First Name(s)	Last Name	Birthday

DM505 Database Design and Programming

24h Take-Home Exam

Handed out: April 7, 2015, 9:00

First name(s): ______________________________

Last name: ______________________________

Birthday: ______________________________

Organization of the Exam

You have at most 24 hours to answer this take-home exam. You may use the course book, the slides from the lecture, and your own notes on the lectures and the exercises. You are specifically not allowed to discuss the tasks with any other person but the teaching assistants and the lecturer. For questions, contact the lecturer by e-mail (jan.baumbach@imada.sdu.dk) or phone (+45-65-50-23-09) from 9:00 until 18:00.

You must answer in English language. Your solution must be handed in as **a single PDF** file clearly stating your full name and birth date on all pages. You have to hand in using SDU Assignment **before 09:00, April 8, 2015**. In case of irrecoverable technical problems with SDU's Blackboard system (and only in this case), you may hand in the exam electronically by sending an email to the lecturer.

You may use external tools, such as Microsoft Powerpoint, dia (http://projects.gnome.org/dia/), or OpenOffice Draw (http://www.openoffice.org/) for E/R diagrams, trees and hashes. For obtaining a single PDF file from multiple PDF files, free solutions such as PDFSAM (http://www.pdfsam.org/) or PDFMerge (http://www.pdfmerge.com/, online version free until 15 MB) are available. If you have access to a scanner, you can alternatively hand-write your solutions on the exam pages.

First Name(s)	Last Name	Birthday

Task 1 – E/R model and the relational model

You have been hired at a game company and you have to create a database for their new roleplaying adventure game.

The company describes what data objects they want to be modeled in their game:

Player having *name, item, money, zone id*

Monster having *name, money, zone id*

Boss having *name, item, money, zone id*

Item having *name, value*

Zone having *id*

They also describe the relationships between these objects:

- A player can only be in one zone at a time. Many players can be in the same zone.
- Different players can be in different zones.
- Each zone has exactly one boss, but many monsters.
- Each player/boss/monster can only have one item at a time, but many players/bosses/monsters can have a copy of the same item.

Solve the following tasks:

First Name(s)	Last Name	Birthday

(a) Create an E/R diagram capturing the objects and relationships described above. Describe all your design decisions and constraints. Please use the notation for E/R diagrams introduced during the lecture.

First Name(s)	Last Name	Birthday

(b) Transfer the E/R diagram from (a) to the relational schema/model.

First Name(s)	Last Name	Birthday

Task 2 – Normalization Theory

In the following tasks assume the relation *R(A, B, C, D, E)* with the following functional dependencies:

AB → C, BC → D, CD → A, E → D

Solve the following tasks:

(a) List all keys of R and explain why there can be no other.

First Name(s)	Last Name	Birthday

(b) Show that R is not in BCNF, i.e. show that there is at least one BCNF violation. Then decompose R until it is in BCNF. Document the steps of the decomposition process and the resulting relations.

First Name(s)	Last Name	Birthday

(c) Analyze whether R is 3NF. If it is, show that there are no 3NF violations. If it is not, show that there is at least one 3NF violation and subsequently decompose the relations such that the resulting relations are in 3NF.

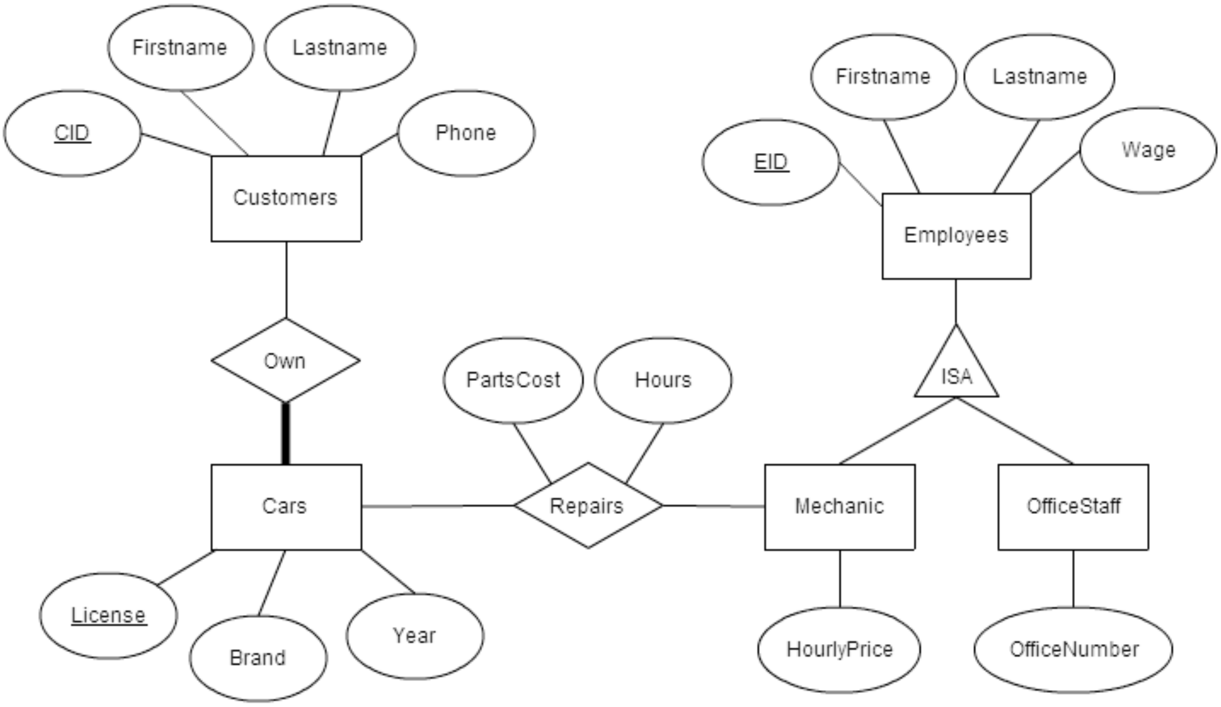
First Name(s)	Last Name	Birthday

(d) What types of anomalies exist? For each type: Give any valid example in which it might occur (you may invent a new database design, if you like).

First Name(s)	Last Name	Birthday

Task 3 – SQL for Data-Definition, Queries, and Modifications

A small garage uses a database to keep track of its customers, employees and the car repairs they do. To this end, the following database design is given:



This can be translated into the following relational model:

Cars(License, Brand, Year);

Customers(CID, Firstname, Lastname, Phone);

Own(CID, License);

Repairs(License, EID, PartCost, Hours);

Employees(EID, Firstname, Lastname, Wage);

OfficeStaff(EID, OfficeNumber);

Mechanic(EID, HourlyPrice);

Solve the following tasks:

First Name(s)	Last Name	Birthday

Create SQL statements performing the following tasks:

(a) Create the Mechanic-table while taking into account that the *HourlyPrice* may not be NULL and must be above 10 EUR (to avoid trouble with the union).

(b) Find the name of the mechanic(s) who carried out the single most expensive repair. The price of a repair is given by *PartsCost* + *Hours* * *HourlyPrice*.

First Name(s)	Last Name	Birthday

(c) Find the average *PartsCost* for repairs on cars, made in the year 2000 (i.e. *Cars.Year* = 2000).

(d) Update the the *PartsCost* to be 100 EUR for all repairs of cars of the brand “Mercedes”.

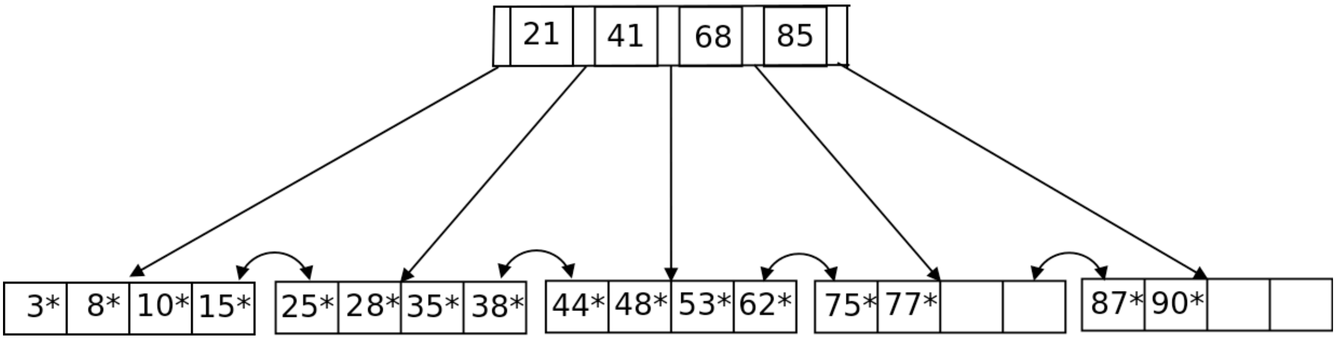
First Name(s)	Last Name	Birthday

Task 4 – Indexes using B$^+$-trees and Hash Tables.

You can use drawings to document the intermediate steps in the following tasks. It is fine to show only the relevant parts of the tree or hash table that you are working on. The final state should be the resulting index structure.

Solve the following tasks:

Given is a B+tree with n = 4 as follows.



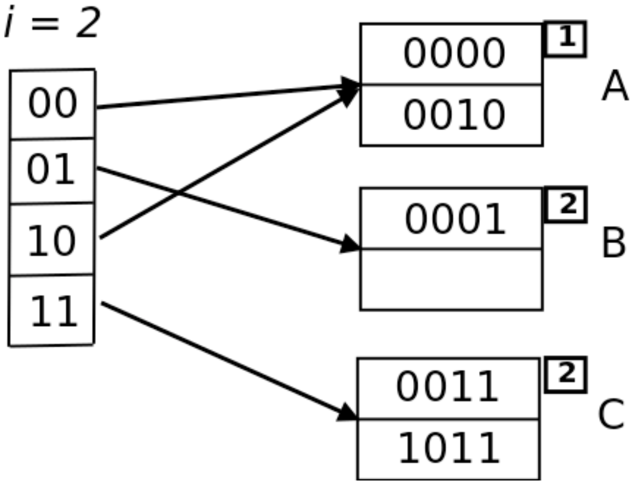
(a) Insert the key 17 into the tree above. Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

(b) Delete key 87 from the original tree (not the result for (a)). Document each intermediate step in the process and draw the final result

First Name(s)	Last Name	Birthday

Given is an extensible Hash table with k = 4, f = 2 as follows.



(c) Insert a key with the hash value 1000 into the extensible hash table above. Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

(d) Insert key 1111 into the original hash table (not the resulting one from (c)). Document each intermediate step in the process and draw the final result.