Task 1: School Database

1. \*\*Define Tables and Columns\*\*:

- \*\*Students Table\*\*: Include columns such as Student ID, First Name, Last Name, Birth Date, and Enrollment Year.

- \*\*Professors Table\*\*: Include columns such as Professor ID, First Name, Last Name, Department, and Hire Date.

- \*\*Subjects Table\*\*: Include columns like Subject ID, Subject Name, and Professor ID (indicating which professor teaches the subject).

2. \*\*Set Constraints\*\*:

- Define primary keys for each table.

- Use appropriate data types for each column.

- Add any necessary constraints to maintain data integrity, such as \*\*NOT NULL\*\* constraints where needed.

**-----Students**

**CREATE TABLE Students(**

**Student\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50) NOT NULL,**

**last\_name VARCHAR(50) NOT NULL,**

**data\_na\_raganje DATE NOT NULL UNIQUE ,**

**godina\_na\_upis CHAR(4),**

**);**

**-----Professors Table**

**CREATE TABLE Nastavnik(**

**Professor\_ID INT PRIMARY KEY,**

**first\_name\_p VARCHAR(30) NOT NULL,**

**last\_name\_p VARCHAR(50) NOT NULL,**

**department VARCHAR(50) NOT NULL,**

**godina\_na\_vrabotuvanje CHAR(4),**

**);**

**----Subjects Table**

**CREATE TABLE Subject(**

**Subject\_id INT PRIMARY KEY,**

**subject\_name VARCHAR(50) NOT NULL,**

**Professor\_ID INT,**

**);**

3. \*\*Relationships\*\*:

- Establish relationships between the tables, such as associating each subject with a professor through the \*\*Professor ID\*\* column.

**FOREIGN KEY(Professor\_ID) REFERENCES Nastavnik(Professor\_ID);**

4. \*\*Filtering Queries\*\*:

- Write the following SQL queries to retrieve specific data from the School Database:

- \*\*Query 1\*\*: Retrieve a list of all students enrolled after a specific year (e.g., 2020).

SELECT \* from Students WHERE godina\_na\_upis >1998;

- \*\*Query 2\*\*: Find the subjects taught by a particular professor by using the professor's ID.

SELECT \*FROM Subject AS s

LEFT JOIN Nastavnik AS n

ON s.Professor\_ID = n.Professor\_ID;

- \*\*Query 3\*\*: List all students' names and enrollment years, filtered by department (for example, list all students under a department where professors belong).

**#### Task 2: Bank Database**

1. \*\*Database Overview\*\*:

- Think through and outline the tables that the bank database should contain.

- Each table should focus on storing essential information about clients and their banking activities, including:

- \*\*Clients\*\*: General client details.

- \*\*Accounts\*\*: Basic account information.

- \*\*Savings\*\*: Details specific to savings accounts, if applicable.

- \*\*Debit Cards\*\*: Information regarding each client’s debit card(s).

- \*\*Credits\*\*: Details on any loans or credit accounts.

2. \*\*Design Decision\*\*:

- Define each table's purpose and its key columns. Decide on the relationships between tables (e.g., a client may have multiple accounts or debit cards).

- Create a clear structure that ensures data integrity and ease of access.

CREATE TABLE Clients(

Client\_id INT PRIMARY KEY,

ime VARCHAR(30),

prezime VARCHAR(30),

telefonski\_broj VARCHAR(30),

email VARCHAR (50),

godini VARCHAR(40),

adresa\_na\_ziveenje VARCHAR(50)

);

CREATE TABLE Accounts(

Client\_id INT,

accounts\_broj INT PRIMARY KEY,

accounts\_balance DECIMAL(10,2),

saving\_account\_period VARCHAR(50),

saving\_account\_balance DECIMAL(10,2),

FOREIGN KEY(Client\_id) REFERENCES Clients(Client\_id)

);

CREATE TABLE produkti(

accounts\_broj INT,

debit\_card VARCHAR (30),

debit\_card\_issed DATE,

credit\_broj INT,

credit\_issued DATE,

balance\_credit DECIMAL (15,2),

FOREIGN KEY(accounts\_broj) REFERENCES Accounts(accounts\_broj)

);

INSERT INTO Clients (Client\_id,ime, prezime, telefonski\_broj, email, godini, adresa\_na\_ziveenje)

VALUES

(1,'Andrej','Nikolovski', '071234567', 'andrej.nikolovski@gmail.com', 25,'Orce\_Nikolov'),

(2, 'Ana', 'Petrovska', '072356789','ana.petrovska@email.com', 25, 'Pero\_nakov'),

(3, 'Marko', 'Stojanov', '078123456','marko.stojanov@email.com', 30, 'Invazija'),

(4, 'Elena', 'Kostovska', '070987654', 'elena.kostovska@email.com', 22,'Franjo\_Kluz'),

(5, 'Igor', 'Dimitrov', '072345678', 'igor.dimitrov@email.com', 28, NULL),

(6, 'Viktor', 'Stefanov', '076543210', 'viktor.stefanov@email.com', 40, NULL);

INSERT INTO Accounts (Client\_id, accounts\_broj,accounts\_balance, saving\_account\_period, saving\_account\_balance )

VALUES

(1, 2, 1500000000,'dvanaeset\_meseci\_period',1000000),

(1, 1, 170000000,'dvaesetipet\_meseci\_period',250000000),

(3, 3, 2400000000,'trinaeset\_meseci\_period', 1000000000),

(4, 4, 1580000000, 'dvesetipet\_meseci\_period', 3500000),

(4, 6, 170000000,'trinaeset\_meseci\_period', 240000000),

(6, 5, 160000000,'osum\_meseci\_period', 29000000000);

INSERT INTO produkti( accounts\_broj, debit\_card, debit\_card\_issed,credit\_broj,credit\_issued,balance\_credit)

VALUES

(1,'edna\_debitna', '2024-08-02', 5, '2023-08-01', 15000000),

(2,'dve\_debitni', '2023-10-05', 4, '1999-01-02', 800000),

(3,'tri\_debitni', '2021-05-02', 2, '1998-09-01', 700000),

(NULL,'edna\_debitna', '2024-06-03', 1, '2000-08-05', 60000),

(4,'dve\_debitni', '2021-08-09', 2, '2018-10-06', 100000000),

(5, 'tri\_debitni', '2023-07-04', 8,' 2005-06-20',180000000),

(6,'dve\_debitni','2000-06-20', 1, '2023-06-05',28000000);

3. \*\*Filtering Queries\*\*:

- Write the following SQL queries to retrieve specific data from the Bank Database:

- \*\*Query 1\*\*: List all clients who have more than a specific amount in their savings account (e.g., $10,000).

SELECT \* FROM Clients AS C

LEFT JOIN Accounts AS A

ON C.Client\_id=A.Client\_id

WHERE saving\_account\_balance>10000000;

- \*\*Query 2\*\*: Retrieve all debit cards issued within a specific date range.

SELECT \* FROM Accounts AS A

LEFT JOIN produkti AS P

ON A.accounts\_broj=P.accounts\_broj

WHERE debit\_card\_issed >= '2021-10-01';

- \*\*Query 3\*\*: List all clients who have taken credit, along with the credit amount and start date, filtering for credits issued after a specific date (e.g., after 2023).

SELECT\*FROM Clients AS C

LEFT JOIN Accounts AS A

ON C.Client\_id= A.Client\_id

RIGHT JOIN produkti AS P

ON A.accounts\_broj=P.accounts\_broj

WHERE credit\_issued >'2000-08-01';