



Курсова работа по База данни

Тема №21 Разработете база данни за система за каталог и продажба на лекарства. В каталога се показва вида на продуктите, болести, за които се прилагат, стандартни дозировки, указания за употреба, цена и т.н. Пази се профил на купувачите (клиентите), както и история на направените от тях покупки. Също така се пази информация за фармацевта, който е продал лекарството, данни за рецептата и т.н.

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

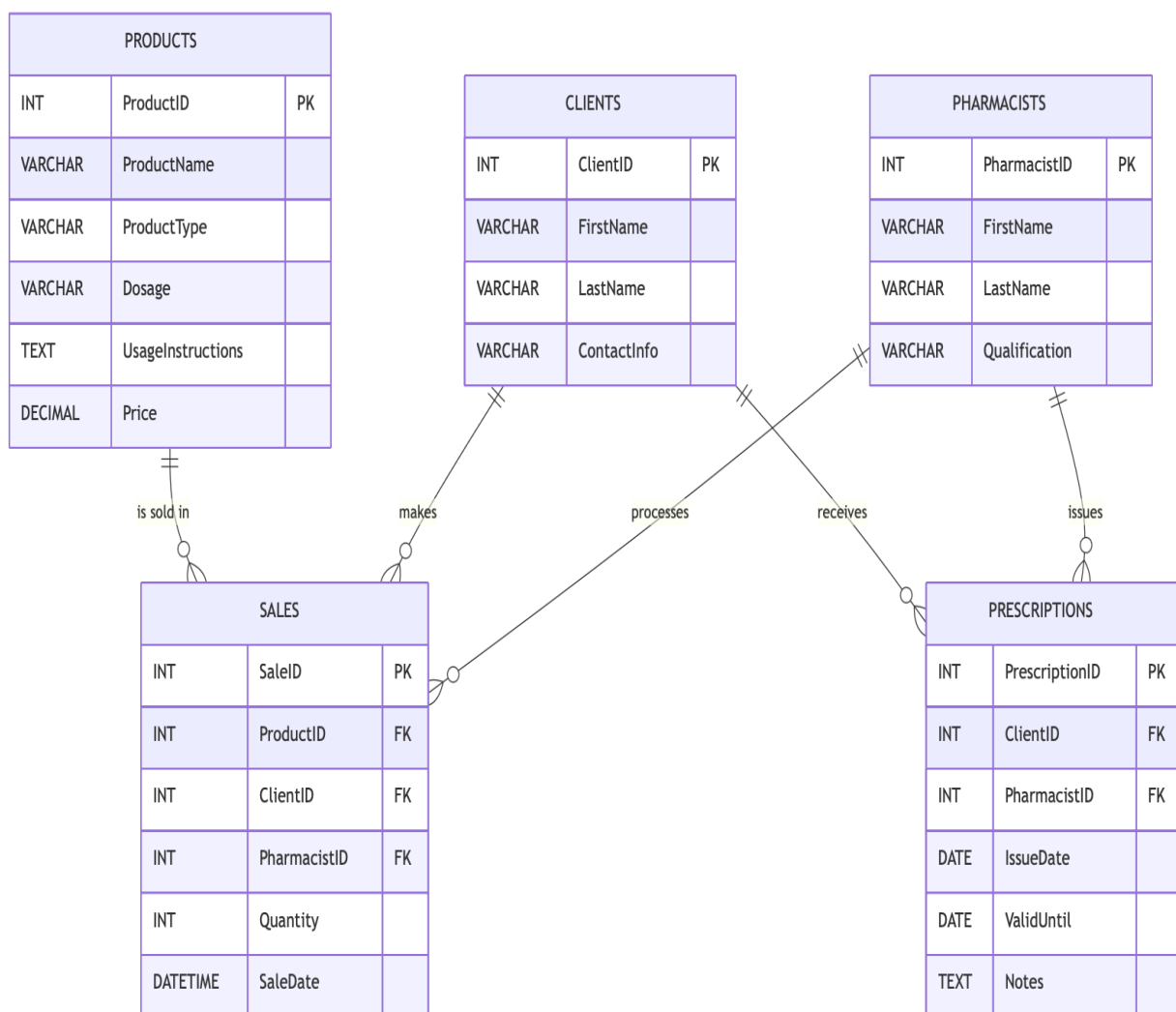
Вашата работа трябва да включва: задание, ER-диаграма, CREATE TABLE заявки, всички останали заявки, решения на задачите от 2 до 9 и резултатите от тях.

студент: Джан Зекериев
фак. номер: 381223016
група94б.....

преподавател:
подпис:

дата: 08.05.2025
гр. София

1.ER диаграма



LOGS		
INT	LogID	PK
VARCHAR	Description	
DATETIME	ChangeDate	

2. Създаване на таблици (CREATE TABLE)

-- 1. Таблица Products: справочник на лекарствата

```
CREATE TABLE Products (
  ProductID INT AUTO_INCREMENT PRIMARY KEY,
  ProductName VARCHAR(100) NOT NULL,
  ProductType VARCHAR(50),
  Dosage VARCHAR(50),
  UsageInstructions TEXT,
  Price DECIMAL(10,2) NOT NULL
);
```

-- 2. Таблица Clients: данни за клиентите

```
CREATE TABLE Clients (
  ClientID INT AUTO_INCREMENT PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  ContactInfo VARCHAR(100)
);
```

-- 3. Таблица Pharmacists: фармацевтите

```
CREATE TABLE Pharmacists (
  PharmacistID INT AUTO_INCREMENT PRIMARY KEY,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  Qualification VARCHAR(100)
);
```

-- 4. Таблица Sales: запис на всяка продажба

```
CREATE TABLE Sales (
```

```

SaleID INT AUTO_INCREMENT PRIMARY KEY,
ProductID INT NOT NULL,
ClientID INT NOT NULL,
PharmacistID INT NOT NULL,
Quantity INT DEFAULT 1,
SaleDate DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
FOREIGN KEY (PharmacistID) REFERENCES Pharmacists(PharmacistID)
);

```

-- 5. Таблица Prescriptions: издадени рецепти

```

CREATE TABLE Prescriptions (
  PrescriptionID INT AUTO_INCREMENT PRIMARY KEY,
  ClientID INT NOT NULL,
  PharmacistID INT NOT NULL,
  IssueDate DATE NOT NULL,
  ValidUntil DATE,
  Notes TEXT,
  FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
  FOREIGN KEY (PharmacistID) REFERENCES Pharmacists(PharmacistID)
);

```

-- 6. Таблица Logs: аудит лог на операциите (примерно за тригер)

```

CREATE TABLE Logs (
  LogID INT AUTO_INCREMENT PRIMARY KEY,
  Description VARCHAR(255),
  ChangeDate DATETIME DEFAULT CURRENT_TIMESTAMP
);

```

Бележка: NOT NULL гарантира валидност на данните, а DEFAULT стойности улесняват вмъкването и гарантират автоматично попълване (напр. текуща дата).

Примерни данни (INSERT)

1. SELECT с ограничение (WHERE)

Избират всички продукти с цена над 12 лв.:

```

1  SELECT ProductName, Price
2  FROM Products
3  WHERE Price > 12.00;
4

```

00% 1:4

Result Grid Filter Rows: Search Export:

ProductName	Price
Amoxicillin	15.00
Amoxicillin	15.00

2. Агрегатна функция + GROUP BY

Брой продукти по тип:

```

1  SELECT ProductType, COUNT(*) AS CountByType
2  FROM Products
3  GROUP BY ProductType;

```

00% 22:3

Result Grid Filter Rows: Search Export:

ProductType	CountByType
Analgesic	2
Antibiotic	2
Painkiller	3

3. INNER JOIN

Списък на продажбите с имена на продукти и клиенти:

```

SELECT S.SaleID, P.ProductName, C.FirstName, C.LastName, S.Quantity, S.SaleDate
FROM Sales S
INNER JOIN Products P ON S.ProductID = P.ProductID
INNER JOIN Clients C ON S.ClientID = C.ClientID;

```

51:4

Grid

Filter Rows:

Search

Export:

eID	ProductName	FirstName	LastName	Quantity	SaleDate	
	Aspirin	Ivan	Petrov	2	2025-05-01 10:15:00	
	Amoxicillin	Maria	Kostova	1	2025-05-02 14:30:00	
	Paracetamol	Georgi	Dimitrov	4	2025-05-03 09:20:00	
	Aspirin	Ivan	Petrov	2	2025-05-01 10:15:00	
	Amoxicillin	Maria	Kostova	1	2025-05-02 14:30:00	
	Paracetamol	Georgi	Dimitrov	4	2025-05-03 09:20:00	
	Aspirin	Ivan	Petrov	2	2025-05-08 10:03:45	

4. LEFT OUTER JOIN

Всички клиенти и техните (възможни) продажби:

```

1 • SELECT C.FirstName, C.LastName, S.SaleID, S.SaleDate
2 FROM Clients C
3 LEFT JOIN Sales S ON C.ClientID = S.ClientID;

```

100%

46:3

Result Grid

Filter Rows:

Export:

	FirstName	LastName	SaleID	SaleDate	
	Ivan	Petrov	1	2025-05-01 10:15:00	
	Maria	Kostova	2	2025-05-02 14:30:00	
	Georgi	Dimitrov	3	2025-05-03 09:20:00	
	Ivan	Petrov	4	2025-05-01 10:15:00	
	Maria	Kostova	5	2025-05-02 14:30:00	
	Georgi	Dimitrov	6	2025-05-03 09:20:00	
	Ivan	Petrov	7	2025-05-08 10:03:45	




5. Вложен SELECT

Най-скъпият продукт:

```

1 • SELECT ProductName, Price
2 FROM Products
3 WHERE Price = (SELECT MAX(Price) FROM Products);

```

100%	↕	49:3	
Result Grid   Filter Rows: <input type="text" value="Search"/> Export: 			
ProductName	Price		
Amoxicillin	15.00		




6. Сложен SELECT с JOIN + агрегат

Обща сума, похарчена от всеки клиент:

```

3 FROM Sales S
4 JOIN Clients C ON S.ClientID = C.ClientID
5 JOIN Products P ON S.ProductID = P.ProductID
6 GROUP BY C.ClientID;

```

100%	↕	21:6	
Result Grid   Filter Rows: <input type="text" value="Search"/> Export: 			
FirstName	LastName	TotalSpent	
Ivan	Petrov	63.00	
Maria	Kostova	30.00	
Georgi	Dimitrov	70.00	

6. Тригер (TRIGGER)

-- Създаваме/гарантираме съществуването на Logs

```
) CREATE TABLE IF NOT EXISTS Logs (  
    LogID INT AUTO_INCREMENT PRIMARY KEY,  
    Description VARCHAR(255),  
    ChangeDate DATETIME DEFAULT CURRENT_TIMESTAMP  
) ;
```

-- Превключваме DELIMITER, за да ползваме вътрешни запетаи

DELIMITER //

CREATE TRIGGER after_sale_insert

AFTER INSERT ON Sales

FOR EACH ROW

```
) BEGIN  
    INSERT INTO Logs (Description, ChangeDate)  
    VALUES (  
        CONCAT(  
            'New sale: SaleID=', NEW.SaleID,  
            ', ProductID=', NEW.ProductID,  
            ', ClientID=', NEW.ClientID,  
            ', Qty=', NEW.Quantity  
        ),  
        NOW()  
    );  
END;
```

//

DELIMITER ;

7. Съхранена процедура (STORED PROCEDURE)

```
DELIMITER //
```


```
CREATE PROCEDURE GetClientPurchaseSummary (IN p_ClientID INT)
) BEGIN
    -- Връща името на клиента и общата сума, изхарчена от него
    SELECT C.FirstName, C.LastName,
           SUM(S.Quantity * P.Price) AS TotalSpent
    FROM Sales S
    JOIN Clients C ON S.ClientID = C.ClientID
    JOIN Products P ON S.ProductID = P.ProductID
    WHERE C.ClientID = p_ClientID
    GROUP BY C.ClientID;
END;
//
```

```
DELIMITER ;
```

Извикване:

1	CALL GetClientPurchaseSummary(1);
---	-----------------------------------

34:1

Result Grid		Filter Rows:	<input type="text" value="Search"/>	Export:
-------------	---	--------------	-------------------------------------	---------

FirstName	LastName	TotalSpent
Ivan	Petrov	63.00

8. Демонстрация на курсор (CURSOR)

```
DELIMITER //
```

```
CREATE PROCEDURE ListProducts()
```

```
BEGIN
```

```
    DECLARE done    BOOLEAN DEFAULT FALSE;
```

```
    DECLARE p_name  VARCHAR(100);
```

```
    DECLARE p_price DECIMAL(10,2);
```

```
    -- Декларираме курсор за избора на име и цена
```

```
    DECLARE product_cursor CURSOR FOR
```

```
        SELECT ProductName, Price FROM Products;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

```
    OPEN product_cursor;
```

```
read_loop: LOOP
```

```
    FETCH product_cursor INTO p_name, p_price;
```

```
    IF done THEN
```

```
        LEAVE read_loop;
```

```
    END IF;
```

```
    -- Показва текущия запис
```

```
    SELECT p_name AS Product, p_price AS Price;
```

```
END LOOP;
```

```
    CLOSE product_cursor;
```

```
END;
```

```
//
```

```
DELIMITER ;
```