DM / CASE ASSIGNMENT

(*Mandatory assignment*)

REQUIRED DELIVERABLES

1. Database Design documentation (One Word or PDF document: ***DM_Case_Design****.docx or.pdf*)
1.1 ER diagram, 1.2 Relational schema, 1.3 Repository, and 1.4 Database diagram (in STEP 2)

2. SQL Scripts (four ***.sql*** files)

2.1 SQL script to create the database structure *in SQL Server*
2.2 SQL script to create the indexes
2.3 SQL script to populate the database with a *reasonable amount of proper data* for testing
2.4 SQL script to test the database (*see STEP 2 below*)

3. Project Closing Report (One Word or PDF document: ***DM_Case_Closing_Report****.docx or .pdf*)

3.1 Evaluation of the product and the process
3.2 Working time records per each person per each task

THE DEVELOPMENT PROCESS / Group work (*Group size: 2-3 students, 2 is the preferred size*)

STEP 1: Database Modelling

Administrative

- Record and ***report working time by person and task*** (otherwise, your points will be reduced)

Conceptual database modelling

- Create a conceptual model of the database and visualise it as an ***ER diagram***.
- Write entity type definitions to the ***repository*** (*see the separate repository template*)

Logical database design

- Derive a ***relational schema*** from your ER diagram.
- Validate the relations with the BCNF rule. Fix the relations if they are not in BCNF.
- Define integrity constraints for your relations. Document them in the ***repository***.

→ **Submit STEP 1 results** to Moodle in a **single ZIP file** (***DM_Case_Step1_****Surname_Surname.zip*)

Submit the results ***before*** you move on to STEP 2!
* You might need to improve the design later. This will be included in the final submission ☺

STEP 2: Physical design, database implementation in SQL Server and testing

Administrative

- Record working time ***by each person and task*** (otherwise, your points will be reduced)

Physical design and implementation

- Write an SQL script to create the database structure in SQL Server
- In SMSS, create a database diagram of your database tables
- Design a basic set of indexes on your tables and write an SQL script to create them

Testing

- Design test data and write an SQL script to insert it into the database.
- Write an SQL script for the 15 user transactions listed at the end of this document.
- Test your implementation by executing the SQL statements and write a short test report.

STEP 3: Evaluate your work

Write a ***project closing report*** where you discuss the questions below.

1. How well does the product meet customer's requirements?
2. What went well? What was difficult?
3. What did we learn? What will we do better next time?

Include in the report the ***working time records per each person***.

STEP 4: Submit everything to Moodle in a ***single ZIP file*** (***DM_Case_Final_****Surname_Surname.zip*)

THE CASE - Cardiff Festival Association

The *Cardiff Festival Association* is an association organizing concerts, dance and theatre performances and other cultural events. So far, tickets have been sold in person at the Cardiff library. Now the association is keen on offering web pages, where clients could see upcoming events and book tickets.

When a client makes a booking he/she should get a unique booking number. The booked tickets should still be purchased at Cardiff Library within three days from booking.

Further details

- The *Cardiff Festival Association* has two venues to hold its events. The smaller hall accommodates 250 people. The larger hall accommodates 3500 people. It is possible that other venues may become available in the future.
- The seats are not numbered.
- The fire safety regulations do not allow any overbooking.
- All tickets to a certain event are sold at the same price.
- Different events may be priced differently.
- An event can also include performances from several different artists.
- Popular performers visit Cardiff often. The Cardiff Festival Association has an extensive registry of artists – domestic and foreign – with their contact information (email, phone) and special requests for catering and refreshments.

- The booked tickets are identified and purchased by booking number.
- A booking can only contain tickets to one event.
- A booking can be cancelled, unless it has already been already purchased (payed for).
- The number of tickets in a booking can be changed, unless the tickets are already purchased.
- Once the tickets are purchased, they can neither be changed nor refunded (except for event cancellation).
- If the event is cancelled, the client can receive a refund of the amount paid for the ticket. The refund is given in person at Cardiff Library.
- All the tickets included in the same booking must be purchased at the same time.
- Only the client's *phone number* is registered in a booking. No other client information is stored in the database.

The new database application should help to complete the tasks (user transactions) listed below[1].

1. Booking a ticket
2. Cancelling a ticket booking
3. Changing the number of tickets in a booking
4. Changing the status of booking to sold when tickets are purchased (the booking is payed for)
5. Removing the unpurchased bookings from the database after three days from booking[2].
6. Cancelling an event (in an extreme exceptional case)
7. Refunding a client in a case of a cancelled event.

Examples of the queries (user transactions) the new application should support are the following:

1. What is Ed Sheeran's contact email?
2. What dance performances are coming up next month?
3. What concerts are coming up this month?
4. When will *Saara Aalto*[3] perform in Cardiff and what are her special requests for catering?
5. How many tickets have been sold to *Riverdance's* dance performance ''Riverdance 2020" that takes place on 10.7.2020?
6. How many tickets are there left for *ZZ Top's*[4] concert on 22.1.2020?
7. How much money has the Cardiff Festival Association got from sold tickets this year?
8. Which artist has sold the highest number of tickets this year. Please notice that the artist can have performed several times this year. All the artist's performances count here.

NB! In this case assignment, you will use the user transactions mentioned above to

1. Validate the design of the database
2. Test the physical implementation of the database.

INSTRUCTION

Validate and test your database structure by ***writing and executing the SQL statements*** for the 15 user transactions mentioned above.

[1] In this case assignment, your task is just to ***write the required SQL statements*** for the listed user transactions. There are 7 + 8 SQL statements to be written.

[2] In this case assignment, you can test this by executing an update manually. The query should get the current date from the DBMS.

[3] *A famous female singer who is known from her X-Factor UK and Eurovision song contest performances.*

[4] *The beard band where Beard does not have a beard.*