Лекция 2. Создание и заполнение БД

1. Создать базу данных при помощи скрипта.

2. Создать минимум 2 таблицы в созданной базе согласно теме при помощи скрипта. Например: таблицы users и

city. Таблицы должны иметь PRIMARY KEY и FOREIGN KEY (поле user\_city)

3. Добавить новое поле в таблицу users при помощи ALTER\_TABLE

4. Заполнить таблицы данными при помощи команды INSERT. Таблица Users должна содержать не менее 10

записей. Таблица City - не менее 5 записей. Обновить при помощи команды UPDATE по одной записи из

каждой таблицы

5. B CREATE TABLE должны быть определены значения полей по умолчанию, ограничения уникальности и

проверочные ограничения (в CREATE TABLE для базы данных должно быть определено хотя бы по одному

ограничению каждого вида). Название и количество полей придумываем сами основываясь на предметной

области таблиц.

6. B ALTER TABLE должны быть определены ограничения уникальности и проверочные ограничения (в ALTER

TABLE для базы данных должно быть определено хотя бы по одному ограничению каждого вида).

7. Удаляем все таблицы и базу данных.

В отчете (.doc) должен быть представлен текст SQL-запросов для создания всех таблиц разработанной базы

данных или файл с самими запросами в формате .sql Не забываем приложить схему самой БД!!!

DROP DATABASE IF EXISTS online\_banking;

CREATE DATABASE online\_banking;

USE online\_banking;

ALTER TABLE accounts

DROP FOREIGN KEY fk\_client\_id;

DROP TABLE IF EXISTS clients;

CREATE TABLE IF NOT EXISTS clients (

client\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(200) NOT NULL,

last\_name VARCHAR(200) NOT NULL,

patronymic VARCHAR(200) NOT NULL,

phone\_number VARCHAR(10) NOT NULL,

email VARCHAR(200) NOT NULL,

birthday DATE NOT NULL,

residential\_address VARCHAR(200) NOT NULL

);

DROP TABLE IF EXISTS accounts;

CREATE TABLE IF NOT EXISTS accounts (

account\_number INT PRIMARY KEY AUTO\_INCREMENT,

account\_type VARCHAR(200) NOT NULL,

currency VARCHAR(200) NOT NULL,

client\_id INT

);

ALTER TABLE clients

ADD CONSTRAINT unique\_email UNIQUE (email);

ALTER TABLE accounts

ADD CONSTRAINT fk\_client\_id FOREIGN KEY (client\_id) REFERENCES clients(client\_id);

INSERT INTO clients (name, last\_name, patronymic, phone\_number, email, birthday, residential\_address)

VALUES

('Иван', 'Иванов', 'Иванович', '1234567890', 'ivan@example.com', '1990-01-01', 'ул. Ленина, д. 1'),

('Петр', 'Петров', 'Петрович', '9876543210', 'petr@example.com', '1985-05-05', 'ул. Пушкина, д. 10'),

('Мария', 'Сидорова', 'Ивановна', '5555555555', 'maria@example.com', '1995-10-10', 'ул. Гагарина, д. 5'),

('Алексей', 'Алексеев', 'Алексеевич', '1111111111', 'alex@example.com', '1988-08-08', 'ул. Кирова, д. 8'),

('Елена', 'Еленова', 'Еленовна', '2222222222', 'elena@example.com', '1992-02-02', 'ул. Сталина, д. 20'),

('Дмитрий', 'Дмитриев', 'Дмитриевич', '3333333333', 'dmitriy@example.com', '1980-12-12', 'ул. Лермонтова, д. 12'),

('Ольга', 'Ольгова', 'Ольговна', '4444444444', 'olga@example.com', '1983-03-03', 'ул. Ленина, д. 30'),

('Николай', 'Николаев', 'Николаевич', '6666666666', 'nikolay@example.com', '1975-07-07', 'ул. Гоголя, д. 7'),

('Анна', 'Иванова', 'Петровна', '7777777777', 'anna@example.com', '1987-04-15', 'ул. Кирова, д. 15'),

('Сергей', 'Сергеев', 'Сергеевич', '8888888888', 'sergey@example.com', '1978-11-25', 'ул. Пушкина, д. 5');

INSERT INTO accounts (account\_type, currency, client\_id)

VALUES

('Сберегательный', 'RUB', 1),

('Текущий', 'USD', 2),

('Кредитный', 'EUR', 3),

('Кредитный', 'RUB', 4),

('Сберегательный', 'USD', 5),

('Текущий', 'EUR', 6),

('Кредитный', 'RUB', 7),

('Сберегательный', 'USD', 8),

('Текущий', 'EUR', 9),

('Кредитный', 'RUB', 10);

UPDATE clients

SET phone\_number = '1111222233'

WHERE client\_id = 1;

UPDATE accounts

SET account\_type = 'Депозитный'

WHERE account\_number = 1;

SELECT \*

FROM clients;

SELECT name, last\_name, patronymic

FROM clients;

SELECT currency

FROM accounts

WHERE currency IN ('USD');

SELECT currency

FROM accounts

WHERE currency IN ('RUB')

AND client\_id > 6;

SELECT currency

FROM accounts

WHERE currency IN ('EUR')

OR client\_id > 6;

SELECT name, last\_name, patronymic

FROM clients

ORDER BY last\_name, name, patronymic;

SELECT name, last\_name, patronymic

FROM clients

ORDER BY last\_name, name, patronymic

LIMIT 5;

SELECT name, last\_name, birthday

FROM clients

WHERE MONTH(birthday) = MONTH(CURRENT\_DATE());

SELECT \*

FROM clients

where phone\_number > 1111111111

order by name;

SELECT \*

FROM clients

where phone\_number > 1111111111

and phone\_number < 6666666666

order by name;

SELECT account\_type, count(client\_id)

FROM accounts

GROUP BY account\_type;

SELECT account\_type, count(client\_id)

FROM accounts

GROUP BY account\_type

HAVING count(client\_id) > 2;

