|  |  |
| --- | --- |
|  | **ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**  **ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ** |

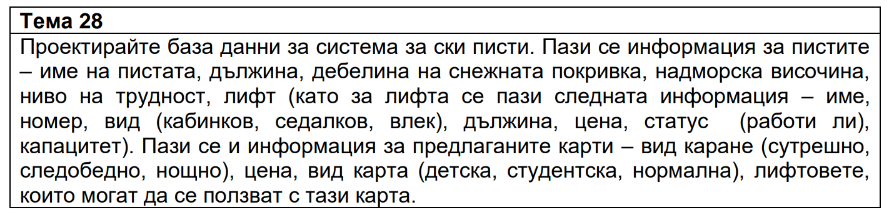
**Курсов Проект по Бази данни**

**Tема №28 Ski resort**

***Изготвил****: Ангел Вълков*

Фак. № 121222160

Група: 40б



1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки.

2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.

3. Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.

4. Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.

5. Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор.

6. Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.

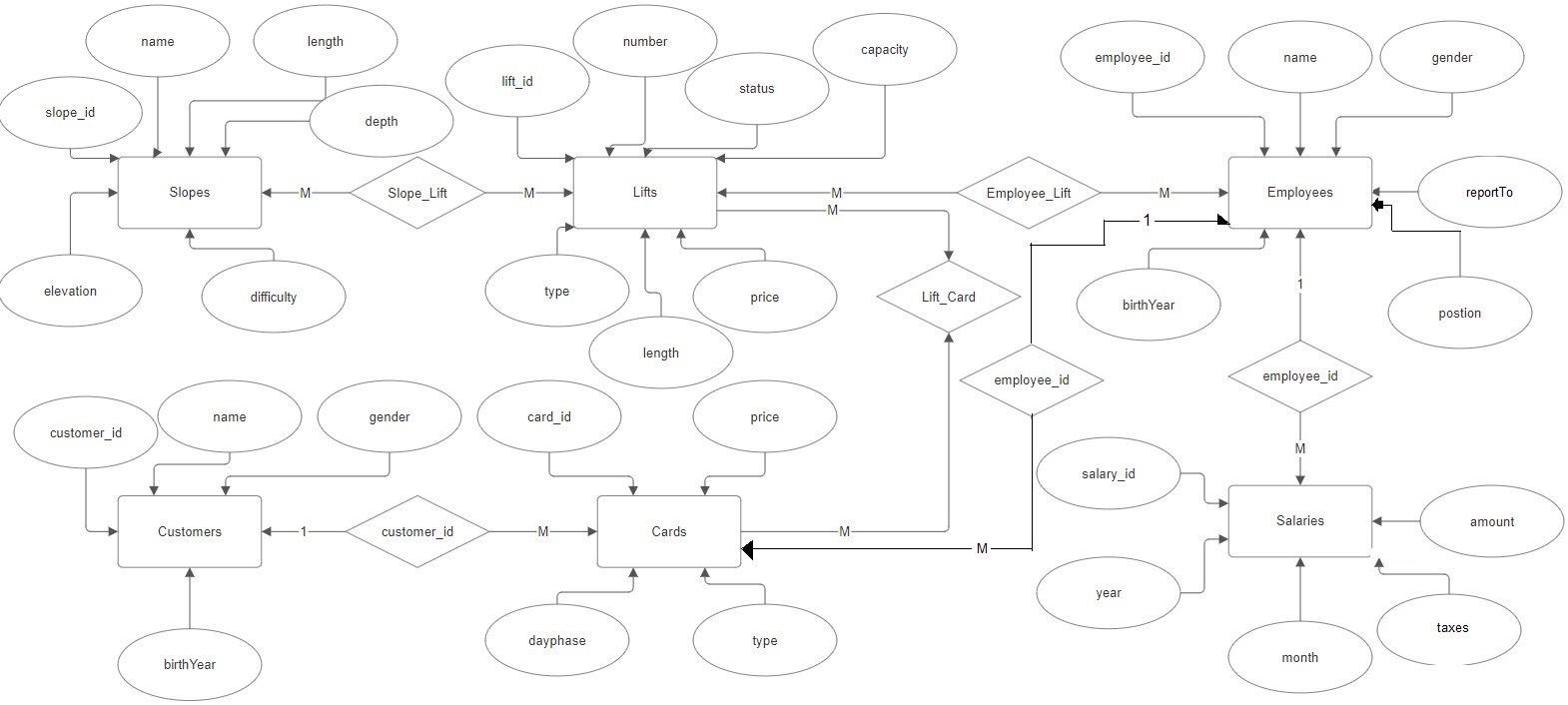
7. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.

8. Създайте тригер по ваш избор.

9. Създайте процедура, в която демонстрирате използване на курсор.

1. **Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL:**

Основните обекти, за които трябва да схраняваме информация, според заданието са: Cards, Lifts и Slopes. Допълнително ще създадем още няколко таблици. Първата ще отразява заплатите на всеки работник. В нея ще се съдържа информация за това кой месец и година, за кой работник и количеството В друга таблица ще съхраняваме работниците, които работят в курорта и на какви позиции са и съответно цялата информация за тях. И в таблицата клиенти ще ще имаме информация за всички клиенти, за тяхната възраст, години, пол и ще бъде вързани към таблиеца карти с връзка едно към много . В селдващата таблица ще имаме самите Карти, като те ще могат да бъдат собственост на един клиент или един работник. И ще пазят информация за това в каква фаза на деня могат да бъдат използвани, дата на създаване, дата на край, тип, цена и за кои лифтове важи, като връзката ще е много към много. В таблицата за лифтовете ще имаме всички параметри от заданието и ще са вързани с таблицата за писти с вързка мнго към много, а в таблицата писти също имаме всичко от условието за тях. Имаме две помощни таблици, които осъществяват връзката много към много между таблиците за карти, писти и и писти. За проектирането на базата ще използваме модела ER-диаграма (Entity Relationship Diagram):



**Create table заявките, които са необходими за да се създаде въпросната база са следните:**

create database SkiResort;

use SkiResort;

CREATE TABLE Customers (

customer\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

gender ENUM("Male", "Female", "Other") NOT NULL,

birth\_year YEAR NOT NULL

);

CREATE TABLE Employees (

employee\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

gender ENUM("Male", "Female", "Other") NOT NULL,

birth\_year YEAR NOT NULL,

position ENUM("Ski Instructor", "Lift Operator", "Receptionist", "Snowmaker", "Security", "Hygienist", "Manager", "Other") NOT NULL

);

CREATE TABLE Cards (

card\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

employee\_id INT,

type ENUM("Child", "Student", "Normal") NOT NULL,

price DECIMAL(5,2) NOT NULL,

dayphase ENUM("Morning", "Afternoon", "Evening", "UnLimit") NOT NULL,

start\_day DATE NOT NULL,

end\_day DATE NOT NULL,

CONSTRAINT FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

CONSTRAINT FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id),

CONSTRAINT only\_one\_card\_owner CHECK (

(customer\_id IS NULL AND employee\_id IS NOT NULL) OR

(employee\_id IS NULL AND customer\_id IS NOT NULL)

)

);

CREATE TABLE Salaries (

salary\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

amount DECIMAL(10,2) NOT NULL,

year YEAR NOT NULL,

month TINYINT NOT NULL CHECK (month >= 1 AND month <= 12),

employee\_id INT NOT NULL,

CONSTRAINT FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id),

UNIQUE(year, month, employee\_id)

);

CREATE TABLE Lifts (

lift\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

number INT NOT NULL,

status ENUM("Open", "Closed", "Ruined", "Under construction") NOT NULL,

length INT NOT NULL,

capacity INT NOT NULL,

type ENUM("Chairlift","Gondola","Drag") NOT NULL,

price DECIMAL(10,2) NOT NULL

);

CREATE TABLE Slopes (

slope\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

length INT NOT NULL,

depth INT NOT NULL,

difficulty ENUM("Easy", "Medium", "Hard", "Professional") NOT NULL,

elevation INT NOT NULL

);

CREATE TABLE Lift\_Cards (

customer\_id INT NOT NULL,

lift\_id INT NOT NULL,

CONSTRAINT FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

CONSTRAINT FOREIGN KEY (lift\_id) REFERENCES Lifts(lift\_id),

PRIMARY KEY(customer\_id, lift\_id)

);

CREATE TABLE Slope\_Lift (

slope\_id INT NOT NULL,

lift\_id INT NOT NULL,

CONSTRAINT FOREIGN KEY (slope\_id) REFERENCES Slopes(slope\_id),

CONSTRAINT FOREIGN KEY (lift\_id) REFERENCES Lifts(lift\_id),

PRIMARY KEY(slope\_id, lift\_id)

);

CREATE TABLE Employee\_Lift (

employee\_id INT NOT NULL,

lift\_id INT NOT NULL,

CONSTRAINT FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id),

CONSTRAINT FOREIGN KEY (lift\_id) REFERENCES Lifts(lift\_id),

PRIMARY KEY(employee\_id, lift\_id)

);

Alter Table Employees

ADD column reportTo INT after name;

Alter Table Employees

ADD CONSTRAINT FOREIGN KEY (reportTo) REFERENCES Employees(employee\_id);

**Добавяме тестови данни в таблиците:**

INSERT INTO Customers (name, gender, birth\_year) VALUES

('John Doe', 'Male', 1990),

('Jane Smith', 'Female', 1985),

('Alex Johnson', 'Other', 1995);

INSERT INTO Employees (name, gender, birth\_year, position) VALUES

('Michael Brown', 'Male', 1980, 'Ski Instructor'),

('Emily Davis', 'Female', 1992, 'Lift Operator'),

('Chris Lee', 'Other', 1988, 'Receptionist');

INSERT INTO Cards (customer\_id, type, price, dayphase, start\_day, end\_day) VALUES

(1, 'Child', 25.00, 'Morning', '2024-01-01', '2024-01-01'),

(2, 'Student', 35.00, 'Afternoon', '2024-01-02', '2024-01-02'),

(3, 'Normal', 45.00, 'UnLimit', '2024-01-03', '2024-01-05');

INSERT INTO Cards (employee\_id, type, price, dayphase, start\_day, end\_day) VALUES

(3, 'Normal', 35.00, 'UnLimit', '2024-01-03', '2024-01-05');

INSERT INTO Salaries (amount, year, month, employee\_id) VALUES

(3000.00, 2024, 1, 1),

(2500.00, 2024, 1, 2),

(2800.00, 2024, 1, 3);

INSERT INTO Lifts (name, number, status, length, capacity, type, price) VALUES

('Mountain Lift', 1, 'Open', 1000, 4, 'Chairlift', 10.00),

('Valley Lift', 2, 'Closed', 800, 6, 'Gondola', 15.00),

('Slope Lift', 3, 'Open', 1200, 2, 'Drag', 5.00);

INSERT INTO Slopes (name, length, depth, difficulty, elevation) VALUES

('Beginner Slope', 1500, 50, 'Easy', 200),

('Intermediate Slope', 2000, 80, 'Medium', 400),

('Advanced Slope', 2500, 100, 'Hard', 600);

INSERT INTO Lift\_Cards (customer\_id, lift\_id) VALUES

(1, 1),

(2, 2),

(3, 3);

INSERT INTO Slope\_Lift (slope\_id, lift\_id) VALUES

(1, 1),

(2, 2),

(3, 3);

INSERT INTO Employee\_Lift (employee\_id, lift\_id) VALUES

(1, 1),

(2, 2),

(3, 3);

INSERT INTO Customers (name, gender, birth\_year) VALUES

('Sarah Johnson', 'Female', 1993),

('David Wilson', 'Male', 1987),

('Taylor Brown', 'Other', 2000),

('Emma White', 'Female', 1998),

('Ryan Martinez', 'Male', 1995);

INSERT INTO Employees (name, gender, birth\_year, position) VALUES

('Olivia Moore', 'Female', 1990, 'Snowmaker'),

('Noah Taylor', 'Male', 1985, 'Security'),

('Ava Anderson', 'Female', 1997, 'Hygienist'),

('Ethan Clark', 'Male', 1989, 'Manager');

INSERT INTO Cards (customer\_id, type, price, dayphase, start\_day, end\_day) VALUES

(4, 'Student', 30.00, 'Afternoon', '2024-01-03', '2024-01-03'),

(5, 'Normal', 40.00, 'UnLimit', '2024-01-02', '2024-01-04'),

(6, 'Child', 20.00, 'Morning', '2024-01-01', '2024-01-01'),

(7, 'Normal', 45.00, 'UnLimit', '2024-01-02', '2024-01-05'),

(8, 'Student', 35.00, 'Afternoon', '2024-01-03', '2024-01-03');

INSERT INTO Salaries (amount, year, month, employee\_id) VALUES

(2800.00, 2023, 2, 4),

(2700.00, 2023, 3, 5),

(2600.00, 2024, 1, 6),

(2900.00, 2024, 2, 6),

(3000.00, 2024, 3, 2);

INSERT INTO Lifts (name, number, status, length, capacity, type, price) VALUES

('Peak Lift', 4, 'Open', 1500, 4, 'Chairlift', 12.00),

('Summit Lift', 5, 'Closed', 1000, 6, 'Gondola', 18.00),

('Valley Lift 2', 6, 'Open', 1200, 3, 'Drag', 6.00);

INSERT INTO Slopes (name, length, depth, difficulty, elevation) VALUES

('Expert Slope', 3000, 120, 'Professional', 800),

('Beginner Slope 2', 1800, 60, 'Easy', 250),

('Advanced Slope 2', 2200, 90, 'Hard', 550);

INSERT INTO Lift\_Cards (customer\_id, lift\_id) VALUES

(4, 4),

(5, 5),

(6, 6),

(7, 4),

(8, 6);

INSERT INTO Slope\_Lift (slope\_id, lift\_id) VALUES

(4, 4),

(5, 5),

(6, 6);

INSERT INTO Employee\_Lift (employee\_id, lift\_id) VALUES

(4, 4),

(5, 5),

(6, 6),

(3, 4),

(5, 3);

-- bad insert Card is owned by employee and customer

INSERT INTO Cards (customer\_id, employee\_id, type, price, dayphase, start\_day, end\_day) VALUES

(4, 3, 'Student', 30.00, 'Afternoon', '2024-01-03', '2024-01-03');

INSERT INTO Customers (name, gender, birth\_year) VALUES

('Sophia Williams', 'Female', 1991),

('Matthew Taylor', 'Male', 1986),

('Chloe Brown', 'Female', 1994),

('Daniel Anderson', 'Male', 1997),

('Grace Wilson', 'Female', 1990);

INSERT INTO Employees (name, gender, birth\_year, position) VALUES

('Liam Jones', 'Male', 1983, 'Snowmaker'),

('Charlotte Martinez', 'Female', 1995, 'Security'),

('Mason Miller', 'Male', 1989, 'Hygienist'),

('Amelia Thompson', 'Female', 1987, 'Manager');

INSERT INTO Cards (customer\_id, type, price, dayphase, start\_day, end\_day) VALUES

(9, 'Normal', 40.00, 'UnLimit', '2024-01-03', '2024-01-05'),

(10, 'Student', 35.00, 'Afternoon', '2024-01-04', '2024-01-04'),

(11, 'Child', 20.00, 'Morning', '2024-01-05', '2024-01-05'),

(12, 'Normal', 45.00, 'UnLimit', '2024-01-06', '2024-01-08'),

(13, 'Student', 30.00, 'Afternoon', '2024-01-07', '2024-01-07');

INSERT INTO Cards (employee\_id, type, price, dayphase, start\_day, end\_day) VALUES

(8, 'Normal', 35.00, 'UnLimit', '2024-01-03', '2024-01-05'),

(9, 'Normal', 45.00, 'UnLimit', '2024-01-04', '2024-01-06'),

(10, 'Normal', 40.00, 'UnLimit', '2024-01-05', '2024-01-07');

INSERT INTO Salaries (amount, year, month, employee\_id) VALUES

(2800.00, 2024, 2, 8),

(2900.00, 2024, 2, 9),

(3000.00, 2024, 2, 10),

(3100.00, 2024, 2, 3),

(3200.00, 2024, 2, 4);

INSERT INTO Lifts (name, number, status, length, capacity, type, price) VALUES

('Ridge Lift', 7, 'Open', 1300, 5, 'Chairlift', 14.00),

('Sunset Lift', 8, 'Closed', 900, 6, 'Gondola', 20.00),

('Meadow Lift', 9, 'Open', 1100, 4, 'Drag', 8.00);

INSERT INTO Slopes (name, length, depth, difficulty, elevation) VALUES

('Pro Slope', 2800, 100, 'Professional', 900),

('Intermediate Slope 2', 1900, 70, 'Medium', 350),

('Hardcore Slope', 2400, 110, 'Hard', 600);

INSERT INTO Lift\_Cards (customer\_id, lift\_id) VALUES

(9, 7),

(10, 8),

(11, 9),

(12, 7),

(13, 9);

INSERT INTO Slope\_Lift (slope\_id, lift\_id) VALUES

(7, 7),

(8, 8),

(9, 9);

INSERT INTO Employee\_Lift (employee\_id, lift\_id) VALUES

(8, 7),

(9, 8),

(10, 9),

(4, 8),

(3, 7);

INSERT INTO Customers (name, gender, birth\_year) VALUES

('Lily Johnson', 'Female', 1992),

('Benjamin Garcia', 'Male', 1988),

('Avery Davis', 'Female', 1996),

('Samuel Rodriguez', 'Male', 1993),

('Harper Martinez', 'Female', 1989);

INSERT INTO Cards (customer\_id, type, price, dayphase, start\_day, end\_day) VALUES

(14, 'Normal', 40.00, 'UnLimit', '2024-01-03', '2024-01-05'),

(15, 'Student', 35.00, 'Afternoon', '2024-01-04', '2024-01-04'),

(16, 'Child', 20.00, 'Morning', '2024-01-05', '2024-01-05'),

(17, 'Normal', 45.00, 'UnLimit', '2024-01-06', '2024-01-08'),

(18, 'Student', 30.00, 'Afternoon', '2024-01-07', '2024-01-07');

INSERT INTO Customers (name, gender, birth\_year) VALUES

('Evelyn Lopez', 'Female', 1990),

('William Lee', 'Male', 1985),

('Ella Hill', 'Female', 1998),

('Alexander Clark', 'Male', 1991),

('Scarlett Lewis', 'Female', 1994);

INSERT INTO Cards (customer\_id, type, price, dayphase, start\_day, end\_day) VALUES

(19, 'Normal', 40.00, 'UnLimit', '2024-01-03', '2024-01-05'),

(20, 'Student', 35.00, 'Afternoon', '2024-01-04', '2024-01-04'),

(21, 'Child', 20.00, 'Morning', '2024-01-05', '2024-01-05'),

(22, 'Normal', 45.00, 'UnLimit', '2024-01-06', '2024-01-08'),

(23, 'Student', 30.00, 'Afternoon', '2024-01-07', '2024-01-07');

INSERT INTO Employees (name, gender, birth\_year, position) VALUES

('Petar Ivanov', 'Male', 2003, 'Other'),

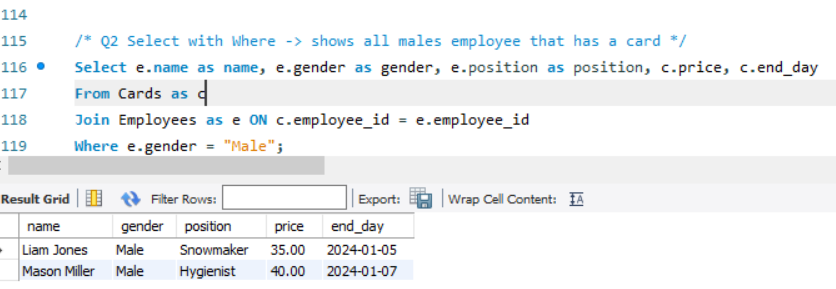
('Angel Valkov', 'Male', 1985, 'Other');

INSERT INTO Employee\_Lift (employee\_id, lift\_id) VALUES

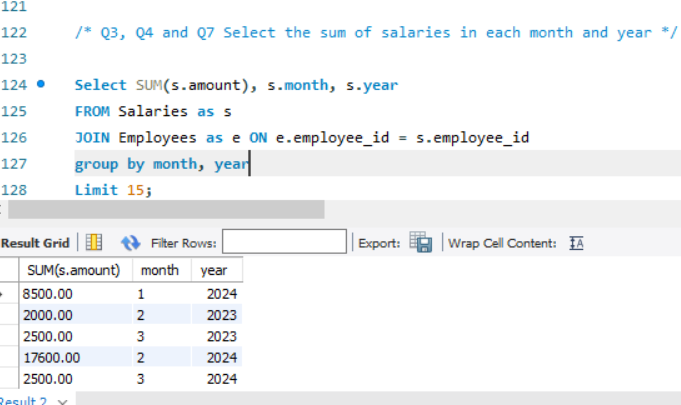
(12, 7),

(13, 8);

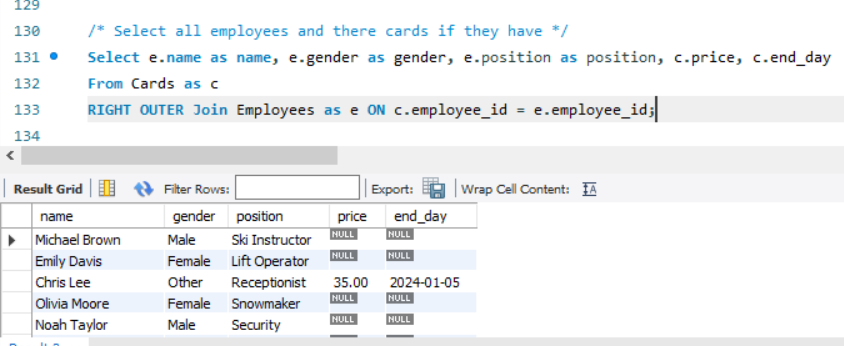
**2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор:**

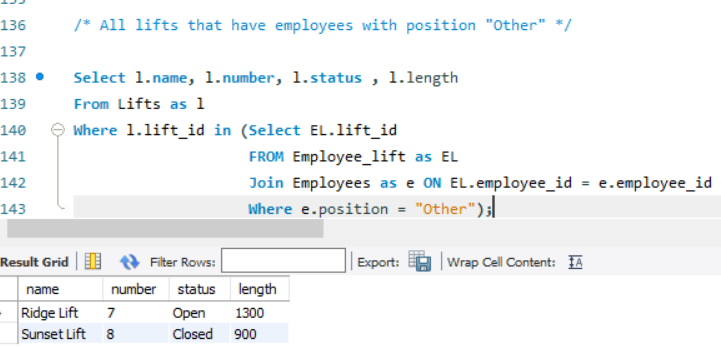


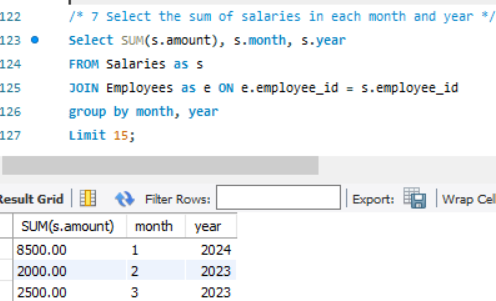
**3.Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.**

**4. Напишете заявка, в която демонстрирате INNER JOIN по ваш избор.** 

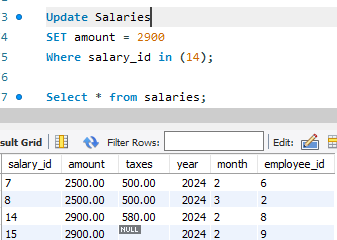
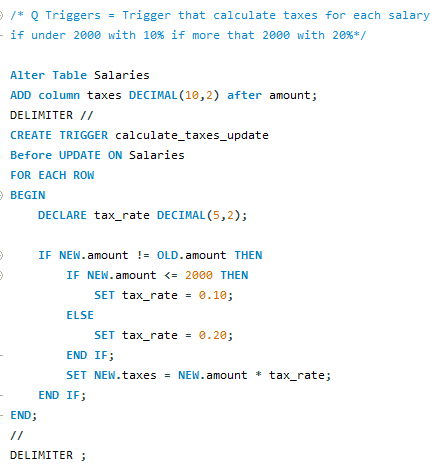
**5. Напишете заявка, в която демонстрирате OUTER JOIN по ваш избор**



**6.Напишете заявка, в която демонстрирате вложен SELECT по ваш избор.****7. 7.**

**7.Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функ**

**8. Създайте тригер по ваш избор.**



**9.Създайте процедура, в която демонстрирате използване на курсор**.