

2.2 Проектирование схемы данных

Объектная модель данных — это графическое представление ключевых сущностей системы, их атрибутов и взаимосвязей, задающее логическую структуру данных. Она служит основой для проектирования физической схемы базы данных, помогает определить необходимые таблицы и связи между ними и тем самым облегчает понимание того, как информация будет организована и обрабатываться в ходе реализации проекта. Объектная модель базы данных представлена на рисунке 2.5:

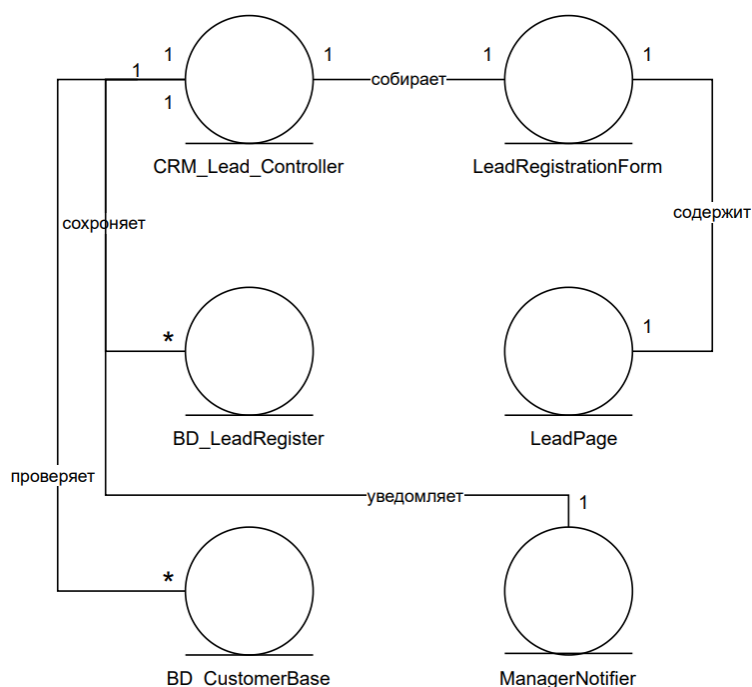


Рисунок 2.5 – Объектная схема данных

Далее на основе объектной модели была разработана схема данных, включающая таблицы, отношения, ключи, а также параметры связей между структурными элементами.

Схема данных, часто называемая ER-диаграммой, служит фундаментальной основой для построения физической базы данных. Она четко детерминирует все сущности (будущие таблицы), их атрибуты (поля), а также тип и кардинальность связей, существующих между ними [8, с. 14]. Детализированная схема данных представлена на Рисунке 2.6:

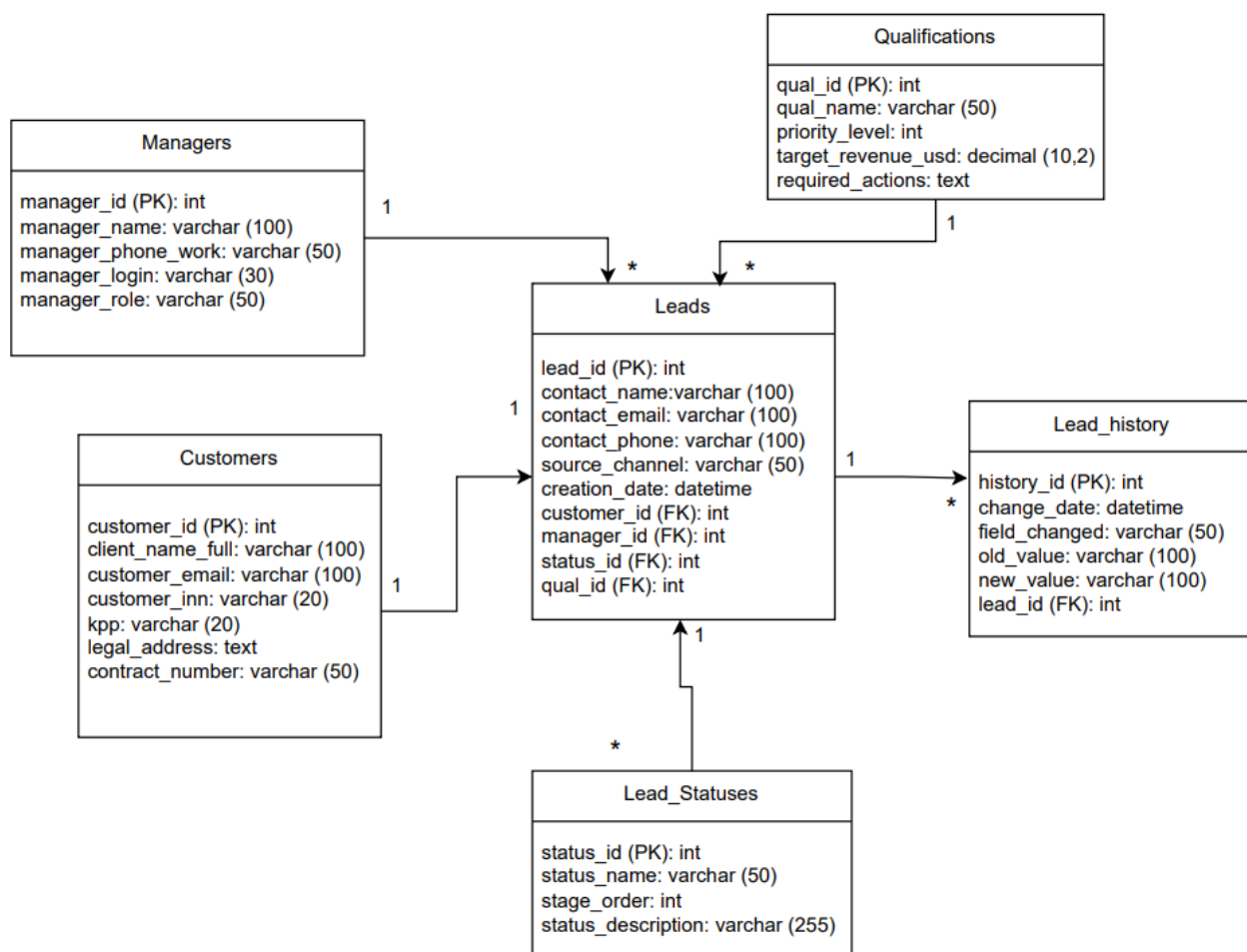


Рисунок 2.6 – Схема данных

У каждой из разработанных таблиц есть свои атрибуты, которые отражены в таблице 2.6.

Таблица 2.6 – Атрибуты схемы данных

№	Таблица	Название атрибута	Тип данных	Назначение
1	Leads	lead_id	Int	Уникальный ID нового запроса
		contact_email	Varchar(100)	Почта компании
		contact_name	Varchar(100)	ФИО или название компании
		contact_phone	Varchar(100)	Контактный телефон
		source_channel	Varchar(50)	Источник поступления (сайт, звонок, партнер)
		creation_date	Datetime	Дата и время регистрации запроса

Окончание таблицы 2.6

2	Customers	customer_id	Int	Уникальный ID существующего клиента
		customer_email	Varchar(100)	Email существующего клиента
		customer_inn	Varchar(20)	ИНН существующего клиента
		client_name_full	Varchar(100)	Полное юридическое название
		kpp	Varchar(20)	КПП организации
		legal_address	Text	Полный юридический адрес
		contract_number	Varchar(50)	Номер действующего договора
3	Managers	manager_id	Int	Уникальный ID сотрудника
		manager_name	Varchar(100)	ФИО менеджера
		manager_phone	Varchar(100)	Номер менеджера
		manager_login	Varchar(30)	Логин менеджера
		manager_role	Varchar(50)	Должность/роль
4	Lead_Statuses	status_id	Int	Уникальный ID статуса
		status_name	Varchar(50)	Название статуса (Новый, Квалифицирован)
		stage_order	Int	Порядок прохождения стадии в воронке
		status_description	Varchar(255)	Детальное описание
5	Qualifications	qual_id	Int	Уникальный ID квалификации
		qual_name	Varchar(50)	Название квалификации
		priority_level	Int	Уровень приоритета
		target_revenue_usd	Decimal (10, 2)	Потенциальный доход
		required_actions	Text	Необходимые действия для перехода

Окончание таблицы 2.6

6	Lead_History	history_id	Int	Уникальный ID записи истории
		change_date	Date	Дата и время изменения
		old_value	Varchar(100)	Старое значение
		new_value	Varchar(100)	Новое значение
		field_changed	Varchar(50)	Название измененного поля

В процессе формирования схемы данных, помимо структуры таблиц, были разработаны дополнительные элементы управления, а именно триггер и хранимая процедура процедура.

Разработанный триггер предназначен для предотвращения несанкционированного изменения критически важного поля «status_id», после того как Лид был перевед в статус «Квалифицирован», во избежание утраты важной информации о его жизненном цикле. Триггер сработает в момент попытки обновления данного поля.

```

CREATE TRIGGER trg_PreventStatusChange
ON LEADS
FOR UPDATE AS
BEGIN
    IF EXISTS (
        SELECT *
        FROM inserted i
        JOIN deleted d ON i.lead_id = d.lead_id
        WHERE d.status_id = 3 AND i.status_id <> 3
    )
    BEGIN
        PRINT 'Запрещено откатывать статус "Квалифицирован" для
сохранения истории.'
        ROLLBACK TRAN -- Откат всей транзакции
    END
END
GO

```

Хранимая процедура разработана для быстрого получения ключевой информации о лиде и его ответственном менеджере по уникальному идентификатору лида.

Процедура по ID Лида выведет его текущий статус и ФИО менеджера, который назначен ответственным за дальнейшую работу.

```
CREATE PROC prLeadManagerStatusInfo @LeadID INT
AS
SELECT
    LS.status_name,
    M.manager_name
FROM LEADS L
JOIN LEAD_STATUSES LS ON L.status_id = LS.status_id
JOIN MANAGERS M ON L.manager_id = M.manager_id
WHERE L.lead_id = @LeadID
GO
```

Листинг кода для создания таблиц и первичных ключей представлен в Приложении Б. Разработанная схема данных, включающая целостную структуру таблиц и элементы управления (триггер, процедура), может быть использована в будущем для создания продуктивной базы данных. SQL-код позволит практически полностью воссоздать базу данных и все ее элементы, описанные в схеме данных.