НАО «КАРАГАНДИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМЕНИ АБЫЛКАСА САГИНОВА»

Кафедра информационных технологий и безопасности

**Лабораторная работа № 2**

**Дисциплина:** "Введение в базы данных"

**Тема:** Создание баз данных и таблиц в среде MS SQL Server. Информационное наполнение.

**Принял:**

Преподаватель: Жакина М.М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)      (дата)*

**Выполнил:**

ст.гр.СИБ-23-7  Абуев С.Б

Караганда 2025

**Лабораторная работа №2**

**Тема:** Создание баз данных и таблиц в среде MS SQL Server. Информационное наполнение.  
  
**Цель:**

- усвоить способы создания таблиц, умолчаний, правил, ограничений БД средствами СУБД MS SQL Server;  
- усвоить способы создания индексов, отношений и схемы отношений (диаграммы) базы данных средствами СУБД MS SQL Server;  
- усвоить способы создания вода, удаления, редактирования данных в таблицах БД MS SQL Server.

CREATE DATABASE lab2

ON

(

NAME = 'lab2',

FILENAME = 'C:\Users\User\Desktop\databases\srsp\databases\mdf\lab2.mdf',

SIZE = 10MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB)

LOG ON

(

NAME = 'lab2\_log',

FILENAME = 'C:\Users\User\Desktop\databases\srsp\databases\logs\lab2.ldf',

SIZE = 20MB,

MAXSIZE = 100MB,

FILEGROWTH = 5MB

)

GO

USE lab2

GO

CREATE TYPE nv FROM NVARCHAR(100) NOT NULL

go

CREATE TYPE nv1 FROM NVARCHAR(100)

go

CREATE TYPE bint from BIGINT NOT NULL

go

CREATE TYPE bint1 from BIGINT

go

CREATE TYPE date1 from DATE NOT NULL

go

CREATE TABLE clients

(

client\_id BIGINT IDENTITY(1,1) PRIMARY KEY,

name nv,

bin\_iin bint UNIQUE,

mailing\_address nv,

bank\_code INT,

account\_number bint UNIQUE,

isindividual bint DEFAULT 1,

CONSTRAINT chk\_client CHECK (

(isindividual = 1 AND LEN(CAST(bin\_iin AS VARCHAR)) = 12)

OR (isindividual = 0 AND LEN(CAST(bin\_iin AS VARCHAR)) = 9)

)

)

GO

CREATE TABLE contracts

(

contract\_id BIGINT IDENTITY(1,1) PRIMARY KEY,

client\_bin\_iin bint,

contract\_number bint UNIQUE,

contract\_date date1,

contract\_duration date1,

CONSTRAINT fk\_contract\_client FOREIGN KEY (client\_bin\_iin) REFERENCES clients(bin\_iin)

)

GO

CREATE TABLE contract\_services

(

contract\_service\_id INT IDENTITY(1,1) PRIMARY KEY,

contract\_number bint,

service\_code bint,

tariff\_plan\_code int

)

GO

CREATE TABLE services

(

service\_code bint IDENTITY(1,1) PRIMARY KEY,

service\_category\_code bint

)

GO

CREATE TABLE service\_categories

(

category\_code bint IDENTITY(1,1) PRIMARY KEY,

name nv,

note nv

)

GO

CREATE TABLE tariff\_plan

(

tariffplancode INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

unit nv,

price DECIMAL(10,2) NOT NULL,

tariffvalidityperiod nv

)

GO

CREATE TABLE usage

(

usageid INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

usagedate date1,

sessionstarttime DATETIME,

sessionendtime DATETIME,

clientbin\_iin bint,

bytessent bint1,

bytesreceived bint1,

servicecode bint

)

GO

CREATE TABLE bank

(

bankcode INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

bankname nv NOT NULL UNIQUE,

address nv1

)

GO

CREATE TABLE payment

(

paymentid INT IDENTITY(1,1) PRIMARY KEY,

clientbin\_iin bint,

contractnumber bint,

paymentdate date1,

amount DECIMAL(10,2) NOT NULL,

bankcode INT

)

GO

CREATE TABLE account

(

accountid INT IDENTITY(1,1) PRIMARY KEY,

clientbin\_iin bint,

openingbalance DECIMAL(10,2) NOT NULL,

incomingamount DECIMAL(10,2) NOT NULL,

costofservices DECIMAL(10,2) NOT NULL,

month INT NOT NULL,

year INT NOT NULL

)

GO

ALTER TABLE clients

ADD CONSTRAINT fk\_client\_bank FOREIGN KEY (bank\_code) REFERENCES bank(bankcode)

ON UPDATE CASCADE

ON DELETE SET NULL;

GO

ALTER TABLE contract\_services

ADD CONSTRAINT fk\_cs\_contract FOREIGN KEY (contract\_number) REFERENCES contracts(contract\_number)

ON UPDATE CASCADE

ON DELETE NO ACTION,

CONSTRAINT fk\_cs\_service FOREIGN KEY (service\_code) REFERENCES services(service\_code)

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_cs\_tariff FOREIGN KEY (tariff\_plan\_code) REFERENCES tariff\_plan(tariffplancode)

ON UPDATE NO ACTION

ON DELETE NO ACTION;

GO

ALTER TABLE services

ADD CONSTRAINT fk\_service\_category FOREIGN KEY (service\_category\_code) REFERENCES service\_categories(category\_code)

ON UPDATE NO ACTION

ON DELETE NO ACTION

GO

ALTER TABLE usage

ADD CONSTRAINT fk\_usage\_client FOREIGN KEY (clientbin\_iin) REFERENCES clients(bin\_iin)

ON UPDATE CASCADE

ON DELETE NO ACTION,

CONSTRAINT fk\_usage\_service FOREIGN KEY (servicecode) REFERENCES services(service\_code)

ON UPDATE NO ACTION

ON DELETE NO ACTION

GO

ALTER TABLE payment

ADD CONSTRAINT fk\_payment\_client FOREIGN KEY (clientbin\_iin) REFERENCES clients(bin\_iin)

ON UPDATE CASCADE

ON DELETE NO ACTION,

CONSTRAINT fk\_payment\_contract FOREIGN KEY (contractnumber) REFERENCES contracts(contract\_number)

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT fk\_payment\_bank FOREIGN KEY (bankcode) REFERENCES bank(bankcode)

ON UPDATE NO ACTION

ON DELETE NO ACTION

GO

ALTER TABLE account

ADD CONSTRAINT fk\_account\_client FOREIGN KEY (clientbin\_iin) REFERENCES clients(bin\_iin)

ON UPDATE CASCADE

ON DELETE NO ACTION

GO

CREATE INDEX idx\_contracts\_client\_bin\_iin ON contracts(client\_bin\_iin);

CREATE INDEX idx\_cs\_contract\_number ON contract\_services(contract\_number);

CREATE INDEX idx\_cs\_service\_code ON contract\_services(service\_code);

CREATE INDEX idx\_cs\_tariff\_plan\_code ON contract\_services(tariff\_plan\_code);

CREATE INDEX idx\_usage\_client ON usage(clientbin\_iin);

CREATE INDEX idx\_usage\_service ON usage(servicecode);

CREATE INDEX idx\_payment\_client ON payment(clientbin\_iin);

CREATE INDEX idx\_payment\_contract ON payment(contractnumber);

CREATE INDEX idx\_payment\_bank ON payment(bankcode);

CREATE INDEX idx\_account\_client ON account(clientbin\_iin);

-- Банки

INSERT INTO bank (bankname, address) VALUES

('KazBank', 'Almaty, Abay 10'),

('HalykBank', 'Astana, Satpayev 45'),

('FreedomBank', 'Almaty, Dostyk 100'),

('EurasianBank', 'Shymkent, Tole bi 90'),

('ForteBank', 'Kokshetau, Abay 20'),

('KaspiBank', 'Almaty, Nazarbayev 50'),

('ATFBank', 'Kostanay, Bayseitova 5'),

('SberBank', 'Astana, Kunaev 2'),

('CenterCredit', 'Almaty, Timiryazev 80'),

('JusanBank', 'Atyrau, Satpayev 12');

-- Клиенты

INSERT INTO clients (name, bin\_iin, mailing\_address, bank\_code, account\_number, isindividual)

VALUES

('Aidos A.', 123456789012, 'Almaty', 1, 111111111111, 1),

('Diana K.', 123456789013, 'Astana', 2, 111111111112, 1),

('TOO "MegaNet"', 987654321, 'Shymkent', 3, 111111111113, 0),

('Talgat M.', 123456789014, 'Almaty', 4, 111111111114, 1),

('TOO "DataLine"', 987654322, 'Astana', 5, 111111111115, 0),

('Dana B.', 123456789015, 'Kokshetau', 6, 111111111116, 1),

('TOO "TeleCom"', 987654323, 'Aktau', 7, 111111111117, 0),

('Serik S.', 123456789016, 'Atyrau', 8, 111111111118, 1),

('TOO "NetCity"', 987654324, 'Kostanay', 9, 111111111119, 0),

('Aruzhan Z.', 123456789017, 'Astana', 10, 111111111120, 1);

-- Категории услуг

INSERT INTO service\_categories (name, note) VALUES

('Internet', 'High-speed internet'),

('Mobile', 'Mobile communication'),

('TV', 'Cable television');

-- Услуги

INSERT INTO services (service\_category\_code) VALUES

(1), (1), (2), (2), (3), (3);

-- Тарифные планы

INSERT INTO tariff\_plan (unit, price, tariffvalidityperiod) VALUES

('GB', 1000.00, 'Monthly'),

('Minute', 500.00, 'Monthly'),

('Channel', 800.00, 'Monthly'),

('GB', 2000.00, 'Weekly'),

('Minute', 1000.00, 'Weekly');

-- Контракты

INSERT INTO contracts (client\_bin\_iin, contract\_number, contract\_date, contract\_duration)

VALUES

(123456789012, 50001, '2024-01-01', '2025-01-01'),

(987654321, 50002, '2024-02-01', '2025-02-01'),

(123456789013, 50003, '2024-03-01', '2025-03-01'),

(987654322, 50004, '2024-04-01', '2025-04-01'),

(123456789014, 50005, '2024-05-01', '2025-05-01'),

(987654323, 50006, '2024-06-01', '2025-06-01'),

(123456789015, 50007, '2024-07-01', '2025-07-01'),

(987654324, 50008, '2024-08-01', '2025-08-01'),

(123456789016, 50009, '2024-09-01', '2025-09-01'),

(123456789017, 50010, '2024-10-01', '2025-10-01');

-- Услуги по контрактам (20 записей)

INSERT INTO contract\_services (contract\_number, service\_code, tariff\_plan\_code)

VALUES

(50001, 1, 1), (50001, 2, 2),

(50002, 3, 2), (50002, 4, 5),

(50003, 5, 3), (50004, 1, 4),

(50005, 2, 1), (50006, 3, 5),

(50007, 4, 2), (50008, 5, 3),

(50009, 6, 4), (50010, 1, 1),

(50001, 3, 2), (50002, 5, 3),

(50003, 6, 4), (50004, 2, 1),

(50005, 4, 2), (50006, 5, 3),

(50007, 6, 4), (50008, 1, 1);

-- Usage (20+ записей)

INSERT INTO usage (usagedate, sessionstarttime, sessionendtime, clientbin\_iin, bytessent, bytesreceived, servicecode)

VALUES

('2025-01-01', '2025-01-01 10:00', '2025-01-01 10:30', 123456789012, 2000, 3000, 1),

('2025-01-02', '2025-01-02 12:00', '2025-01-02 13:00', 123456789013, 1500, 2500, 2),

('2025-01-03', '2025-01-03 14:00', '2025-01-03 15:00', 987654321, 3000, 4000, 3),

('2025-01-04', '2025-01-04 09:00', '2025-01-04 09:30', 987654322, 500, 800, 4),

('2025-01-05', '2025-01-05 08:00', '2025-01-05 08:45', 123456789014, 2500, 3200, 5),

('2025-01-06', '2025-01-06 16:00', '2025-01-06 17:15', 987654323, 1000, 1500, 6),

('2025-01-07', '2025-01-07 18:00', '2025-01-07 19:00', 123456789015, 1800, 2100, 1),

('2025-01-08', '2025-01-08 11:00', '2025-01-08 12:00', 987654324, 2400, 2700, 2),

('2025-01-09', '2025-01-09 15:00', '2025-01-09 15:30', 123456789016, 900, 1200, 3),

('2025-01-10', '2025-01-10 17:00', '2025-01-10 17:45', 123456789017, 1500, 1700, 4);

-- Платежи (20 записей)

INSERT INTO payment (clientbin\_iin, contractnumber, paymentdate, amount, bankcode)

VALUES

(123456789012, 50001, '2025-01-15', 5000, 1),

(987654321, 50002, '2025-02-15', 10000, 2),

(123456789013, 50003, '2025-03-15', 8000, 3),

(987654322, 50004, '2025-04-15', 6000, 4),

(123456789014, 50005, '2025-05-15', 12000, 5),

(987654323, 50006, '2025-06-15', 4000, 6),

(123456789015, 50007, '2025-07-15', 7000, 7),

(987654324, 50008, '2025-08-15', 9000, 8),

(123456789016, 50009, '2025-09-15', 5000, 9),

(123456789017, 50010, '2025-10-15', 11000, 10);

-- Account (20 записей)

INSERT INTO account (clientbin\_iin, openingbalance, incomingamount, costofservices, month, year)

VALUES

(123456789012, 10000, 5000, 3000, 1, 2025),

(987654321, 15000, 7000, 4000, 2, 2025),

(123456789013, 9000, 6000, 3500, 3, 2025),

(987654322, 11000, 4000, 2500, 4, 2025),

(123456789014, 8000, 5000, 3000, 5, 2025),

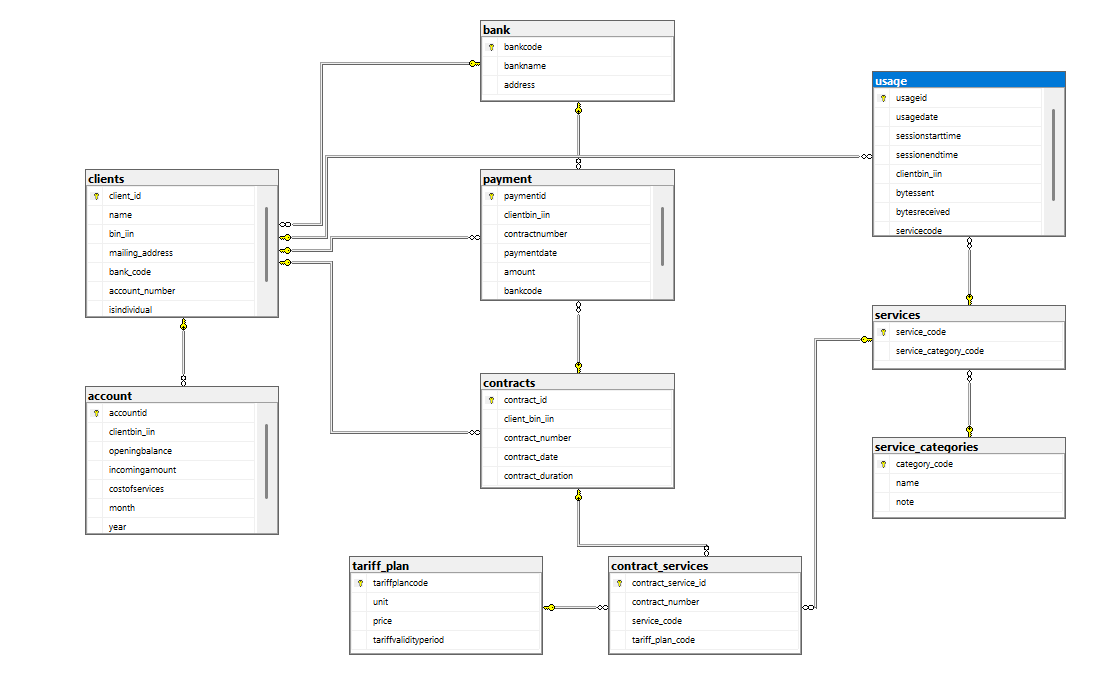
(987654323, 9500, 6000, 3200, 6, 2025),

(123456789015, 7000, 4500, 2800, 7, 2025),

(987654324, 12000, 8000, 3600, 8, 2025),

(123456789016, 10000, 4000, 2000, 9, 2025),

(123456789017, 11000, 7000, 3300, 10, 2025);



**Контрольные вопросы**:

### **1. Перечень поддерживаемых типов данных**

В SQL Server:

* **Числовые:** INT, BIGINT, SMALLINT, TINYINT, DECIMAL, NUMERIC, FLOAT, REAL, MONEY, SMALLMONEY
* **Строковые:** CHAR, VARCHAR, NCHAR, NVARCHAR, TEXT, NTEXT
* **Дата и время:** DATE, DATETIME, SMALLDATETIME, TIME, DATETIME2, DATETIMEOFFSET
* **Логические:** BIT
* **Бинарные:** BINARY, VARBINARY, IMAGE
* **Прочие:** UNIQUEIDENTIFIER, XML, GEOGRAPHY, HIERARCHYID, CURSOR, TABLE

### **2. Пользовательский тип данных**

**Назначение:** унифицировать типы в разных таблицах (например, везде один тип для BIN/IIN).  
**Создание:**

CREATE TYPE nv FROM NVARCHAR(100) NOT NULL;

**Удаление:**

DROP TYPE nv;

### **3. Способы определения таблиц**

* Через **DDL-команду** CREATE TABLE:
* CREATE TABLE clients (...);
* Через **графический интерфейс** SSMS (Design Table).
* Через **скрипт на основе существующей таблицы** (SELECT INTO).

### **4. Назначение умолчаний, правил и ограничений**

* **Умолчания (DEFAULT):** значение по умолчанию, если не указано другое.
* **Ограничения (CONSTRAINTS):** обеспечивают корректность данных.
* **Правила (RULE):** старый механизм, аналог CHECK.

### **5. Перечень поддерживаемых умолчаний, ограничений, правил**

* DEFAULT — умолчание
* CHECK — логическое условие
* UNIQUE — уникальность
* PRIMARY KEY — первичный ключ
* FOREIGN KEY — внешний ключ
* NOT NULL — запрет NULL
* RULE — пользовательское правило (устарело)

### **6. Способы назначения умолчаний, ограничений, правил**

* В момент создания таблицы:
* salary DECIMAL(10,2) DEFAULT 0 CHECK (salary >= 0)
* После создания таблицы:
* ALTER TABLE employees ADD CONSTRAINT chk\_salary CHECK (salary > 0);

### **7. Что представляют собой Null-значение**

NULL — это **отсутствие значения**, а не ноль и не пустая строка.  
При сравнении NULL = NULL → **FALSE**, поэтому используется IS NULL.

### **8. Свойство колонки IDENTITY**

Позволяет **автоматически генерировать значения** для поля при вставке.

id INT IDENTITY(1,1)

→ начало с 1, шаг 1.  
Используется для первичных ключей.

### **9. Назначение механизма отношений. Виды отношений**

Назначение — **связывать таблицы** и обеспечивать целостность данных.  
**Виды:**

* Один-к-одному (1:1)
* Один-ко-многим (1:N)
* Многие-ко-многим (M:N, через промежуточную таблицу)

### **10. Способы создания отношений**

* Через FOREIGN KEY при создании таблицы или через ALTER TABLE

ALTER TABLE orders ADD CONSTRAINT fk\_client FOREIGN KEY (client\_id)

REFERENCES clients(client\_id)

* Через визуальный конструктор диаграмм в SSMS.

### **11. Ссылочная целостность данных. Способы её поддержания**

Обеспечивает, чтобы **ссылки между таблицами были корректны**.  
Поддерживается с помощью FOREIGN KEY и правил:

* ON DELETE CASCADE
* ON UPDATE CASCADE
* SET NULL
* NO ACTION

### **12. Способы отображения зависимостей между таблицами**

* Через **Database Diagrams** в SSMS
* Через **sys.foreign\_keys** / **sys.objects**
* Через визуальные ER-диаграммы (например, в dbdiagram.io)

### **13. Способы построения и изменения схемы БД**

* CREATE TABLE, ALTER TABLE, DROP TABLE
* CREATE SCHEMA, ALTER SCHEMA
* В SSMS через GUI (Design, Diagrams)

### **14. Назначение и классификация индексов**

**Назначение:** ускоряют поиск и сортировку данных.  
**Классификация:**

* По уникальности: UNIQUE / обычный
* По структуре: кластерный (CLUSTERED) и некластерный (NONCLUSTERED)
* По составу: одиночный / составной

### **15. Способы создания индексов**

CREATE INDEX idx\_client\_bank ON clients(bank\_code);

CREATE UNIQUE INDEX uidx\_contract ON contracts(contract\_number);

или при создании PK/UNIQUE они создаются автоматически.

### **16. Способы ввода данных в таблицы. Ограничения целостности**

* Ввод через INSERT INTO
* Через SSMS (Edit Top 200 Rows)
* Через импорты (BULK INSERT, bcp, SSIS)  
  Ограничения целостности (PK, FK, CHECK, UNIQUE) проверяются **при вставке**.

### **17. Понятие транзакции. Свойства транзакций**

**Транзакция** — группа операций, выполняемых как единое целое.  
**Свойства ACID:**

* **A** — Atomicity (атомарность)
* **C** — Consistency (согласованность)
* **I** — Isolation (изоляция)
* **D** — Durability (долговечность)

BEGIN TRAN

UPDATE ...

COMMIT TRAN

### **18. Удаление данных, усечение таблиц, удаление таблиц**

* Удаление данных:
* DELETE FROM table WHERE condition;
* Удаление всех данных:
* TRUNCATE TABLE table;
* Удаление таблицы:

DROP TABLE table;