Examination – Data Management

This folder contains multiple files, which is going to be filled by you with the solutions to each assignment. This file describes the tasks you must perform, while the other files are referred to via the tasks described below. Each task has a percentage assigned to it that gives you an indicator of how much the task weighs towards your grade for the examination. Be aware that correct submission also counts as a percentage of your grade. **Remember that all SQL submitted at this exam must run in PostgreSQL – other dialects are not acceptable**. Mind your time usage on each task (each task has a time recommendation attached). The following sections describe the tasks you need to perform.

Task 1: Relational Databases Part 1 (55% ~ 66 min)

Subtask 1: ER and EER diagrams

This subtask concerns itself with generating an ER/EER diagram from a textual context.

For this task, make sure you either open the Task1Subtask1.drawio file (if you have the offline version of draw.io), or download your diagram and override the Task1Subtask1.drawio file. If you are using other tools, make sure you export it into a commonly readable format such as PNG, JPG, or PDF, and call the file Task1Subtask1 when done. You are allowed to draw the diagram by hand and attach a picture of it instead but be very sure everything is easily readable as we cannot grade anything we cannot read.

The text below is the output of an interview. This is the only documentation you have, so if something is unclear, you will have to make the decision on how to understand it - just make a note in the diagram explaining your reasoning.

You are allowed to describe identifying relationships and weak entities with a thick border (I recommend 4pt), just as was done in the solution from DM04. Remember to be as precise as possible and include all the types of descriptions you have learned (some might not be applicable for the case though). This includes strong and weak entities, inheritance, relationships, cardinalities, keys and more.

Interview Resuls (in Danish):

John er ejer af ”Make Me Mobile!”, der er en køreskole der henvender sig til private kunder og erhvervskunder som ønsker at anskaffe kørekort til en almindelig bil eller taxa. John har to andre kørelærer ansat, og overvejer at tilføje flere til sin stab i fremtiden.

Nuværende samler John Navn, CPR-nummer, Telefon nummer og E-mail på sine privatkunder. For hans erhvervskunder samler han desuden også CVR-nummer, Firmanavn og firmaets telefon nummer. Han oplyser at firmaer der betaler for et taxakørekort, betaler for en specifik persons kørekort. Derfor er det ekstra oplysninger han indsamler hvis det er et firma der betaler. Han har også flere kunder der først kommer privat og får et kørekort, og senere kommer igen og får taxa kørekort igennem et firma. Det er desuden vigtigt at ”Make Me Mobile!” Kan se forskel på hvilet type kørekort en person er ved at tage, og om det er blevet tildelt, og på hvilken dato det blev tildelt. Kørekort typerne er Privat bil, eller Taxa kørekort. John oplyser desuden at de skal gemme alle testforsøg (teori og praksis) der er lavet på hvert individ med dato og resultat, da dette er et lovkrav. Han skal også kunne se hvilke datoer der har været køretimer (som altid har samme længde), og hvilken kørelærer der har udført køretimen, da dette også er et lovkrav.

Subtask 2: Mapping to Tables

This subtask concerns itself with mapping an ER diagram (or EER) to tables in a database.

For this task, make sure you either open the Task1Subtask2and3.sql file. The file is empty, and you will have to fill it out with the database create script for the ER diagram you made in ***Task1 Subtask 1.*** If you were not able to create the diagram, use the interview result text as the basis for your tables. Once completed, add two driving instructors, and three customers of which one is a company type. Each customer should have at least one driving lesson, and one test result.

Subtask 3: Querying a database

This subtask concerns itself with querying databases. Open Task1Subtask2and3.sql, and add the following two queries at the end of the file:

1. Query for all driving lessons performed by the first driving instructor, that are paid by a company. The script results should only show the dates and the driving instructor name.
2. Create a query that counts all the driving lessons made by the second driving instructor. It should output only the name, and a number with the amount of driving lessons given.

Task 2: Relational Databases Part 2 - Normalization (20% ~ 24 min)

This task concerns itself with normalizing a given dataset to tables in SQL, and making a query within the normalized tables

The output of this task should be saved in the Task2.sql file, which is currently empty. Either open the Task2.sql file directly and save to that file using your editor (default pgAdmin) or copy your code from pgAdmin into the Task2.sql file. If you choose to copy it, then copy it often to make sure you do not lose your work. If you would like to explain your choices, please do so as SQL comments.

Movie Title	Actors	Director	Genre	Release Year	Box Office Date
Die Hard	Bruce Willis, Bonnie Bedelia	John McTiernan	Action, Thriller	1988	20 july
Hostage	Bruce Willis, Kevin Pollak	Bruce Willis	Action, Chrime, Drama	2005	11 march
Die Hard	Bruce Willis, Bonnie Bedelia	John McTiernan	Action	1989	Denmark, 19 May
1922	Thomas Jane, Molly Parker	Zak Hilditch	Crime, Drama & Horror	2017	Netflix
Special Delivery	Unknown	Unknown	Propaganda, Short Film, War	1946	Null
Journey to the Edge of the Universe	Null	Yavar Abbas	Documentary, Nature	2008	Null

Tasks are as follows:

- Normalize the dataset above to at least third normal form.
- Create the tables in SQL in Task2.sql.
- You do **NOT** have to insert the data into the tables, just create the tables, and the relationships.

Task 3: NoSQL Databases (20% ~ 24 min)

Subtask 1: JSON

Open the file called “Task3Subtask1” from the zip file. Create a json object that contains the following content:

1. The file must contain an array of two objects.
2. Each object signifies a product and has the following properties: Name (string), Manufacturer (string), Price (number)
3. Also, each of the objects have a property called Owners, which contain an array of strings with the current and previous owners’ names.
4. Fill all properties with text and numbers that make sense for the given products and owners. Use your imagination.

Subtask 2: NoSQL Databases

For this subtask, open the file Task3Subtask2.txt and answer the questions contained within the file. Write your answers instead of the "<<ANSWER HERE>>" tags.

The questions within the file are as follows:

1. Based on the interview results with John from "Make Me Mobile!" from Task 1 Subtask 1, should they use a relational database or NoSQL database? Explain your reasoning for why they should make that choice, and what the benefits would be to go in the other direction of your recommendation.
2. Explain the difference between Horizontal and Vertical Partitioning/Sharding, and how the concept is different from Horizontal and Vertical Scaling.

Submission (5% ~ 6 min)

To submit the assignment, make sure you have a folder containing the following files:

- Task1Subtask1.drawio (alternatively .jpg, .png, or .pdf)
- Task1Subtask2and3.sql
- Task2.sql
- Task3Subtask1.json
- Task3Subtask2.txt

Check each file has your work contained within, and then Zip the file. Now rename the file to your SDU username and postfix it with " – DM", such as "abcd12 - DM.zip". After these steps are complete, go to the assignment location from where you downloaded the counting activity, and upload the zip file. For your own security, please also try and download the file again and unzip it to verify the version stored on the server works as intended.