Московско финансово-юридическая академия МФЮА г. Ярославля

Лабораторная работа №3 по дисциплине:

«Основы проектирования баз данных»

Работу выполнил:

Студент группы 31ИПо8381

Корпачёв Д.Д

Преподователь:

Фурмавнин С.А

Ярославль, 2025 год

Цель работы:

Научиться наполнять базу данных в СУБД MS SQL Management Studio/PostreSQL/MySQL, созданную в проекте, вносить изменения в данные и удалять их. Научиться строить SQL запросы INSERT, UPDATE и DELETE.

Теоретические сведения:

UPDATE –

SET –

SELECT –

FROM –

JOIN –

WHERE –

LIKE / NOT LIKE –

Структура 1 задания:

Цель первого задания:

Написать скрипт добавления в вашу базу данных осмысленных данных с использованием INSERT. Суммарно в базу данных необходимо добавить не менее 50 записей.

INSERT-запросы пишутся в виде:

INSERT INTO TABLE\_NAME(field\_1, field\_2, field\_3) VALUES

(field\_1\_data , field\_2\_data, field\_3\_data),

(field\_1\_data , field\_2\_data, field\_3\_data),

(field\_1\_data , field\_2\_data, field\_3\_data)

Мои INSERT`ы:

INSERT INTO Patients(patient\_id, patient\_fio, patient\_birthday, patient\_gender, service\_discount, patient\_address)

VALUES

(0, 'Горбин', '2000-01-01', 'M', 200.50, 'Рыбинск. Ул. Авдеева'),

(1, 'Стукачёв', '2000-02-02', 'M', 100.25, 'Рыбинск. Ул. Гаванская'),

(2, 'Сишарпов', '2000-03-03', 'M', 195.50, 'Рыбинск. Ул. Батова'),

(3, 'Иванов', '2000-04-04', 'M', 225.35, 'Ярославль. Ул. Республиканская'),

(4, 'Носов', '2000-05-05', 'M', 115.75, 'Ярославль. Ул. Пушкина'),

(5, 'Ремонтов', '2000-06-06', 'M', 200.15, 'Ярославль Ул. Свободы');

INSERT INTO Doctors(doctor\_id, doctor\_fio, specialty, category\_name, price\_per\_visit)

VALUES

(0, 'Петров П.П.', 'Терапевт', 'Первая', 100),

(1, 'Сидорова С.С', 'Терапевт', 'Вторая', 110),

(2, 'Кузнецов К.К.', 'Пульмонолог', 'Третья', 120),

(3, 'Иванова И.И.', 'Эндокринолог', 'Четвёртая', 130),

(4, 'Морозов М.М.', 'Кардиолог', 'Пятая', 140),

(5, 'Тихонов Т.Т.', 'Аллерголог', 'Шестая', 150);

INSERT INTO Appointments(appointment\_id, purpose, appointment\_date, coupon\_number, visit\_cost)

VALUES

(0, 'Консультация по гриппу', '2023-10-01', 12345, 300.00),

(1, 'Диагностика ОРВИ', '2023-10-02', 12346, 150.00),

(2, 'Лечение пневмонии', '2023-10-03', 12347, 400.00),

(3, 'Обследование диабета', '2023-10-04', 12348, 500.00),

(4, 'Консультация по гипертонии', '2023-10-05', 12349, 200.00),

(5, 'Обследование аллергии', '2023-10-06', 12350, 50.00);

INSERT INTO MedicalCards(med\_card\_id, patient\_id, opening\_date, appointment\_id)

VALUES

(0, 0, '2023-10-01', 0),

(1, 1, '2023-10-02', 1),

(2, 2, '2023-10-03', 2),

(3, 3, '2023-10-04', 3),

(4, 4, '2023-10-05', 4),

(5, 5, '2023-10-06', 5);

INSERT INTO MedicalRecordEntries(record\_number, med\_card\_id, diagnosis\_name, doctor\_id)

VALUES

(0, 0, 'Грипп', 0),

(1, 1, 'ОРВИ', 1),

(2, 2, 'Пневмония', 2),

(3, 3, 'Диабет', 3),

(4, 4, 'Гипертония', 4),

(5, 5, 'Аллергия', 5);

Структура 2 задания:

Цель второго задания:

Написать скрипт изменения имеющихся данных с

использованием UPDATE. Подробности задания о том, какие

данные необходимо изменить и на что, уточняется у преподавателя

и является индивидуальным заданием по каждому варианту. В

рамках этого упражнения необходимо написать не менее 5 запросов.

UPDATE-запросы пишутся в виде:

UPDATE TABLE\_NAME

SET <имя поля, которое нужно изменить> = some\_condition;

WHERE name\_field = some\_condition;

Задания, данные мне преподователем звучат так:

1. Все посещения к доктору Сидоровой, снизить стоимость на 10%

2. У всех пациентов, родившехся до 01.04 увеличить скидку в 1,5

3. Всем иногородним скидка 10%

Мой запрос для 1 задания:

UPDATE Appointments

SET visit\_cost = visit\_cost \* 1.1

WHERE appointment\_id IN (

SELECT app.appointment\_id

FROM Appointments app

JOIN MedicalCards mc ON app.appointment\_id = mc.appointment\_id

JOIN MedicalRecordEntries mre ON mc.med\_card\_id = mre.med\_card\_id

WHERE mre.doctor\_id = 1

);

Разберём построчно:

1. UPDATE Appointments – начало запроса, будем обновлять таблицу Appointments
2. SET visit\_cost = visit\_cost \* 1.1 – устанавливаем значение для поля visit\_cost
3. WHERE appointment\_id IN ( - указываем условие, где мы будем обновлять только те записи, где appointment\_id находится в результате подзапроса
4. SELECT app.appointment\_id – в подзапросе выбираем поле appointment\_id от псевдонима app таблицы Appointments
5. FROM Appointments app – объявляем псевдоним app от таблицы Appointments и берём данные из Appointments
6. JOIN MedicalCards mc ON app.appointment\_id = mc.appointment\_id – соединяем с таблицей MedicalCards (псевдоним mc) по совпадению appointment\_id в обеих таблицах
7. JOIN MedicalRecordEntries mre ON mc.med\_card\_id = mre.med\_card\_id – соединяем с таблицей MedicalRecordEntries (псевдоним mre) по совпадению med\_card\_id в обеих таблицах
8. WHERE mre.doctor\_id = 1 – запрос будет фильтровать данные по условию, что номер доктора равен одному
9. ); - конец запроса

Т.е запрос должен обновить поле visit\_cost, если номер доктора – это 1 (Доктор Сидорова), и понизить стоимость посещения на 10%, если такой доктор есть && есть посещение к ней

Мой запрос для второго задания:

UPDATE Patients

SET service\_discount = service\_discount - service\_discount \* 0.015

WHERE patient\_birthday < '2000-04-04';

Разберём запрос построчно:

1. UPDATE Patients – будем обновлять данные в таблице Patients в результате запроса
2. SET service\_discount = service\_discount - service\_discount \* 0.015 – устанавливаем значение для поля service\_discount
3. WHERE patient\_birthday < '2000-04-04'; - условие, которое говорит, что ДР Пациента должно быть меньше 2000г 4 апреля и конец запроса

Запрос должен обновить поле service\_discount (скидка), при условии, что ДР пациента должно быть раньше, чем 4 апреля 2000г, скидка должна повыситься в 1,5 раза

Мой запрос для третьего задания:

UPDATE Patients

SET service\_discount = service\_discount - service\_discount \* 0.1

WHERE patient\_address NOT LIKE '%Ярославль%';

Разберём запрос построчно:

1. UPDATE Patients – будем обновлять данные в таблице Patients в результате запроса
2. SET service\_discount = service\_discount - service\_discount \* 0.1 – устанавливаем значение полю service\_discount
3. WHERE patient\_address NOT LIKE '%Ярославль%'; - условие, которое говорит, что пациент должен быть иногородним

Запрос должен обновить данные в поле service\_discount, если пациент – иногородный,

Структура 3 задания: