CO2102 - Database Project

Guidelines

- Please find the deadline information on the CO2102 Blackboard Homepage.
- Resubmission before the deadline is allowed. Late submission will cause a penalty on the overall mark as per the University's regulations.
- This project weighs 50%, and is calculated out of 100. Read the marking scheme section at the end for more details.
- After you complete and execute all the tasks using Workbench, copy and paste your solution onto the Project Answer Template Word file, and save it. The template is located under the "DB Project" folder on Blackboard.
- You should submit via TurnItIn on Blackboard. The submission is located under the folder "DB Project".
- Your submission should only include a **single** file, titled: *YourStudentIDNumber*.docx (e.g. *109776523.docx*).
- Do not use your handwriting to draw the ERD, and use software to achieve this.
- Once you are happy to proceed with the submission, click the submit button. Please note that if you do not follow the submission guidelines your work might not be marked.
- Be aware of **PLAGIARISM**. This assignment is an individual piece of work. Any similarities reported by the plagiarism detection software between your work and any external/online or internal resources, the plagiarised parts will automatically be flagged and reported, and marks will be deducted in accordance to the University's guidelines.

Question 1

Create an ERD using the below business rules that can be implemented for Omega United. Your diagram should illustrate all the attributes, entities, keys, and relationships. For all entities, make sure to accurately represent each relationship name, type, and degree.

Omega United is a nonprofit organization that provides aid to people after natural disasters, such as a Covid-19 pandemic. Individuals volunteer their time to carry out the tasks of the organization. For each volunteer, their name, address, and telephone number are tracked. Each volunteer may be assigned to several tasks during the time that they are doing volunteer work, and some tasks require many volunteers. It is possible for a volunteer to be in the system without having been assigned a task yet. It is possible to have tasks that no one has been assigned. When a volunteer is assigned to a task, the system should track the start time and end time of that assignment.

For each task, there is a task code, task description, task type, and a task status. For example, there may be a task with task code “101”, description of “answer the telephone”, a type of "recurring”, and a status of “ongoing”. There could be another task with a code of “102” description of “prepare 5000 packages of basic medical supplies,” a type of “Packing,” and a status of “open.”.

For all tasks of type “packing,” there is a packing list that specifies the contents of the packages. There are many different packing lists to produce different packages, such as basic medical packages, child care packages, food packages, etc. Each packing list has a packing list ID number, packing list name, and a packing list description, which describes the items that ideally go into making that type of package. Every packing task is associated with only one packing list. A packing list may not be associated with any tasks, or may be associated with many tasks. Tasks that are not packing tasks are not associated with any packing list.

Packing tasks result in the creation of packages. Each individual package of supplies that is produced by the organization, is tracked. Each package is assigned an ID number. The

date the package was created, and total weight of the package is recorded. A given package is associated with only one task. Some tasks (e.g., “answer the phones”) will not have produced any packages, while other tasks (e.g., “prepare 5000 packages of basic medical supplies”) will be associated with many packages.

The packing list describes the ideal contents of each package, but it is not always possible to include the ideal number of each item. Therefore, the actual items included in each package should be tracked. A package can contain many different items, and a given item can be used in many different packages.

For each item that the organization provides, there is an item ID number, item description, item value, and item quantity on hand stored in the system. Along with tracking the actual items that are placed in each package, the quantity of each item placed in the package must be tracked too. For example, a packing list may state that basic medical packages should include 100 bandages, 4 bottles of iodine, and 4 bottles of Hydroxy-chloroquine. However, because of the limited supply of items, a given package may include only 10 bandages, 1 bottle of iodine, and no Hydroxy-chloroquine. The fact that this package includes bandages and iodine needs to be recorded along with the quantity of each item that is included. It is possible for the organization to have items donated that have not been included in any package yet, but every package will contain at least one item.

Question 2

In this question you will create a full database schema for a pharmaceutical company. The following sections describe the full schema and components of the required database.

A pharmaceutical company includes full information about its employees. This covers the employee number, last name, first name, extension, email address, office code, job title, and the other employee in which he/she reports to *(Hint: multiple employees can only report to one employee, however one/several employees cannot report to multiple employees at the same time).*

The company also stores information about their pharmaceutical retail customers, which covers the customer number, customer full name (last name and first name), phone number, address line 1, address line 2, city, county, postal code, country, sales amount, representative employee number (in which they deal with directly), and the credit limit number of that customer.

Furthermore, the pharmaceutical company database stores information about the drugs they produce and prescription orders. This covers the order number, order date, required date, shipped date, order status, comments, and the pharmaceutical customer number that relates to this order. In addition, information about the individual drugs should cover the product code, product name, product line, scale-weight, vendor, description, quantity in stock, the buying price, and the MSRP (manufacturer's suggested retail price).

In relation to the employees information mentioned previously, the employees are assigned to offices within the pharmaceutical company, and this information includes the office code, City, phone number, address line 1, address line 2, county, country, postal code, and territory. In addition, with regards to the pharmaceutical retail customers, the payments information are stored in correspondence to each customer. This includes the customer number, cheque number, payment date, and the amount paid by each customer.

The mutual information between the drugs/products and the orders include the order number, product code, quantity ordered, price, and the order line number. Finally, this database stores

the information about pharmaceutical product lines, which includes the product line text, description, website, and the product image.

A product line can include multiple products, however, one product cannot belong to more than one product line. Furthermore, an order can include many products, and a product could also be related to multiple orders. In addition, one retail customer can make multiple payments, however one payment can only be related to one customer at a given time. Several employees within the pharmaceutical company can report to one employee only, and they cannot report to multiple employees at once. One office can have more than one employee, yet one employee should not be assigned to more than one office.

Task 1

Create an ER diagram using the class diagram method to represent the above description. Your diagram should illustrate all the attributes, entities, keys, and relationships. For all entities, make sure to accurately represent each relationship name, type, and degree.

Task 2

Write SQL DDL commands to create the previous database schema. Make sure to represent your exact ER diagram from the previous task, and achieve the relationships, data types, and the correct attributes for the corresponding entities. You must also make sure that the order of your SQL commands is correct to ensure data integrity.

Task 3

Write MySQL commands to insert 4 rows in each table you have previously created. You should create your own unique data (*BE AWARE OF PLAGIARISM*). In addition, you should make sure that the entered data makes sense against each table. In particular, make sure to comply with the correct data types, constraint rules, and the relationships in which you have set previously. You must also make sure that the order of your SQL commands is correct to ensure data integrity when executing the SQL script.

Task 4

Create a view called “Customer_Order__Restricted_Info” , which shows the customer id, customer full name (first name and last name as one column separated by space), customer full address (which shows address line 1, address line 2, postal code, and country, all as one column and separated by spaces), and separate information about the order status, and quantity ordered, which are relevant to each customer with a credit limit of more than 1000, and an order shipped date that took place before 2010.

Marking Scheme: *How is this Project Marked?*

- Each section of the work will be assessed in accordance to the marks assigned to it in the marking scheme. This is outlined below in accordance with each section/task.
- In order to ensure consistency and fairness, individual feedback will be provided on the system against each given mark.
- Your MySQL solution should run successfully on MySQL Workbench. Failed to do so, 50% of the overall mark will be deducted, and then your solution will be marked accordingly.
- If your solution to Question 2 -> Task 1 is incorrect, you will not lose all the marks for Tasks 2, 3, and 4. In line with this, only 25% of each task will be deducted, and then your solution will be marked accordingly.
- This assignment weighs 50%, and is calculated out of 100.
- The mark distribution is assigned as follows:
    - Question 1
        - Task 1: 35% (See ERD Rubric below)
    - Questions 2
        - Task 1: 20% (See ERD Rubric below)
        - Task 2: 25%
        - Task 3: 5%
        - Task 4: 15%
- A full sample solution of the assignment will be provided after the submission date.

ERD Rubric

Criteria (weight)	100% - 70% Exemplary	70% - 50% Satisfactory	50% - 30% Needs Improvement	Score (Weighted)
Notation (x2)	Diagram uses an appropriate E-R notation. The notation is used correctly for all elements of the diagram.	Diagram uses an appropriate E-R notation. The notation is used correctly for most elements of the diagram.	Diagram does not use an appropriate E-R notation or uses a notation incorrectly for most or all elements.	15%
Complexity (x1)	The required number of tables and foreign key relationships will be needed to implement the database.	As drawn, the required number of tables and foreign key relationship may not be needed, but the required complexity can be achieved with minor changes.	The required number of tables and foreign key relationship will not be needed. It is unclear how the project could satisfy the required complexity.	15%
Professionalism (x1)	Diagram presents a professional appearance. It could be shared with a "real-world" customer without changes.	Diagram largely presents a professional tone. It could be shared with a "real-world" customer with minor revisions.	Diagram is unprofessional. Major revisions would be necessary before sharing the document with a "real-world" customer.	10%
Entity Sets (x2)	Diagram captures all entity sets necessary for a database that would satisfy the initial problem statement.	Diagram captures most entity sets necessary for a database that would satisfy the initial problem statement.	Diagram captures few or none of the entity sets necessary for a database that would satisfy the initial problem statement.	15%
Attributes and Keys (x1)	Diagram captures all attributes and primary keys necessary for a database that would satisfy the initial problem statement.	Diagram captures most attributes and primary keys necessary for a database that would satisfy the initial problem statement.	Diagram captures none or few of the attributes and primary keys necessary for a database that would satisfy the initial problem statement.	15%
Relationships (x2)	Diagram captures all relationships necessary for a database that would satisfy the initial problem statement.	Diagram captures most relationships necessary for a database that would satisfy the initial problem statement.	Diagram captures none or few of the relationships necessary for a database that would satisfy the initial problem statement.	15%
Constraints (x1)	Diagram captures all cardinality and participation constraints necessary for a database that would satisfy the initial problem statement.	Diagram captures most of the cardinality and participation constraints necessary for a database that would satisfy the initial problem statement.	Diagram captures none or few of the cardinality and participation constraints necessary for a database that would satisfy the initial problem statement.	15%