MCAST
Malta College of Arts, Science & Technology
INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

Database Programming II
IICT6203

Assignment 2 (Home)

Developing DBMS Applications

Instructions to Students

The following instructions are to be carefully read before starting this assignment. It is recommended to read the assignment and ask for clarification if you do not understand any of the questions.

- This home assignment must be completed by the **28th of May 2017**.
- This assignment has a total weight of **66%**.
- Copying is strictly prohibited, and it will be penalised through a referral and other disciplinary procedures as per MCAST Policies, Procedures and Regulations.
- **All** your work should be zipped in a single file and named as **Class_Name_Surname.zip**. This must be uploaded to TurnitIn before the deadline.
- This assignment will then be presented in class in an interview format.

SCENARIO

A local video game company has produced a game that allows gamers to generate revenue whilst playing their game. They have decided that it was time to generate statistics about the gamers and their balances. They have generated a JSON file with the client's details (persondata.json).

The main aim is to load this JSON file into a database, making sure it meets the given design. Then automatically generate statistics for the uploaded data. A front-end will be required, this can be coded either in Java or C#.

KU7 Show how to query XML data within a DB.

Q1 – *JSON Data Loading*: In this question you must load the data found in a JSON file to a database table. Your answer for this question must include:

- A stored procedure named *main.loadJSON* that is capable of loading the data found in the file *persondata.json* to the *main.jsontable* table.
- Each *person* element within the given JSON file should be loaded in a sepearate row of the *main.jsontable* table.

KU3 Identify methods of optimizing a given database structure.

Q2 – *Converting to a Relational Model*: In this question you are to convert the data which was loaded in the *main.jsontable* table to suit a relational model. Your answer must include a trigger named *main.afterParsing*.

- It must populate the existing relational tables named *main.country,* and *main.people* using the data inserted in *mian.jsontable*.
- Make sure that any records which have NULL value in any of their fields are to be omitted.
- Do **not** delete the records loaded in *main.jsontable* once the records have been transferred.

KU6 Recall the code needed to support transaction locking.

Q3 – *Statistic Generation*: In this question you need to generate the total balance for 3 age brackets using the *main.people* table. Your answer must include a procedure named *main.generateStatistics*. The age brackets are calculated by splitting the groups into 3 equal buckets ranging from the oldest to the youngest person. Each bucket must have an integer range. The generated results must be inserted into table *main.stats.* It is important that the statistics are generated on records that are committed and no repeatable reads are permitted.

The first run should have the following output:

	statId	dateCollected	totalPeople	bracket1	bracket2	bracket3
1	4037A4D1-A957-42BE-B172-1B3A108BF223	2018-04-19 13:48:59.267	989	418442.74	381701.10	449008.85

KU5 Recall the code needed to handle errors in a given SQL statement.

Q4 – *Integration step*: In this question, you are to prepare a procedure that loads the JSON file, populates the relational database, and generates the statistics. Your answer must include a stored procedure named *main.mainRun*. If any errors are encountered during the running of this procedure, it must throw the error message provided by SQL Server.

SE2 Evaluate a database system and propose manners of making it more resilient.

Q5 – *Improvements*: Make changes to the procedure *main.mainRun* so that it accepts a filename (along with its path) for the JSON file.

AA4 Develop an application with CRUD functionalities that communicates with a DB.

SE3 Develop a series of queries that cater for concurrency issues.

Q6 – *DBMS Application:* You are to develop an application using C# or Java that interacts with the DBMS to manipulate data found in the database.

Your application should include the following:

- A menu with three options: *Load JSON*, *Show Statistics* and *Show Country Total*.
- A connection to the local database instance using SQL Server Authentication (not Integrated Authentication).
- The first option in the menu (Load JSON) must call the stored procedure *main.mainRun*.
- The second option in the menu (Show Statistics) must return the latest record in table *main.stats*. Only the total people, and the brackets should be displayed.
- The third option in the menu (Show Country Total) must ask for a country and registration year, then return the total balance for that country in that year.
- For each option make sure to use appropriate transaction isolation levels.
- This application must have appropriate error handling and messaging.

KU4 | Recall the syntax to create an index and how to check its impact on a DB.

AA3 | Interpret query execution plans and identify optimization techniques

Q7 – Improving Performance: In this question, you are to improve the performance of the database by identifying and implementing the relevant database objects. These are the requirements:

a. Prepare a query that makes use of *main.person* and *main.country*. The query must have an applied context.
b. Run the query and produce a query execution plan.
c. Make required changes to the database to try to improve performance.
d. Run the query again and produce a query execution plan again.
e. Evaluate and discuss the difference between the two query execution plans. This evaluation must contain the total I/O and CPU cost for each case.

AA5 | Prepare a number of reports generated from XML data stored within a DB.

Q8 – *Report Generation:* You are required to generate the following two reports.

Report 1 – Show the average balance per birthyear. The result must look as follows.

```
[{"BirthYear":1983,"AvgBalance":1267.579583},
 {"BirthYear":1964,"AvgBalance":1291.527058},
 {"BirthYear":1956,"AvgBalance":1279.204285},
 {"BirthYear":1972,"AvgBalance":1044.722777},
…
{"BirthYear":1977,"AvgBalance":1285.464210},
{"BirthYear":1957,"AvgBalance":1222.733000}]
```

Report 2 – Show the contents of the *main.stats* table excluding the primary key. The result must look as follows.

```
<stats>
  <row>
    <DateCollected>2018-04-19</DateCollected>
    <totalPeople>989</totalPeople>
    <bracket1>418442.74</bracket1>
    <bracket2>381701.10</bracket2>
    <bracket3>449008.85</bracket3>
  </row>
</stats>
```

Marking Scheme

KU7 Show how to query XML data within a DB.		
	Maximum	Awarded
Q1 – Data loaded from file into table.	5	
Total	5	

KU3 Identify methods of optimizing a given database structure.		
	Maximum	Awarded
Q2 - Countries populated correctly.	1.5	
Q2 – People populated correctly.	1.5	
Q2 – Necessary precautions taken for each table (1 mark each)	2	
Total	5	

KU6 Recall the code needed to support transaction locking.		
	Maximum	Awarded
Q3 – Correct transaction isolation level used.	1	
Q3 – Statistics generated correctly.	4	
Total	5	

KU5 Recall the code needed to handle errors in a given SQL statement.		
	Maximum	Awarded
Q4 – Procedure works correctly.	3	
Q4 – Custom error message.	2	
Total	5	

SE2 Evaluate a database system and propose manners of making it more resilient.		
	Maximum	Awarded
Q5 – Necessary changes made to accept file as a parameter.	10	
Total	10	

AA4 Develop an application with CRUD functionalities that communicates with a DB.		
	Maximum	Awarded
Q6 – Application connects to DBMS.	4	
Q6 – Correctly handled errors in application.	3	
Total	7	

SE3 Develop a series of queries that cater for concurrency issues.		
	Maximum	Awarded
Q6– Menu option 1 implemented correctly	2	
Q6– Menu option 2 implemented correctly	2	
Q6 – Menu option 3 implemented correctly	3	
Q6 – Correct transaction isolation level (1 each)	3	
Total	10	

KU4 Recall the syntax to create an index and how to check its impact on a DB.		
	Maximum	Awarded
Q7 – Correct code provided for selectivity ratio.	3	
Q7 – Correct database object created.	2	
Total	5	

AA3 Interpret query execution plans and identify optimization techniques		
	Maximum	Awarded
Q7 – Correct query that benefits from database object.	3	
Q7 – Correctly compared statistics.	4	
Total	7	

AA5 Prepare a number of reports generated from XML data stored within a DB.		
	Maximum	Awarded
Q8 – Report 1 generated successfully.	3.5	
Q8 – Report 2 generated successfully.	3.5	
Total	7	