

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**

Мартын Владислав Алексеевич

Проектирование и разработка базы данных «Дом творчества молодежи»

**КУРСОВАЯ РАБОТА**

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 | | | | | | | |
|  |  |  | | | | Мартын В. А. | | | |
|  | | | | | | |  |  | (подпись) | | | |  | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | |  |  | Руководитель | | | | | | | |
|  | | | | | | |  |  | ст. преподаватель | | | | | | | |
|  | | | | | | |  |  | ученая степень, должность | | | | |  | Краморенко Н. В. | |
|  | | | | | | |  |  | (подпись) | | | | |  | (ФИО) | |
|  | | | | | | | | | | | | | | | | |
| Регистрационный № | | | | | | |  |  | Защищен с оценкой | | | | | | | |
|  | | | | | | |  |  |  | | | | | | | |
|  | | | |  |  | |  |  | « |  | » |  |  | | | 2024 г. |
| (подпись) | | | |  | (ФИО) | |  |  |  | | | | | | | |
| « |  | » |  | | | 2024 г. |  |  |  | | | | | | | |

г. Владивосток

2024

**Оглавление**

[Введение 3](#_Toc163293876)

[1 Анализ предметной области 4](#_Toc163293877)

[2 Построение концептуальной модели базы данных 5](#_Toc163293878)

[2.1 Взаимосвязь объектов при решении задач ПО 6](#_Toc163293879)

[2.2 Концептуальная модель данных 12](#_Toc163293880)

[3 Реализация проекта базы данных 14](#_Toc163293881)

[3.1 Создание таблиц 14](#_Toc163293882)

[3.2 Разработка системы запросов в БД 19](#_Toc163293883)

[4 Работа со средой разработки СУБД 22](#_Toc163293887)

[5 Руководство пользователя 24](#_Toc163293888)

[Заключение 30](#_Toc163293889)

[Список используемых источников 31](#_Toc163293890)

**Введение**

В современном обществе вопросы развития и поддержки молодежи занимают центральное место. Одним из ключевых аспектов этого развития является создание и функционирование учреждений, специализирующихся на творческом развитии. В этом контексте особую роль играют дома творчества молодежи - уникальные пространства, где юные таланты находят возможности для самовыражения, творческого роста и профессионального развития.

Объектом исследования является Дом творчества молодежи.

Предметом исследования является возможность автоматизации процессов ведения учета информации о детях, посещающих кружки детского творчества, а также о преподавателях, ведущих занятия в Доме творчества молодежи.

Цель: проектирование и разработка базы данных для эффективного учета и управления информацией в Доме творчества молодежи.

Задачи:

1. Провести анализ предметной области;
2. Построить концептуальную модели базы данных;
3. Построить реляционную модели 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 создаётся для информационного обслуживания руководства организации, сотрудников и для ведения учета детей и групп творчества. БД должна содержать данные о преподавателях, детях, студиях и групп творчества.

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

* каждый преподаватель может вести занятия в нескольких группах;
* каждый преподаватель может вести занятия только по своей специализации;
* каждый преподаватель может владеть несколькими специализациями;
* каждый кружок или студия имеет несколько детских групп;
* каждая студия или кружок имеет номер и название;
* каждая группа имеет номер и название;
* каждый ребенок может переходить из группы в группу.

**2 Построение концептуальной модели базы данных**

Для создания концептуальной модели необходимо выделить сущности предметной области:

1. **Семейные положения.** Атрибуты: Код положения, Название положения.
2. **Населенные пункты.** Атрибуты: Код пункта, Название пункта.
3. **Преподаватели.** Атрибуты: Номер паспорта, Фамилия, Имя отчество, Дата рождения, Пол, Семейное положение, Образование, Адрес, Код пункта, Номер телефона.
4. **Специализации преподавателей.** Атрибуты: Номер паспорта, Код специализации.
5. **Специализации.** Атрибуты: Код специализации, Название специализации.
6. **Группы.** Атрибуты: Номер группы, Название группы, Номер паспорта преподавателя, Номер студии.
7. **Переходы.** Атрибуты: Номер свидетельства, Дата начала, Из группы, Дата окончания, В группу.
8. **Родители.** Атрибуты: Номер паспорта, Номер свидетельства, Номер телефона, Адрес.
9. **Образование.** Атрибуты: Код образования, Название образования.
10. **Учебные заведения.** Атрибуты: Код заведения, Название заведения, Тип заведения.
11. **Студии.** Атрибуты: Номер студии, Название студии.
12. **Дети.** Атрибуты: Номер свидетельства, Фамилия, Имя отчество, Дата рождения, Адрес, Номер телефона.
13. **Школьники.** Атрибуты: Номер свидетельства, Номер заведения, Класс.

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

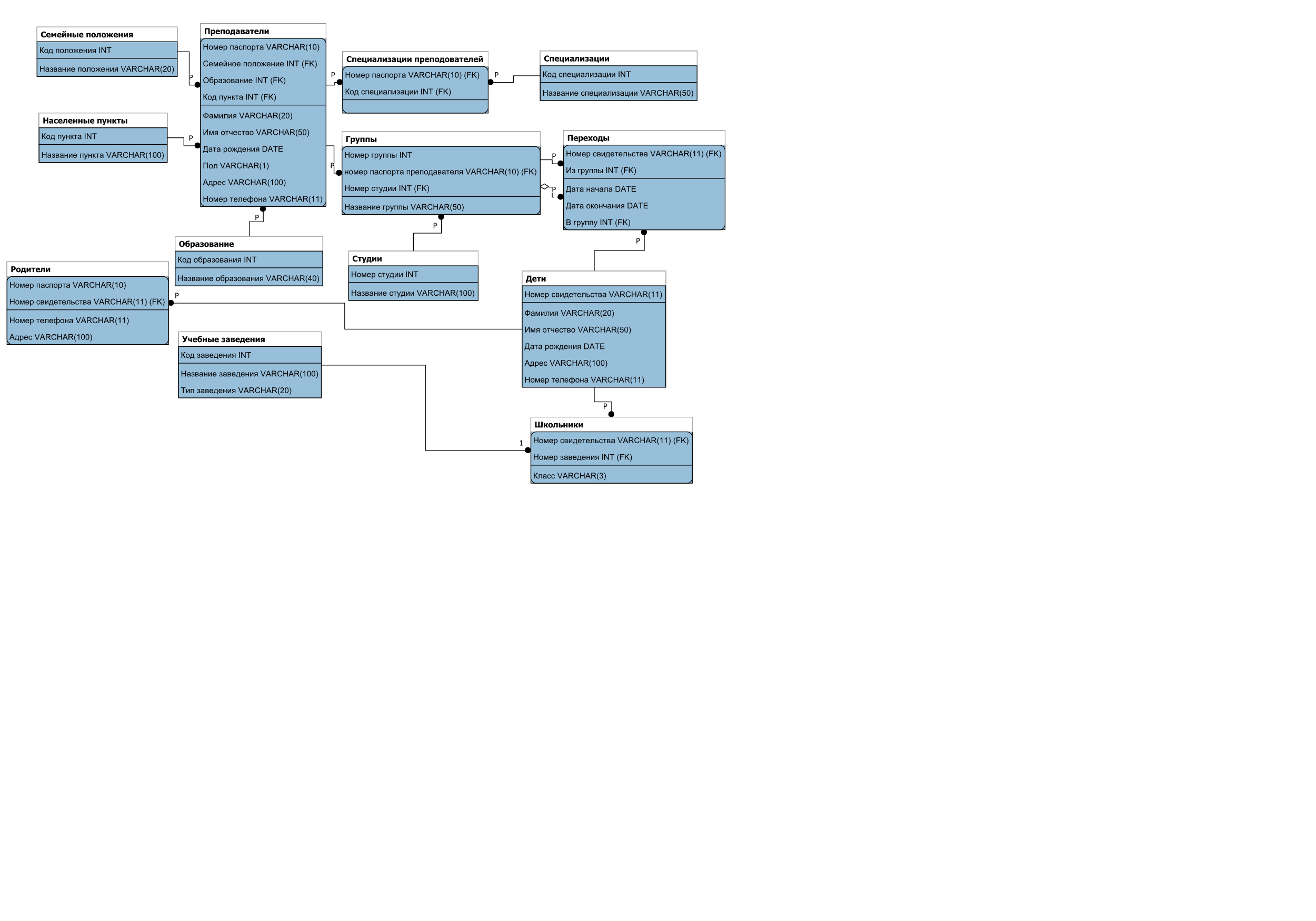


Рисунок 1 - EER–диаграмма предметной области «Дом творчества молодежи»

**2.1 Взаимосвязь объектов при решении задач ПО**

Потенциальным ключом отношения СЕМЕЙНЫЕ ПОЛОЖЕНИЯ (Таблица 1) является поле: Название положения. Он занимает много места, поэтому введем суррогатный первичный ключ Код положения.

Таблица 1 – Схема отношения СЕМЕЙНЫЕ ПОЛОЖЕНИЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код положения | Код положения | INT | **суррогатный первичный ключ** |
| Название положения | Название положения | VARCHAR(20) | Обязательное поле |

Потенциальным ключом отношения НАСЕЛЕННЫЕ ПУНКТЫ (Таблица 2) является поле: Название пункта. Он занимает много места, поэтому введем суррогатный первичный ключ Код пункта.

Таблица 2 – Схема отношения НАСЕЛЕННЫЕ ПУНКТЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Код пункта | Код пункта | INT | **суррогатный первичный ключ** |
| Название пункта | Название пункта | VARCHAR(100) | Обязательное поле |

Потенциальными ключами отношения ПРЕПОДАВАТЕЛИ (Таблица 3) являются поля: Номер паспорта, фамилия, имя отчество, дата рождения, пол, семейное положение, образование, адрес, код пункта, номер телефона. В качестве ключа возьмем номер паспорта как уникальное значение.

Таблица 3 – Схема отношения ПРЕПОДАВАТЕЛИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер паспорта | Номер паспорта | VARCHAR(10) | **первичный ключ** |
| фамилия | фамилия | VARCHAR(20) | Обязательное поле |
| Имя отчество | Имя отчество | VARCHAR(50) | Обязательное поле |
| Дата рождения | Дата рождения | DATE | Обязательное поле |
| Пол | Пол | VARCHAR(1) | Обязательное поле |
| Семейное положение | Семейное положение | INT | Обязательное поле |
| Образование | Образование | INT | Обязательное поле |

*Окончание таблицы 3*

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Адрес | Адрес | VARCHAR(100) | Обязательное поле |
| Код пункта | Код пункта | INT | Обязательное поле |
| Номер телефона | Номер телефона | VARCHAR(11) | Обязательное поле |

Потенциальными ключами отношения СПЕЦИАЛИЗАЦИИ ПРЕПОДОВАТЕЛЕЙ (Таблица 4) являются поля: номер паспорта, код специализации. В качестве ключа примем составной ключ: номер паспорта-код специализации.

Таблица 4 – Схема отношения СПЕЦИАЛИЗАЦИИ ПРЕПОДОВАТЕЛЕЙ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер паспорта | Номер паспорта | VARCHAR(10) | **первичный ключ** |
| Код специализации | Код специализации | INT | **Первичный ключ** |

Потенциальным ключом отношения СПЕЦИАЛИЗАЦИИ (Таблица 5) является поле: название специализации. Введем суррогатный ключ: код специализации.

Таблица 5 – Схема отношения СПЕЦИАЛИЗАЦИИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| код специализации | код специализации | INT | **первичный ключ** |
| название специализации | название специализации | VARCHAR(50) | Обязательное поле |

Потенциальными ключами отношения ГРУППЫ (Таблица 6) являются поля: Название группы, номер паспорта преподавателя, номер студии. Введем суррогатный ключ: номер группы.

Таблица 6 – Схема отношения ГРУППЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| номер группы | номер группы | INT | **Суррогатный первичный ключ** |
| Название группы | Название группы | VARCHAR(50) | Обязательное поле |
| номер паспорта преподавателя | номер паспорта преподавателя | VARCHAR(10) | Обязательное поле |
| номер студии | номер студии | INT | Обязательное поле |

Потенциальными ключами отношения ПЕРЕХОДЫ (Таблица 7) являются поля: Номер свидетельства, дата начала, из группы, дата окончания, в группу. Выберем ключом – номер свидетельства.

Таблица 7 – Схема ПЕРЕХОДЫ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер свидетельства | Номер свидетельства | VARCHAR(11) | **Первичный ключ** |
| дата начала | дата начала | DATE | Обязательное поле |
| из группы | из группы | INT | Обязательное поле |
| дата окончания | дата окончания | DATE | Обязательное поле |
| в группу | в группу | INT | Обязательное поле |

Потенциальными ключами отношения РОДИТЕЛИ (Таблица 8) являются поля: номер паспорта, номер свидетельства, номер телефона, адрес. Выберем ключом – номер паспорта.

Таблица 8 – Схема РОДИТЕЛИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| номер паспорта | номер паспорта | VARCHAR(10) | **Первичный ключ** |
| номер свидетельства | номер свидетельства | VARCHAR(11) | Обязательное поле |
| номер телефона | номер телефона | VARCHAR(11) | Обязательное поле |
| адрес | адрес | VARCHAR(100) | Обязательное поле |

Потенциальным ключом отношения ОБРАЗОВАНИЕ (Таблица 9) является поле: название образования. Введем суррогатный ключ – код образования.

Таблица 9 – Схема ОБРАЗОВАНИЕ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| код образования | код образования | INT | **Суррогатный первичный ключ** |
| название образования | название образования | VARCHAR(40) | Обязательное поле |

Потенциальными ключами отношения УЧЕБНЫЕ ЗАВЕДЕНИЯ (Таблица 10) являются поля: название заведения, тип заведения. Введем суррогатный ключ – код заведения.

Таблица 10 – Схема УЧЕБНЫЕ ЗАВЕДЕНИЯ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| код заведения | код заведения | INT | **Суррогатный первичный ключ** |

*Окончание таблицы 10*

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| название заведения | название заведения | VARCHAR(100) | Обязательное поле |
| тип заведения | тип заведения | VARCHAR(20) | Обязательное поле |

Потенциальным ключом отношения СТУДИИ (Таблица 11) является поле: название студии. Введем суррогатный ключ – код студии.

Таблица 11 – Схема СТУДИИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| код студии | код студии | INT | **Суррогатный первичный ключ** |
| название студии | название студии | VARCHAR(100) | Обязательное поле |

Потенциальными ключами отношения ДЕТИ (Таблица 12) являются поля: номер свидетельства, фамилия, имя отчество, дата рождения, адрес, номер телефона. Выберем в качестве ключа – номер свидетельства.

Таблица 12 – Схема ДЕТИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| номер свидетельства | номер свидетельства | VARCHAR(11) | **Первичный ключ** |
| фамилия | фамилия | VARCHAR(20) | Обязательное поле |
| имя отчество | имя отчество | VARCHAR(50) | Обязательное поле |
| дата рождения | дата рождения | DATE | Обязательное поле |

*Окончание таблицы 12*

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| адрес | адрес | VARCHAR(100) | Обязательное поле |
| номер телефона | номер телефона | VARCHAR(11) | Обязательное поле |

Потенциальными ключами отношения ШКОЛЬНИКИ (Таблица 13) являются поля: номер свидетельства, номер заведения, класс. Выберем в качестве ключа – номер свидетельства.

Таблица 13 – Схема ШКОЛЬНИКИ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| номер свидетельства | номер свидетельства | VARCHAR(11) | **Первичный ключ** |
| номер заведения | номер заведения | INT | Обязательное поле |
| класс | класс | VARCHAR(3) | Обязательное поле |

**2.2 Концептуальная модель данных**

1НФ. Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы. В отношениях каждое поле содержит только одно атомарное значение, следовательно, отношения находятся в I НФ.

2НФ. 2НФ предполагает, что каждый неключевой атрибут должен полностью зависеть от всего первичного ключа, а не от его части. Составные ключи вынесены в отдельные таблицы, следовательно, отношения находятся в I НФ.

3НФ. Во всех отношениях отсутствуют транзитивные зависимости для каждого не ключевого атрибута по отношению к ключевому, следовательно, можно сказать, что отношения находится в 3НФ.

4НФ. Отношение ПРЕПОДАВАТЕЛИ, РОДИТЕЛИ и ДЕТИ нарушают 4НФ, т.к. не всякий телефон привязан к конкретному адресу. Но выделять Телефоны и адреса в отдельное отношение не стоит, т.к. эти сведения носят справочный характер и не требуется их автоматическая обработка.

Полученные отношения соответствуют 4 нормальной форме. Схема базы данных после нормализации соответствует схеме до нормализации представленной на рисунке 1.

**3 Реализация проекта базы данных**

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

**3.1 Создание таблиц**

1. Отношение Семейные положения

CREATE TABLE IF NOT EXISTS `Center`.`Семейные положения` (

`Код положения` INT NOT NULL,

`Название положения` VARCHAR(20) NOT NULL,

PRIMARY KEY (`Код положения`))

ENGINE = InnoDB;

1. Отношение Образование

CREATE TABLE IF NOT EXISTS `Center`.`Образование` (

`Код образования` INT NOT NULL,

`Название образования` VARCHAR(40) NOT NULL,

PRIMARY KEY (`Код образования`))

ENGINE = InnoDB;

1. Отношение Населенные пункты

CREATE TABLE IF NOT EXISTS `Center`.`Населенные пункты` (

`Код пункта` INT NOT NULL,

`Название пункта` VARCHAR(100) NOT NULL,

PRIMARY KEY (`Код пункта`))

ENGINE = InnoDB;

1. Отношение Преподаватели

CREATE TABLE IF NOT EXISTS `Center`.`Преподаватели` (

`Номер паспорта` VARCHAR(10) NOT NULL,

`Фамилия` VARCHAR(20) NOT NULL,

`Имя отчество` VARCHAR(50) NOT NULL,

`Дата рождения` DATE NOT NULL,

`Пол` VARCHAR(1) NOT NULL,

`Семейное положение` INT NOT NULL,

`Образование` INT NOT NULL,

`Адрес` VARCHAR(100) NOT NULL,

`Код пункта` INT NOT NULL,

`Номер телефона` VARCHAR(11) NOT NULL,

PRIMARY KEY (`Номер паспорта`, `Семейное положение`, `Образование`, `Код пункта`),

INDEX `FK\_Status\_idx` (`Семейное положение` ASC) VISIBLE,

INDEX `FK\_Degree\_idx` (`Образование` ASC) VISIBLE,

INDEX `FK\_point\_idx` (`Код пункта` ASC) VISIBLE,

CONSTRAINT `FK\_Status`

FOREIGN KEY (`Семейное положение`)

REFERENCES `Center`.`Семейные положения` (`Код положения`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_Degree`

FOREIGN KEY (`Образование`)

REFERENCES `Center`.`Образование` (`Код образования`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_point`

FOREIGN KEY (`Код пункта`)

REFERENCES `Center`.`Населенные пункты` (`Код пункта`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

1. Отношение Дети

CREATE TABLE IF NOT EXISTS `Center`.`Дети` (

`Номер свидетельства` VARCHAR(11) NOT NULL,

`Фамилия` VARCHAR(20) NOT NULL,

`Имя отчество` VARCHAR(50) NOT NULL,

`Дата рождения` DATE NOT NULL,

`Адрес` VARCHAR(100) NOT NULL,

`Номер телефона` VARCHAR(11) NOT NULL,

PRIMARY KEY (`Номер свидетельства`))

ENGINE = InnoDB;

1. Отношение Учебные заведения

CREATE TABLE IF NOT EXISTS `Center`.`Учебные заведения` (

`Код заведения` INT NOT NULL,

`Название заведения` VARCHAR(100) NOT NULL,

`Тип заведения` VARCHAR(20) NOT NULL,

PRIMARY KEY (`Код заведения`))

ENGINE = InnoDB;

1. Отношение Школьники

CREATE TABLE IF NOT EXISTS `Center`.`Школьники` (

`Номер свидетельства` VARCHAR(11) NOT NULL,

`Номер заведения` INT NOT NULL,

`Класс` VARCHAR(3) NOT NULL,

PRIMARY KEY (`Номер свидетельства`, `Номер заведения`),

INDEX `FK\_emb\_idx` (`Номер заведения` ASC) VISIBLE,

CONSTRAINT `FK\_student`

FOREIGN KEY (`Номер свидетельства`)

REFERENCES `Center`.`Дети` (`Номер свидетельства`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `FK\_emb`

FOREIGN KEY (`Номер заведения`)

REFERENCES `Center`.`Учебные заведения` (`Код заведения`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

1. Отношение Студии

CREATE TABLE IF NOT EXISTS `Center`.`Студии` (

`Номер студии` INT NOT NULL,

`Название студии` VARCHAR(100) NOT NULL,

PRIMARY KEY (`Номер студии`))

ENGINE = InnoDB;

1. Отношение Группы

CREATE TABLE IF NOT EXISTS `Center`.`Группы` (

`Номер группы` INT NOT NULL,

`Название группы` VARCHAR(50) NOT NULL,

`номер паспорта преподавателя` VARCHAR(10) NOT NULL,

`Номер студии` INT NOT NULL,

PRIMARY KEY (`Номер группы`, `номер паспорта преподавателя`, `Номер студии`),

INDEX `FK\_Tech\_idx` (`номер паспорта преподавателя` ASC) VISIBLE,

INDEX `FK\_Studio\_idx` (`Номер студии` ASC) VISIBLE,

CONSTRAINT `FK\_Tech`

FOREIGN KEY (`номер паспорта преподавателя`)

REFERENCES `Center`.`Преподаватели` (`Номер паспорта`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_Studio`

FOREIGN KEY (`Номер студии`)

REFERENCES `Center`.`Студии` (`Номер студии`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

1. Отношение Переходы

CREATE TABLE IF NOT EXISTS `Center`.`Переходы` (

`Номер свидетельства` VARCHAR(11) NOT NULL,

`Дата начала` DATE NOT NULL,

`Из группы` INT NOT NULL,

`Дата окончания` DATE NULL,

`В группу` INT NULL,

PRIMARY KEY (`Номер свидетельства`, `Из группы`),

INDEX `FK\_From\_idx` (`Из группы` ASC) VISIBLE,

INDEX `FK\_To\_idx` (`В группу` ASC) VISIBLE,

CONSTRAINT `FK\_Child`

FOREIGN KEY (`Номер свидетельства`)

REFERENCES `Center`.`Дети` (`Номер свидетельства`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_From`

FOREIGN KEY (`Из группы`)

REFERENCES `Center`.`Группы` (`Номер группы`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_To`

FOREIGN KEY (`В группу`)

REFERENCES `Center`.`Группы` (`Номер группы`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

1. Отношение Специализации

CREATE TABLE IF NOT EXISTS `Center`.`Специализации` (

`Код специализации` INT NOT NULL,

`Название специализации` VARCHAR(50) NOT NULL,

PRIMARY KEY (`Код специализации`))

ENGINE = InnoDB;

1. Отношение Специализации преподователей

CREATE TABLE IF NOT EXISTS `Center`.`Специализации преподователей` (

`Номер паспорта` VARCHAR(10) NOT NULL,

`Код специализации` INT NOT NULL,

PRIMARY KEY (`Номер паспорта`, `Код специализации`),

INDEX `FK\_spec\_idx` (`Код специализации` ASC) VISIBLE,

CONSTRAINT `FK\_Spec`

FOREIGN KEY (`Код специализации`)

REFERENCES `Center`.`Специализации` (`Код специализации`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `FK\_Teach`

FOREIGN KEY (`Номер паспорта`)

REFERENCES `Center`.`Преподаватели` (`Номер паспорта`)

ON DELETE RESTRICT

ON UPDATE RESTRICT)

ENGINE = InnoDB;

1. Отношение Родители

CREATE TABLE IF NOT EXISTS `Center`.`Родители` (

`Номер паспорта` VARCHAR(10) NOT NULL,

`Номер свидетельства` VARCHAR(11) NOT NULL,

`Номер телефона` VARCHAR(11) NOT NULL,

`Адрес` VARCHAR(100) NOT NULL,

PRIMARY KEY (`Номер паспорта`, `Номер свидетельства`),

INDEX `FK\_child\_idx` (`Номер свидетельства` ASC) VISIBLE,

CONSTRAINT `FK\_child\_doc`

FOREIGN KEY (`Номер свидетельства`)

REFERENCES `Center`.`Дети` (`Номер свидетельства`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

**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.

Добавление данных в таблицу Населенные пункты:

INSERT INTO [населенные пункты] ( [Код пункта], [Название пункта] )

VALUES ([1], [Абаза]);

Добавление данных в таблицу Образование:

INSERT INTO образование ( [Код образования], [Название образования] )

VALUES ([1], [высшее]);

Добавление данных в таблицу Семейные положения:

INSERT INTO [семейные положения] ( [Код положения], [Название положения] )

VALUES ([1], [в браке]);

Добавление данных в таблицу Специализации:

INSERT INTO специализации ( [Код специализации], [Название специализации] )

VALUES ([1], [танцы]);

Добавление данных в таблицу Студии:

INSERT INTO студии ( [Номер студии], [Название студии] )

VALUES ([1], [радуга]);

**3.2.2 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 все группы преподавателя с номером паспорта 11111111:

SELECT группы.[Номер группы], группы.[Название группы], преподаватели.[Фамилия], преподаватели.[Имя отчество]

FROM группы INNER JOIN преподаватели ON группы.[номер паспорта преподавателя]=преподаватели.[Номер паспорта]

WHERE группы.[номер паспорта преподавателя]='11111111';

Найдем переходы детей из групп за 2024 год:

SELECT переходы.[Номер свидетельства], переходы.[Дата начала], переходы.[Из группы], переходы.[Дата окончания], переходы.[В группу]

FROM переходы

WHERE Year(переходы.[Дата начала])= 2024;

Найдем школьников из 3А класса:

SELECT дети.[Фамилия], дети.[Имя отчество], школьники.[Класс]

FROM школьники INNER JOIN дети ON школьники.[Номер свидетельства] = дети.[Номер свидетельства]

WHERE школьники.[Класс] = '3А';

**3.2.3 Запросы на корректировку**

Корректирующие запросы могут изменять как все записи таблицы, так и определенное их подмножество – это будет зависеть от условия отбора.

Изменим класс школьников:

UPDATE школьники SET школьники.Класс = '4А'

WHERE школьники.[Класс] = '3А';

Удалим специализацию преподователя:

DELETE [специализации преподователей].\*

FROM [специализации преподователей]

WHERE [специализации преподователей].[Код специализации]=1;

**4 Работа со средой разработки СУБД**

Для реализации построенной модели данных используем СУБД Access.

Access относится к СУБД реляционного типа. Поддерживает формирование запросов на языке SQL. В Access встроен ODBC, позволяющий СУБД связываться с другими базами данных. Также в 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. Построенный на языке SQL запрос – инструмент для выбора по некоторому критерию подмножества из множества записей в таблице;
3. Формы. СУБД Access позволяет создавать формы для удобного представления данных для пользователя;
4. Отчеты. Отчеты используются для форматирования, сведения и показа данных. В них всегда будет отображена текущая информация базы данных;
5. Макросы. Это нечто вроде упрощенного языка программирования, с помощью которого можно сделать базу данных более функциональной.

Для того чтобы связаться с MySQL необходимо:

1. В Access выбрать «Внешние данные» и выбрать подключение через «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;
4. MS Access подключается к серверу Myqsl и отображает список таблиц, на которые можно ссылаться;
5. После завершения процесса можно приступать к созданию интерфейсов и запросы к связанным таблицам.

**5 Руководство пользователя**

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

Вначале пользователь видит ГКФ (рисунок 2) :

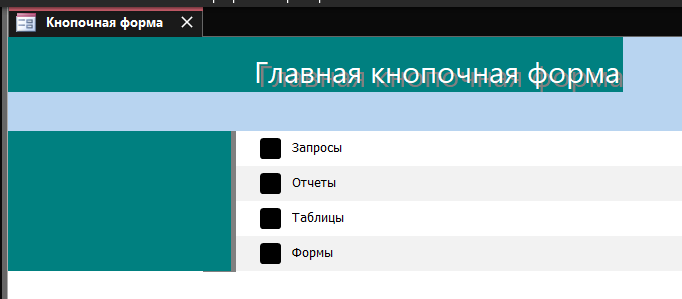


Рисунок 2 - Главная кнопочная форма

Для просмотра содержимого таблиц ему необходимо перейти на вкладку «таблицы» (рисунок 3).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 3 – Таблицы

Открывается окно с теми таблицами которые можно просмотреть.

Далее при клике по названию одной из таблиц, пользователь переходит к непосредственному содержанию таблицы.

Для возвращения к главному меню необходимо закрыть таблицу и нажать на кнопку «назад».

Далее при нажатии на кнопку «формы» (рисунок 4) пользователю открывается окно с таблицами в которых он может изменять данные или добавлять новые записи.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 4 – Формы

Для перехода между записями таблицы используются соответствующие кнопки. Для добавления нового элемента существует кнопка «Добавить». Для выхода из формы служит кнопка с замещающим текст рисунком.

На каждой форме пользователь видит поля и текстовые сообщения для ввода информации кнопки: перейти на предыдущую и следующую запись, добавить запись в таблицу, вывести таблицу с информацией и выход из формы.

Форма «группы» (рисунок 5) имеет вспомогательные таблицы: преподаватели, студии, для понимания заполняемой информации.

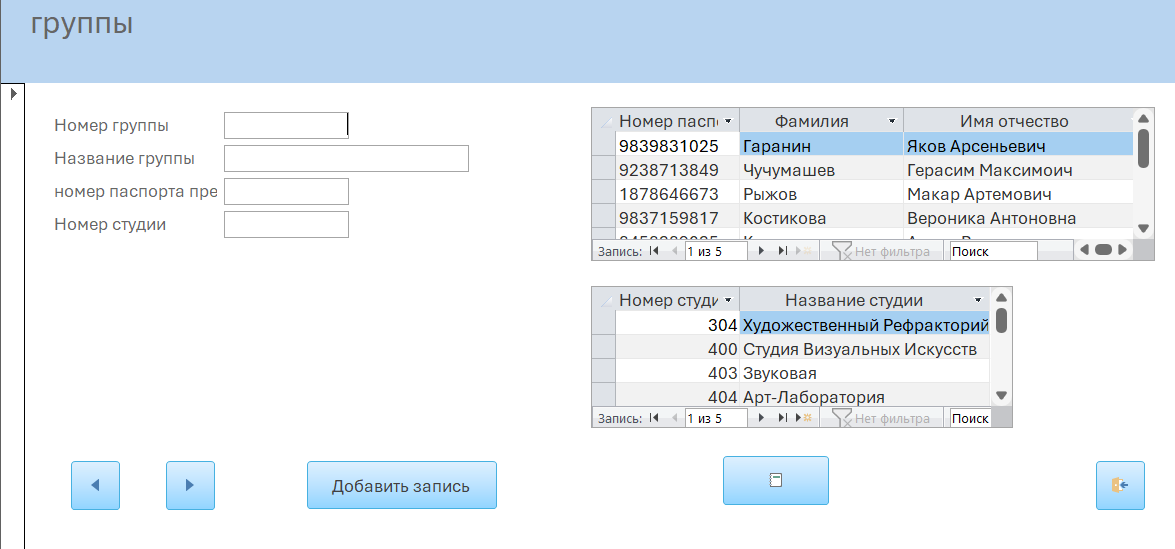


Рисунок 5 – Форма группы

Форма «преподаватели» (рисунок 6) имеет вспомогательные таблицы: семейные положения, виды образования, населенные пункты, специализации, специализации преподавателей для понимания заполняемой информации. Изображение выглядит как текст, число, программное обеспечение, диаграмма

Автоматически созданное описание

Рисунок 6 – Форма группы

Форма «школьники» (рисунок 7) имеет вспомогательные таблицы: дети, учебные заведения, для понимания заполняемой информации.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 7 – Форма группы

Форма «переходы» (рисунок 8) имеет вспомогательные таблицы: дети, группы, для понимания заполняемой информации.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 8 – Форма группы

Форма «родители» (рисунок 9) имеет вспомогательную таблицу: дети, для понимания заполняемой информации.

Изображение выглядит как текст, Шрифт, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 9 – Форма группы

Вкладка отчеты (рисунок 10), на которую пользователь также может перейти по клику, необходима для компактного и систематического отображения запросов пользователя.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 10 – Отчеты

Для просмотра результата запросов необходимо перейти на вкладку «запросы» (рисунок 11).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 11 – Запросы

**Заключение**

В ходе курсовой работы были решены следующие задачи:

1. провести анализ предметной области;
2. построить концептуальную модели базы данных;
3. построить реляционную модели 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.

Была достигнута автоматизация некоторых процессов, протекающих в организации, в частности учета детей, сотрудников и групп детского творчества. База данных была спроектирована с расчетом на расширение и дополнение, следовательно, в дальнейшем можно автоматизировать некоторые другие процессы данной предметной области.

**Список используемых источников**

1. Основные сведения о базах данных [Электронный ресурс]. - Режим доступа: https://support.microsoft.com/ru-ru/topic/%D0%BE%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 09.01.2024);

2. MySQL учебник [Электронный ресурс]. - Режим доступа: https://oracleplsql.ru/mysql-manual.html (дата обращения 01.01.2024);

3. MySQL официальная документация [Электронный ресурс]. - Режим доступа: https://docs.oracle.com/cd/E17952\_01/mysql-8.3-en/index.html (дата обращения 06.01.2024).