## Лекция 2. Обзор современных инструментальных средств разработки схемы базы данных

При выборе технологии построения информационной системы, содержащей в своем составе базу данных, нужно тщательно оценивать и прогнозировать ее потенциальные потребности в средствах управления данными. Конечно, любую информационную систему можно основывать на использовании промышленной, большой и мощной СУБД (такой, как например Oracle). Но вполне может оказаться так, что в действительности приложение будет использовать доли процентов общих возможностей СУБД. Накладные расходы (затраты на дополнительную аппаратуру, лицензирование дорогостоящего программного продукта, увеличение общего времени выполнения операций) могут оказаться неоправданными.

При разработке информационных систем для локальных вычислительных сетей с использованием технологии клиент/сервер оправданно и целесообразно в качестве СУБД применять свободно распространяемую СУБД FireBird, которую можно устанавливать практически на компьютеры с любой платформой (Unix, Linux, Windows и пр.). Эта СУБД, для своей установки, не требует покупки специального сервера (так например, СУБД MS SQL Server требует для своей установки сервер Windows Server 2003) и обладает большим количеством других преимуществ.

Обслуживание СУБД FireBird осуществляется с использованием инструментов администрирования IBExpert, которые для граждан и предприятий, использующих русскоязычную операционную систему Windows, также являются бесплатными.

Методология проектирования информационных систем, включающих базы данных, предусматривает проектирование базы данных и приложения для работы с ней. Данное методическое пособие посвящено вопросам разработки баз данных. Проектирование баз данных таких систем предполагает разбиение всего процесса на несколько фаз, каждая из которых может состоять из нескольких этапов. На каждом этапе разработчик использует набор технических приемов позволяющих решать задачи данной стадии разработки.

Весь процесс разработки разделяется на три основные фазы: концептуальное, логическое и физическое проектирование.

*Концептуальное проектирование БД* необходимо для создания информационной модели предприятия (предметной области), не зависящей от каких- либо физических условий реализации. К последним относятся: тип СУБД, --- программ приложения, используемый язык программирования, конкретная вычислительная платформа и другие физические особенности реализации.

*Логическое проектирование БД* необходимо для создания информационной модели предприятия на основе разработанного концептуальной модели с учетом используемого типа СУБД (но не конкретной СУБД и прочих физических условий реализации).

*Физическое проектирование БД* - это процесс создания описания конкретной реализации БД с учетом особенностей выбранной СУБД. Эта фаза заканчивается созданием конкретной БД для создаваемого приложения, на основании разработанной ранее логической модели.

Проектирование базы данных может предусматривать выбор наиболее подходящего инструмента автоматизированного проектирования - CASE-инструмента (Computer-Aided Software Engineering).

В самом широком смысле термин CASE- инструмент применим к любым средствам автоматизированного проектирования и создания программ.

CASE- инструменты могут включать следующие компоненты:

* словарь данных, предназначенный для хранения информации в данных, используемых в создаваемом приложении;
* инструменты проектирования, обеспечивающие проведение анализа данных;
* инструменты разработки модели данных предприятия (модели бизнес-процесса), а также концептуальных и логических моделей данных;
* инструменты, позволяющие создавать прототипы приложений.

Использование CASE-инструментов позволяет существенно повысить производительность труда при разработке приложений баз данных.

## 2.1. Концептуальная модель данных

Концептуальная модель данных отображает обобщающее представление о данных, не зависимое от типа выбранной СУБД. Она описывает то, какие данные хранятся в базе данных, а также связи, существующие между ними. Фактически это полное представление требований к данным со стороны организации, у которой работают пользователи.

Концептуальная модель данных состоит из сущностей со своими атрибутами и n-арных связей и используется как средство построения и представления информационных потребностей предприятия.

**Сущность**: информационный объект, относящийся к деятельности предприятия

**Атрибут**: характеристика сущности

**Связь**: связь сущностей между собой, обычно между двумя сущностями, а в общем – между n сущностями; осуществляется через связь экземпляров одной сущности с экземплярами другой сущности

**Роль**: определяется с каждой стороны связи. Определяет смысл участия соответствующей сущности в данной связи (например, родительская сущность, дочерняя сущность)

**Кардинальность связи**: максимальное количество экземпляров одной сущности, связанных с одним экземпляром другой сущности

**Управляемость** **ролью**: показывает, что данная сущность является дочерней сущностью родительской сущности

**Ограничения роли**: механизм поддержания целостности связей

**Ключ**: может быть первичным или потенциальным.

**Зависимость (подчиненность) ключа**: определяется для первичных ключей и суперключей.

**Сущность представляет собой любой абстрактный или конкретный объект организации, чьи характеристики определяются с помощью атрибутов.**

**При создании сущности, *Open ModelSphere* автоматически присваивает ей имя «Сущность»*.* Во избежание неприятностей при создании сложных моделей, настоятельно рекомендуется присваивать сущностям смысловые имена как можно раньше, сразу после создания сущностей. Для этого следует использовать функцию редактирования содержимого созданной сущности.**

**Атрибут обычно содержит одно значение. Каждая сущность должна иметь, по крайней мере, один атрибут.**

***Open ModelSphere* имеет функцию редактирования, которая позволяет добавлять атрибуты непосредственно в графическое представление сущности. При создании атрибута, *Open ModelSphere* автоматически присваивает атрибуту имя «**Атрибут»**.**

Первичные ключи можно создавать с помощью двух способов: используя панель инструментов, или, непосредственно вводя информацию в окне свойств сущностей модели.

Например: BOOKS ID, FIRMS ID, PAYMENTS ID, NAKLS ID.

В ***Open ModelSphere***, первичные ключи подчеркнуты, чтобы отличить их от других атрибутов.

В рассматриваемом примере, все первичные ключи состоят из одного атрибута. Например, в сущности FIRMS использован атрибут FIRMS ID для идентификации каждой фирмы, атрибут BOOKS ID однозначно идентифицирует каждую книгу сущности BOOKS.

***Open ModelSphere*** предоставляет два типа связей: с использованием прямых углов и использованием любых углов . Оба типа играют одну и ту же роль, различаются только их изображения на диаграмме.

При создании связи устанавливаются кардинальности по умолчанию: показатель кардинальности равен **0,N** для родительской сущности, от которой начинается связь, и **1,1** – для дочерней сущности, на которой эта связь заканчивается. Если дочерняя сущность является слабой сущностью, то ее кардинальность должна быть подчеркнутой (такая связь называется **идентифицирующей**).



Рисунок 1. Окончательный вид концептуальной модели