string no longer than 30 characters.
- **last_name**, which is a string no longer than 30 characters.
- **DOB**, which is the Date of Birth of a student.
- **major_id**, which contains the ID of the major: an alphanumeric code that is 10 characters long.

Other Information:

- stu_id is the primary key of this table.
- major_id attribute is a foreign key that refers to an attribute named "id" in a table named "Majors".
- If the "id" in the "Majors" table is changed (updated), the change should cause a reaction in the "Students" table to maintain the consistency between these two tables.

[7 marks]

(b) Study the relational schema below and write SQL statements to answer the questions that follow.

Hotel(hotelNo, hotelName, city)
Room(roomNo, *hotelNo*, type, price)
Guest(guestNo, guestFirstName, guestLastname, guestAddress)
Booking(*hotelNo*, *guestNo*, dateFrom, dateTo, *roomNo*, booking_price)
In this case, it assumes that room type can be single, double, or family.

- Select all the guests whose last name start with an "D". **[3 marks]**
- List all double or family rooms with a price above $100.00 per night, in ascending order of price. **[3 marks]**
- List the number of rooms in each hotel in Beijing. **[4 marks]**
- Insert a new row into "Hotel" table with the following details: **[4 marks]**

  hotelNo: 654321
  hotelName: BDIC-2013J
  city: Dublin

- Decrease the price of all double rooms by 10%. **[4 marks]**

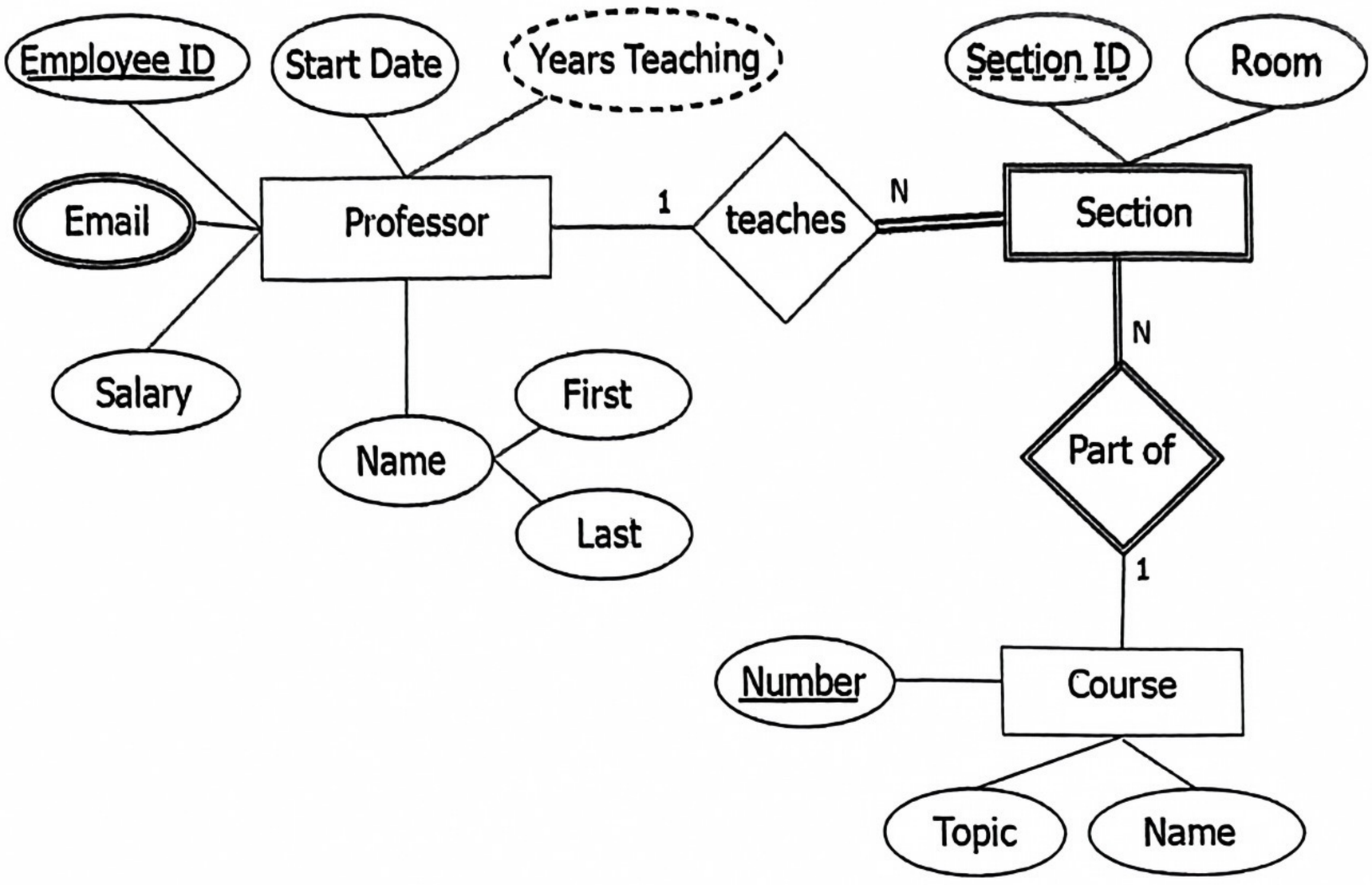
[Total 25 marks]

d1.

Obtained score

Question 3:

Study the Entity Relationship diagram below and answer the questions that follow.



(a) In the diagram, the "Years Teaching" attribute in the "Professor" entity type is shown with a dashed line. What is the meaning of this dashed line, and how is this different from other attributes? It is a derived attribute. It is can be calculated based on startDate.
[3 marks]

(b) In the diagram, the "Email" attribute in the "Professor" entity type is shown with a double line. What is the meaning of this double line, and how is this different from other attributes?
It means this attribute can have multiple values.
[3 marks]
A professor could have multiple email addresses

(c) In the diagram, the "Section" entity type is shown with a double line. What is the meaning of this double line, and how is this different from other entities?
It is weak entity types, which can not exit without a relationship to Course.
[4 marks]

(d) Map the Entity Relationship diagram to a relational model. In your answer, describe in detail the process that you use.

[15 marks]

Professor: Employee ID, Start Date, Salary, Years Teaching, First, Last

Course: Number, Topic, Name

Section: SectionID, CourseNum, Room, EmployeeID

[Total 25 marks]

Obtained score

Question 4:

Below is the definition of a table **t_employees** and source code to access this table by using JDBC. Examine the code and answer the questions below:

Table **t_employees**

ID	INT
Name	VARCHAR(30)
Department	VARCHAR(20)

```
public class Employee{
        private int id;
        private String name;
        private String department;
        public Employee(int eid, String n, String d){
                this.id = eid;
                this.name = n;
                this.department = d;
        }
        public int getId(){
                return this.id;
        }
        public void setId(int id){
                this.id = id
        }
        public String getName(){
                return this.name;
        }
        public void setName(String name){
                this.name=name;
        }
        public String getDepartment(){
                return this.department;
        }
        public void setDepartment(String d){
                this.department = d;
        }
}
```

n = getConn();

mmmmStatement(query)) {


```
import java.sql.*
import java.util.ArrayList;
import java.util.List;
public class DBHelper {
        public static Connection getConn() throws SQLException{
                String url = "jdbc:mysql://localhost:3306/db_employee";
                Connection conn = DriverManager.getConnection(url);
                return conn;
        }
        public static List<Employee> getEmployeesByPageNo(int n) {
                //TODO
        }
        public static void deleteEmployee(int eid){
                //TODO
        }
        public static void updateEmployee(Employee e){
                //TODO
        }
}
```

(a) Use an example to explain what an SQL Injection Attack is? How can it be avoided? **[5 marks]**

If user enter password: `Pfasd' OR '1'='1'. Now, a user can gain access without knowing password

PreparedStatement

User input is dangerous.

If: String query = "SELECT username from users WHERE username = '" + Uname + "' AND password = '" + pass + "'";

(b) Assume that employees are displayed page by page in the system and each page displays 8 employees. Complete the code above filling the method **getEmployeesbyPageNo(int n)** to retrieve employees from the table, which are displayed on Page n.

[6 marks]

(c) Complete the code above filling the method **deleteEmployee(int eid)** to delete the employee with given eid from the table.

[5 marks]

(d) Complete the code above filling the method **updateEmployee(Employee e)** to update the employee information into the database.

[5 marks]

(e) Explain what ORM stands for and what is is used for?

Object Relational Mapping is a programming technique for converting data between relational databases and oop languages. **[4 marks]**

[Total 25 marks]

Q4

(b).

```
List<Employee> employees = new ArrayList<>();
int pagesize = 8;
int offset = (n-1) * pagesize;

String query = "SELECT id, * FROM t_employees LiMIT ?, ?";

try( Connection conn = getConn();
PreparedStatement pre = conn.prepareStatement(query)) {

Pre.setInt (1, offset);
Pre.setInt(2, PageSize);
try(ResultSet result = pre.executeQuery()) {
    While(result.next()) {
        int id = result.getInt("Id");
        String name = result.getString("Name");
        String department = result.getString("Department");
        Employee employee = new Employee (id, name, department);
        employees.add (employee);
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
return employees;
```

Q 4

1C.

```
String query = " DELET FROM t_employees WHRP WHERE ID = ?";
try ( Connection conn = getConnl );
    PreparedStatement pre = conn.prepareStatement (query) ) {
    Pre. setInt(1, eid);
int rowsDeleted = pre. executeUpdatel);
if (rowsDeleted >0) {
    Sout (" Succesful");
} else {
    ↑ Sout (" Nothing he deleted");
    }
} Catch (SQLException e) {
    e. printStackTrace();
}
```

```
d1:

String query = " UPDATE ~~er~~ t_employees SET Name = ?, Department = ?
                                              WHERE ID = ? ";
try ( Connection conn = getConn();
      PreparedStatement pre = conn.prepareStatement( query )) {

      pre.setString (1, e.getName());
      Pre.setString(2, e.getDepartment());
      pre.setInt(3, e.getId());

      int rowsUpdated = pre.executeUpdate();
      if( rowsUpdated > 0) {
         Sout(" Successful");
      } else {
         Sout(" Noting was updated");
      }

catch( SQLException ex) {
      ex.printStackTrace();
```

```
public class DBHelper {
        public static Connection getConn() throws SQLException{
                String url = "jdbc:mysql://localhost:3306/db_employee";
                DriverManager.getConnection(url);
```

Q3

d.

1. Mapping Regular Entity Types

Professor : EmployeeID, Start Date, Salary, First, Last, Years Teaching

Cources : Number, Topic, Name

2. Mapping Weak Entity Types

~~Dependent~~ :

Section : ~~EmployeeID~~, SectionID, Room, CourseNum

3. Mapping 1:1 Relationships

NO

4. Mapping 1:N Relationships

Section: ~~EmployeeID~~, SectionID, Room, CourseNum, EmployeeID,

5. Mapping M:N Relationships

No.

6. Mapping Multivalued Attributes

Professor_email : EmployeeID, Email