

2.2 Проектирование схемы данных

Объектная модель данных – это схематичное графическое изображение технической структуры проекта: объектов и связей между ними. Она служит логическим отображением предметной области и позволяет представить данные в виде структурированных объектов, что обеспечивает удобство дальнейшей разработки, сопровождения и масштабирования программного решения. На рисунке 2.5 представлена данная модель.

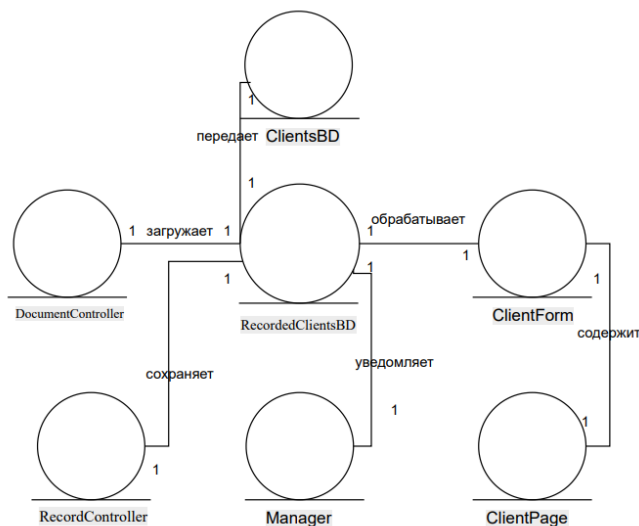


Рисунок 2.5 – Объектная модель системы

На ее основе была сформирована схема данных, определяющая структуру таблиц, поля, отношения и ограничения целостности. Схема данных должна обеспечить корректное хранение информации как о лидах, так и о зарегистрированных заказчиках, а также связях между заказчиками и их документами. На рисунке 2.6 представлена данная модель.

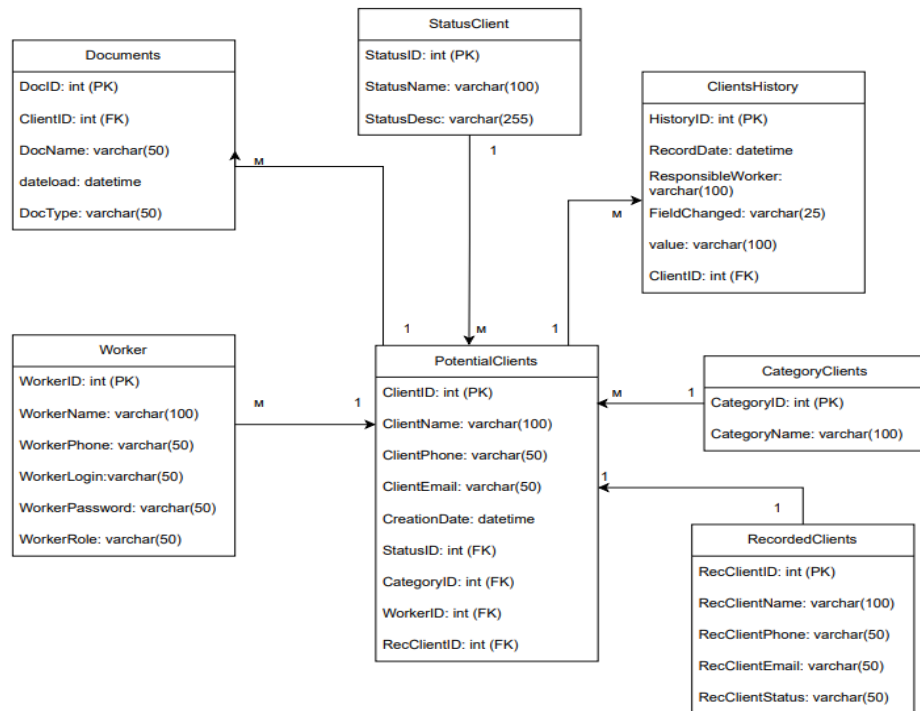


Рисунок 2.6 – Схема данных

SQL-триггеры широко применяются для обеспечения целостности данных, введения бизнес-правил и автоматизации повседневных операций, что повышает эффективность управления базой данных и уменьшает риск ошибок в данных [15]. Был создан триггер, который отслеживает изменения в таблице PotentialClients и автоматически создаёт записи в ClientsHistory, если изменился статус или ответственный за заказчика рабочий.

```

CREATE TRIGGER trg_client_history_update
ON PotentialClients
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    IF UPDATE(StatusID)
    BEGIN
        INSERT INTO ClientsHistory (RecordDate, FieldChanged, [Value], ClientID, ResponsibleWorker)
        SELECT
            GETDATE(),
            'StatusID',
            CAST(d.StatusID AS VARCHAR(20)),
            d.ClientID,
            w.WorkerName
        FROM deleted d
        JOIN inserted i ON d.ClientID = i.ClientID
        JOIN Worker w ON i.WorkerID = w.WorkerID
    END
  
```

```
WHERE d.StatusID <> i.StatusID;  
END  
END;
```

Ниже приведён SQL-триггер, который автоматически проставляет дату создания лида при вставке новой записи в таблицу PotentialClients.

```
CREATE TRIGGER trg_PotentialClients_set_creation_date  
ON PotentialClients  
AFTER INSERT  
AS  
BEGIN  
    SET NOCOUNT ON;  
  
    UPDATE pc  
    SET pc.CreationDate = GETDATE()  
    FROM PotentialClients pc  
    INNER JOIN inserted i ON pc.ClientID = i.ClientID  
    WHERE i.CreationDate IS NULL;  
END;
```

В приложении Б представлен код для создания таблиц, ключей, связей и других элементов базы данных