Examination Information

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 that indicates how much the task weighs toward your grade for the examination. **Remember that all SQL submitted at this exam must run in PostgreSQL – other dialects are unacceptable**. Mind your time used on each task (each task has a time recommendation attached), but remember there are multiple subtasks for each task! The following sections describe the tasks you need to perform. **READ the entire document before solving the assignments!**

Each task can have subtasks – remember to read them all before you start. The tasks present in this exam:

- Task 1
  - Subtask 1 – ER and EER diagrams
  - Subtask 2 – Mapping to Tables
  - Subtask 3 – Querying the Database
- Task 2
  - Subtask 1 - Normalization
- Task 3
  - Subtask 1 – Database Comparisons
- Task 4
  - Subtask 1 – JSON objects
- Submission

Task 1: Relational Databases – Part 1 (3 subtasks) (60% ~ 72 min)

Subtask 1: ER and EER diagrams (30 min)

This subtask concerns 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 use other tools, 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 sure everything is easily readable, as we cannot grade anything we cannot read.

The text below is the output of an interview. This is your only documentation, so if something needs to be clarified, you must decide how to understand it - make a note in the diagram explaining your reasoning. Remember to be as precise as possible and include all the types of descriptions you have learned (some might not apply to the case, though). This includes strong and weak entities, inheritance, relationships, cardinalities, keys, etc.

Interview Results (in Danish):

Cleandestine er et internationalt renseri for hemmelige sikkerhedsorganisationer. Cleandestine går meget op i service for deres kunder, hemmeligholdelse, samt at kundernes aktiviteter støttes. Firmaet har eksisteret siden 1985, og har nu 52 renserier fordelt over Europa, og forventer at ekspandere til USA i løbet af de næste år. For at dække behovene fra den store mængde af hemmelige sikkerheds services som bruger deres rengørings services, skal de i dag skabe en ny applikation (læs; database) for at understøtte deres services beskrevet herunder.

Basisfunktionerne der skal understøttes, er følgende. Man kan ikke gå ind fra gaden og købe services fra Cleandestine, man skal være oprettes som organisation. Når man ringer ind, identificerer man sig som agent af en registreret organisation med et kodeord. De forskellige agenter har forskellige kodeord. Under opkaldet skal placering (noget der skal ske på stedet), afhentningssted, eller indleveringssted, samt tid aftales. Derudover skal typen af rensning også gemmes. Eksempler på typer kan fx. være "Komplet service" som inkluderer fjerneste af lig, blod og mere, og i den anden ende af skalaen er der "Almindelig tøj rens". Systemet skal understøtte flere forskellige typer så services kan udvides i fremtiden. Når rensningen er overstået, er der nogle typer af rens, fx af tøj, som kræver tilbagelevering til agenten. Her skal der så også registreres lokation for afhentning (materiale kan sendes imellem renserier), eller om der skal leveres i feltet (evt. kode for genkendelse af agenten skal også gemmes).

Ud over ovenstående funktioner, skal systemet også gemme medarbejderne af Cleandestine, deres kodenavn, samt hvilke remedier de bruger til en evt. levering. Herunder findes der to typer fartøjer (som fx fly, helikoptere, samt motorcykler), og andre remedier, som fx droner, faldskærme, og klovnekostymer. Af og til skal der bruges mere end et fartøj og remedie til levering. Hver medarbejder har også mindst et ekspertiseområde som matcher med typen af rens der skal foretages; dette skal også registreres, så det er lettere at finde den rigtige til opgaven.

Eksempelvis skal man kunne ringe ind, fortælle at man kommer fra NSA, give en personlig kode, og bestille servicen "vådt arbejde" i Yemen med levering i

Maryland. Hertil kan man fortælle at det kræver fly og faldskærm. Cleandestine vil dermed kunne matche opgaven op med James, der er specialist i vådt arbejde.

Subtask 2: Mapping to Tables (30 min)

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

For this task, make sure you open the Task1Subtask2and3.sql file. The file is empty, and you must fill it out with the database to create the script for the ER diagram you made in ***Task 1 Subtask 1.*** If you could not create the diagram, use the interview result text as the basis for your tables.

Subtask 3: Querying a database (12 min)

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

1. Add two rows to each table that is used in point 2 and 3 database (one employee needs to have used a car for delivery).
2. Query for all pickups before the current date.
3. Query for the codename of each employee of Cleandestine that used a Car, and the codename of the customer.

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

This task concerns itself with normalizing a given dataset (the excel image below) to tables in SQL and making a query within the normalized tables.

The output of this task should be saved in the Task2.txt file, which is currently empty. Open the Task2.txt file directly and save to that file using your editor. Feel free to write comments.

Vought International

Hero	Villan	Event	Nemesis	Power Origin
Homelander, age 40	Serge AKA Frenchie	Alien Invasion	Billie Butcher, Hughie Cambell	Lab Experiment
Starlight, Age 23	Hughie Cambell	Alien Invasion, Basket Court	Homelander	Injection
The Deep, 5 Arrests	Billy Butcher	Basket Court	Fishermen	Injection
Queen Maive	Kimiko Miyashiro	Alien Invasion	Homelander	Injection
Black Noir, Room 3461	Stormfront	Epic Encounter at Brightford, Alien Invasion		Injection
A-Train, age 34		Epic Encounter at Brightford	Himself	Injection

Tasks are as follows (read all steps first!):

- Normalize the dataset to at least the third normal form.
- Write the normalized schema in Task2.txt using the template/shorthand form (for example ***TableName(PK id, prop1, prop2, FK prop3)***)
    - Alternatively, you are allowed to use "create table" statements if you are not comfortable with the template/shorthand notation.
- You do **NOT** have to insert the data into tables!
- **Remember** the relationships with primary and foreign keys. Also, show the relationships in the shorthand format at the end: **"Table1" M -> N "Table2"**.

- Feel free to write notes/reasoning for your choices in the document.

Task 3: Database Comparisons (10% ~ 12 min)

Open the file called "Task3.txt" from the zip file. The file is empty. Write a few paragraphs that explain the following:

- What are the commonalities and differences between these three database types?
  - Relational Database
  - Column Oriented Database (Wide-Column database)
  - Graph Database

Task 4: JSON files (10% ~ 12 min)

Open the file called "Task4.json" from the zip file. The file is empty. Write a JSON Object with the following information (Remember to fill each object with some information so it is not empty):

- The root of the file should be a list of objects.
- Each object represents a house with the properties: Address, Zip, City, Occupants, and Owners.
- Occupants and Owners are of the type Person, with the properties: First Name, Last Name, Spouse (of type person), Age.

Remember to fill in the information in the objects as it helps me check the object's validity.

Submission

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

- Task1Subtask1.drawio (alternatively .jpg, .png, or .pdf)
- Task1Subtask2and3.sql
- Task2.txt
- Task3.txt
- Task4.json

Instructions:

1. Check that each file has your work contained within it.
2. Zip the files without the folder containing them (mark all files and zip them together).
3. Name the zip file after your SDU Username, such as "abcd17.zip.
4. Go to the assignment location from where you downloaded the exam.
5. Upload the zip file.
6. For your security, download the file again and unzip it to verify that the version stored on the server works as intended (and that you included the correct files – yes, this happens!).