Databases Laboratory 2: The Entity-Relationship Model

Lucca Heineck, luccou25@student.hh.se, DS4001-2025
Vinicius Beuther, vinbeu25@student.hh.se, DS4001-2025

Feb. 2025

1 Introduction

The objective of this lab is to practice how to design databases via Entity-Relationship (ER) Models. You will practice skills in:

- Translating domain descriptions to ER Models.
- Identifying different components in the domain and how they are related to the ER model.

In general, you will practice implementing databases given a domain description.

2 Task 1 - The Company-Employee Example

2.1 Task 1.a

Translate the following domain description to an ER diagram and implement it with an SQL statement:

"Create a database of companies of similar organizational structure. The companies have employees, and employees are divided into different departments within the company. Each division in the same company should have a unique name. Some employees have supervisors. Some employees are assigned as managers with special titles. Managers have access to company cars. Please also make an inventory of company cars and who can have access to it, and for which time period they have the access."

2.1.1 Q1: Entity Relationship Model

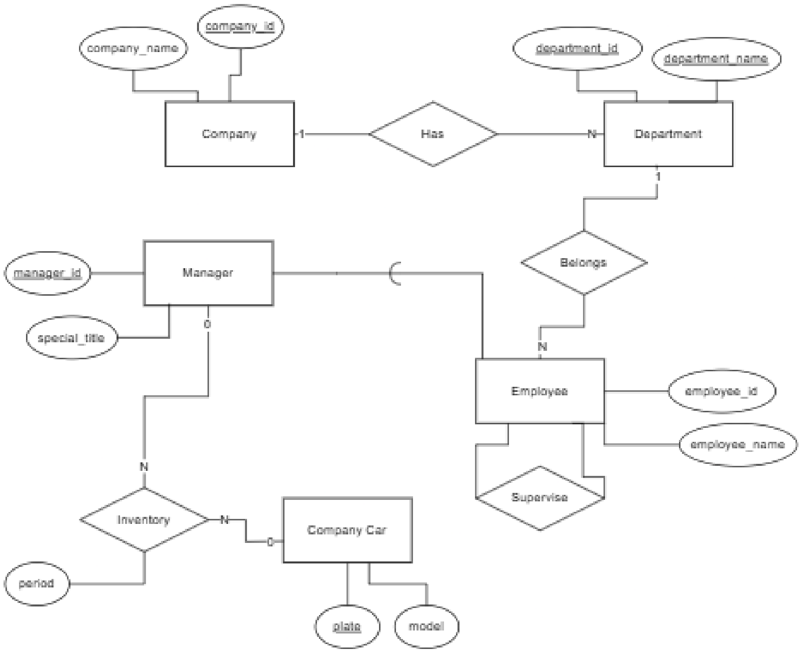

Figure 1: Company and Employees ER Diagram

2.1.2 Q2: SQL Statements

```
CREATE DATABASE hh_db_lab2;
USE hh_db_lab2;

# Lab 02

# Task 01a
# Creating Company table
CREATE TABLE IF NOT EXISTS Company(
        company_id INT AUTO_INCREMENT NOT NULL UNIQUE,
    company_name VARCHAR(80) NOT NULL UNIQUE,
    PRIMARY KEY(company_id)
);

# Inserting dummy data
INSERT INTO Company(company_name) VALUES
('Oxford'),
('Condor'),
('Tuper');

# Creating Department table
CREATE TABLE IF NOT EXISTS Department(
        department_id INT AUTO_INCREMENT NOT NULL UNIQUE,
    department_name VARCHAR(50),
    company_id INT NOT NULL,
    PRIMARY KEY(department_id),
    FOREIGN KEY(company_id) REFERENCES Company(company_id)
);
```

```
# Inserting dummy data
INSERT INTO Department(department_name, company_id) VALUES
(’Reception’, (SELECT company_id FROM Company WHERE company_name = ’Oxford’)),
(’Package’, (SELECT company_id FROM Company WHERE company_name = ’Condor’)),
(’CNC’, (SELECT company_id FROM Company WHERE company_name = ’Tuper’))
;

# Creating Employee table
CREATE TABLE IF NOT EXISTS Employee(
        employee_id INT AUTO_INCREMENT NOT NULL UNIQUE,
    employee_name VARCHAR(70) NOT NULL,
    department_id INT NOT NULL,
    PRIMARY KEY(employee_id),
    FOREIGN KEY(department_id) REFERENCES Department(department_id)
);

# Inserting dummy data
INSERT INTO Employee(employee_name, department_id) VALUES
(’Elsio’,
        (SELECT department_id FROM Department
        WHERE department_name = ’Packing’
    AND
    company_id = (SELECT company_id FROM Company WHERE company_name = ’Oxford’)
    )
);

# Creating Manager table
CREATE TABLE IF NOT EXISTS Manager(
        manager_id INT NOT NULL AUTO_INCREMENT UNIQUE,
    special_title VARCHAR(80) NOT NULL,
    FOREIGN KEY(manager_id) REFERENCES Manager(manager_id),
    PRIMARY KEY(manager_id)
);

# Inserting dummy data in Manager table
INSERT INTO Manager() VALUES(
        (SELECT employee_id FROM Employee WHERE employee_name = ’Vinicius’),
    ’Packing Supervisor’
);

# Creating Supervise table
CREATE TABLE IF NOT EXISTS Supervise(
        manager_id INT NOT NULL,
    employee_id INT NOT NULL,
    PRIMARY KEY(manager_id, employee_id),
    FOREIGN KEY(manager_id) REFERENCES Manager(manager_id),
    FOREIGN KEY(employee_id) REFERENCES Employee(employee_id)
);

# Inserting dummy data
INSERT INTO Supervise(manager_id, employee_id) VALUES(
        (SELECT m.manager_id
                FROM Manager as m
                JOIN Employee as e
        ON m.manager_id = e.employee_id
        WHERE e.employee_name = ’Vinicius’
    ),
    (SELECT e.employee_id FROM Employee e WHERE e.employee_name = ’Felipe’)
);
```

```
# Creating Company car table
CREATE TABLE IF NOT EXISTS Company_Car(
        car_plate VARCHAR(8) UNIQUE NOT NULL PRIMARY KEY,
    model VARCHAR(20) NOT NULL
);

# Inserting dummy data
INSERT INTO Company_Car(car_plate, model) VALUES
('ICK-1901','Omega'),
('MAE-1420', 'Kadett')
;

# Creating Inventory table
CREATE TABLE IF NOT EXISTS Inventory(
        car_plate VARCHAR(8) NOT NULL,
    manager_id INT NOT NULL,
    start_period TIMESTAMP DEFAULT(CURRENT_TIMESTAMP),
    end_period TIMESTAMP NOT NULL,
        PRIMARY KEY(car_plate, start_period, end_period),
    FOREIGN KEY(car_plate) REFERENCES Company_Car(car_plate),
    FOREIGN KEY(manager_id) REFERENCES Manager(manager_id)
);

# Inserting dummy data
INSERT INTO Inventory(car_plate, manager_id, end_period) VALUES(
        (SELECT car_plate FROM Company_Car WHERE model = 'Omega'),
    2,
    '2025-02-20'
);
```

3 Task 2 - The Boardgame Cafe Example

3.1 Task 2.a

Translate the following domain description to an ER diagram and implement it with an SQL statement:

”The database should store information about the board game café facility, registered customers, and their access record to the facility. Both current and past customers should be included in the database. Each customer has a name and an email address. Not every customer is a current member. The database should keep track of which customers are members currently, and the time period of their membership.

Each board game café is located in a city, with an address and a name. Two cafés can have the same name, but only if they are in different cities. You can assume cities have unique names, and cafés can be established in any city. The database should also keep the access record (including the café visited and when) of each customer.”

3.1.1 Q1: Entity Relationship Model

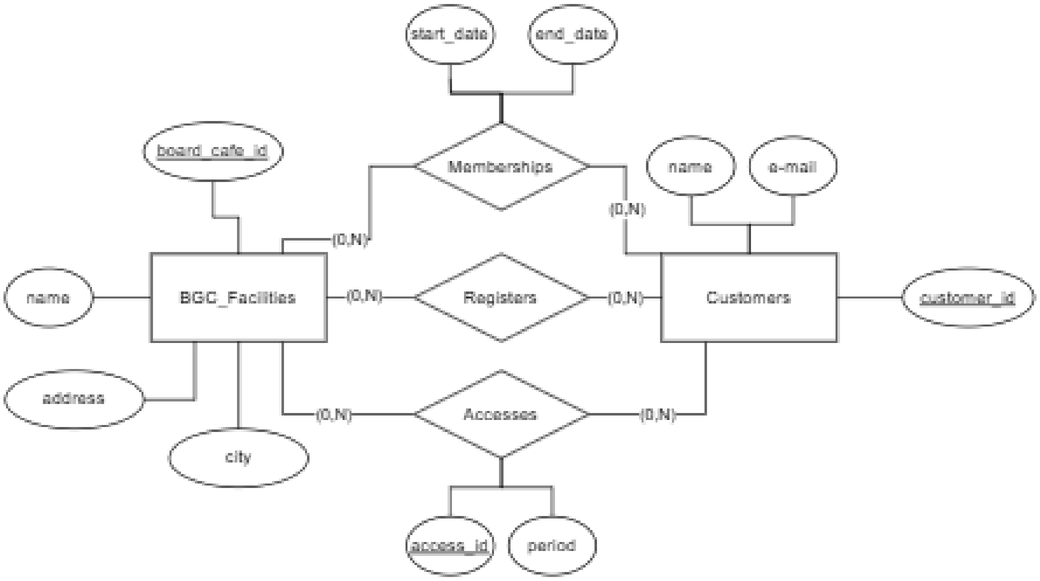

Figure 2: Board Game Cafe ER Diagram

3.1.2 Q2: SQL Statements

```
CREATE DATABASE hh_db_lab2;
USE hh_db_lab2;

# Lab 02

# Task 2

# Creating Board Game Cafe Facilities table
CREATE TABLE BGC_Facilities (
    id_board_cafe INT AUTO_INCREMENT PRIMARY key not null UNIQUE,
    address VARCHAR(255) NOT NULL,
    city VARCHAR(100) NOT NULL,
    name VARCHAR(150) NOT NULL
);

# Inserting dummy data into BGC_Facilities
INSERT INTO BGC_Facilities (address, city, name) VALUES
(’123 Main St’, ’New York’, ’Board Cafe A’),
(’456 Maple Ave’, ’Los Angeles’, ’Board Cafe B’),
(’789 Oak St’, ’Chicago’, ’Board Cafe C’);

# Creating Customers table
CREATE TABLE Customers (
    id_customer INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE
);

# Inserting dummy data into Customers
INSERT INTO Customers (name, email) VALUES
```

```
('Alice Johnson', 'alice@example.com'),
('Bob Smith', 'bob@example.com'),
('Charlie Brown', 'charlie@example.com');

# Creating Memberships table (Many-to-Many relationship between BGC_Facilities and Customers)
CREATE TABLE Memberships (
    id_board_cafe INT NOT NULL,
    id_customer INT NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE,
    PRIMARY KEY (id_board_cafe, id_customer),
    FOREIGN KEY (id_board_cafe) REFERENCES BGC_Facilities(id_board_cafe) ON DELETE CASCADE,
    FOREIGN KEY (id_customer) REFERENCES Customers(id_customer) ON DELETE CASCADE
);

# Inserting dummy data into Memberships with start and end dates
INSERT INTO Memberships (id_board_cafe, id_customer, start_date, end_date) VALUES
((SELECT id_board_cafe FROM BGC_Facilities WHERE name = 'Board Cafe A'),
 (SELECT id_customer FROM Customers WHERE name = 'Alice Johnson'), '2025-01-01', '2025-12-31'),
((SELECT id_board_cafe FROM BGC_Facilities WHERE name = 'Board Cafe B'),
 (SELECT id_customer FROM Customers WHERE name = 'Bob Smith'), '2025-02-01', '2025-11-30');

# Creating Registers table (Tracks customer registrations to cafes)
CREATE TABLE Registers (
    id_board_cafe INT NOT NULL,
    id_customer INT NOT NULL,
    PRIMARY KEY (id_board_cafe, id_customer),
    FOREIGN KEY (id_board_cafe) REFERENCES BGC_Facilities(id_board_cafe) ON DELETE CASCADE,
    FOREIGN KEY (id_customer) REFERENCES Customers(id_customer) ON DELETE CASCADE
);

# Inserting dummy data into Registers
INSERT INTO Registers (id_board_cafe, id_customer) VALUES
((SELECT id_board_cafe FROM BGC_Facilities WHERE name = 'Board Cafe A'),
 (SELECT id_customer FROM Customers WHERE name = 'Charlie Brown'));

# Creating Accesses table (Tracks customer access periods to cafes)
CREATE TABLE Accesses (
    access_id INT AUTO_INCREMENT PRIMARY KEY,
    id_board_cafe INT NOT NULL,
    id_customer INT NOT NULL,
    period_start DATE NOT NULL,
    period_end DATE NOT NULL,
    FOREIGN KEY (id_board_cafe) REFERENCES BGC_Facilities(id_board_cafe) ON DELETE CASCADE,
    FOREIGN KEY (id_customer) REFERENCES Customers(id_customer) ON DELETE CASCADE
);

# Inserting dummy data into Accesses
INSERT INTO Accesses (id_board_cafe, id_customer, period_start, period_end) VALUES
((SELECT id_board_cafe FROM BGC_Facilities WHERE name = 'Board Cafe C'),
 (SELECT id_customer FROM Customers WHERE name = 'Alice Johnson'), '2024-01-01', '2024-12-31');
```

4 Conclusion

Creating the ER diagrams and corresponding SQL scripts taught us valuable lessons in database design, such as how to model relationships between entities and apply normalization principles to avoid redundancy. We gained hands-on experience with SQL syntax, including defining primary and foreign keys, and learned the importance of translating conceptual models into practical implementations. This task enhanced our problem-solving skills, especially when troubleshooting SQL issues, and gave us a better understanding of how efficient database design contributes to application performance and reliability.