

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 _____ ИТР _____

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)  
);
```

2. 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)  
);
```

3. 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),
```

					МИВУ 09.03.04 – N.00n	Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

```

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)
);

```

4. 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)
);

```

5. 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
))
);

```

6. 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,

```

					МИВУ 09.03.04 – N.00n	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

```
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, 'Тригорьев 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;
```

DepartmentName	EmployeeCount
Отдел маркетинга	2
Отдел продаж	3
Отдел разработки	5

7. 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)
);
```

					МИВУ 09.03.04 – N.00n	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

```
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;
```

ClientName	OrderCount	TotalAmount
Иванов	2	300
Козлов	2	350
Макаров	1	150
Петров	1	150
Сидоров	3	750

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

					МИВУ 09.03.04 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5