Проектирование баз данных

"Сложная система, спроектированная наспех, никогда не работает, и исправить её, чтобы заставить работать, невозможно".

Законы Мерфи. 16-й закон системантики

Этапы проектирования БД

I. Информационно-логическое проектирование (предыдущая лекция)
- анализ предметной области;
- построение модели предметной области;
- определение границ информационной поддержки;
- определение групп пользователей.

II. Определение требований к операционной обстановке:
- выбор аппаратной платформы;
- выбор операционной системы.

III. Выбор СУБД и других инструментальных программных средств.
- выбор СУБД;
- выбор версии СУБД и архитектуры, в которой она будет работать.

IV. Логическое проектирование БД (даталогическое):
- преобразование схемы предметной области в схему базы данных;
- создание схем отношений;
- нормализация отношений.

V. Физическое проектирование БД:
- реализация проекта на DDL-языке выбранной СУБД;
- создание дополнительных объектов БД (индексов, представлений, триггеров и др.).

Определение требований к операционной обстановке

На этом этапе производится:

- ✓ оценка требований к вычислительным ресурсам, необходимым для функционирования системы;
- ✓ выбор типа и конфигурации ЭВМ;
- ✓ выбор типа и версии операционной системы (ОС).

Выбор зависит от таких показателей, как:

- ✓ примерный объём данных в БД;
- ✓ динамика роста объёма данных;
- ✓ характер запросов к данным (извлечение и обновление отдельных записей, обработка групп записей, обработка отдельных отношений или соединение отношений);
- ✓ интенсивность запросов к данным по типам запросов;
- ✓ требования ко времени отклика системы по типам запросов;
- ✓ режим работы (интерактивный, пакетный или режим реального времени).

Выбор СУБД

Наиболее важные критерии выбора СУБД:

- ✓ тип модели данных, которую поддерживает данная СУБД, адекватность модели данных структуре рассматриваемой ПО;
- ✓ характеристики производительности СУБД;
- ✓ запас функциональных возможностей для дальнейшего развития информационной системы;
- ✓ степень оснащённости СУБД инструментарием для персонала администрирования данными;
- ✓ удобство и надежность СУБД в эксплуатации;
- ✓ наличие специалистов по работе с конкретной СУБД;
- ✓ стоимость СУБД и дополнительного программного обеспечения.

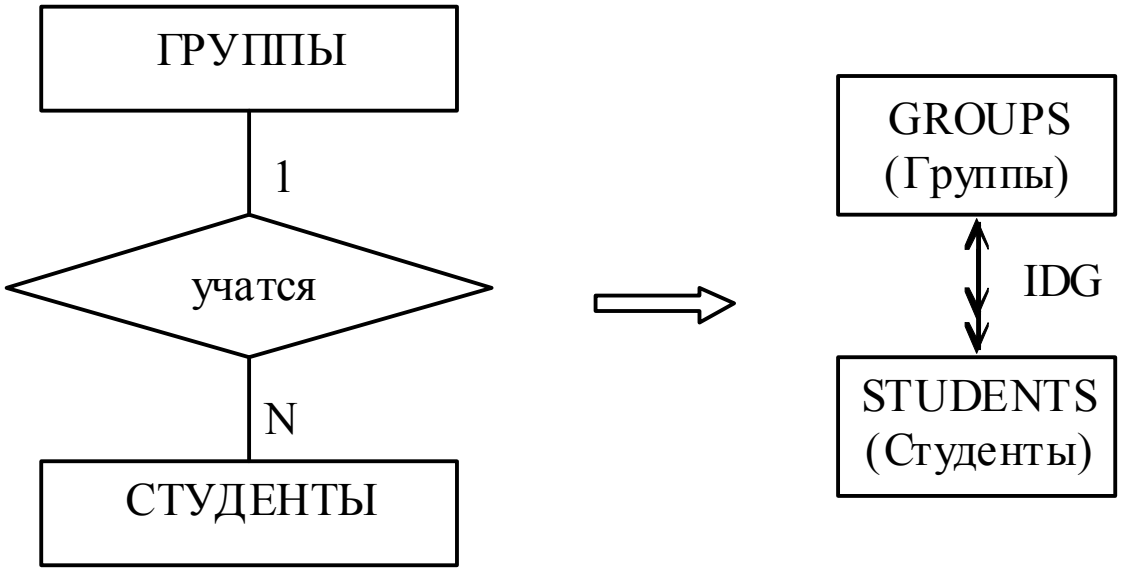
Также может выбираться дополнительное программное обеспечение (например, CASE-средства: ERWin, BPWin и т.п.).

Логическое проектирование РБД

Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

1. Каждый тип сущности преобразуется в таблицу БД. В таблицу вносятся все атрибуты, относящиеся к данному типу сущности.
2. Бинарная связь 1:n (между сущностями **разных типов**) реализуется с помощью внешнего ключа между двумя таблицами

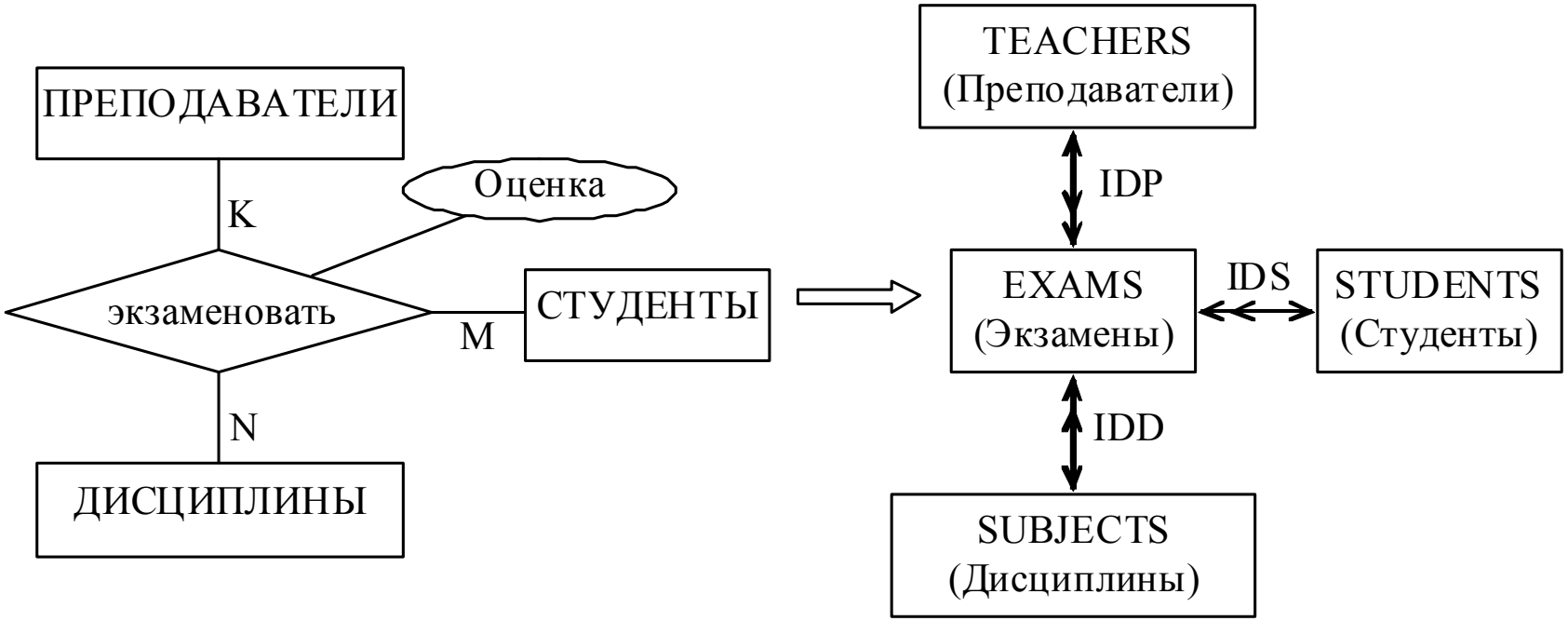


Логическое проектирование РБД

Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

3. Каждая связь со степенью больше двух и связь, имеющая атрибуты, преобразуется в таблицу БД.

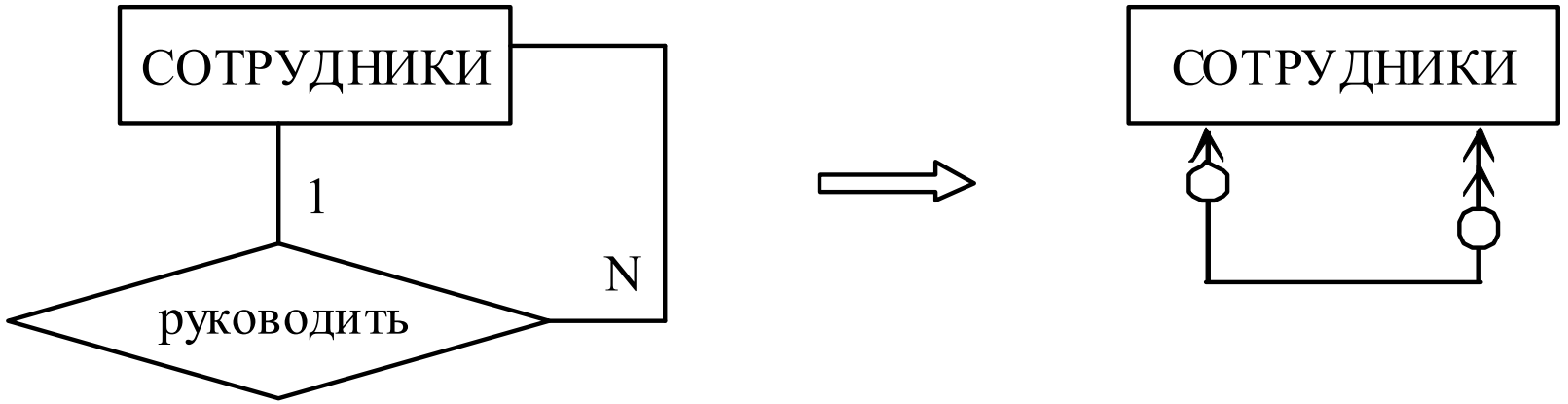


Логическое проектирование РБД

Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

4. Связь 1:1 реализуется в рамках одной таблицы. Исключение из этого правила составляют ситуации, когда связанные сущности существуют независимо друг от друга.
5. Унарная связь 1:n (между сущностями **одного типа**) реализуется с помощью внешнего ключа, определённого в той же таблице, что и первичный ключ.

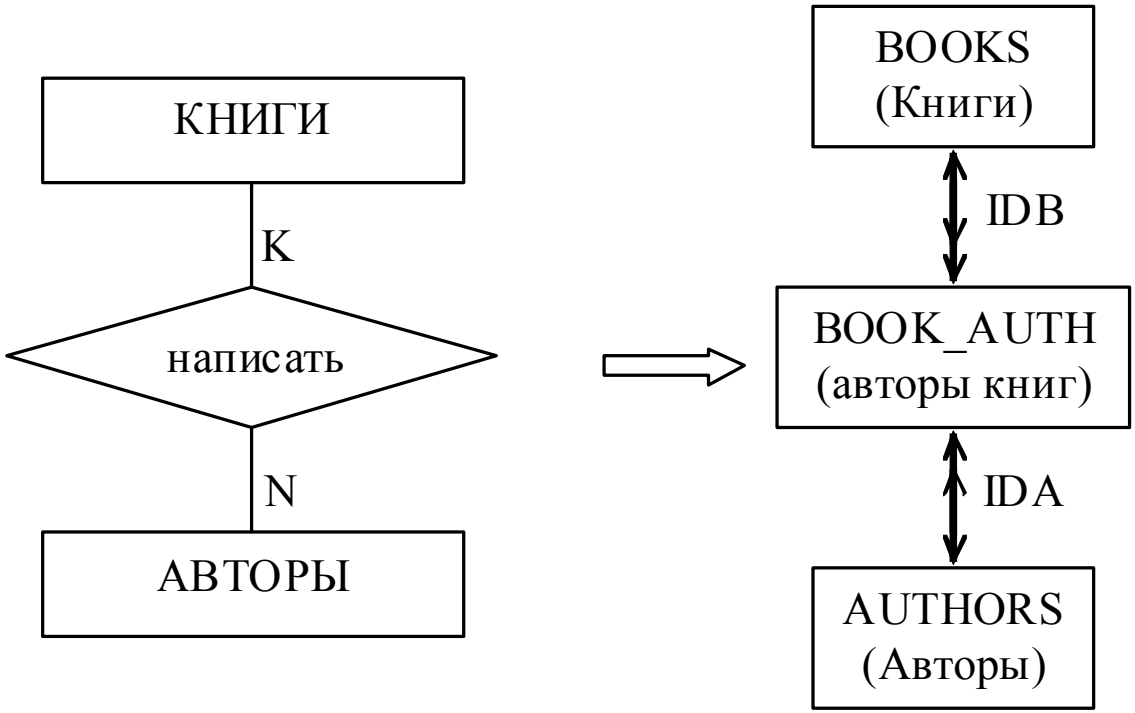


Логическое проектирование РБД

Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

6. Бинарная связь типа n:m реализуется с помощью промежуточной таблицы.

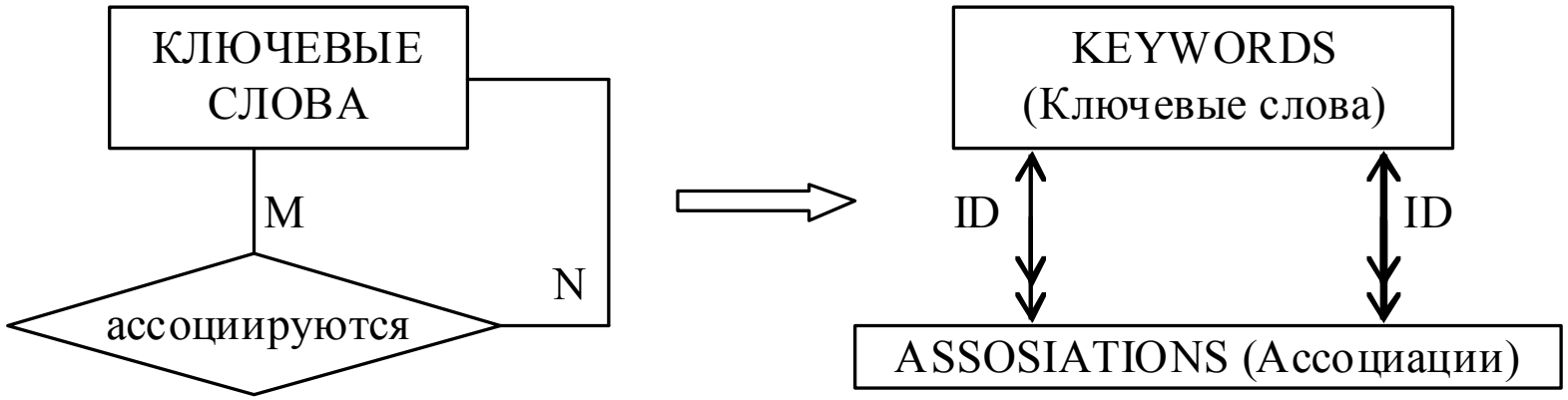


Логическое проектирование РБД

Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

7. Унарная связь n:m реализуется с помощью промежуточной таблицы.



На этом этапе возможно ещё выявление нереализуемых и необычных связей (связи 1:n, обязательные в обе стороны; взаимоисключающие связи и др.).

Логическое проектирование РБД

Составление схем отношений.

Определение первичных ключей (ПК):

1. При наличии потенциальных ключей ПК выбирается из них. Обычно, берется тот ключ, по которому чаще всего происходит обращение к данным. Если такого нет, то выбирается ключ, занимающий меньше памяти.
2. Если потенциальных ключей нет, назначается суррогатный ПК (он не несет смысловой нагрузки). Некоторые СУБД позволяют определять значения такого ключа как AUTOINCREMENT, т.е. числовое поле, значение которого начинается с 1 и автоматически увеличивается на 1 при добавлении новой записи.
3. Составной ПК назначается в том случае, если необходимо реализовать ограничение целостности "уникальность".

Логическое проектирование РБД

Определение типов данных атрибутов**. Общие рекомендации:**

- ✓ Для коротких символьных значений и символьных строк фиксированной длины следует выбирать тип CHAR.
- ✓ Для символьных строк переменной длины нужно выбирать тип VARCHAR с указанием максимально возможной длины хранимого значения.
- ✓ Для числовых атрибутов, не участвующих в сложных расчётах, нужно использовать основной числовой тип реляционных СУБД – тип NUMBER, указывая реально необходимое количество разрядов.
- ✓ Для числовых атрибутов, которые участвуют в сложных расчётах, следует использовать такие числовые типы, которые хранят данные в машинном (двоичном) представлении.
- ✓ Для числовых атрибутов, имеющих ведущие нули, следует выбирать тип CHAR, а не числовой тип, иначе ведущие нули будут потеряны.
- ✓ Для хранения дат нужно выбирать тип DATE или его варианты (DATETIME, например).
- ✓ Для хранения больших объектов (графических, звуковых и т.п.) следует выбирать специальные типы данных, перечень которых зависит от СУБД.
- ✓ Для семантически одинаковых полей разных таблиц нужно выбирать одинаковые типы данных. Во многих СУБД для упрощения типизации данных можно создать специальные типы данных (create type) и использовать их в качестве типов полей таблиц.

Логическое проектирование РБД

Определение и реализация ограничений целостности:

Если какое-либо ограничение целостности может быть включено в структуру БД (на языке DCL), то его надо реализовать именно так!

Рассмотрим различные типы ограничений целостности в языке SQL:

- ✓ Уникальность значения первичного ключа (PRIMARY KEY).
- ✓ Уникальность ключевого поля или комбинации значений ключевых полей (UNIQUE).
- ✓ Обязательность/необязательность значения (NOT NULL/NULL).
- ✓ Задание условия на значения атрибутов (CHECK).
- ✓ Определение домена атрибута на основе значений другого атрибута: множество значений некоторого атрибута отношения является подмножеством значений другого атрибута этого или другого отношения (внешний ключ, FOREIGN KEY).

Логическое проектирование РБД

Определение и реализация ограничений целостности.

Если какое-либо ограничение целостности (ОЦ) нельзя реализовать средствами DCL, то возможны следующие способы его реализации:

- ✓ С помощью процедурных объектов БД (триггеров, trigger).
- ✓ Программно (т.е. через приложение). Для большей гарантии соблюдения ОЦ желательно проектировать программу так, чтобы внесение изменений в данные и проверка ОЦ выполнялись в одном единственном месте.
- ✓ Вручную. Ручная процедура обязательно должна быть описана в документации (в руководстве пользователя).

Необходимо обратить особое внимание на поля таблиц, для которых домен определён как список возможных значений. Это ограничение целостности можно реализовать в виде: CHECK(<поле> IN (<список значений>)).

Но такой подход имеет следующий недостаток: добавление нового значения в список потребует изменения схемы отношения (команда ALTER TABLE). Можно поступить до-другому: вынести этот список значений в отдельное отношение. Например, список типов образования (начальное, неполное среднее, среднее, средне-специальное, незаконченное высшее, высшее) для таблицы *СОТРУДНИКИ*. Таблица *ТИПЫ ОБРАЗОВАНИЯ* будет состоять из одного поля *Название типа*, определённого как первичный ключ. Тогда поле *Образование* таблицы *СОТРУДНИКИ* станет внешним ключом.

Физическое проектирование РБД

При использовании СУБД Oracle примерная последовательность создания объектов БД следующая:

1. Создание БД (create database).
2. Создание пользователей (create user).
3. Создание пользовательских типов (create type).
4. Создание кластеров и таблиц (create cluster, create table).
5. Создание представлений (create view).
6. Создание синонимов (create synonym).
7. Создание последовательностей (create sequence).
8. Назначение прав доступа (grant).
9. Заливка данных (Oracle Loader, imp.exe,…).
10. Создание индексов (create index).
11. Создание процедур и функций (create procedure, create function).
12. Создание триггеров (create trigger).