

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. READ the entire document before solving the assignments!

Task 1: Relational Databases Part 1

(60% ~ 72 min)

Subtask 1: ER and EER diagrams (30 min)

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. 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 Results (in Danish):

Jan har en virksomhed ved navn "Jan og Søns Kunstige Billeder" der lever af at lave kunstrig intelligens til redigering af billeder. Virksomheden er nuværende på to mand og har kun 4 kunder. Han ved dog at virksomheden over det næste år regner med at ekspandere til over 50 kunder igennem et samarbejde med en ny partner. For at kunne samle billeder til at træne hans modeller til kunstig intelligens skal han have lavet en database til at gemme kundernes billeder, da hver kunde skal levere flere hundrede tusinde billeder til træningen.

Jan ønsker at bruge databasen til at holde et register over hans kunders navne, adresse, telefonnummer, og email adresse (en telefon og email per kunde). Hver kunde giver Jans virksomhed en licens til at bruge billederne. Licensen kan være enten "Fri brug", "Fri brug til træning", "Kun til brug af kundens modeller". Hvert billede han får fra en kunde, skal gemmes hvor billedet ligger i filsystemet, hvilken type dette billede er (fx jpg eller png), samt dato for modtagelse af denne. Desuden er det nødvendigt for Jan at kunne analysere billederne og definere en mængde af etiketter som beskriver indholdet af billedet. Dette kunne fx være "Indendørs", "Køkken", "Bil" eller mange andre etiketter. Jan har også brug for at en kunde kan angive to typer af filer for et enkelt billede. Det ene skal være originalbilledet, og det andet skal være et redigeret billede som kunden normalt sender til deres egne kunder. Originalbilleder og redigerede billeder har altid samme etiketter, og deler generelt de samme informationer. Til sidst nævner Jan at nogle billeder bliver sendt ind som et samarbejde imellem kunder, og at det ville være rart for ham hvis den type billeder kun lå i systemet en gang selv om det var relateret til flere kunder.

Subtask 2: Mapping to Tables (30 min)

This subtask concerns itself with 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 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.

Subtask 3: Querying a database (12 min)

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

1. Add two entries to each table.
2. Query for all labels (etiketter) that has been associated to at least one image.
3. Query for all images that has two owners. Include the owner's names and email, as well as the image location in the file system.

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.txt file, which is currently empty. Either open the Task2.txt file directly and save to that file using your editor. Feel free to write comments.

New Owner	Brand	Model	Registration Number	Producer	Sales Person	Sales Date
Kastanjevej 34, 5000 Odense	BMW	BMW i4, 2018	HONKTWICE	BMW	André Jensen, 310.000 kr, +45 21 34 45 48	may 2020
Arne Henrik Olsen	Seat	Leon	FF23455	VW	unknown	19 june 2018
Dennis Hansen	VW	VW Golf, 2001	AA81543	VAG Group	Dennis Hansen, 45.000 kr, +45 21 34 45 45	2016
Anne Hansen, +45 54 45 33 44	VW	VW Golf	AA81543	VAG Group	Dennis Hansen, 10.000 kr, +45 21 34 45 45	january 2021
null	Citroen	C1 2019	VK42344	Peugeot-Citroën SA	null	null

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

- Normalize the dataset above to first normal form.
- Normalize the first normal form to the second normal form.
- Normalize the second normal form to the third normal form.
- Create all normal forms in Task2.txt using the template form (for example **TableName(PK id, prop1, prop2, FK prop3)**) or use "create table" statements (your choice).
- You do **NOT** have to insert the data into the tables (if you choose to create tables).
- Remember the relationships with primary and foreign keys.
- Feel free to write notes/reasoning for your choices in the document.

Task 3: Database choices (10% ~ 24 min)

Open the file called "Task3.txt" from the zip file. Below is a statement from a customer. What database would you choose for this customer? Your answer, together with your considerations, has to be written in Task3.txt. **Make sure to motivate your choice:**

Dan k r et stort firma med 18000 kunder som sammens tter relationer imellem kunder og firmaer. Dette inkluderer hvilke firmaer der er ejet af hinanden, hvem der s der i direktionskredsen, og hvem der ejer dem. Desuden holder firmaet styr p  hvilke kunder der k ber ydelser p  tv rs af de forskellige firmaer, men ogs  hvad deres politiske holdninger er, og hvordan de stemmer. Der er nuv rende 7 millioner kunder registreret i systemet. Dan lever godt af at udnytte denne viden til at s lge disse informationer til alt fra stater til andre firmaer der pr ver at m lrette deres salg.

Task 4: Regular Expressions

(10% ~ 24 min)

Open the file called “Task4.txt” from the zip file. The file contains text under the line. You will have to create two regular expressions that ONLY match the stated assignment. Write the answers in the Task4.txt file in the top next to the questions in the top. The assignments are as follows:

1. Match only quoted text.
2. Match only lines that end with a question mark.

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.txt

Instructions:

1. Complete both the DM and VOP assignments
2. Check each file has your work contained within it.
3. Zip the files.
4. Rename the file to “DM.zip”.
5. Leave VOP.zip next to it, and zip both files together in one file.
6. Name the zip file after your SDU Username such as “abcd17.zip”.
7. Go to the assignment location from where you downloaded the counting activity.
8. Upload the zip file.
9. For your own security, download the file again and unzip it to verify the version stored on the server works as intended (and that you included the right files – yes this happens!).