|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ и информационные технологии»***\_\_\_\_\_

**ЛАБОРАТОРНЫЕ РАБОТЫ 1**

**ДИСЦИПЛИНА: «ПРОЕКТИРОВАНИЕ СИСТЕМ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. \_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_Губин Е.В.\_\_\_\_\_)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_Глебов С.А.\_\_\_\_\_)  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2024

**Цель:** формирование практических навыков разработки базы данных.

**Задачи:**

1. По имеющейся даталогической модели разработать базу данных под управлением выбранной СУБД
2. Определить стратегии автоматической поддержки ссылочной целостности.
3. Проверить функциональность.

**Проектирование базы данных**

Предметной областью базы данных является телефонная компания. Реализованы следующие сущности:

* Customers – описывает клиентов телефонной компании, в качестве PRIMARY KEY выбран номер лицевого счёта клиента, так как он является уникальным.
* Phone\_numbers – определяет номера телефонов клиентов. По реализуемой идее у одного клиента может быть несколько номеров телефона, поэтому реализована связь с таблицей customers 1:m.
* Traffic\_plans – описывает все тарифные планы у телефонной компании. Один тарифный план может быть у нескольких номеров телефона, поэтому реализована связь с таблицей phone\_numbers 1:m.
* Customer\_to\_customer и customer\_to\_other – две таблицы, описывающие действия (сообщения и звонки) в двух форматах: связь с клиентом этой же телефонной компании и связь с клиентом другой компании. В первом случае определены свойства источника и его собеседника (реализована связь 1:m с таблицей phone\_numbers), а во втором случае определяется номер телефона клиента другой компании, клиент этой компании (реализована связь 1:m с таблицей phone\_numbers).
* Messages – описывает сущность сообщений. Имеет два поля, которые определяют вышеописанный формат связи, при этом у одной записи одно из них должно быть null, а другое непустым.
* Calls – Описывает сущность звонка, Имеет те же два поля, что и таблица messages.

**ER-диаграмма:**

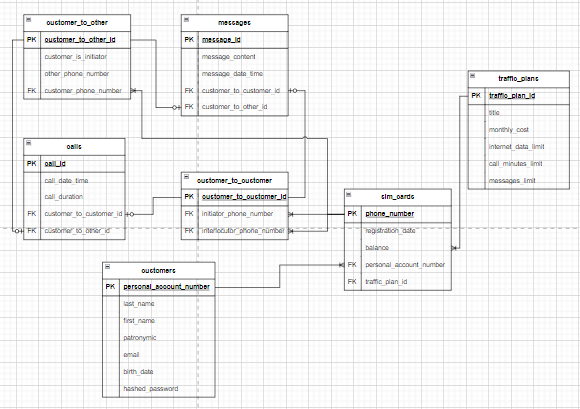


Рисунок 1ER-диаграмма

**Реализация в postgresql**

**Создание таблиц:**

CREATE TABLE customers (

personal\_account\_number BIGINT PRIMARY KEY,

last\_name VARCHAR(50) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

patronymic VARCHAR(50),

email TEXT UNIQUE NOT NULL,

birth\_date DATE NOT NULL,

hashed\_password VARCHAR(50) NOT NULL

);

CREATE TABLE traffic\_plans (

traffic\_plan\_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

title VARCHAR(50) NOT NULL,

monthly\_cost REAL NOT NULL,

internet\_data\_limit REAL,

call\_minutes\_limit INT,

messages\_limit INT

);

CREATE TABLE sim\_cards (

phone\_number VARCHAR(50) PRIMARY KEY,

registration\_date DATE NOT NULL,

balance REAL NOT NULL,

traffic\_plan\_id INT NOT NULL,

personal\_account\_number BIGINT NOT NULL,

FOREIGN KEY (traffic\_plan\_id) REFERENCES traffic\_plans (traffic\_plan\_id),

FOREIGN KEY (personal\_account\_number) REFERENCES customers (personal\_account\_number)

);

CREATE TABLE customer\_to\_customer (

customer\_to\_customer\_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

initiator\_phone\_number VARCHAR(50) NOT NULL,

interlocutor\_phone\_number VARCHAR(50) NOT NULL,

FOREIGN KEY (initiator\_phone\_number) REFERENCES sim\_cards (phone\_number),

FOREIGN KEY (interlocutor\_phone\_number) REFERENCES sim\_cards (phone\_number)

);

CREATE TABLE customer\_to\_other (

customer\_to\_other\_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

customer\_is\_initiator BOOL NOT NULL,

other\_phone\_number VARCHAR(50) NOT NULL,

customer\_phone\_number VARCHAR(50) NOT NULL,

FOREIGN KEY (customer\_phone\_number) REFERENCES sim\_cards (phone\_number)

);

CREATE TABLE messages (

message\_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

message\_content VARCHAR(255) NOT NULL,

message\_date\_time TIMESTAMP,

customer\_to\_customer\_id INT,

customer\_to\_other\_id INT,

FOREIGN KEY (customer\_to\_customer\_id) REFERENCES customer\_to\_customer (customer\_to\_customer\_id),

FOREIGN KEY (customer\_to\_other\_id) REFERENCES customer\_to\_other (customer\_to\_other\_id)

);

CREATE TABLE calls (

call\_id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

call\_date\_time TIMESTAMP NOT NULL,

call\_duration INT NOT NULL,

customer\_to\_customer\_id INT,

customer\_to\_other\_id INT,

FOREIGN KEY (customer\_to\_customer\_id) REFERENCES customer\_to\_customer (customer\_to\_customer\_id),

FOREIGN KEY (customer\_to\_other\_id) REFERENCES customer\_to\_other (customer\_to\_other\_id)

);

**Заполнение базы данных:**

INSERT INTO customers (personal\_account\_number, last\_name, first\_name, patronymic, email, birth\_date, hashed\_password)

VALUES

(1000000001, 'Иванов', 'Иван', 'Иванович', 'ivanov@example.com', '1990-01-15', 'hashed\_password\_1'),

(1000000002, 'Петров', 'Петр', 'Петрович', 'petrov@example.com', '1985-05-20', 'hashed\_password\_2'),

(1000000003, 'Сидорова', 'Анна', 'Сергеевна', 'sidorova@example.com', '1992-03-10', 'hashed\_password\_3');

INSERT INTO traffic\_plans (title, monthly\_cost, internet\_data\_limit, call\_minutes\_limit, messages\_limit)

VALUES

('Базовый', 300.00, 5000, 300, 100),

('Стандартный', 500.00, 10000, 600, 200),

('Премиум', 700.00, 15000, 1200, 300);

INSERT INTO sim\_cards (phone\_number, registration\_date, balance, traffic\_plan\_id, personal\_account\_number)

VALUES

('89001234567', '2023-01-01', 200.00, 1, 1000000001),

('89007654321', '2023-02-01', 300.00, 2, 1000000002),

('89009876543', '2023-03-01', 230.00, 3, 1000000003);

INSERT INTO customer\_to\_customer (initiator\_phone\_number, interlocutor\_phone\_number)

VALUES

('89001234567', '89007654321'),

('89007654321', '89009876543');

INSERT INTO customer\_to\_other (customer\_is\_initiator, other\_phone\_number, customer\_phone\_number)

VALUES

(TRUE, '89001112233', '89001234567'),

(FALSE, '89004445566', '89007654321');

INSERT INTO messages (message\_content, message\_date\_time, customer\_to\_customer\_id, customer\_to\_other\_id)

VALUES

('Привет! Как дела?', NOW(), 1, NULL),

('Все отлично! А у тебя?', NOW(), 1, NULL),

('Не могу говорить сейчас.', NOW(), NULL, 1);

INSERT INTO calls (call\_date\_time, call\_duration, customer\_to\_customer\_id, customer\_to\_other\_id)

VALUES

(NOW(), 120, 1, NULL),

(NOW(), 90, NULL, 1);

**Вывод таблиц:**

SELECT \* FROM customers;

SELECT \* FROM phone\_numbers;

SELECT \* FROM traffic\_plans;

SELECT \* FROM customer\_to\_customer;

SELECT \* FROM customer\_to\_other;

SELECT \* FROM calls;

SELECT \* FROM messages;

**Создание функций и запросов:**

CREATE FUNCTION get\_messages\_rest(ph\_number VARCHAR(50))

RETURNS INT

LANGUAGE plpgsql

AS

$$

DECLARE

rest INT;

BEGIN

SELECT tp.messages\_limit

INTO rest

FROM traffic\_plans tp

WHERE

tp.traffic\_plan\_id = (SELECT sc.traffic\_plan\_id FROM sim\_cards sc WHERE sc.phone\_number = ph\_number);

SELECT rest - COUNT(\*)

INTO rest

FROM

(SELECT ctc.customer\_to\_customer\_id FROM customer\_to\_customer ctc WHERE ctc.initiator\_phone\_number = ph\_number

AND ctc.customer\_to\_customer\_id IN (SELECT mes.customer\_to\_customer\_id FROM messages mes));

SELECT rest - COUNT(\*)

INTO rest

FROM

(SELECT cto.customer\_to\_other\_id FROM customer\_to\_other cto WHERE cto.customer\_phone\_number = ph\_number

AND customer\_is\_initiator = TRUE AND cto.customer\_to\_other\_id IN (SELECT mes.customer\_to\_other\_id FROM messages mes));

RETURN rest;

END;

$$;

SELECT get\_messages\_rest('89001234567');

CREATE FUNCTION get\_call\_minutes\_rest(ph\_number VARCHAR(50))

RETURNS INT

LANGUAGE plpgsql

AS

$$

DECLARE

rest INT;

BEGIN

SELECT tp.call\_minutes\_limit

INTO rest

FROM traffic\_plans tp

WHERE

tp.traffic\_plan\_id = (SELECT sc.traffic\_plan\_id FROM sim\_cards sc WHERE sc.phone\_number = ph\_number);

SELECT rest - SUM(cl.call\_duration)

INTO rest

FROM calls cl

WHERE

cl.customer\_to\_customer\_id IN (SELECT ctc.customer\_to\_customer\_id FROM customer\_to\_customer ctc WHERE ctc.initiator\_phone\_number = ph\_number)

OR cl.customer\_to\_other\_id IN (SELECT cto.customer\_to\_other\_id FROM customer\_to\_other cto WHERE cto.customer\_is\_initiator = TRUE AND cto.customer\_phone\_number = ph\_number);

RETURN rest;

END;

$$;

SELECT get\_call\_minutes\_rest('89001234567');

SELECT \* FROM messages mes

WHERE

mes.customer\_to\_customer\_id IN (SELECT ctc.customer\_to\_customer\_id FROM customer\_to\_customer ctc WHERE ctc.initiator\_phone\_number = '89001234567')

OR mes.customer\_to\_other\_id IN (SELECT cto.customer\_to\_other\_id FROM customer\_to\_other cto WHERE customer\_is\_initiator = TRUE AND customer\_phone\_number = '89001234567');

**Результаты выполнения лабораторной работы:**

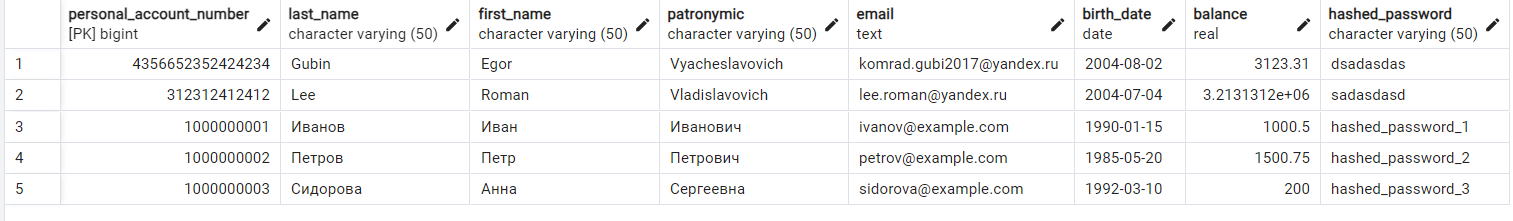
****

Рисунок 2 Запрос на вывод всех клиентов

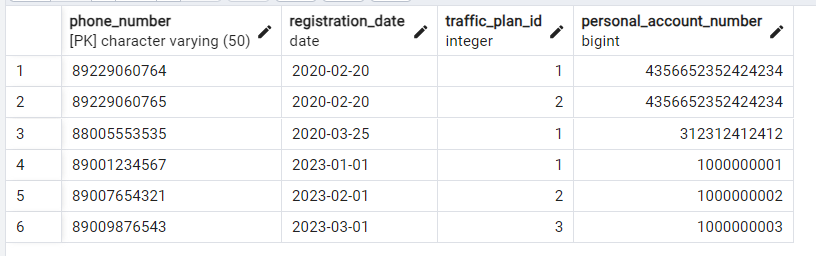
****

Рисунок 3 Запрос на вывод всех номеров телефона

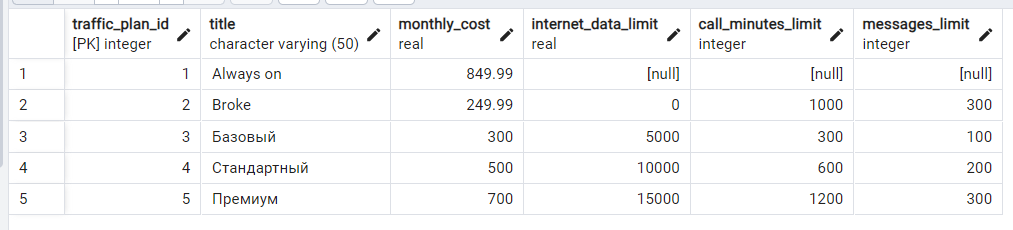
****

Рисунок 4 Запрос на вывод всех тарифных планов

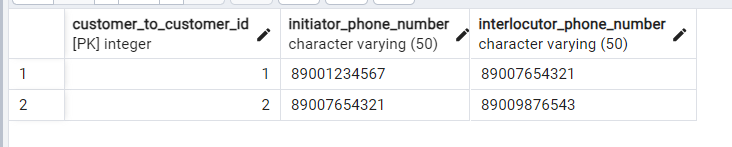
****

Рисунок 5 Запрос на вывод экземпляров формата клиент – клиент

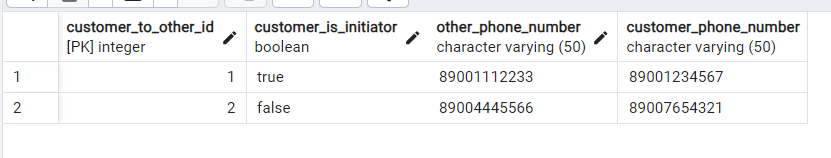
****

Рисунок 6 Запрос на вывод экземпляров клиент - клиент другой компании

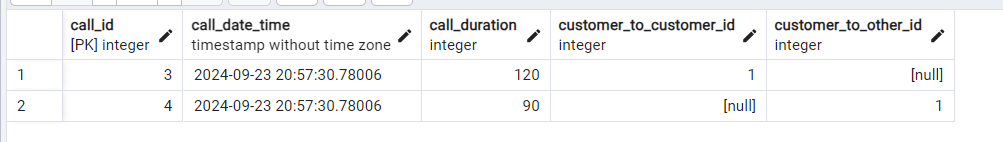
****

Рисунок 7 Запрос на вывод всех звонков

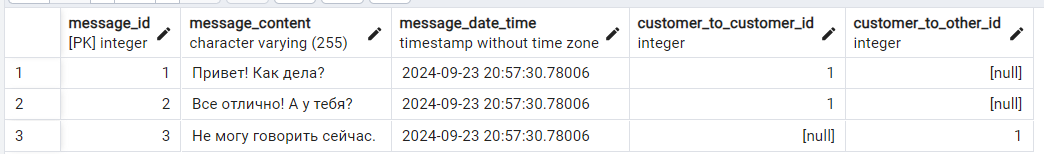


Рисунок 8 Запрос на вывод всех сообщений

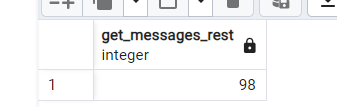


Рисунок 9 Запрос на остаток сообщений по номеру телефона



Рисунок 10 Запрос на остаток минут по звонкам по номеру телефона

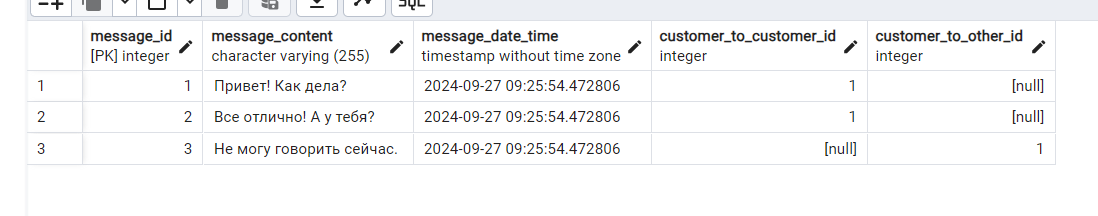


Рисунок 11 Запрос на отправленные сообщения

**Вывод:** в ходе лабораторной работы была спроектирована база данных предметной области телефонная компания, произведена нормализация. Произведено заполнение базы данных.