

# Технология разработки баз данных

*СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НАУКИ И  
ТЕХНОЛОГИЙ ИМЕНИ АКАДЕМИКА М.Ф. РЕШЕТНЕВА  
АЭРОКОСМИЧЕСКИЙ КОЛЛЕДЖ*

## *Преобразование ER-модели в реляционную модель*

Концептуальная модель позволяет понять суть разрабатываемой базы данных, но она не подходит для непосредственной реализации структуры БД. Необходимо преобразование концептуальной модели в логическую с учетом особенностей выбранной СУБД.

Если выбор СУБД остановился на реляционной СУБД, то особенности реляционной модели данных достаточно жестко описывает стандарт языка SQL, поэтому переход к реляционной модели, в общем случае, можно свести к ряду последовательных преобразований.

Для ER-модели существует алгоритм однозначного преобразования в реляционную модель данных, что позволило разработать множество инструментальных систем (**САПР – система автоматизированного проектирования**), поддерживающих процесс разработки информационных систем, базирующихся на технологии баз данных.

**Алгоритм перехода от ER-модели к реляционной модели данных обычно сводится к следующим шагам:**

1. Каждой сущности ставится в соответствие отношение РМД. Имена отношений могут быть ограничены требованиями конкретной СУБД, они ограничены по длине и не должны содержать пробелов и некоторых специальных символов.

2. Каждый атрибут сущности становится атрибутом соответствующего отношения, на их имена также могут накладываться некоторые ограничения (например, многие СУБД не поддерживают кириллицу). Для каждого атрибута задается конкретный допустимый в СУБД тип данных и обязательность или необязательность данного атрибута. На рисунке, упрощенно, показан процесс преобразования, с учетом выбранной СУБД (в правой части отношения указаны типы данных принятые в СУБД MySQL для соответствующих полей).

Книги
Номер
Автор
Название
Издательство
Стоимость



Books	
IDb	INT(10)
bAuthor	VARCHAR(255)
bName	VARCHAR(255)
bPublisher	VARCHAR(255)
bCost	INT(10)

3. Первичный ключ сущности становится первичным ключом соответствующего отношения (на рисунке первичный ключ выделен желтым цветом). Атрибуты, входящие в первичный ключ отношения, автоматически получают свойство обязательности и уникальности.

4. В каждое отношение, соответствующее подчиненной сущности, добавляется набор атрибутов первичного ключа главной сущности. В отношении, соответствующем подчиненной сущности, этот набор атрибутов становится *внешним ключом* (атрибут IDb таблицы «Склад»).

Books (Книги)	
<b>IDb</b>	INT(10)
bAuthor	VARCHAR(255)
bName	VARCHAR(255)
blzd	VARCHAR(255)
bCost	INT(10)



Store (Склад)	
<b>IDs</b>	INT(10)
IDb	INT(10)
sCount	INT(10)

5. Для необязательных типов связи на физическом уровне у атрибутов, соответствующих внешнему ключу, устанавливается свойство допустимости неопределенных значений. При обязательном типе связи атрибуты получают свойство отсутствия неопределенных значений.

6. При реализации связи *многие-ко-многим*, допустимой в концептуальной модели, производится ее преобразование к связям *один-ко-многим*, например, через промежуточное отношение (связующее

Books (Книги)	
<b>IDb</b>	INT(10)
bAuthor	VARCHAR(255)
bName	VARCHAR(255)
bIzd	VARCHAR(255)
bCost	INT(10)

Zakaz (Связующая таблица «Заказ»)	
<b>IDp</b>	INT(10)
IDb	INT(10)
IDc	INT(10)

Client (Клиент)	
<b>IDc</b>	INT(10)
cName	VARCHAR(255)
cPhone	VARCHAR(255)



После выполнения преобразований необходимо убедиться в корректности полученной схемы БД, в противном случае в таблицах могут оказаться нежелательные функциональные зависимости, которые впоследствии могут привести к возникновению различных аномалий. Проверить полученную схему БД, на отсутствие нежелательных функциональных зависимостей, можно используя правила нормализации.

На данном этапе выполняется *описание внешних моделей* в терминах РСУБД. Под внешними моделями подразумевается совокупность моделей данных для каждого приложения, использующего БД. Например, для БД «Книжный магазин» можно выделить три основных внешних модели: для приложения продавца, администрации и клиента. Каждое из этих приложений будет решать свои прикладные задачи, используя для этого не всю БД, а лишь требуемую ее часть.

После создания схемы БД и проверки ее корректности необходимо описать так называемые *бизнес-правила*.

**Бизнес-правила** (БП) задают ограничения на значения данных в БД. Они также определяют механизмы, согласно которым при изменении одних данных изменяются и связанные с ними данные в той же или других таблицах БД. Таким образом, бизнес правила определяют условия поддержания БД в целостном состоянии. На первом этапе проектирования, т.е. при описании предметной области определяются и ограничительные условия, правда, на данном этапе они реализуются уже в рамках реляционной модели данных.



Преподаватель: Яскина А.Г.

© Сибирский государственный университет  
науки и технологий имени академика М. Ф.  
Решетнева, 2017