**Методические указания для Лабораторной работы №1**

**Основные сведения**

В рамках выбранной сферы деятельности необходимо определить набор реляционных таблиц и их логические связи. Для каждого атрибута необходимо указать тип и размер данных, а также установить ограничения целостности. Каждая таблица должна иметь первичный ключ, а также потенциальные и внешние ключи.

Разработка логической модели, используя метод "сущность-связь" (ER-метод), включает следующие шаги:

1) Построение ER-диаграммы, которая включает все сущности и связи, интересующие предметную область.

2) Анализ связей и определение их характеристик, таких как степень связи, мощность и класс принадлежности.

3) Создание набора предварительных отношений и назначение предполагаемого первичного ключа для каждого отношения.

4) Создание списка неключевых атрибутов и их распределение по предварительным отношениям.

5) Проверка соответствия всех полученных отношений нормальной форме Бойса-Кодда.

6) Построение модели данных.

**Практическая работа**

При выполнении лабораторной работы необходимо осуществить следующие шаги:

1. Создать логическую модель данных, соответствующую конкретной предметной области, согласно стандарту IDEF1X.

2. Перейти к построению физической модели, основываясь на разработанной логической модели данных.

3. Подготовить подробный отчет, в котором будут описаны все выполненные этапы работы по лаборатории.

**Пример выполнения работы**

* Анализируя диалект SQL в СУБД PostgreSQL, мы рассмотрим пример использования учебной базы данных *project*, занимающейся предоставлением данных о проектах IT компании. При работе с данной базой данных необходимо учесть следующую информацию:
* клиенты (ФИО, телефон, компания-заказчик);
* проекты(заказчик, начало и конец проекта,задание);
* отдел (айди, задача);
* сотрудник (ФИО, должность, квалификация, номер отдела);

Логическая модель данных предметной области в стандарте IDEF1X представлена на рис. 1. Выделены сущности *КЛИЕНТ, ПРОЕКТ, ОТДЕЛ, СОТРУДНИК, КВАЛИФИКАЦИЯ*.

**Виды связей**

***Один к одному*** ***(One-to-One)***. Пример: таблица "Паспорт" связана с таблицей "Человек" через уникальный идентификатор. У каждого человека есть только один паспорт, и каждый паспорт принадлежит только одному человеку.

***Один ко многим (One-to-Many)***. Пример: таблица "Категория" связана с таблицей "Продукт". Каждая категория может содержать много продуктов, но каждый продукт принадлежит только одной категории.

***Многие ко многим (Many-to-Many)***. Пример: таблицы "Студент" и "Курс" связаны через промежуточную таблицу "Регистрация". Студенты могут записываться на несколько курсов, и курс может иметь несколько зарегистрированных студентов.

***Многие к одному (Many-to-One)***. Пример: таблица "Студент" связана с таблицей "Университет". Множество студентов могут принадлежать к одному университету, но каждый студент связан только с одним университетом.

**Виды отношений**

***Идентифицирующее отношение***, где первая сущность однозначно определяет вторую сущность, причем ее первичный ключ наследуется в качестве первичного ключа второй сущности (внешний ключ).

***Неидентифицирующее отношение***, где первая сущность связана со второй сущностью, но однозначно не определяет ее. Первичный ключ первой сущности наследуется в качестве не ключевого атрибута второй сущности.

***Отношение «многие-ко-многим»*** (неспецифическое). Обе сущности имеют формальную связь, но наследования атрибутов не происходит.

***Отношения категоризации***. Это отношения между двумя и более сущностями, в которых каждый экземпляр одной сущности, называемой общей, связан в точности с одним экземпляром сущности, называемой сущностью-категорией.

**Первичный ключ или primary key(PK)**

Первичный ключ — особенное поле в SQL-таблице, которое позволяет однозначно идентифицировать каждую запись в ней. Как правило, эти поля используются для хранения уникальных идентификаторов объектов, которые перечислены в таблице, например, это может быть ID клиента или товара.

Первичный ключ имеет несколько свойств:

* каждая запись в таком поле должна быть уникальной;
* запись в поле не должна быть пустой;
* в одной таблице может быть только один ключ (существуют также составные ключи, которые могут включать в себя несколько полей, однако в этой статье мы не будем их рассматривать).

**Внешний ключ или foreign key**

Внешний ключ нужен для того, чтобы связать две разные SQL-таблицы между собой. Внешний ключ таблицы должен соответствует значению первичного ключа таблицы, с которой он связан. Это помогает сохранять согласованность базы данных путем обеспечения так называемой «ссылочной целостности» (referential integrity).

В примере на рис.1 используется связь *Многие ко многим* поэтому требуется создать связующую таблицу, которая будет связывать проекты и отделы, это нужно для корректного связывания сущностей при реализации таблиц.

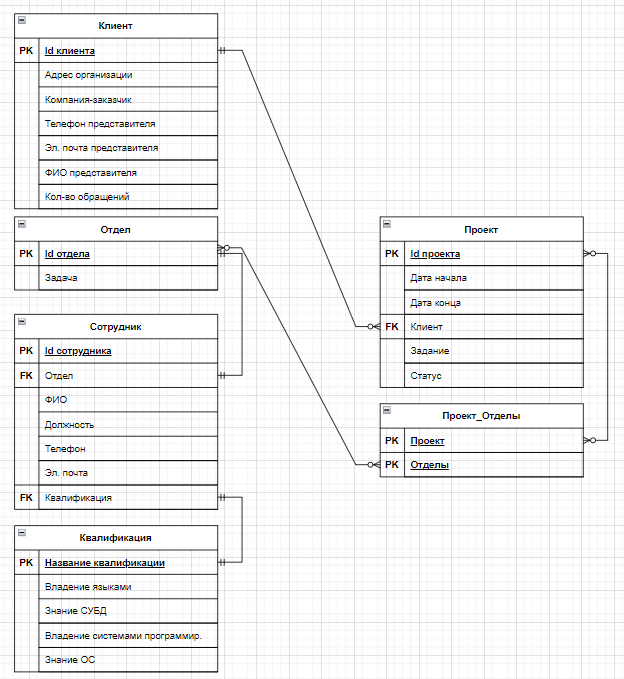


Рис. 1. Логическая модель данных предметной области

Физическая модель данных предметной области в стандарте IDEF1X для целевой СУБД PostgreSQL представлена на рис. 2.

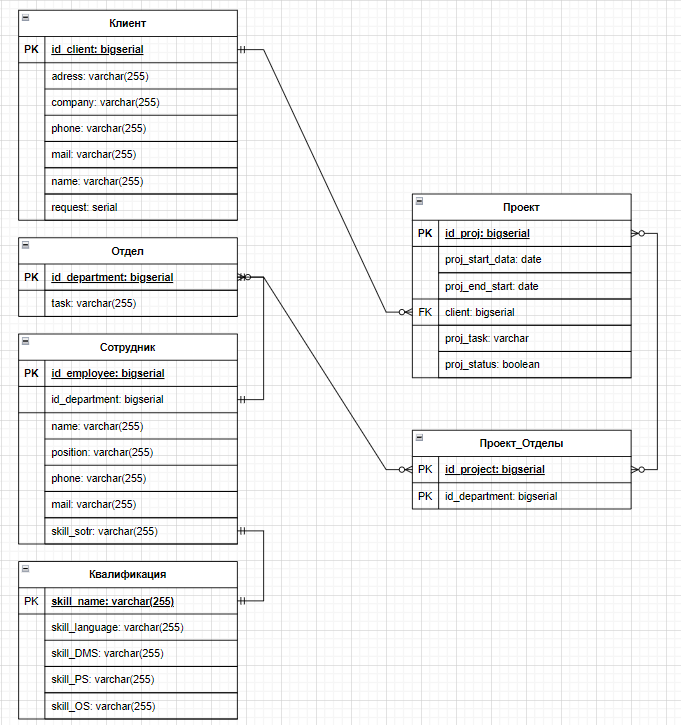


Рис. 2. Физическая модель предметной области

База данных project состоит из шести таблиц:

* *client* – список зарегистрированных клиентов;
* *otdel* – список отделов;
* *sotrudnik* – список сотрудников;
* *project* – список проектов.
* *skill* – список навыков.
* *Project\_Departament* – таблица для связи Проекта и нескольких отделов

Таблица *client* состоит из четырех полей:

* *id\_client* – уникальный код клиента;
* *name* – ФИО представителя;
* *company* – компания представителя;
* *phone –* телефон представителя;
* *address* – адресс компании- клиента;
* *mail* – электронная почта представителя;
* *request* – количество обращений.

Таблица *otdel* состоит из четырех полей:

* *id\_otd* – уникальный код отдела;
* *task*  – задача отдела;

Таблица *sotrudnik* состоит из трёх полей:

* *id\_sotr* – уникальный код сотрудника;
* *name* – ФИО сотрудника;
* *position* – Должность сотрудника;
* *phone*  - Телефон сотрудника;
* mail – Электронная почта сотрудника;
* skill – Квалификация сотрудника;

Таблица *skill* вклю­чает пять полей:

* *skill* – название квалификации;
* *skill\_language* – знание языков программирования;
* *skill\_DMS* – знание СУБД;
* *skill\_SP – знание систем программирования*;
* *skill\_OS* – знание операционных систем.

Таблица *project* вклю­чает девять полей:

* *id\_project* – уникальный номер проекта;
* *id\_otdel* – номера работающих над проектом отделов;
* *id\_client* – id клиента;
* *start\_date* – начало проекта;
* *end\_date* – конец проекта;
* *task*  – задача проекта;
* *status –* Статус проекта;

1. **Таблица сущностей.**

Табл. 1. Пример таблицы сущностей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название сущности | Количество | Изменение количества сущностей | Идентификатор | Ограничение доступа | Тип связи |
| Проект | 900 | 15% | Project | Team leader, project manager | М:1 (Клиент)  М:М(Отдел) |
| Клиент | 900 | 1% | Client | Team leader, project manager | 1:М (Проект) |
| Отдел | 10 | 3% | Otdel | Team leader | 1:М(Сотрудник) |
| Сотрудник | 100 | 20% | Sotrudnik | Team leader, HR | М:1 (Отдел)  1:1 (Квалификация) |
| Квалификация | 8 | 5% | Skill | Team leader, HR | 1:1 (Сотрудник) |

Примечание.

***Название сущности*** – это название выделенной в предметной области сущности.

***Количество*** – предлагаемое количество экземпляров сущности.

***Изменение количества сущностей*** – процент изменения сущностей в единицу времени (в таблице за единицу времени взят 1 год).

***Идентификатор*** – имя файла данных в базе данных, в котором будут храниться экземпляры данной сущности.

***Ограничение доступа*** – список пользователей, которым разрешен доступ к данной сущности.

***Тип связи*** – тип связи для каждой сущности, с которой связана данная сущность.

1. **Таблицы атрибутов.**

Таблица атрибутов сущности Проект

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID проекта | Числ | 0-9999 | Team leader, project manager | Ключ |
| Id отдела | Числ | 0-99999 | Team leader, project manager | Атрибут для связки |
| Сроки | дд\\мм\\гг |  | Team leader, project manager |  |
| Id клиента | Числ | 0-99999 | Team leader, project manager | Атрибут для связки |
| Задание | Строк. | 0-255 | Агент |  |
| Статус | Строк. | 0-255 | Агент |  |
| Начало проекта | дд\\мм\\гг |  | Team leader, project manager |  |
| Конец проекта | дд\\мм\\гг |  | Team leader, project manager |  |

Таблица атрибутов сущности КЛИЕНТ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID клиента | Числ | 0-9999 | Team leader, project manager | Ключ |
| ФИО представителя | Строк. | 0-255 | Team leader, project manager |  |
| Телефон представителя | Строк. | 0-255 | Team leader, project manager |  |
| Адрес организации | Строк. | 0-255 | Team leader, project manager |  |
| Электронная почта  представителя | Строк. | 0-255 | Team leader, project manager |  |
| Количество обращений | Числ | 0-9999 | Team leader, project manager |  |
| Компания-заказчик | Строк. | 0-255 | Team leader, project manager |  |

Таблица атрибутов сущности Отдел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID отдела | Числ. | 0-9999 | Team leader | Ключ |
| Id сотрудника | Числ | 0-9999 | Team leader | Атрибут для связки |
| Задание | Строк. | 0-255 | Team leader |  |

Таблица атрибутов сущности Сотрудник

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| ID сотрудника | Числ. | 0-9999 | Team leader | Ключ |
| ФИО | Строк | 0-255 | Team leader |  |
| Должность | Строк. | 0-255 | Team leader |  |
| Телефон | Числ. | 11 | Team leader |  |
| Эл. почта | Строк. | 0-255 | Team leader |  |
| Квалификация | Строк. | 0-255 | Team leader |  |

Таблица атрибутов сущности Квалификация

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя атрибута | Формат | Область допустимых значений | Ограничение доступа | Роль атрибута |
| Название квалификации | Строк. | 0-255 | Team leader, HR | Ключ |
| Владение языками | Строк | 0-255 | Team leader, HR |  |
| Знание СУБД | Строк. | 0-255 | Team leader, HR |  |
| Владение системами программирования | Строк. | 0-255 | Team leader, HR |  |
| Знание ОС | Строк. | 0-255 | Team leader, HR |  |

1. **Список возможных запросов.**
2. Выдать по id клиента проекты.
3. Выдать по id отдела сотрудников*.*