

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ
по лабораторной работе № 2
на тему
СОЗДАНИЕ РЕЛЯЦИОННОЙ СХЕМЫ ДАННЫХ
«ШКОЛА»

Студент:

А.Н. Климович

Преподаватель:

Д.В. Куприянова

МИНСК 2024

1 ЦЕЛЬ РАБОТЫ

Выполнить логическое проектирование БД путем построения реляционной схемы данных по ранее спроектированной ER-модели «Школа». Требуется преобразовать ER- диаграмму в реляционную схему данных (в виде UML-диаграммы).

2 ИСХОДНЫЕ ДАННЫЕ

В ходе лабораторной работы будут выполнены следующие задачи:

1. Проверить ER-диаграмму, созданную в лабораторной работе №1.
2. Выполнить преобразование ER-диаграммы в реляционную модель в двух вариантах: бумажном и автоматизированном.
3. Сравнить полученные диаграммы и, если есть расхождения в полученных реляционных диаграммах, найти несоответствия и устранить их.

На рисунке 1.1 приведена ER-диаграмма модели «Школа», разработанная в ходе лабораторной работы № 1.

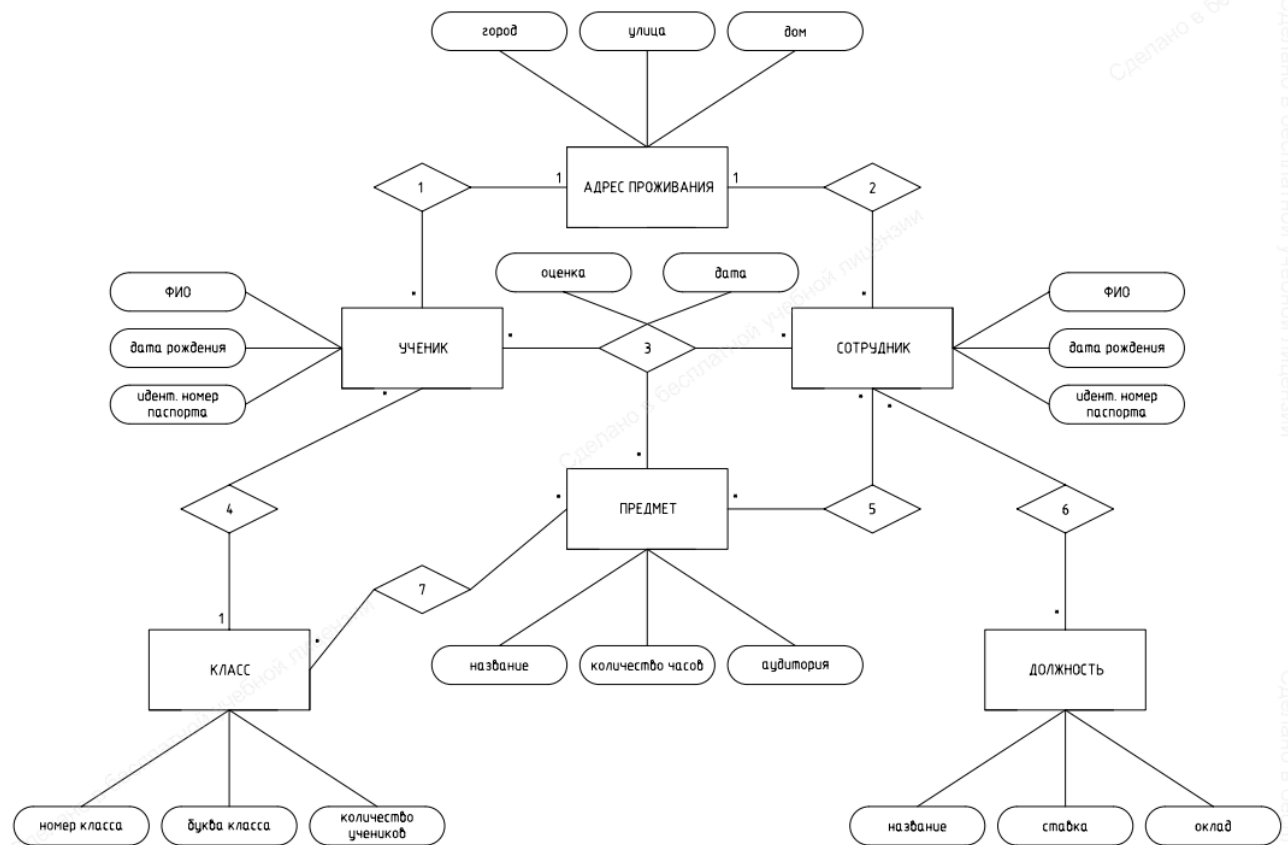


Рисунок 1.1 – ER-диаграмма модели «Школа»

3 ВЫПОЛНЕНИЕ РАБОТЫ

3.1 Алгоритм перевода ER-модели в реляционную модель

Порядок перевода ER-модели в реляционную модель выполняется с помощью алгоритма, состоящего из шести шагов:

1. Каждый *объект* на ER-диаграмме превращается в реляционное отношение (таблицу), имя объекта становится именем таблицы (следует указать понятное имя).

2. Каждый *атрибут объекта* становится столбцом таблицы с тем же именем и требуемым типом данных.

3. Уникальные (*ключевые*) атрибуты объекта превращаются в первичный ключ таблицы (при наличии нескольких возможных уникальных идентификаторов, выбирается наиболее подходящий для использования; если таковых атрибутов нет или они плохо подходят для долговременного использования в БД, то желательно создать *суррогатный ключ*). *Каждая таблица в БД должна иметь первичный ключ.*

4. Связи «*один-ко-многим*» (в том числе и связи «*один-к-одному*») становятся ссылками в уже существующих таблицах, при этом внешний ключ добавляется в виде столбца (столбцов) в таблицу, соответствующую объекту со стороны «*многие*» связи. Внешние ключи должны ссылаться только на первичные ключи целевых таблиц.

5. Связи «*многие-ко-многим*» реализуются каждая через отдельную промежуточную таблицу:

- эта промежуточная таблица обязательно будет содержать столбцы внешних ключей, ссылающиеся на соответствующие объекты связи;

- первичный ключ промежуточной таблицы для исключения дубликатов должен быть составным и включать в себя все внешние ключи на объекты, участвующие в связи.

6. Если связь имеет *дополнительные* атрибуты, то, как и в случае атрибутов объектов, они становятся столбцом соответствующей таблицы:

- для связей «*один-ко-многим*» – в таблице со стороны «*многие*» (там, где расположен внешний ключ);

- для связей «*многие-ко-многим*» – в промежуточной таблице (при этом атрибуты, расширяющие комбинацию в связи (например, «дата»), также должны войти в состав *составного* первичного ключа промежуточной таблицы).

3.2 Перевод ER-модели в реляционную модель

Результаты этапов 1-3 алгоритма перевода ER-модели в реляционную модель приведены на рисунке 3.1.

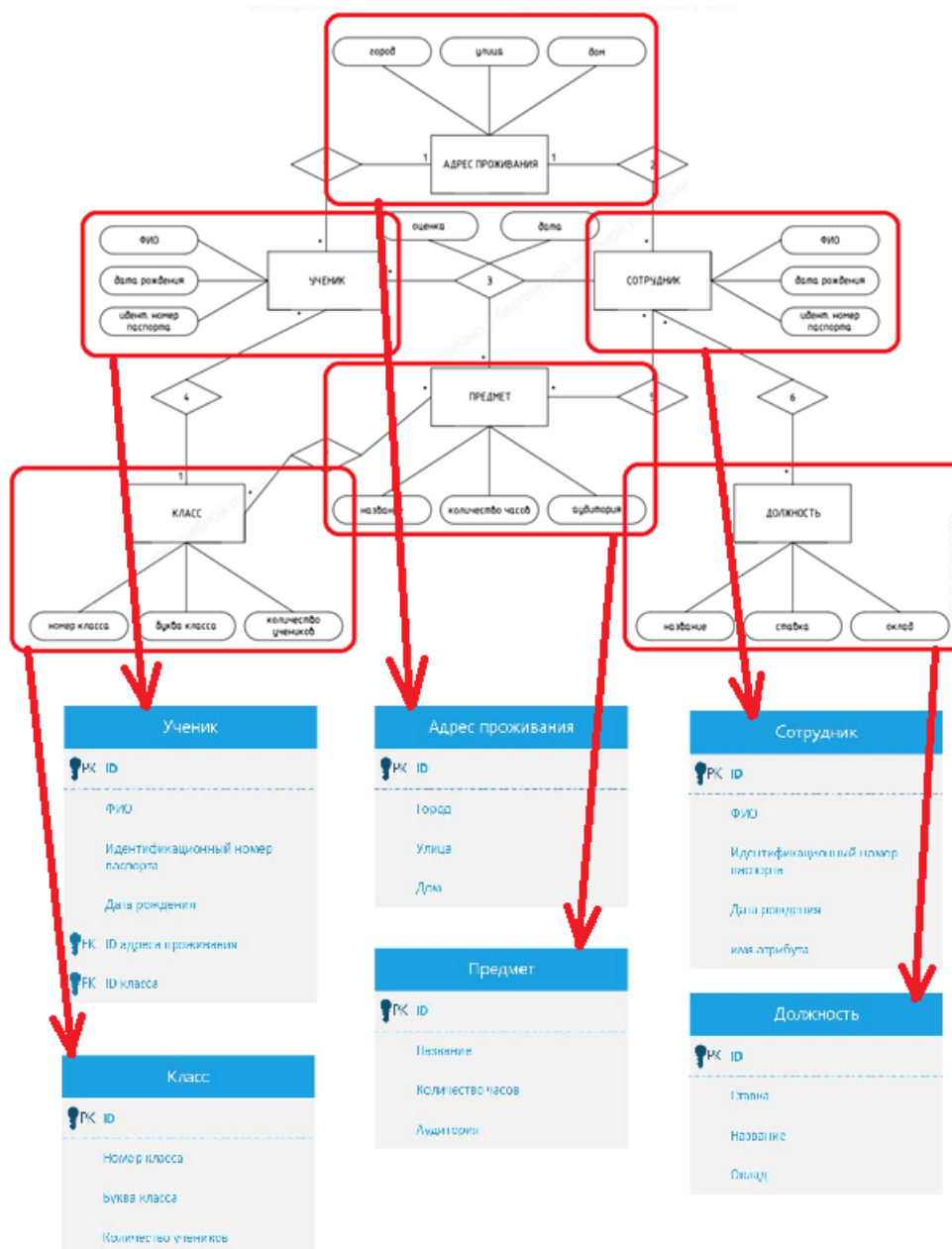


Рисунок 3.1 – Выделение объектов на ER-диаграмме

Выделенные объекты становятся таблицами с названиями, которые совпадают с именем сущностей, а атрибуты – столбцами таблиц. Таким образом, получим шесть таблиц (см. рисунок 3.2):

1. Таблица «ученик» со столбцами:
 - ФИО;
 - идентификационный номер паспорта;
 - дата рождения.
2. Таблица «сотрудник» со столбцами:
 - ФИО;
 - идентификационный номер паспорта (первичный ключ);
 - дата рождения.

3. Таблица «адрес проживания» со столбцами:

- город;
- улица;
- дом.

4. Таблица «предмет» со столбцами:

- название;
- количество часов;
- аудитория.

5. Таблица «класс» со столбцами:

- номер класса;
- буква класса;
- количество учеников.

6. Таблица «должность» со столбцами:

- название;
- ставка;
- оклад.

Также необходимо выделить/добавить в каждой таблице первичный ключ. Кроме того, отношение «один-ко-многим» порождает внешние ключи. С учетом этого получим таблицы, приведенные на рисунке 3.3.

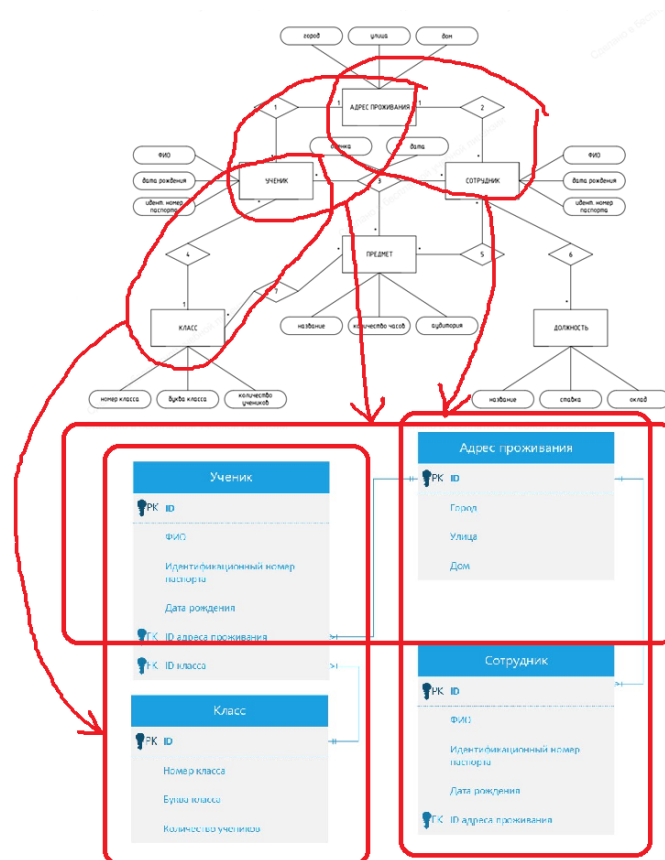


Рисунок 3.3 – Основные таблицы реляционной модели с первичными и внешними ключами

Отношения «многие-ко-многим» порождают четыре промежуточные таблицы:

1. Промежуточная таблица «расписание занятий», связывающая таблицы «класс» и «предмет», со столбцами:

- ID (первичный ключ);
- ID класса (внешний ключ);
- ID предмета (внешний ключ).

2. Промежуточная таблица «журнал оценок», связывающая таблицы «ученик», «предмет» и «сотрудник», со столбцами:

- ID (первичный ключ);
- ID ученика (внешний ключ);
- ID предмета (внешний ключ);
- ID сотрудника (внешний ключ);
- оценка;
- дата.

3. Промежуточная таблица «знание предмета», связывающая таблицы «сотрудник» и «предмет», со столбцами:

- ID сотрудника (первичный и внешний ключ);
- ID предмета (внешний ключ).

4. Промежуточная таблица «должность сотрудника», связывающая таблицы «сотрудник» и «должность», со столбцами:

- ID сотрудника (первичный и внешний ключ);
- ID должности (внешний ключ)

Эти таблицы приведены на рисунке 3.4.

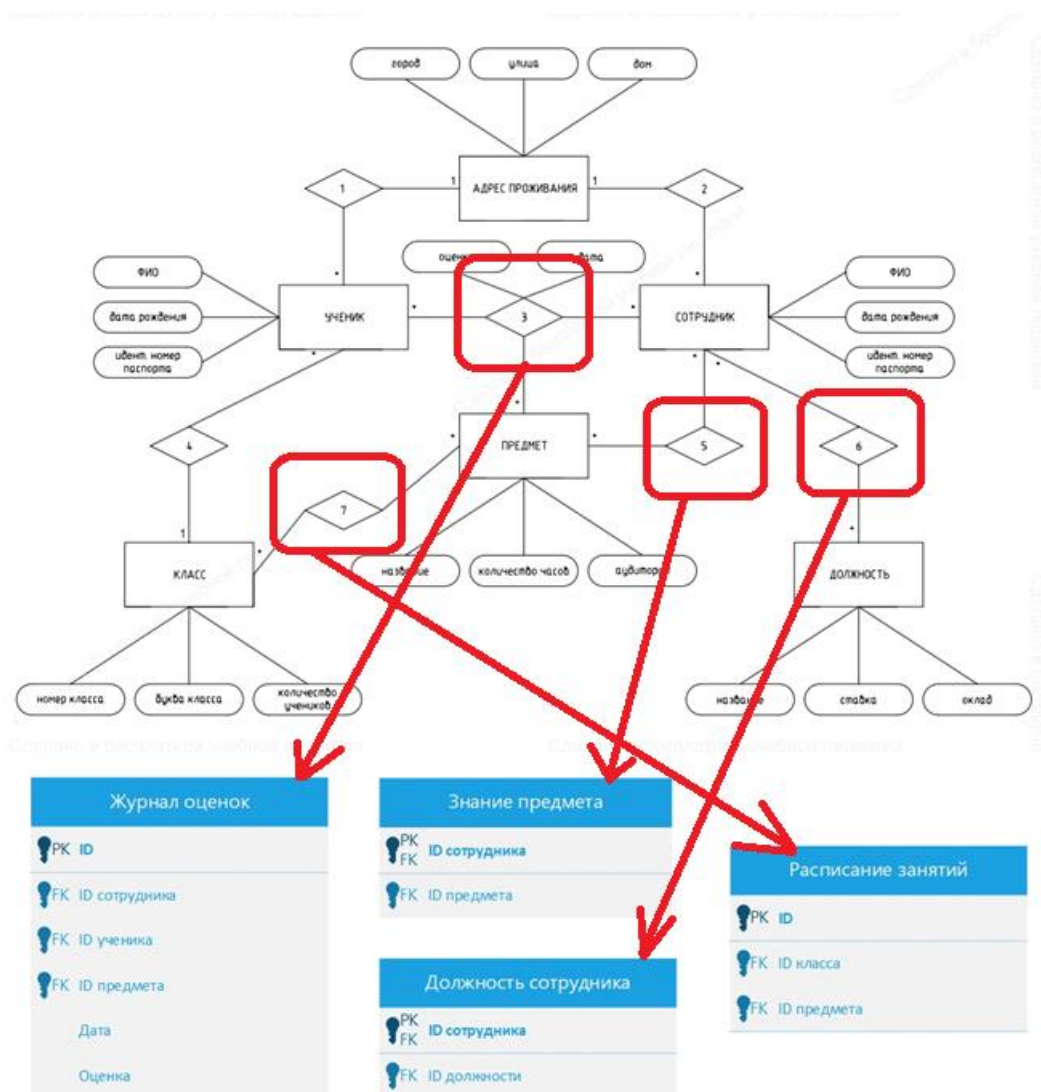


Рисунок 3.4 – Промежуточные таблицы реляционной модели

Последнее действие – объединить основные таблицы с промежуточными и показать отношения между ними соединительными линиями. Для этого выберем нотацию Crow's Foot («вороньи лапки») (см. рисунок 3.5).

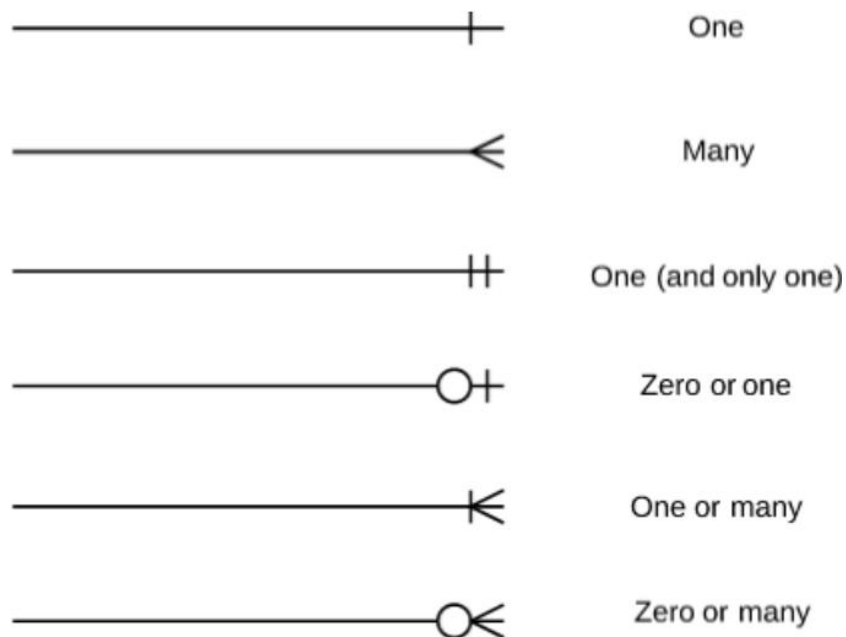


Рисунок 3.5 – Соединительные линии в нотации Crow's Foot

В итоге преобразование ER-диаграммы в реляционную модель с помощью программы Visio показано на рисунке 3.6.

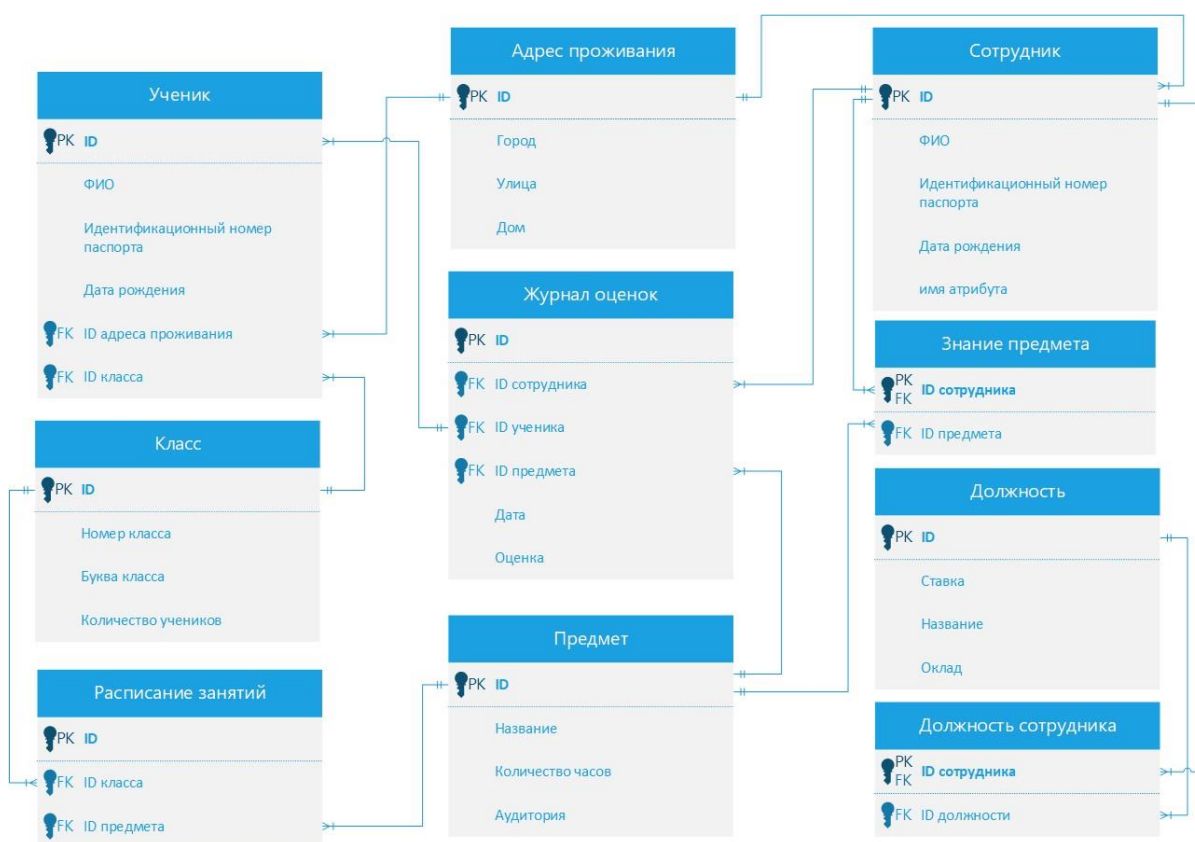


Рисунок 3.6 – Реляционная модель («бумажный» вариант)

Реляционная модель в «автоматизированном» варианте из ER-диаграммы будет получена в виде UML-диаграммы. Это можно сделать с помощью программы pgAdmin, которая была установлена вместе с PostgreSQL в предыдущей лабораторной работе.

При запуске pgAdmin требуется ввести пароль (см. рисунок 3.7).



Рисунок 3.7 – Ввод пароля в pgAdmin

Создаем базу данных «school». Далее нажимаем на вкладку «Tools->ERD Tool», после чего открывается редактор, в котором можно проектировать диаграмму. Для этого были созданы аналогичные таблицы реляционной модели «бумажного» варианта, установлены связи и присвоены типы данных всем столбцам таблицы.

Пример создания таблицы приведен на рисунке 3.8 и 3.9.

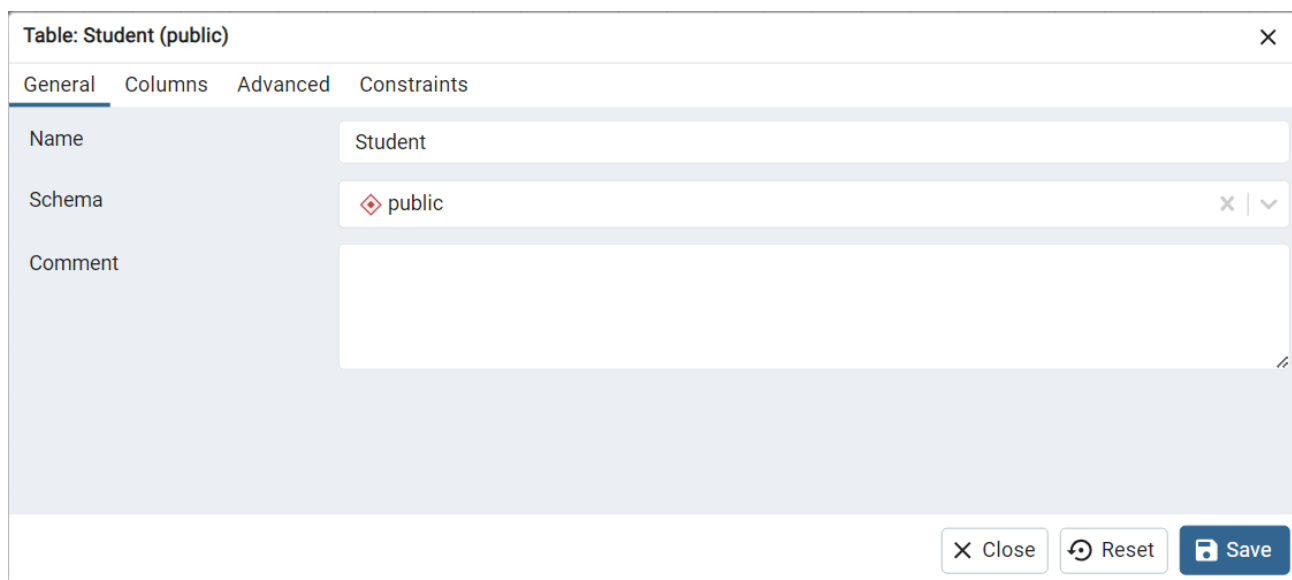


Рисунок 3.8 – Создание таблицы (общая информация)

Table: Student (public)

General Columns Advanced Constraints

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	ID	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	Full name	character varying	80		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Pasport ID	character	14		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Date of Birth	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Class ID	bigint			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Close Reset Save

Рисунок 3.9 – Создание столбцов таблицы

В итоге получим UML-диаграмму, которая показана на рисунке 3.10.

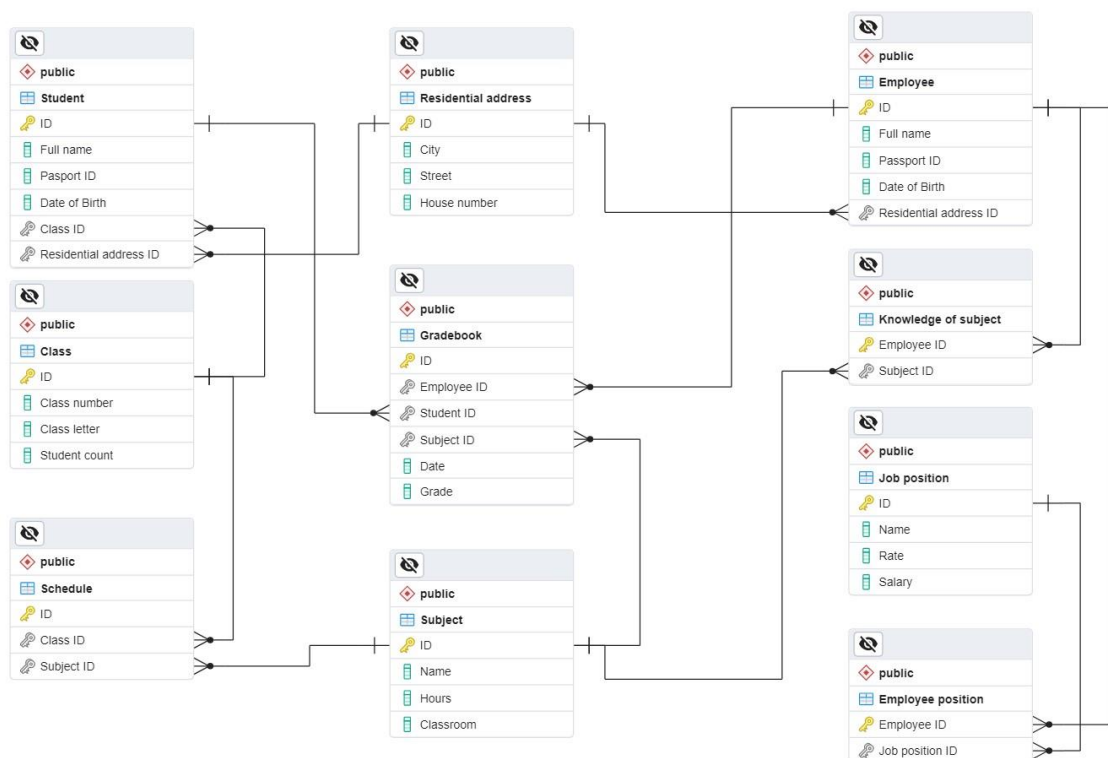


Рисунок 3.10 – Реляционная модель («автоматизированный» вариант)

4 ВЫВОД

В ходе лабораторной работы было выполнено преобразование ER-диаграммы, разработанную в ходе лабораторной работы № 1, в реляционную схему данных в двух вариантах: бумажном и автоматизированном. Также

были нормализованы реляционные отношения. Расхождений в полученных реляционных диаграммах выявлено не было.