

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**(ШКОЛА)**

**Департамент программной инженерии и искусственного интеллекта**

ВАЯЙ МИХАИЛ СЕРГЕЕВИЧ

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ

«ОПЛАТА ЗА МЕЖДУГОРОДНИЕ РАЗГОВОРЫ»

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Проектирование и разработка баз данных»

по образовательной программе подготовки бакалавров по направлению

09.03.04 «Программная инженерия», профиль «Программная инженерия»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |  |  | Студент группы Б9121-09.03.04 | | | | | | | |
|  |  |  | | | | Ваяй М.С. | | | |
|  | | | | | | |  |  | (подпись) | | | |  | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | |  |  | Руководитель | | | | | | | |
|  | | | | | | |  |  | старший преподаватель | | | | | | | |
|  | | | | | | |  |  | ученая степень, должность | | | | |  | Краморенко Н.В. | |
|  | | | | | | |  |  | (подпись) | | | | |  | (ФИО) | |
|  | | | | | | | | | | | | | | | | |
| Регистрационный № | | | | | | |  |  | Защищен с оценкой | | | | | | | |
|  | | | | | | |  |  |  | | | | | | | |
|  | | | |  |  | |  |  | « |  | » |  |  | | | 2024 г. |
| (подпись) | | | |  | (ФИО) | |  |  |  | | | | | | | |
| « |  | » |  | | | 2024 г. |  |  |  | | | | | | | |

г. Владивосток

2024

**Оглавление**

[Оглавление 2](#_Toc157782113)

[Введение 3](#_Toc157782114)

[1 Анализ предметной области 4](#_Toc157782115)

[2 Построение концептуальной модели базы данных 5](#_Toc157782116)

[3 Реализация проекта базы данных 9](#_Toc157782117)

[3.1 Создание таблиц 9](#_Toc157782118)

[3.2 Разработка системы запросов в БД 11](#_Toc157782119)

[3.2.1. Запросы на добавление 11](#_Toc157782120)

[3.2.2 Запросы на вызов триггеров 12](#_Toc157782121)

[3.2.3 Запросы на выборку 14](#_Toc157782122)

[3.2.4 Запросы на корректировку 17](#_Toc157782123)

[4 Работа со средой разработки СУБД 18](#_Toc157782124)

[5 Руководство пользователя 22](#_Toc157782125)

[Заключение 26](#_Toc157782126)

[Список литературы 27](#_Toc157782127)

**Введение**

В условиях современной телефонной связи любое предприятие, оказывающее услуги междугородней телефонной связи, нередко сталкивается с необходимость эффективного учета клиентов, тарифов и операций по оплате предоставленных услуг. Это требует системы управления данными, способной оперативно реагировать на изменения в тарифах, обрабатывать актуальную информацию о разговорах и счетах, а также обеспечивать надежное хранение данных о клиентах и услугах.

Объектом исследования является предприятие междугородней телефонной связи (МТС), клиенты и процессы, связанные с учетом тарифов и операций по оплате междугородних разговоров.

Предметом исследования является возможность автоматизации процессов ведения учета клиентов, тарифов и операций по оплате междугородних разговоров, в частности, хранение информации о клиентах, тарифах, а также данные о проведенных разговорах и операциях по оплате.

Задачи:

1. Провести анализ предметной области с целью выделить основные термины предметной области и их смысл, связи между ними.
2. Построить концептуальную модель базы данных.
3. Построить реляционную модель базы данных, преобразовать EER-модель в реляционную модель в СУБД MySQL.
4. Разработать систему запросов в БД.
5. Создать диалоговый интерфейс взаимодействия БД с пользователями.
6. Описать режимы работы пользователя.

**1 Анализ предметной области**

База данных создается для информационного обслуживания клиентов предприятия, осуществляющих междугородние телефонные звонки, и для ведения учета оплаты тарифов, услуг. БД должна содержать данные о клиентах, звонках, городах, тарифах, услугах и их оплате.

В соответствии с предметной областью система строится с учётом следующих особенностей:

1. Технолог вносит информацию о новых клиентах, пользующихся услугами это: номер телефона клиента, ФИО, адрес, дата регистрации.
2. Технолог изменяет цены за пользование услугами, эти данные хранятся в следующем виде: дата, название населенного пункта, с которым осуществляется связь, стоимость одной минуты разговора, льготная стоимость (с 20:00 до 6:00).
3. Оператор МТС фиксирует дату разговора, город, с которым произошел разговора, номер абонента, длительность разговора.
4. Оператор отсылает абоненту квитанцию об оплате.
5. Квитанция хранит в себе следующие данные: номер квитанции, дата выставления, сумму к оплате, дата оплаты, номер разговора (звонка).

**2 Построение концептуальной модели базы данных**

Для создания концептуальной модели необходимо выделить сущности предметной области:

1. **Клиент**. Атрибуты: номер телефона, ФИО, дата регистрации, адрес регистрации.
2. **Звонок**. Атрибуты: код, дата звонка, время звонка, длительность звонка.
3. **Тариф**. Атрибуты: код, стоимость минуты, льготная стоимость минуты, дата тарифа
4. **Город**. Атрибуты: код, название города.
5. **Квитанция**. Атрибуты: код, дата выставления, дата оплаты, сумма к оплате.

Выделенные сущности можно связать следующим образом:

1. Сущности Город и Клиент: не идентифицирующая связь «один ко многим». В одном городе может находиться несколько клиентов. Сущность Клиент хранит код города в качестве атрибута.
2. Сущности Город и Тариф: идентифицирующая связь «один ко многим». Для одного города может быть установлено несколько тарифов. Сущность Тариф хранит код города в качестве атрибута.
3. Сущности Звонок и Квитанция: не идентифицирующая связь «один к одному». Несколько квитанций создается для каждого звонка. Сущность Квитанция хранит код звонка в качестве атрибута.
4. Сущности Клиент и Звонок: не идентифицирующая связь «один ко многим». Каждый клиент может осуществлять несколько звонков. Сущность Звонок хранит номер телефона в качестве атрибута.
5. Сущности Тариф и Звонок: не идентифицирующая связь «один ко многим». По одному тарифу может быть совершено несколько звонков. Сущность Звонок хранит код тарифа в качестве атрибута.

Исходя из выявленных сущностей и связей, построим EER–диаграмму (Рисунок 1).

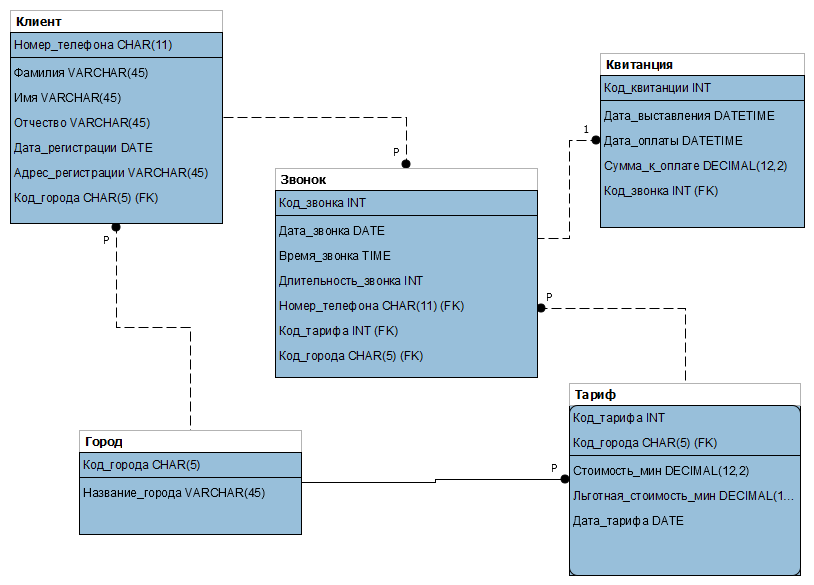


Рисунок 1 - EER–диаграмма предметной области «Оплата за междугородние разговоры»

Установим следующую взаимосвязь между объектов при решении задач ПО.

Потенциальным ключом отношения ГОРОД является поле: название города. Он занимает достаточно много места, и к тому же может меняться. Введем суррогатный первичный ключ Код города.

Таблица 1 – Схема отношения ГОРОД

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код города | Код\_города | CHAR(5) | **Суррогатный первичный ключ** |
| Название города | Название\_города | VARCHAR(45) | Обязательное поле |

Потенциальным ключом отношения ТАРИФ является поле: номер тарифа.

Таблица 2 – Схема отношения ТАРИФ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер тарифа | Код\_тарифа | INT | **Первичный ключ** |
| Код города | Код\_города | CHAR(5) | **Первичный ключ (к таблице Город)** |
| Стоимость минуты | Стоимость\_мин | DECIMAL(12, 2) | Обязательное поле |
| Льготная стоимость | Льготная\_стоимость\_мин | DECIMAL(12, 2) | Обязательное поле |
| Дата тарифа | Дата\_тарифа | DATE | Обязательное поле |

Потенциальным ключом отношения КВИТАНЦИЯ является поле: номер квитанции.

Таблица 3 – Схема отношения КВИТАНЦИЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание | Имя поля | Тип, длина | Примечания |
| Номер квитанции | Код\_квитанции | INT | **Первичный ключ** |
| Дата выставления | Дата\_выставления | DATETIME | Обязательное поле |
| Дата оплаты | Дата\_оплаты | DATETIME | Обязательное поле |
| Сумма к оплате | Сумма\_к\_оплате | DECIMAL(12,2) | Обязательное поле |
| Код звонка | Код\_звонка | INT | Внешний ключ (к таблице Звонок) |

Потенциальным ключом отношения КЛИЕНТ является поле: номер телефона.

Таблица 4 – Схема отношения КЛИЕНТ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание | Имя поля | Тип, длина | Примечания |
| Номер телефона | Номер\_телефона | CHAR(11) | **Первичный ключ** |
| Фамилия | Фамилия | VARCHAR(45) | Обязательное поле |
| Имя | Имя | VARCHAR(45) | Обязательное поле |
| Отчество | Отчество | VARCHAR(45) | Не обязательное поле |
| Дата регистрации | Дата\_регистрации | DATE | Обязательное поле |
| Адрес регистрации | Адрес\_регистрации | VARCHAR(45) | Обязательное поле |
| Код города | Код\_города | CHAR(5) | Внешний ключ (к таблице Город) |

Потенциальным ключом отношения ЗВОНОК является поле: код звонка.

Таблица 5 – Схема отношения ЗВОНОК

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание | Имя поля | Тип, длина | Примечания |
| Номер звонка | Код\_звонка | INT | **Первичный ключ** |
| Дата звонка | Дата\_звонка | DATE | Обязательное поле |
| Время звонка | Время\_звонка | TIME | Обязательное поле |
| Длительность звонка | Длительность\_звонка | INT | Обязательное поле |
| Номер телефона | Номер\_телефона | CHAR(11) | Внешний ключ (к таблице Клиент) |
| Номер тарифа | Код\_тарифа | INT | Внешний ключ (к таблице Тариф) |
| Код города | Код\_города | CHAR(5) | Внешний ключ (к таблице Город) |

Все отношения удовлетворяют требованиям 4НФ.

# **3 Реализация проекта базы данных**

Ниже приведены запросы на создание таблиц на языке SQL в нотации СУБД MySQL.

## **3.1 Создание таблиц**

Для реализации построенной модели данных претендентом является СУБД Access.

1. Отношение Клиент:

CREATE TABLE IF NOT EXISTS mts.`Клиент` (

`Номер\_телефона` CHAR(11) NOT NULL,

`Фамилия` VARCHAR(45) NOT NULL,

`Имя` VARCHAR(45) NOT NULL,

`Отчество` VARCHAR(45) NULL,

`Дата\_регистрации` DATE NOT NULL,

`Адрес\_регистрации` VARCHAR(45) NOT NULL,

`Код\_города` CHAR(5) NOT NULL,

PRIMARY KEY (`Номер\_телефона`),

CONSTRAINT `fk\_Клиент\_Город1`

FOREIGN KEY (`Код\_города`)

REFERENCES `Город` (`Код\_города`)

ON DELETE RESTRICT

ON UPDATE CASCADE);

1. Отношение Тариф:

CREATE TABLE IF NOT EXISTS mts.`Тариф` (

`Код\_тарифа` INT NOT NULL AUTO\_INCREMENT,

`Стоимость\_мин` DECIMAL(12,2) NOT NULL,

`Льготная\_стоимость\_мин` DECIMAL(12,2) NOT NULL,

`Дата\_тарифа` DATE NOT NULL,

`Код\_города` CHAR(5) NOT NULL,

PRIMARY KEY (`Код\_тарифа`, `Код\_города`),

CONSTRAINT `fk\_Тариф\_Город1`

FOREIGN KEY (`Код\_города`)

REFERENCES `Город` (`Код\_города`)

ON DELETE CASCADE

ON UPDATE CASCADE);

1. Отношение Город:

CREATE TABLE IF NOT EXISTS mts.`Город` (

`Код\_города` CHAR(5) NOT NULL,

`Название\_города` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Код\_города`));

1. Отношение Квитанция:

CREATE TABLE IF NOT EXISTS mts.`Квитанция` (

`Код\_квитанции` INT NOT NULL AUTO\_INCREMENT,

`Дата\_выставления` DATETIME NOT NULL,

`Дата\_оплаты` DATETIME NOT NULL,

`Сумма\_к\_оплате` DECIMAL(12,2) NOT NULL,

`Код\_звонка` INT NOT NULL,

PRIMARY KEY (`Код\_квитанции`),

CONSTRAINT `fk\_Квитанция\_Звонок1`

FOREIGN KEY (`Код\_звонка`)

REFERENCES `Звонок` (`Код\_звонка`)

ON DELETE RESTRICT

ON UPDATE CASCADE);

1. Отношение Звонок:

CREATE TABLE IF NOT EXISTS mts.`Звонок` (

`Код\_звонка` INT NOT NULL AUTO\_INCREMENT,

`Дата\_звонка` DATE NOT NULL,

`Время\_звонка` TIME NOT NULL,

`Длительность\_звонка` INT NOT NULL,

`Номер\_телефона` CHAR(11) NOT NULL,

`Код\_тарифа` INT NOT NULL,

`Код\_города` CHAR(5) NOT NULL,

PRIMARY KEY (`Код\_звонка`),

CONSTRAINT `fk\_Звонок\_Клиент1`

FOREIGN KEY (`Номер\_телефона`)

REFERENCES `Клиент` (`Номер\_телефона`)

ON DELETE RESTRICT

ON UPDATE CASCADE,

CONSTRAINT `fk\_Звонок\_Тариф1`

FOREIGN KEY (`Код\_тарифа` , `Код\_города`)

REFERENCES `Тариф` (`Код\_тарифа` , `Код\_города`)

ON DELETE RESTRICT

ON UPDATE CASCADE)

## **3.2 Разработка системы запросов в БД**

## **3.2.1. Запросы на добавление**

Запрос на добавление предназначен для добавления новых записей в существующую таблицу с использованием данных из других источников. Добавляемые записи не удаляются из исходной таблицы. Допустимо добавление записей между таблицами с различной структурой.

1. Добавление данных в таблицу Город:

INSERT INTO mts.`Город`(`Код\_города`, `Название\_города`) VALUES (423, 'Владивосток');

INSERT INTO mts.`Город`(`Код\_города`, `Название\_города`) VALUES (4162, 'Благовещенск(Амурская область)');

1. Добавление данных в таблицу Клиент:

INSERT INTO mts.`Клиент`(`Номер\_телефона`, `Фамилия`, `Имя`, `Отчество`, `Дата\_регистрации`, `Адрес\_регистрации`, `Код\_города`)

VALUES ('12345678901', 'Ваяй', 'Михаил', 'Сергеевич', '2024-01-30', 'ул. Южная, 20', '423');

INSERT INTO mts.`Клиент`(`Номер\_телефона`, `Фамилия`, `Имя`, `Отчество`, `Дата\_регистрации`, `Адрес\_регистрации`, `Код\_города`)

VALUES ('98765432109', 'Иванов', 'Петр', 'Сергеевич', '2023-02-10', 'ул. Маяковского, 10', '4162');

1. Добавление данных в таблицу Тариф:

INSERT INTO mts.`Тариф`(`Стоимость\_мин`, `Льготная\_стоимость\_мин`, `Дата\_тарифа`, `Код\_города`) VALUES (10, 5, '2024-01-20', '423');

INSERT INTO mts.`Тариф`(`Стоимость\_мин`, `Льготная\_стоимость\_мин`, `Дата\_тарифа`, `Код\_города`) VALUES (4, 2, '2023-02-01', '4162');

Запросы на добавление данных в таблицу Квитанция и в таблицу Звонок сформированы с помощью триггеров, которые вызываются после добавления данных во все таблицы, описанные выше.

## **3.2.2 Запросы на вызов триггеров**

Триггер – это особая разновидность хранимых процедур в базе данных. Особенность триггеров заключается в том, что SQL код, написанный в теле триггера, будет исполнен после того, как в базе данных произойдет какое-либо событие. События в базах данных происходят в результате выполнения команд или команд манипуляции данными.

Использование триггеров в данной базе данных «Оплата междугородних телефонных разговор» необходимо для автоматизации заполнения данных после добавления данных в другие таблицы, добавления случайного количества звонков, автоматического составления квитанций.

Триггер на заполнение таблицы Звонок:

DELIMITER //

CREATE TRIGGER mts.AutoInsertCalls

AFTER INSERT ON mts.`Клиент`

FOR EACH ROW

BEGIN

DECLARE i INT DEFAULT 0;

DECLARE random\_date DATE;

-- Генерируем случайную дату звонка, позже даты регистрации клиента

SET random\_date = TIMESTAMPADD(DAY, FLOOR(RAND() \* DATEDIFF(NOW(), NEW.`Дата\_регистрации`)), NEW.`Дата\_регистрации`);

-- Случайное количество звонков от 1 до 5

SET i = FLOOR(RAND() \* 5) + 1;

WHILE i > 0 DO

-- Вставляем случайные данные о звонках для нового клиента

INSERT INTO mts.`Звонок` (`Дата\_звонка`, `Время\_звонка`, `Длительность\_звонка`, `Номер\_телефона`, `Код\_тарифа`, `Код\_города`)

SELECT random\_date, SEC\_TO\_TIME(FLOOR(RAND() \* 86400)), FLOOR(RAND() \* 60) + 1, NEW.`Номер\_телефона`, `Код\_тарифа`, NEW.`Код\_города`

FROM mts.`Тариф`

WHERE `Код\_города` = NEW.`Код\_города`

ORDER BY RAND()

LIMIT 1;

SET i = i - 1;

END WHILE;

END //

DELIMITER ;

Триггер на заполнение таблицы Квитанция:

DELIMITER //

CREATE TRIGGER mts.AutoGenerateInvoice

AFTER INSERT ON mts.`Звонок`

FOR EACH ROW

BEGIN

DECLARE sum\_to\_pay DECIMAL(12,2);

DECLARE end\_time DATETIME;

-- Рассчитываем время завершения звонка

SET end\_time = TIMESTAMP(NEW.`Дата\_звонка`, NEW.`Время\_звонка`) + INTERVAL NEW.`Длительность\_звонка` MINUTE;

-- Рассчитываем сумму к оплате для данного звонка

SELECT

CASE

WHEN TIME(NEW.`Время\_звонка`) >= '20:00' OR TIME(NEW.`Время\_звонка`) < '06:00' THEN NEW.`Длительность\_звонка` \* t.`Льготная\_стоимость\_мин`

ELSE NEW.`Длительность\_звонка` \* t.`Стоимость\_мин`

END INTO sum\_to\_pay

FROM mts.`Тариф` t

WHERE t.`Код\_тарифа` = NEW.`Код\_тарифа`;

-- Вставляем данные в таблицу Квитанция

INSERT INTO mts.`Квитанция` (`Дата\_выставления`, `Дата\_оплаты`, `Сумма\_к\_оплате`, `Код\_звонка`)

VALUES (end\_time, end\_time, sum\_to\_pay, NEW.`Код\_звонка`);

END //

DELIMITER ;

## **3.2.3 Запросы на выборку**

Запросы на выборку – это отбор и обработка данных из таблиц на основании критериев, заданных пользователем, и отображение результатов в виде таблицы.

Найдем клиента с наибольшей суммой к оплате, если есть клиенты с одинаковой наибольшей суммой к оплате, то выводим всех:

SELECT

Клиент.Номер\_телефона,

Клиент.Фамилия,

Клиент.Имя,

Клиент.Отчество,

SUM(Квитанция.Сумма\_к\_оплате) AS Общая\_сумма\_к\_оплате

FROM

mts.Клиент

JOIN

mts.Звонок ON Клиент.Номер\_телефона = Звонок.Номер\_телефона

JOIN

mts.Квитанция ON Звонок.Код\_звонка = Квитанция.Код\_звонка

GROUP BY

Клиент.Номер\_телефона, Клиент.Фамилия, Клиент.Имя, Клиент.Отчество

HAVING

SUM(Квитанция.Сумма\_к\_оплате) = (

SELECT

MAX(сумма)

FROM (

SELECT

SUM(Квитанция.Сумма\_к\_оплате) AS сумма

FROM

mts.Клиент

JOIN

mts.Звонок ON Клиент.Номер\_телефона = Звонок.Номер\_телефона

JOIN

mts.Квитанция ON Звонок.Код\_звонка = Квитанция.Код\_звонка

GROUP BY

Клиент.Номер\_телефона

) AS max\_sums

);

Выведем ФИО клиентов, которые делали звонки с 20:00 по 6:00:

SELECT DISTINCT

Клиент.Фамилия,

Клиент.Имя,

Клиент.Отчество

FROM

mts.Клиент

JOIN

mts.Звонок ON Клиент.Номер\_телефона = Звонок.Номер\_телефона

WHERE

TIME(Звонок.Время\_звонка) >= '20:00' OR TIME(Звонок.Время\_звонка) < '06:00';

Выведем число звонков, длительность которых была больше 5 минут:

SELECT

COUNT(\*) AS Число\_звонков\_длительностью\_больше\_5\_минут

FROM

mts.Звонок

WHERE

Длительность\_звонка > 5;

Найдем тариф, которым чаще всего пользуются клиенты, если есть тарифы с одинаковым числом использований, то выводим все:

SELECT

Код\_тарифа,

COUNT(\*) AS Количество\_использований

FROM

mts.Звонок

GROUP BY

Код\_тарифа

HAVING

COUNT(\*) = (

SELECT

MAX(количество)

FROM (

SELECT

COUNT(\*) AS количество

FROM

mts.Звонок

GROUP BY

Код\_тарифа

) AS max\_counts

);

## **3.2.4 Запросы на корректировку**

Корректирующие запросы (запрос на обновление, удаление, добавление) могут изменять как все записи таблицы, так и определенное их подмножество – это будет зависеть от условия отбор.

Обновим данные клиента:

UPDATE mts.`Клиент`

SET

`Фамилия` = 'Новая фамилия',

`Имя` = 'Новое имя',

`Отчество` = 'Новое отчество',

`Дата\_регистрации` = 'Новая дата регистрации',

`Адрес\_регистрации` = 'Новый адрес регистрации',

`Код\_города` = 'Новый Город'

WHERE

`Номер\_телефона` = 'ваш\_номер\_телефона\_клиента';

Обновим данные тарифа:

UPDATE mts.`Тариф`

SET `Стоимость\_мин` = новая\_стоимость\_мин,

`Льготная\_стоимость\_мин` = новая\_льготная\_стоимость\_мин

WHERE `Код\_тарифа` = ваш\_код\_тарифа\_здесь;

Добавим необязательного поля для хранения возраста клиента:

ALTER TABLE mts.`Клиент`

ADD COLUMN `Возраст` INT;

**4 Работа со средой разработки СУБД**

Для реализации построенной модели данных претендентом является СУБД Access.

СУБД Access разрабатывается и поддерживается с 1992 года, что делается её проверенным инструментом для создания баз данных.

Access относится к СУБД реляционного типа. Поддерживает формирование запросов на языке SQL.

В Access встроен мощный инструмент ODBC, позволяющий СУБД связываться с другими базами данных, например, Oracle.

В СУБД Access встроена поддержка языка Visual Basic for Application и создание макросов на нем, что позволяет средствами СУБД автоматизировать запросы к базе данных, создать пользовательский интерфейс для работы с данными.

Элементы базы данных:

1. Таблицы. Информация в реляционной базе данных Access представлена в виде таблиц, где столбцы представляют собой атрибуты, а строки – экземпляры объектов таблицы.
2. Запрос. Простроенный на языке SQL запрос – инструмент для выбора по некоторому критерию подмножества из множества записей в таблице.
3. Формы. СУБД Access позволяет создавать экранные формы для удобного представления данных для пользователя. Через формы можно отображать данные из таблиц и создавать запросы.
4. Отчеты. Отчеты используются для форматирования, сведения и показа данных. В них всегда будет отображена текущая информация базы данных. Отчеты обычно форматируются таким образом, чтобы их можно было распечатать, но их также можно просматривать на экране, экспортировать в другие программы или выкладывать в сообщения электронной почты.
5. Макросы. Это нечто вроде упрощенного языка программирования, с помощью которого можно сделать базу данных более функциональной.

Файлы баз данных Access, начиная с версии 2007 года, сохраняются в формате ACCDB. Особенностями этого формата являются:

1. Многозначные поля – поля подстановки, позволяющие хранить более одного значения для каждой записи.
2. Вложения – тип данных, позволяющий хранить документы и двоичные файлы различных форматов.
3. Шифрование – возможность зашифровать содержимое базы данных поставить защиту паролем.
4. Отслеживание изменений длинных текстовых полей (полей типа данных MEMO).

Для того чтобы связаться с MySQL необходимо:

1. В Access выбрать «Внешние данные» и выбрать подключение через «ODBC Database». (Рисунок 2)

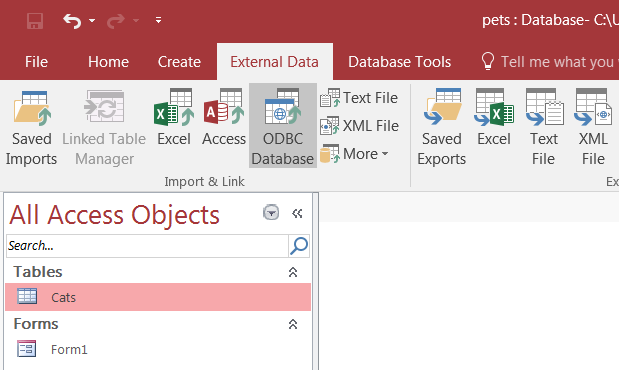


Рисунок 2 - «ODBC Database»

2. В диалоговом окне выберите «Link to the data source by creating a linked table».

3. Затем в диалоговом окне выбора источника данных необходимо соединиться с базой данных (Рисунок 3).

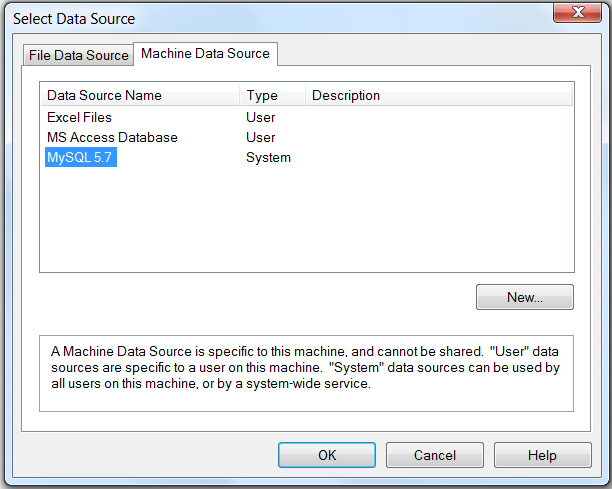


Рисунок 3 - «Select Data Source»

4. MS Access подключается к серверу MySQL и отображает список таблиц, на которые можно ссылаться.

1. После завершения процесса можно приступать к созданию интерфейсов и запросы к связанным таблицам. (Рисунок 4)

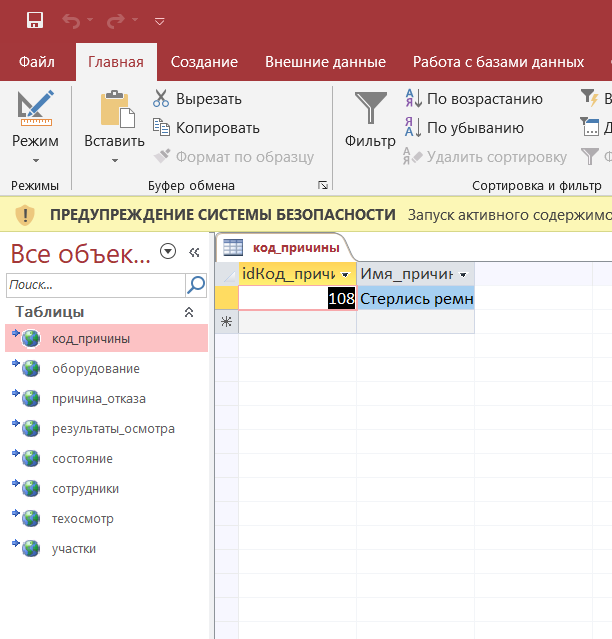


Рисунок 4 – «Внешние данные. Импортированная БД»

Для удобства взаимодействия пользователей с базой данных с помощью MS Access были разработаны некоторые элементы интерфейса:

Главная кнопочная форма - размещает все основные элементы БД;

Отчеты- основаны на запросах пользователя для вывода информации;

Таблицы – просмотр текущих записей каждой таблицы;

Формы – элемент просмотра и добавления записей в базу данных;

Кнопки выхода и добавления записей.

**5 Руководство пользователя**

У пользователя несколько режимов работы: просмотр записей каждой таблицы, внесение изменений в данные и добавление новых записей в каждую таблицу.

В начале пользователь видит Главную кнопочную форму (ГКФ). (Рисунок 5)

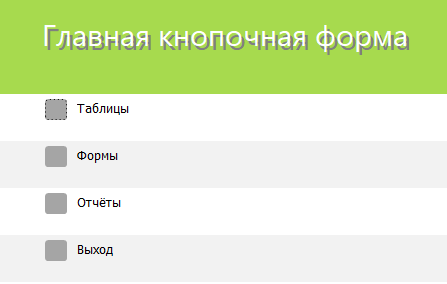


Рисунок 5 – Главная кнопочная форма»

Для просмотра содержимого таблиц ему необходимо перейти на вкладку «Таблицы». (Рисунок 6)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – «Таблицы»

Открывается окно с теми таблица, которые можно просмотреть. Далее при клике по названию одной из таблиц, пользователь переходит к непосредственному содержанию таблицы. (Рисунок 7)

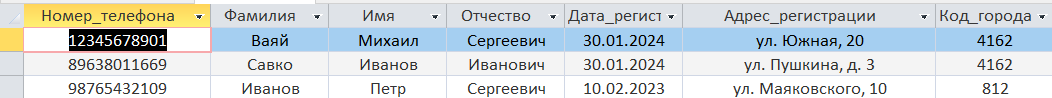


Рисунок 7 – Содержание таблицы «Клиенты»

Для возвращения к главному меню необходимо закрыть таблицу и нажать кнопку «Назад». (Рисунок 8)

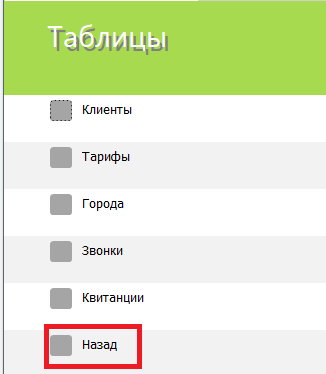


Рисунок 8 – Кнопка «Назад»

Далее при нажатии на кнопку «Формы» пользователю открывается окно с таблицами, в которых он может изменять или добавлять новые записи. (Рисунок 9)

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 9 – Вкладка «Формы»

Если пользователь хочет внести новую информацию, то он может выбрать нужную ему таблицу, откроется форма с возможностью добавления новых записей и редактирования имеющихся. (Рисунок 10)

Изображение выглядит как текст, снимок экрана, число, Шрифт

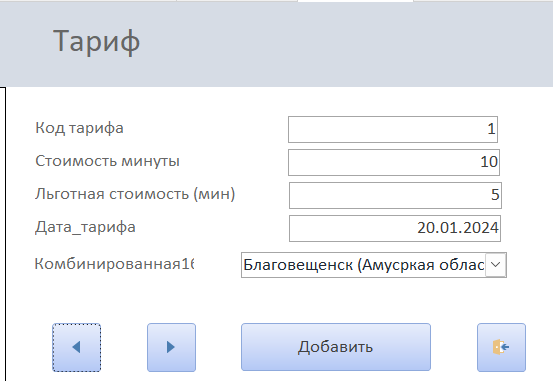
Автоматически созданное описание

Рисунок 10 – Формы для таблиц «Клиенты», «Тарифы»

Для перехода между записями таблицы используются соответствующие кнопки переключения. Для добавления нового элемента используется кнопка «Добавить». (Рисунок 11)

Изображение выглядит как текст, снимок экрана, Шрифт, Прямоугольник

Автоматически созданное описание

Рисунок 11 – Кнопка «Добавить»

Для выхода из формы используется кнопка с заменяющим текст рисунком двери. (Рисунок 12)

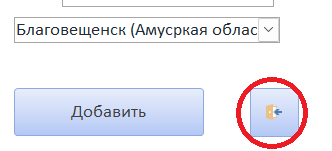


Рисунок 12 – Кнопка «Выход»

При нажатии на кнопку «Добавить» перед пользователем открывается пустая форма с полями для ввода. Для выхода из формы применяется кнопка справа.

Пользователь может добавить данные в таблицы: Клиенты, Тарифы. Остальные таблицы заполняются в соответствии с логикой триггеров.

Вкладка «Отчёты», на которую пользователь также может перейти по клику, необходима для компактного и систематического отображения запросов пользователя. Например, если необходимо просмотреть всех клиентов, то нужно перейти по вкладке «Клиенты МТС». (Рисунок 13)

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 13 – Вкладка «Отчёты»

После перед пользователем открывается отчет с соответствующими данными. (Рисунок 14)

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 14 – Отчёт «Клиенты МТС»

Аналогично отображаются и другие отчеты. При нажатии на кнопку «Назад» мы возвращаемся на ГКФ.

# **Заключение**

В ходе курсовой работы:

1. Проведён анализ предметной области с целью выделить основные термины предметной области и их смысл, связи между ними.
2. Построена концептуальная модель базы данных.
3. Построена реляционная модель базы данных, преобразована EER-модель в реляционную модель в СУБД MySQL.
4. Разработана система запросов в БД.
5. Создан диалоговый интерфейс взаимодействия БД с пользователями.
6. Описан режимы работы пользователя.

Была достигнута автоматизация некоторых процессов ведения учета клиентов, тарифов и операций по оплате междугородних разговоров, в частности, хранение информации о клиентах, тарифах, а также данные о проведенных разговорах и операциях по оплате. База данных была спроектирована с расчетом на расширение и дополнение, следовательно, в дальнейшем можно автоматизировать некоторые другие процессы данной предметной области.

# **Список литературы**

1. Основные сведения о базах данных [Электронный ресурс]. - Режим доступа: [https://support.office.com/ru-ru/article/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5%D1%81%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BE-%D0%B1%D0%B0%D0%B7%D0%B0%D1%85-D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-a849ac16-07c7-4a31-9948-](https://support.office.com/ru-ru/article/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5-%D1%81%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BE-%D0%B1%D0%B0%D0%B7%D0%B0%D1%85-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-a849ac16-07c7-4a31-9948-3c8c94a7c204) [3c8c94a7c204](https://support.office.com/ru-ru/article/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5-%D1%81%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BE-%D0%B1%D0%B0%D0%B7%D0%B0%D1%85-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-a849ac16-07c7-4a31-9948-3c8c94a7c204) (дата обращения 02.02.2024);
2. Спецификации Access [Электронный ресурс]. - Режим доступа: https://support.office.com/[ru-ru/article/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%](https://support.office.com/ru-ru/article/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c) [D0%BA%D0%B0%D1%86%D0%B8%D0%B8-access-0cf3c66f-9cf2-4e32-9568-](https://support.office.com/ru-ru/article/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c) [98c1025bb47c](https://support.office.com/ru-ru/article/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c) (дата обращения 02.02.2024);
3. MySQL официальная документация [Электронный ресурс]. - Режим доступа: [MySQL :: MySQL Documentation](https://dev.mysql.com/doc/) (дата обращения 02.02.2024).