



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ЦЕНТР ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Управление качеством»

РАЗРАБОТКА БАЗЫ ДАННЫХ

**Методические указания к курсовому про-
екту по дисциплине «Управление процес-
сами» для студентов специальности
220501 «Управление качеством»**

Авторы: Александров П.В., Жмайлов Б.Б.

Ростов-на-Дону, 2010



Аннотация

Методические указания содержат пример проектирования базы данных. Предназначена для студентов V курса специальности 220501 «Управление качеством».





Оглавление

ВВЕДЕНИЕ	4
1 ПРОЕКТИРОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ	4
2 РЕАЛИЗАЦИЯ СХЕМЫ ДАННЫХ	5
3 СОЗДАНИЕ БАЗЫ ДАННЫХ	5
4 СОЗДАНИЕ ТАБЛИЦ	7
5 РЕДАКТИРОВАНИЕ ЗАПИСЕЙ В ТАБЛИЦЕ	8
6 РЕАЛИЗАЦИЯ СВЯЗИ МНОГИЕ К ОДНОМУ	9
7 РЕАЛИЗАЦИЯ СВЯЗИ МНОГИЕ КО МНОГИМ	11



ВВЕДЕНИЕ

Исходными данными для проектирования и реализации базы данных (БД) является построенная на предыдущем этапе курсового проекта DFD диаграмма. На ее основании можно определить сущности (таблицы) БД и набор операций которые можно будет выполнять над этими сущностями. Дальнейшая задача заключается в проектировании сущностей (определении их структуры) и установление взаимосвязей между ними. Данный этап является важным, так как ошибка при проектировании схемы данных может быть обнаружена не сразу, а спустя какое-то время, когда уже накоплен массив информации, что может привести к довольно большим проблемам при последующей эксплуатации БД. В качестве платформы СУБД можно использовать любую базу данных, которая поддерживает ANSI стандарт.

Далее необходимо определить стречневые и характеристические сущности, например Производитель (Стрежневая) — товары (характеристическая).

1 ПРОЕКТИРОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ

Рассмотрим проектирование БД на примере деятельности коммерческой фирмы, сотрудники которой взаимодействуют с поставщиками/потребителями. Структура организации состоит из отделов, каждым из которых руководит начальник. Каждый из сотрудников может взаимодействовать с множеством клиентов. В то же время несколько сотрудников могут работать с одним клиентом. Анализ отношений сотрудников с клиентами и отделами с точки зрения теории реляционных БД, позволяет установить тип связей между ними.

На первом этапе необходимо определить структуру сущностей (таблиц).

При определении типов данных нужно примерно представлять размер информации, которая будет храниться в данном столбце, так как неоправданное увеличение его длины может в последующем замедлять работу с базой данных.

Далее необходимо выбрать первичный ключ для таблицы. Первичный ключ (ПК) — это поле (столбец) или группа полей таблицы базы данных, значение которого (или комбинация значений которых) используется в качестве уникального идентификатора записи (строки) этой таблицы. Другими словами, ПК — это набор полей (атрибутов, столбцов) таблицы, совокупность значений которых определена для любой записи (строки) этой таблицы



Управление процессами

и различна для любых двух записей. Вторая нормальная форма (НФ) требует наличие первичного ключа.

1 РЕКОМЕНДАЦИИ ПО ИМЕНОВАНИЮ ТАБЛИЦ БАЗЫ ДАННЫХ И ИХ АТТРИБУТОВ

Во избежании проблемы связанных с последующим обращением к сущностям и их атрибутам (столбцам таблицы) рекомендуем придерживаться следующих правил их именования:

- использовать только латинский алфавит;
- при именовании сущностей использовать англоязычные названия соответствующие русским по смыслу например сотрудники - officers, производитель — producer;

при именовании атрибутов стараться, чтоб название не превышало 8 символов, в противном случае использовать акронимы например дата рождения — date_born, идентификатор родительского объекта — id_parent.

2 РЕАЛИЗАЦИЯ СХЕМЫ ДАННЫХ

Для примера реализации БД, будем использовать СУБД HSQL и OpenOffice.org Base, в качестве клиента к ней. База данных OpenOffice.org Base состоит из таблиц, запросов, форм и отчетов. Запросы служат для извлечения данных из таблиц с применением фильтров, сортировок, агрегации данных. Формы — элементы пользовательского интерфейса. Конечный пользователь БД зачастую не должен иметь непосредственный доступ к таблицам. Формы выступают промежуточным слоем (посредником) между пользователем и таблицами БД. Отчет представляет собой текстовый документ Writer, в котором данные отображаются в организованном и отформатированном виде. OpenOffice.org Base предлагает набор мастеров для создания компонентов базы данных.

3 СОЗДАНИЕ БАЗЫ ДАННЫХ

1. Запустить OpenOffice.org Base (рис. 1). Автоматически должен открыться мастер создания БД (рис 2).



Управление процессами

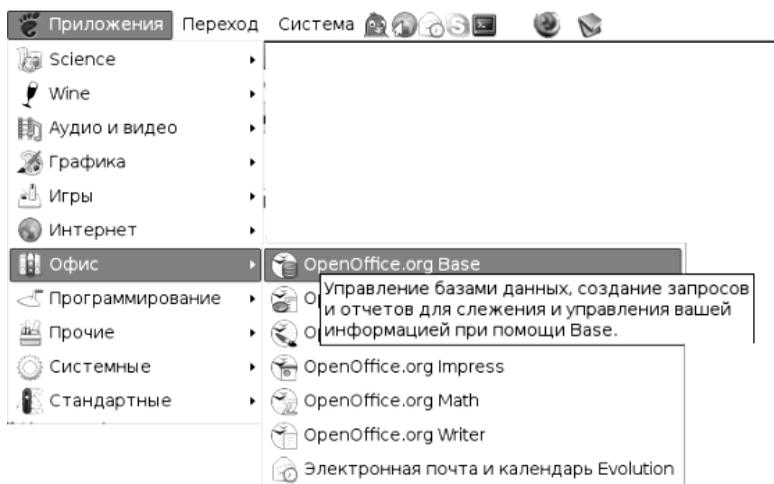


Рисунок 1 - запуск OpenOffice.org Base.

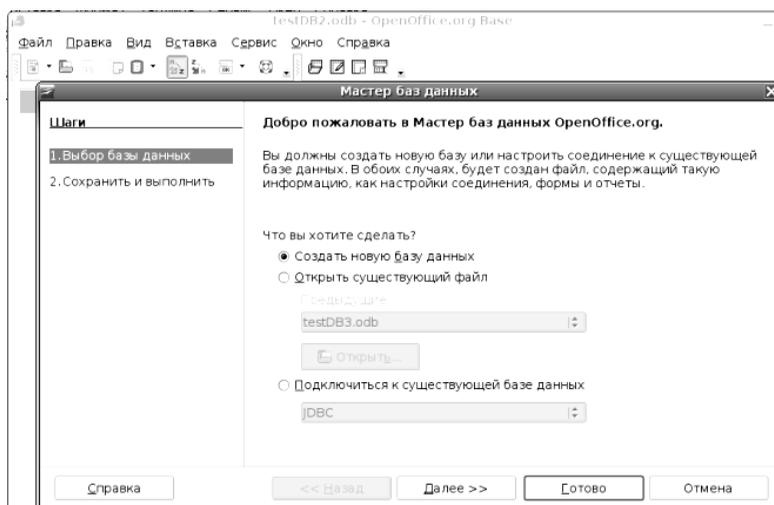


Рисунок 2 - мастер создания базы данных.

2. Отметить пункт **Создать новую базу данных**. Нажать **Далее**.
3. Отказаться от регистрации. Выбрать - **Открыть базу для редактирования**.
4. Нажать кнопку **Готово**, после чего открывается диалог сохранения файла.
5. Выбрать тип - **База данных ODF**. Имя БД выбрать в соответствии с рекомендациями приведенными выше. Сохранить



файл в своей личной директории.

4 СОЗДАНИЕ ТАБЛИЦ

Рассмотрим создание таблицы с информацией о сотрудниках организации. Таблица будет хранить следующие данные: — id, ФИО, дата рождения. OpenOffice.org Base предоставляет средство автоматической генерации значения уникального идентификатора. Например, добавляя запись в таблицу, можно вводить только ФИО сотрудника, а идентификатор генерируется автоматически.

1. Щелкните на значок **Таблицы**. Чтобы создать новую таблицу, выберите Создание таблицы в режиме дизайнера. Откроется дизайнер таблицы. Каждая строка в дизайнера — это атрибут будущей таблицы.

2. Добавить атрибут id. Тип «Целое [Integer]», Описание - «Уникальный идентификатор клиента» (рисунок 3).

3. Добавить атрибут name. Тип - «Текст[VARCHAR]», Описание - «Уникальный идентификатор сотрудника».

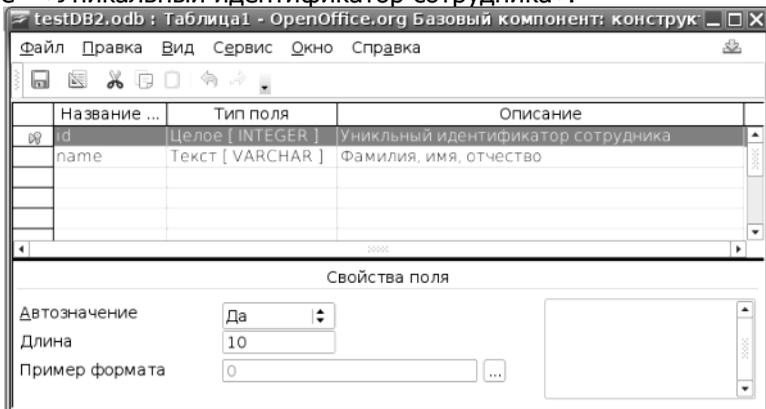


Рисунок 3 - Дизайнер таблиц.

4. Добавить атрибут birth_date. Тип - «Дата [DATE]», Описание - «Дата рождения сотрудника».

5. На строке id, левее колонки «наименование» щелкнуть правой кнопкой мыши. Появится контекстное меню (рисунок 4). Пометим поле id, как первичный ключ.



Управление процессами

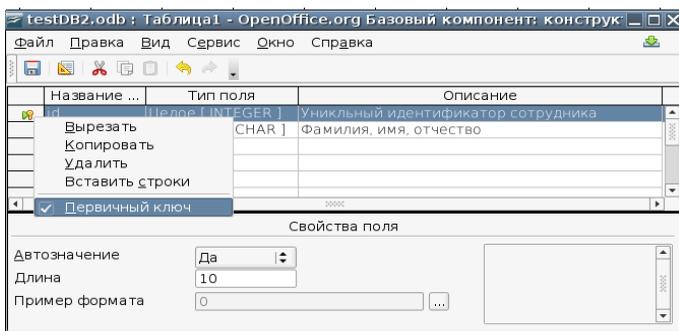


Рисунок 4 - создание первичного ключа.

6. В свойствах поля id отметить **Автозначение — Да**.
7. Выбрать **меню Файл — Сохранить**.
8. Задать для таблицы имя — officers.

По описанному выше принципу создать таблицу departments. Рекомендуемый набор полей и их описание приведено в таблице 1.

Таблица 1 — информация об отделах организации

Наименование	Тип	Описание
Id	Целое [INTEGER]	Уникальный идентификатор отдела
Name	Текст [VARCHAR]	Наименование отдела
Phone_number	Текст [VARCHAR]	Телефонный номер отдела

5 РЕДАКТИРОВАНИЕ ЗАПИСЕЙ В ТАБЛИЦЕ

1. Откроем таблицу officers двойным щелчком мыши
2. Добавить несколько строк. Каждая строка — информация о сотруднике (рисунок 5).

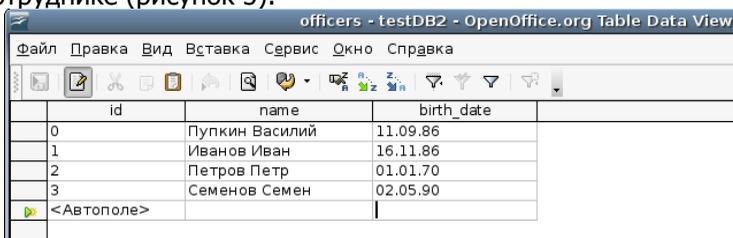


Рисунок 5 - редактирование данных в таблице

Таким же образом добавить информацию в таблицу отделов. Убедитесь, что поля, которые помечены автозаполняемыми,



Управление процессами

получают уникальные значения автоматически. Так на рисунке 5, первая добавленная строка получила идентификатор 0, вторая 1 и т.д. При попытке изменения значения в поле id 1-й строки на 2 и последующего сохранения, выводится ошибка. Помните, что значения первичного ключа должны быть уникальными в пределах таблицы!

6 РЕАЛИЗАЦИЯ СВЯЗИ МНОГИЕ К ОДНОМУ

По условиям нашей задачи, каждый сотрудник относится к некоторому отделу. Для реализации этой связи добавим в таблицу сотрудников поле — идентификатор отдела.

Таблица 2 – поле dep_id таблицы officers

Наименование	Тип	Описание
dep_id	Целое [INTEGER]	Ссылка на идентификатор отдела

Откройте таблицу officers на редактирование данных. Проставьте идентификаторы отделов для каждого сотрудника. Убедитесь, что база данных позволяет присвоить полю dep_id значение, которое отсутствует с таблице departments. В этом случае получается, что сотрудник может относиться к несуществующему отделу. Данная несоответствие называется нарушением ссылочной целостности.

Ссылочная целостность в реляционной базе данных – это согласованность между связанными таблицами. Ссылочная целостность автоматически поддерживается путем комбинирования первичного ключа и внешнего ключа. Для соблюдения ссылочной целостности требуется, чтобы любое поле в таблице, объявленное внешним ключом, могло содержать только значения из поля первичного ключа родительской таблицы необходимо:

1. Открыть конструктор связей. Выбрать пункт меню **Сервис — Связи...** (рисунок 6).
2. На панели инструментов нажать на кнопку **Добавить таблицы**.
3. Добавить таблицы officers и departments.
4. На панели инструментов нажать на кнопку **Создать связь**.



Управление процессами

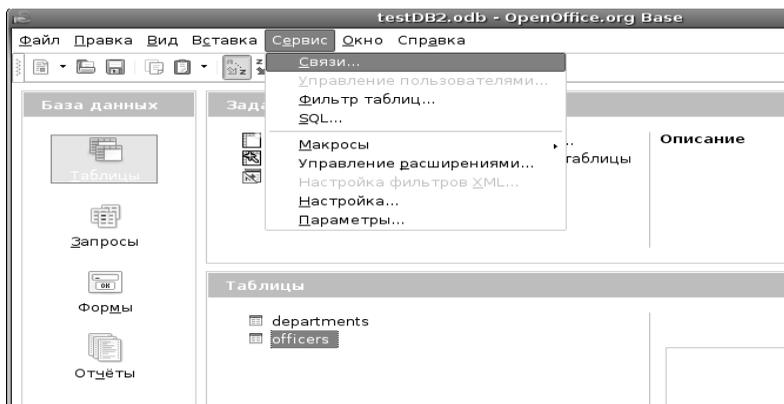


Рисунок 6 — Автоматический контроль ссылочной целостности.

5. В секции Включенные таблицы выбрать officers и departments, как показано на рисунке 7.

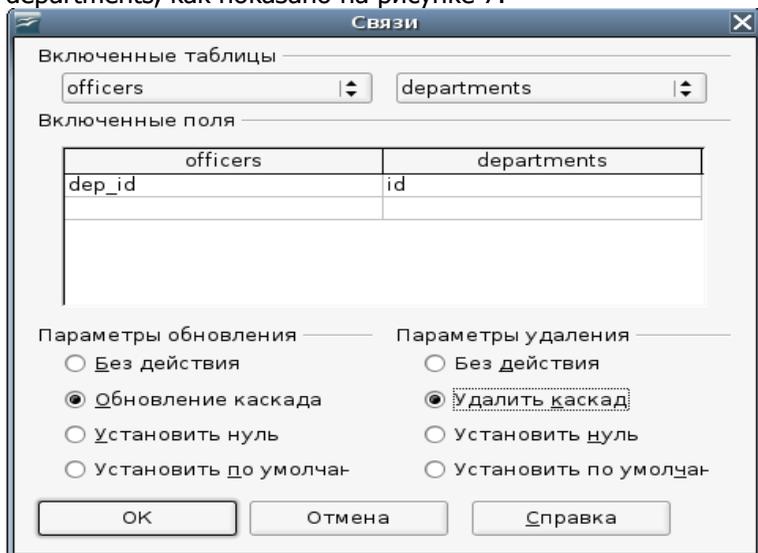


Рисунок 7 — создание внешнего ключа.

6. В секции **Включенные поля** выбрать dep_id для таблицы officers и id для таблицы departments.

7. В секции **Параметры обновления** отметить Обновление каскада.

8. В секции **Параметры удаления** отметить **Удалить каскада**.

9. Нажмите **Ок**. Если все выполнено правильно, в конструкторе связей добавится связь между таблицами (рисунок 8).



Управление процессами

Откройте таблицу сотрудников. Попробуйте задать в поле `dep_id` значение, которое отсутствует в таблице отделов.



Рисунок 8 — Внешний ключ в редакторе связей

6 ПОЛЯ, ОБЯЗАТЕЛЬНЫЕ ДЛЯ ЗАПОЛНЕНИЯ

Откройте таблицу сотрудников в режиме редактирования данных. Добавьте запись не указывая поле `name`. Каждый сотрудник организации обязательно должен иметь имя. Добавим это ограничение.

1. Открыть таблицу сотрудников в режиме редактирования структуры. ПКМ на таблице `officers` → **Изменить**
2. Выделить атрибут «`name`».
3. Отметить его как обязательное (рисунок 9).

7 РЕАЛИЗАЦИЯ СВЯЗИ МНОГИЕ КО МНОГИМ

Рассмотрим связь сотрудник-клиент. По условию задачи, тип связи — многие ко многим. Связь многие ко многим правильно реализовывать через дополнительную таблицу. Дополнительная таблица будет состоять из двух полей: идентификатор сотрудника и идентификатор клиента. Рассмотрим пример.



Управление процессами

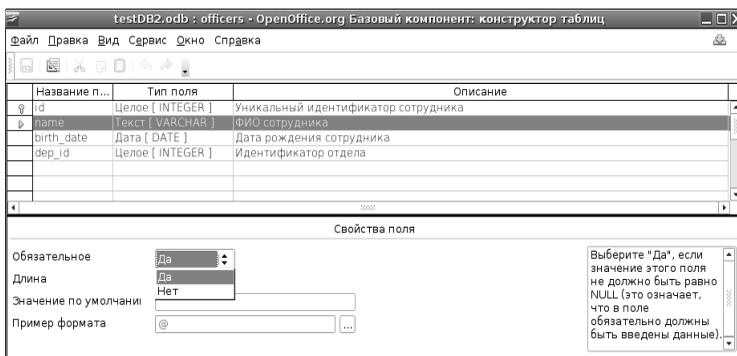


Рисунок 9 — Обязательное поле

Попробуйте добавить сотрудника без имени.

Таблица 3 — Сотрудники

id	name
1	Иванов
2	Петров

Таблица 4 — Клиенты

id	name
1	Калинин
2	Рябинин
3	Тополев

Для того чтобы записать, что сотрудник Иванов работает с клиентами Калининым и Ивановым, а Петров с Рябининым и Ивановым, необходимо добавить в промежуточную таблицу 4 записи.

Таблица 5 – пример содержимого таблицы officers_to_clients.

1	1
1	2
2	2
2	3



Управление процессами

1. Добавить таблицу clients с информацией о клиентах.
2. Добавить таблицу officers_to_clients(таблица 5). Пометить оба поля ключевыми.

Таблица 5 — Структура таблицы связи клиент - сотрудник

Наименование	Тип	Описание
officer_id	Целое [INTEGER]	Ссылка на сотрудника
client_id	Целое [INTEGER]	Ссылка на клиента

3. В дизайнера связей, создать два внешних ключа: officers_to_clients.client_id → clients.id и officers_to_clients.officer_id → officers.id (рисунок 10).

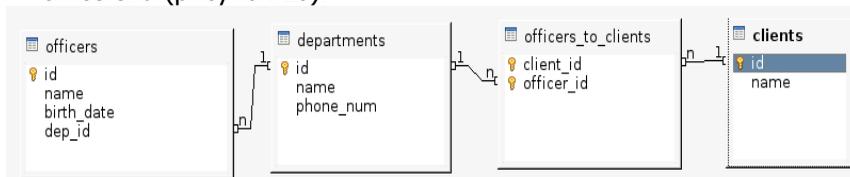


Рисунок 10 — Связь многие ко многим.

Добавьте в таблицу клиентов несколько записей. Свяжите их с сотрудниками.