**База данных** – это совокупность взаимосвязанных данных.

**Нормализация данных** – процесс преобразования таблиц базы данных к нормальной форме; **(Шесть нормальных форм** – 1NF, 2NF, ...6NF). Широкое практическое применение имеют формы 1NF, 2NF, 3NF.

**Первая нормальная форма:** Таблица не должна содержать повторяющихся групп данных**.** Атомарность – каждый столбец должен содержать одно неделимое значение**.** Чтобы таблица соответствовала **1-й нормальной форме** (1NF), необходимо, чтобы все значения ее полей были неделимыми и не вычисляемыми, а все записи – уникальными (не должно быть полностью совпадающих строк).

Устранить повторяющиеся группы в отдельных таблицах. Создать отдельную таблицу для каждого набора связанных данных. Идентифицировать каждый набор связанных данных с помощью первичного ключа.

**Вторая нормальная форма:** Чтобы таблица соответствовала **2-й нормальной форме** (2NF), необходимо, чтобы она находилась в 1-й нормальной форме и все не ключевые поля полностью зависели от ключевого.

Создать отдельные таблицы для наборов значений, относящихся к нескольким записям. Связать эти таблицы с помощью внешнего ключа.

**Третья нормальная форма:** Таблица находится во второй нормальной форме.Отсутствуют транзитивные зависимости. Для перехода к **3-й нормальной форме** (3NF) необходимо обеспечить, чтобы все таблицы находились во 2-й нормальной форме и все не ключевые поля в таблицах не зависели взаимно друг от друга.

MSS поддерживает следующие типы данных: числовые, символьные, для хранения даты и времени, денежные, двоичные и специальные.

| **Тип данных** | **Описание** | **Тип данных** | **Описание** |
| --- | --- | --- | --- |
| **integer**  **(int)** | Целочисленные значения, занимают 4 байта. | **date** | Дата, занимает 3 байта |
| **smallint** | Целочисленные значения, занимают 2 байта | **time[(p)]**  **0 ≤ p ≤ 7** | Время, занимает от 3 до 5 байтов. **p** − колич. знаков после точки в секундах |
| **tinyint** | Неотрицательные целочисленные значения, занимают 1 байт | **smalldatetime** | Дата и время, занимает 2 байта |
| **bigint** | Целочисленные значения, занимают 8 байт | **datetime** | Дата и время, занимает 4 байта |
| **real** | Вещественные числа с плавающей точкой | **datetime2** | Дата и время, занимает 8 байтов |
| **decimal (p[,s] )**  **(dec)**  или  **numeric (p[,s] )** | Вещественные значения с фикс. точкой, **р** − общее количество цифр, **s** − количество цифр после точки. Занимает от 5 до 17 байт. | **сhаr[(n)]** | Строки фиксированной длины из однобайтовых симво­лов, n − количество символов |
| **float[(p)]** | Вещественные числа с плавающей точкой. Если **р** < 25 − одинарная точность (4 байта), при **р** > 25 − двойная точность (8 байтов) | **nchar[(n)]** | Строки фиксированной длины символов Unicode. Каждый символ занимает 2 байта. |
| **money** | Денежные значения, занимают 8 байтов | **varchar[(n)]** | Строки переменной длины из однобайтовых симво­лов |
| **smallmoney** | Денежные значения, занимают 4 байта | **nvarchar[(n)]** | Строки переменной длины символов Unicode − 2 байта |
| **binary(n)** | Задает битовую строку, длиной ровно **n** байтов | **varbinary(n)** | Задает битовую строку, длиной не более **n** байтов |

Виды ключей:

* первичный ключ;
* внешний ключ;
* простой и составной ключ;

Первичный ключ – это поле, которое используется для обеспечения уникальности данных в таблице. Это означает, что значение (информация) в ближайшем будущем выключает в каждой таблице (запись) таблицы, которая может быть представлена.

Внешний ключ — это столбец (или несколько столбцов), который ссылается на первичный ключ таблицы.

Если первичный ключ состоит из единственного атрибута, его называют **простым ключом**.

Если первичный ключ состоит из двух и более атрибутов, его называют **составным ключом**.