Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТP

Кафедра ПИН

*ЛАБОРАТОРНАЯ*

*РАБОТА №7*

По Организация баз данных

Тема Манипулирование данными

Руководитель

Быков А. А.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2023

**Тема: Манипулирование данными**

**Задание на лабораторную работу:**

1. Создайте таблицу EXAM\_MARKS так, чтобы не допускался ввод в таблицу двух записей об оценках одного студента по конкретным экзамену и предмету обучения, и чтобы не допускалось проведение двух экзаменов по любым предметам в один день.

CREATE TABLE EXAM\_MARKS (

student\_id INT NOT NULL,

subject\_id INT NOT NULL,

exam\_date DATE NOT NULL,

exam\_mark INT NOT NULL,

CONSTRAINT PK\_EXAM\_MARKS PRIMARY KEY (student\_id, subject\_id, exam\_date),

CONSTRAINT FK\_EXAM\_MARKS\_STUDENT FOREIGN KEY (student\_id) REFERENCES STUDENTS (student\_id),

CONSTRAINT FK\_EXAM\_MARKS\_SUBJECT FOREIGN KEY (subject\_id) REFERENCES SUBJECTS (subject\_id),

CONSTRAINT CK\_EXAM\_MARKS\_EXAM\_DATE UNIQUE (subject\_id, exam\_date)

);

1. Создайте таблицу предметов обучения SUBJECT так, чтобы количество отводимых на предмет часов по умолчанию было равно 36, не допускались записи с отсутствующим количеством часов, поле SUBJ\_ID являлось первичным ключом таблицы и значения семестров (поле SEMESTER) лежали в диапазоне от 1 до 12.

CREATE TABLE SUBJECT (

SUBJ\_ID INT PRIMARY KEY,

SUBJ\_NAME VARCHAR(50) NOT NULL,

HOURS INT NOT NULL DEFAULT 36,

SEMESTER INT NOT NULL CHECK (SEMESTER >= 1 AND SEMESTER <= 12)

);

1. Создайте таблицу EXAM\_MARKS таким образом, чтобы значения поля EXAM\_\_ID были больше значений поля SUBJ\_ID, а значения поля SUBJ\_ID были больше значений поля STUDENT\_\_ID; пусть также будут запрещены значения NULL в любом из этих трех полей

CREATE TABLE EXAM\_MARKS (

EXAM\_ID INT NOT NULL,

SUBJ\_ID INT NOT NULL,

STUDENT\_ID INT NOT NULL,

EXAM\_MARK INT,

CONSTRAINT PK\_EXAM\_MARKS PRIMARY KEY (EXAM\_ID),

CONSTRAINT FK\_EXAM\_MARKS\_EXAM FOREIGN KEY (EXAM\_ID) REFERENCES EXAMS (EXAM\_ID),

CONSTRAINT FK\_EXAM\_MARKS\_SUBJ FOREIGN KEY (SUBJ\_ID) REFERENCES SUBJECTS (SUBJ\_ID),

CONSTRAINT FK\_EXAM\_MARKS\_STUDENT FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENTS (STUDENT\_ID),

CONSTRAINT CK\_EXAM\_MARKS\_EXAM\_ID CHECK (EXAM\_ID > SUBJ\_ID),

CONSTRAINT CK\_EXAM\_MARKS\_SUBJ\_ID CHECK (SUBJ\_ID > STUDENT\_ID),

CONSTRAINT CK\_EXAM\_MARKS\_NOT\_NULL CHECK (EXAM\_ID IS NOT NULL AND SUBJ\_ID IS NOT NULL AND STUDENT\_ID IS NOT NULL)

);

1. Напишите запрос для создания таблицы EXAM\_MARKS, чтобы не допускался ввод в таблицу двух записей об оценках одного студента по конкретным экзамену и предмету обучения.

CREATE TABLE EXAM\_MARKS (

STUDENT\_ID INT,

EXAM\_ID INT,

SUBJ\_ID INT,

MARK INT,

CONSTRAINT exam\_marks PRIMARY KEY (STUDENT\_ID,EXAM\_ID,

SUBJ\_ID),

CONSTRAINT exam\_marks UNIQUE (STUDENT\_ID,EXAM\_ID, SUBJ\_ID)

);

1. Напишите запрос для создания таблицы UNIVERSITY, чтобы не допускалось добавление нескольких университетов с одинаковым рейтингом и с пустым полем «город».

CREATE TABLE UNIVERSITY (

UNIV\_ID INT PRIMARY KEY,

UNIV\_NAME VARCHAR(50) NOT NULL,

CITY VARCHAR(50),

RATING INT NOT NULL,

CONSTRAINT CK\_UNIVERSITY\_CITY\_NOT\_NULL CHECK (CITY IS NOT NULL),

CONSTRAINT CK\_UNIVERSITY\_RATING\_UNIQUE CHECK (NOT EXISTS (

SELECT 1 FROM UNIVERSITY

WHERE RATING = UNIVERSITY.RATING

AND CITY = UNIVERSITY.CITY

AND UNIV\_ID <> UNIVERSITY.UNIV\_ID

))

);

1. Создайте две таблицы: "Работники" и "Отделы". Свяжите их отношением один-ко-многим. Напишите запрос, который возвращает список всех отделов с указанием количества работников в них.

CREATE TABLE Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(50) NOT NULL

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

EmployeeName VARCHAR(50) NOT NULL,

DepartmentID INT,

CONSTRAINT FK\_Employees\_Departments FOREIGN KEY (DepartmentID) REFERENCES Departments (DepartmentID)

);

INSERT INTO Departments (DepartmentID, DepartmentName)

VALUES (1, 'Отдел продаж'),

(2, 'Отдел маркетинга'),

(3, 'Отдел разработки');

INSERT INTO Employees (EmployeeID, EmployeeName, DepartmentID)

VALUES (1, 'Иванов Иван', 1),

(2, 'Петров Петр', 1),

(3, 'Сидорова Елена', 1),

(4, 'Козлова Анна', 2),

(5, 'Макарова Ольга', 2),

(6, 'Кузнецов Дмитрий', 3),

(7, 'Васильев Алексей', 3),

(8, 'Соколова Мария', 3),

(9, 'Никитина Анна', 3),

(10, 'Григорьев Иван',3);

Запрос:

SELECT Departments.DepartmentName, COUNT(Employees.EmployeeID) as EmployeeCount

FROM Departments

LEFT JOIN Employees ON Departments.DepartmentID = Employees.DepartmentID

GROUP BY Departments.DepartmentName;

1. Создайте две таблицы: "Клиенты" и "Заказы". Свяжите их отношением один-ко-многим. Напишите запрос, который возвращает список всех клиентов, для каждого из которых указывается количество сделанных им заказов и общая стоимость всех его заказов.

CREATE TABLE Clients (

ClientID INT PRIMARY KEY,

ClientName VARCHAR(50) NOT NULL

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

OrderDate DATE NOT NULL,

OrderAmount DECIMAL(10,2) NOT NULL,

ClientID INT,

CONSTRAINT FK\_Orders\_Clients FOREIGN KEY (ClientID) REFERENCES Clients (ClientID)

);

INSERT INTO Clients (ClientID, ClientName)

VALUES (1, 'Иванов'),

(2, 'Петров'),

(3, 'Сидоров'),

(4, 'Козлов'),

(5, 'Макаров');

INSERT INTO Orders (OrderID, OrderDate, OrderAmount, ClientID)

VALUES (1, '2022-01-01', 100.00, 1),

(2, '2022-02-01', 200.00, 1),

(3, '2022-03-01', 150.00, 2),

(4, '2022-04-01', 300.00, 3),

(5, '2022-05-01', 50.00, 3),

(6, '2022-06-01', 400.00, 3),

(7, '2022-07-01', 100.00, 4),

(8, '2022-08-01', 250.00, 4),

(9, '2022-09-01', 150.00, 5);

Запрос:

SELECT Clients.ClientName, COUNT(Orders.OrderID) as OrderCount, SUM(Orders.OrderAmount)as TotalAmount

FROM Clients

LEFT JOIN Orders ON Clients.ClientID = Orders.ClientID

GROUP BY Clients.ClientName;

**Вывод:** в ходе лабораторной работы были использованы методы, задействовующие манипулирование данными.