**Логическое проектирование БД**

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели предметной области. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Будем рассматривать логическое проектирование БД для реляционной модели данных.

Проектирование реляционной БД проходит в том же порядке, что и проектирование БД других моделей, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных.

Они обусловлены отсутствием средств явного представления типов множественных связей между объектами предметной области и неразвитостью средств описания ограничений целостности на уровне модели данных.

На этом этапе выполняются следующие действия:

* удаление связей M \N \
* рекурсивных связей;
* связей с атрибутами;
* множественных атрибутов;
* избыточных связей;
* перепроверка связей типа 1:1.

Далее выполняется нормализация отношений. В рамках реляционной модели данных Э.Ф. Коддом был разработан аппарат нормализации отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей нормальной форме.

Нормализация отношений выполняется на основе анализа первичных ключей и существования функциональных зависимостей между атрибутами.

Функциональная зависимость атрибута В от атрибута А – такая зависимость атрибутов, при которой каждому значению атрибута А соответствует одно значение атрибута В.

Как правило, нормализация выполняется в несколько этапов. Каждый этап соответствует определенной нормальной форме (НФ). При проектировании реляционных БД требования первой нормальной формы (1НФ) должны выполняться всегда, остальные – по желанию проектировщика. Однако, чтобы исключить аномалии обновления и избыточность данных, рекомендуется приводить отношение к третьей нормальной форме (ЗНФ).

В качестве примера рассмотрим отношение «Продажа». Последовательно применяя к отношению правила нормализации, приведем однотабличную БД к «нормальному» виду.

Будем рассматривать только три нормальные формы, так как они чаше всего применяются на практике, рисунок 1.

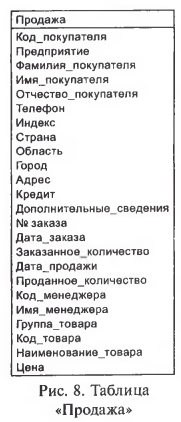


Рисунок 1 – Таблица «Продажа»

Таблица «Продажа» содержит следующую информацию:

1) сведения о покупателе;

2) дату заказа и количество проданного товара;

3) дату выполнения заказа;

4) характеристику проданного товара.

Основная проблема хранения данных в одной таблице – значительное число повторяющейся информации.

Например, сведения о каждом покупателе будут повторяться для каждого сделанного им заказа.

Проблемы, которые при этом возникают:

* затраты времени на ввод повторяющихся записей;
* неоправданное увеличение БД;
* увеличение времени для получения нужной информации.

Первая нормальная форма (1НФ)

Таблица в 1НФ должна удовлетворять следующим требованиям:

1) отсутствие повторяющихся записей;

2) отсутствие повторяющихся групп полей;

3) атрибуты и строки не упорядочены;

4) значения атрибутов атомарны.

Для удовлетворения 1-го условия таблица должна иметь первичный ключ, для 2-го – ее необходимо разбить на сущности. Последующие условия исходят из свойств отношения.

Предлагаются три сущности, которые будут описывать следующую информацию:

* сведения о покупателе;
* когда, кем и для какого покупателя оформлен заказ;
* какие товары вошли в заказ.

Каждой таблице необходимо определить первичный ключ и установить связи, рисунок 2.



Рисунок 2 – Первая нормальная форма

Вторая нормальная форма (2НФ)

О таблице говорят, что она находится во 2НФ, если:

1) она удовлетворяет условиям 1НФ;

2) любое неключевое поле однозначно идентифицируется полным набором ключевых полей.

В таблице «Составзаказа» имеется составной ключ. Данная таблица не является таблицей во 2НФ, так как поля «Наименованиетовара», «Группа\_товара», «Цена» однозначно определяются только одним атрибутом, входящим в состав первичного ключа.

Схема БД изменится, так как таблица «Состав заказа» разобьется на две таблицы: «Состав\_заказа» и «Товар», рисунок 3.



Рисунок 3 – Вторая нормальная форма

Третья нормальная форма (ЗНФ)

О таблице говорят, что она находится в ЗНФ, если:

1) она удовлетворяет условиям 2НФ;

2) ни одно из неключевых полей таблицы не идентифицируется с помощью другого неключевого поля.

Таблица «Заказ» имеет атрибут «Имя менеджера», который однозначно определяется другим неключевым атрибутом «Код\_менеджера». Таблица «Заказ» не является ЗНФ. Необходимо выделить таблицу «Менеджеры» (рис. 11).

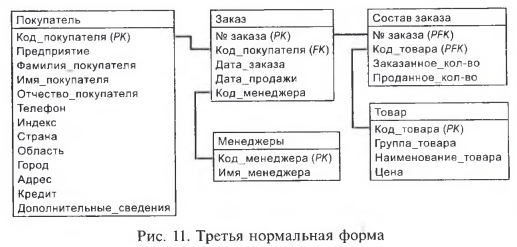


Рисунок 4 – Третья нормальная форма