First Name(s)	Last Name	Birthday

Task 1 – E/R model and the relational model

Your task is to create an exam database for the university administration. They describe that they want the following objects in their system:

Exam having *exam id*, *date*, and *time*

Room having *name*

Course having *title*, and *lecturer*

Exam Assistant having *first name*, *last name*

Lecturer having *first name*, *last name*, and *phone number*

Student having *first name*, *last name*, *CPR number*

They also describe the relationships between these objects:

- Each course is taught by one lecturer and has one exam.
- Different students can be assigned to different rooms for the same exam.
- Each exam is served by a number of exam assistants.
- Keep in mind that both lecturers and exam assistants can potentially be students taking an exam.

First Name(s)	Last Name	Birthday

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

First Name(s)	Last Name	Birthday

(b) Transfer the E/R diagram from (a) to the relational model. Describe the process and the choices you have made.

First Name(s)	Last Name	Birthday

Task 2 – Normalisation Theory

In the following tasks assume the relation $R(A, B, C, D, E)$ with the following functional dependencies: $AB \rightarrow C$, $AC \rightarrow D$, $D \rightarrow B$, $E \rightarrow D$.

(a) List all keys of R and argue 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.

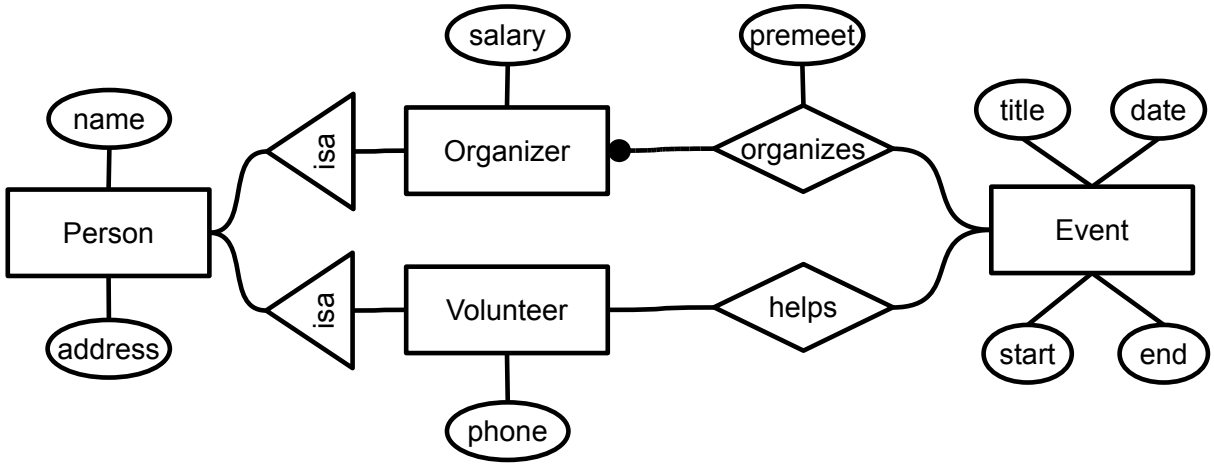
First Name(s)	Last Name	Birthday

(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 decompose the relations such that the resulting relations are in 3NF.

First Name(s)	Last Name	Birthday

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

The following E/R diagram describes the database of an event management organization. Here, for cost reasons, events are organized by exactly one professional organizer and any number of volunteers.



The following relational model is the result of the translation of the above E/R diagram incl. design decisions such as not combining the *Organizes* and *Event* relations:

$Person(\underline{name}, address)$ $\quad$ $Event(\underline{title}, date, start, end)$

$Organizer(\underline{name}, salary)$ $\quad$ $Organizes(\underline{name}, \underline{title}, premeet)$

$Volunteer(\underline{name}, phone)$ $\quad$ $Helps(\underline{name}, \underline{title})$

First Name(s)	Last Name	Birthday

(a) Events always start and end on the same day at exact hours, i.e., "start" is always a smaller integer than "end". Give a `CREATE TABLE` statement for creating the table *Event* that respects this.

(b) Write one `SELECT FROM WHERE` query that lists the names and phone numbers of all volunteers helping at the event 'Rock around the Clock'.

First Name(s)	Last Name	Birthday

(c) Write one `UPDATE` statement that increases the "premeet" time of all organizer-event pairs, where there are no volunteers associated to the same event, by 2 hours.

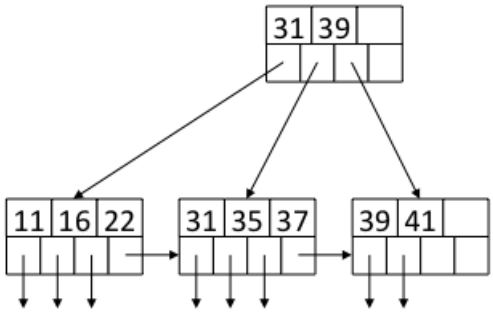
(d) Write one `SELECT FROM WHERE` query that computes the ratio of unpaid hours to paid hours for each event together with the amount paid to the main organizer according to his hourly "salary". Unpaid hours are defined as the duration of the event times the number of volunteers while paid hours are defined as the duration of the event plus the number of premeet hours.

First Name(s)	Last Name	Birthday

Task 4 – Indexes using B+trees, 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 a full drawing of the resulting index structure.

B+tree with $n = 3$



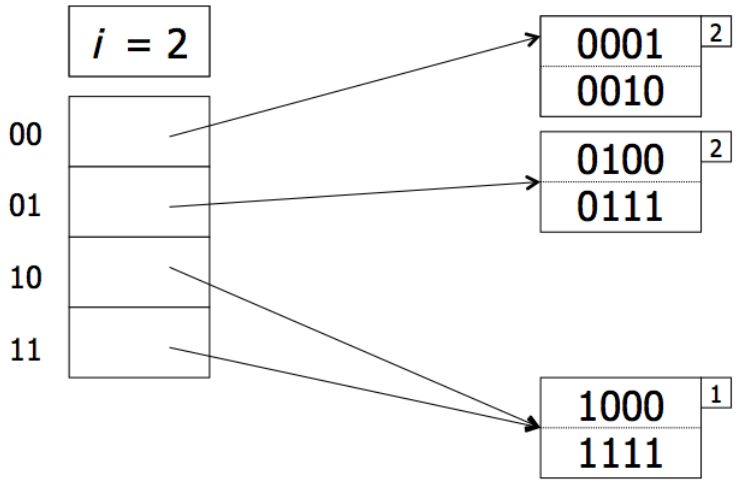
(a) Insert the key 40 into the B+tree above. Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

(b) Insert the key 17 into the original B+tree above (not the result of (a)). Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

Extensible Hash Table with $k = 4, f = 2$



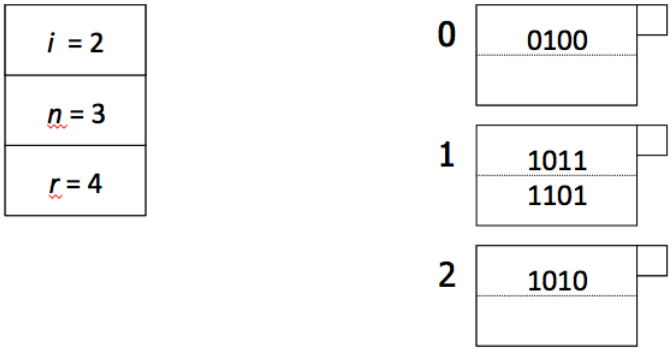
(c) Insert a key with the hash value 1100 into the extensible hash table above. Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

(d) Insert a key with the hash value 0011 into the original extensible hash table above (not the result of (c)). Document each intermediate step in the process and draw the final result.

First Name(s)	Last Name	Birthday

Linear Hash Table with $k = 4, f = 2, p_{max} = 0.8$



(e) Insert a key with the hash value 0010 into the linear hash table above. Document each intermediate step in the process and draw the final result.