PROJECT

WORK REPORT

Database technique

Name: sandra kaljula

ProgRAM: TGDU1 2020

Högskolan Krisitanstad

2021-02-13

TABLE OF CONTENTS

[Description of database 3](#_Toc64533951)

[The three tables 3](#_Toc64533952)

[Department table 3](#_Toc64533953)

[Room table 3](#_Toc64533954)

[Elder table 3](#_Toc64533955)

[Relations 3](#_Toc64533956)

[Department And room 3](#_Toc64533957)

[Room and elder 4](#_Toc64533958)

[2. ER MODEL 4](#_Toc64533959)

[3. Implementation Diagram 5](#_Toc64533960)

[4. EXAMPLE SQL STATEMENTS 5](#_Toc64533961)

[Datafields 5](#_Toc64533962)

[TEN SQL STATEMENTS 6](#_Toc64533963)

[FIRST 6](#_Toc64533964)

[SECOND 7](#_Toc64533965)

[THIRD 7](#_Toc64533966)

[FOURTH 7](#_Toc64533967)

[FIFTH 8](#_Toc64533968)

[SIXTH 8](#_Toc64533969)

[SEVENTH 8](#_Toc64533970)

[EIGHTH 9](#_Toc64533971)

[NINTH 9](#_Toc64533972)

[TENTH 9](#_Toc64533973)

[5. CONCLUSION 10](#_Toc64533974)

# Description of database

The database created is called retirement\_home. The goal of this database was to keep track of all the department’s rooms and elders.

## The three tables

The database consists of 3 tables – department, room and elder. All tables are normalized to the third normal form. The tuples have only one value each, all non-key attributes are fully functional dependent on the primary key and none of the non-key attributes are dependent on other non-key attributes in all tables.

### Department table

The department table has three attributes – id, name and floor. Id is of type int and contains the departments id. Name is of type varchar (45) and contains the name for the department. Floor is of type INT and contains the floor number of the department. Id is the primary key for this table.

### Room table

The room table has four attributes – number, size, view and department\_id. Number is of type int, containing the number of the room. Size is the size of the room in square meters and of type int. View is of type varchar (45) and contains the cardinal direction of the room’s windows. Lastly department\_id, which is a foreign key from the department table that is also of type int and contains the department’s id the room belongs to. Both department\_id and number create a composite primary key for this table.

### Elder table

The elder table has five attributes – ssn, name, moveInDate, room\_number and room\_department\_id. Ssn contains the social security number for the elder and is of type int. Name is of type varchar (45) and contains the name of the elder. MoveInDate is of type date and contains the date the elder moved in. Room\_number and room\_department\_id are the foreign keys from the room table. These attributes contain the room number, respective department id. The primary key in this table is ssn.

## Relations

A retirement home has departments, rooms and elders.

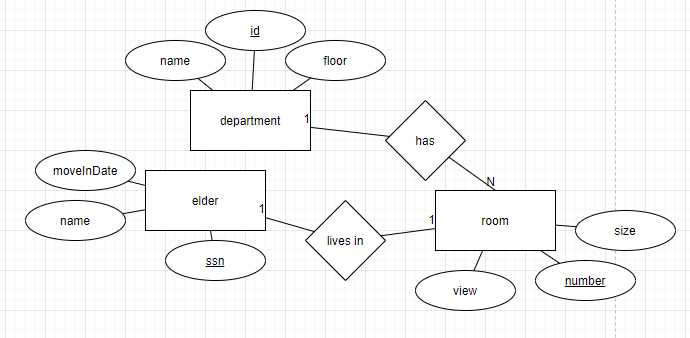
### Department And room

One department has an identifying relationship with many rooms. Every room needs a department to be uniquely identifiable. With both number and the foreign key department\_id being a composite primary key in the room table, there can be several rooms in the retirement home with the same room number, but in different departments. A room cannot exist without a department.

### Room and elder

One room has a non-identifying relationship with one elder. It means that there can only be one elder per room in the retirement home. In the elder table, the room\_number and room\_department\_id are foreign keys, but an elder can be identified by the ssn (social security number) alone. There can be an elder without a room and a room that is empty.

# 2. ER MODEL

 Image 1. The ER model of the database.

# 3. Implementation Diagram

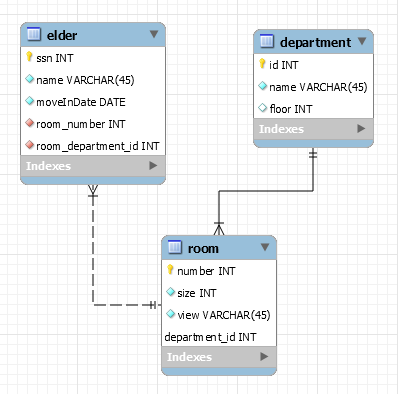


Image 2. The relational model (the EER diagram).

# 4. EXAMPLE SQL STATEMENTS

## Datafields

Images 3,4 and 5 show the datafields prior to any action has been taken.

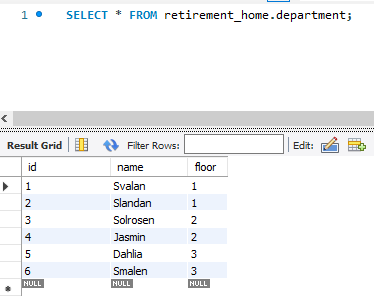


Image 3. Department table’s datafields from the start.

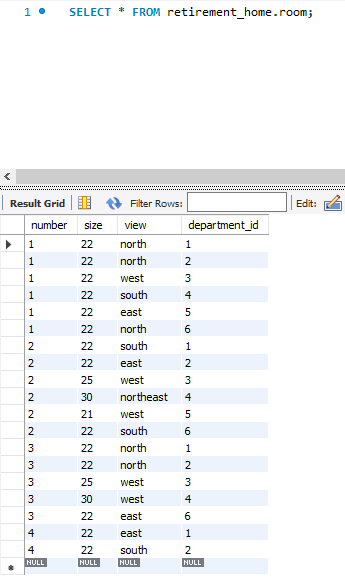


Image 4. Room table’s datafields from the start.

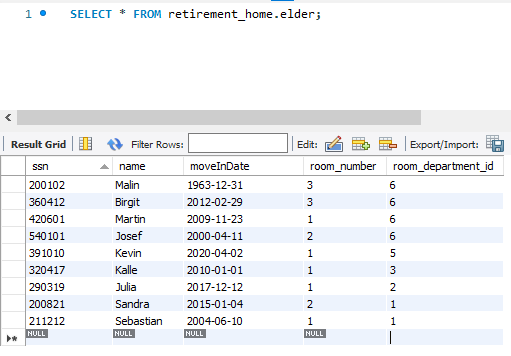


Image 5. Elder table’s datafields from the start.

## TEN SQL STATEMENTS

Ten SQL statements to show how the database can be used.

### FIRST

SELECT department.name, COUNT(room.number) as "rooms in total"

FROM department JOIN room

ON department.id = room.department\_id

GROUP BY department.id;

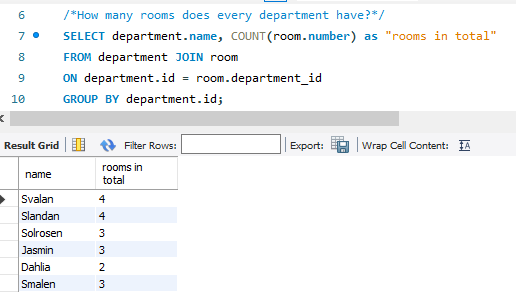


Image 6. Finding out how many rooms there are in total for every department.

### SECOND

SELECT elder.name, department.name as department, department.floor

FROM elder

INNER JOIN department

ON elder.room\_department\_id = department.id

WHERE elder.name = 'Julia';

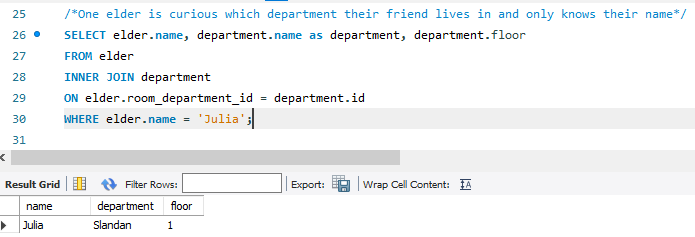


Image 7. Finding an elder named Julia in a department

### THIRD

INSERT INTO elder (ssn, name, moveInDate, room\_number, room\_department\_id)

VALUES (190923, 'Maria', '2021-02-12' , 4, 2);

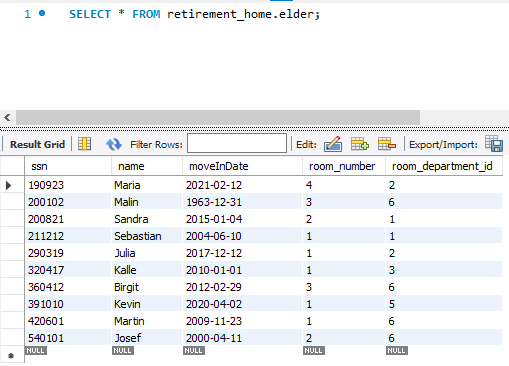


Image 8. A new elder called Maria moves into room 4, department 2.

### FOURTH

DELETE

FROM elder

WHERE ssn = 200821;

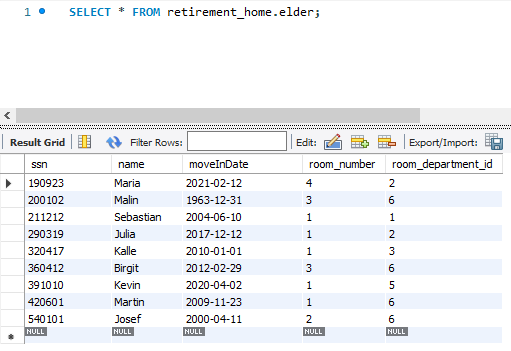


Image 9. An elder with ssn 200821 moves out.

### FIFTH

UPDATE elder

SET ssn = 190924

WHERE ssn = 190923;

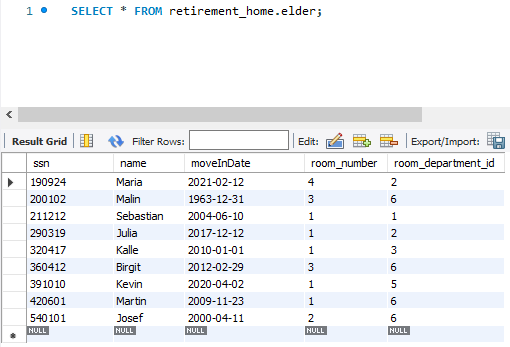


Image 10. Correcting an incorrect ssn value from 190923 to 190924.

### SIXTH

SELECT r.department\_id, r.number

FROM elder AS e

RIGHT JOIN room AS r

ON e.room\_number = r.number AND e.room\_department\_id = r.department\_id

WHERE e.room\_number IS NULL;

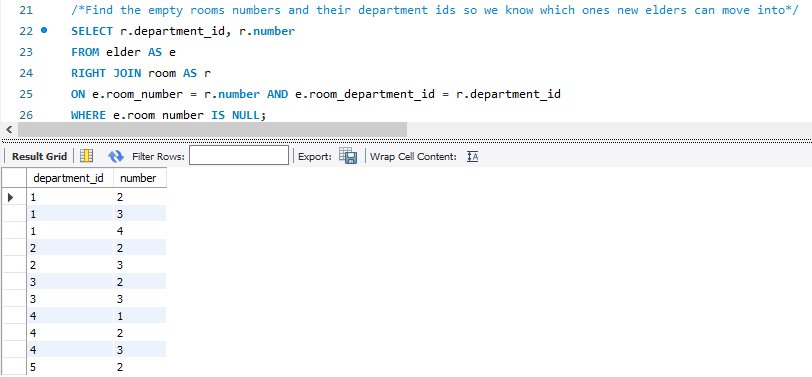


Image 11. Finding empty room numbers and their department ids, which new elders can move into.

### SEVENTH

SELECT name, room\_number, room\_department\_id

FROM elder AS e

LEFT JOIN room AS r

ON e.room\_number = r.number AND e.room\_department\_id = r.department\_id

ORDER BY r.department\_id;

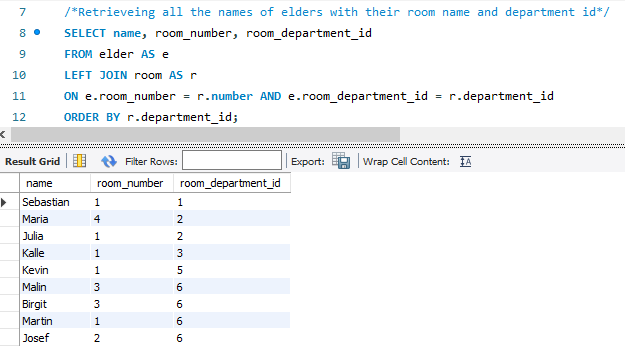


Image 12. Retrieving the names, room numbers and department ids of all elders.

### EIGHTH

SELECT name, ssn, YEAR(CURRENT\_TIMESTAMP) - (SUBSTR(ssn,1,2)+1900) as age

FROM elder

ORDER BY ssn ASC LIMIT 1;



Image 13. Finding the oldest person in the whole retirement home and their age.

### NINTH

SELECT name, ssn, moveInDate, room\_number

FROM elder

WHERE room\_department\_id = 6;

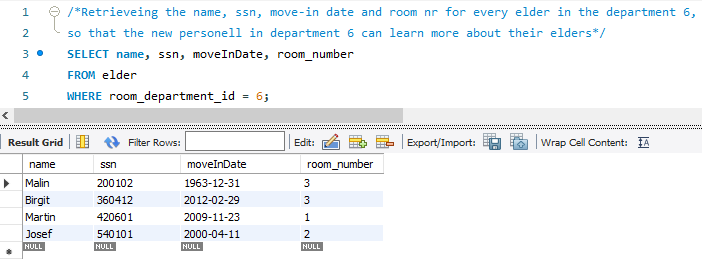


Image 14. Retrieving all information about elders living in department 6.

### TENTH

SELECT name, ssn, moveInDate

FROM elder

WHERE moveInDate in (SELECT MAX(moveInDate) from elder);

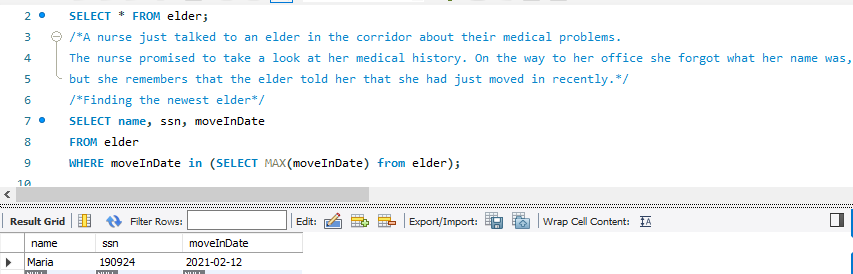


Image 15. Finding the newest elder in the retirement home.

# 5. CONCLUSION

This was my first database project I had built right from the beginning. So, there were a lot of things I learnt from this project. I learned to design a database – create an ER-diagram for a real-world example in my own mini-world. I worked a lot with MYSQL and there I learned to create an EER-diagram, fill in data and write SQL statements to work with the data in the database. I happened to search for instructions and information on the internet on this topic, which in turn teached me to use the vocabulary on this topic.

There were some complications when it came to filling in the data. For example, when creating a room in x-department I would first have to create an x-department to assign a room to it and apply the changes. I did not think of it at first, so I got a bit frustrated there.

Overall, I think the course literature, lectures and help sessions helped me to a great extent. Without this project and the struggles, I had during it, I do not think I would have learned as much about databases as I did thanks to this project.