

В основе реляционных систем лежит
математический аппарат - *реляционная
алгебра.*

Отношение (relation) — это двумерная таблица, которая соответствует некоторой сущности. Каждый столбец таблицы описывает какой-либо атрибут этой сущности. Иногда строки называются записями или кортежами (tuples), а столбцы — атрибутами (attributes).

Синонимы

Реляционная модель	Программист	Пользователь
Отношение	Файл	Таблица
Кортеж (строка)	Запись	Строка
Атрибут	Поле	Столбец

Реляционная база данных - это совокупность отношений или двумерных таблиц.

В таблицах базы данных хранятся не только сами данные, но и описания этих данных — *метаданные*.

Чтобы таблица могла принадлежать
реляционной базе данных, она должна иметь
определенные свойства и удовлетворять
некоторым ограничениям.

- Каждый столбец имеет уникальное имя; порядок столбцов в таблице несуществен.
- Все записи в столбце должны быть одного типа.

НомЗач	Фамилия	Имя	Отчество	ДатРожд
100747	Алексеев	Георгий	Вадимович	27.11.1992
100754	Меленцов	Иван	Антонович	23.06.1993
100757	Назаров	Григорий	Георгиевич	29.10.1992

Основные типы данных делятся на
встроенные и определенные пользователем.

- Числовые.
- Строковые.
- Даты и времени.
- Логические.
- Специальные.

Для столбцов с данными можно устанавливать порядок сортировки.

В значениях столбца можно использовать неизвестное (неопределенное) значение NULL.

В таблице можно создавать ключи:

- а) первичный ключ (primary key),
- б) уникальный ключ (unique),
- с) внешний ключ (foreign key).

Ключ (key) — это группа из одного или более атрибутов, которая уникальным образом идентифицирует строку.

С помощью ключей между таблицами можно создавать связи (отношения, relationship): «внешний ключ/первичный ключ» и «внешний ключ/уникальный ключ». Такие связи между таблицами называются *декларативной целостностью данных* (declarative data integrity).

Кроме ключей в таблице могут существовать другие ограничения (constraint) :

- а) ограничение на значения, помещаемые в столбцы таблицы - CHECK
- б) значение по умолчанию – DEFAULT
- с) допустимость для столбца значения NULL.

Таблица может содержать произвольное количество записей (строк), которое ограничивается только объемом внешней памяти.

Не может быть двух одинаковых строк, и порядок строк не имеет значения.

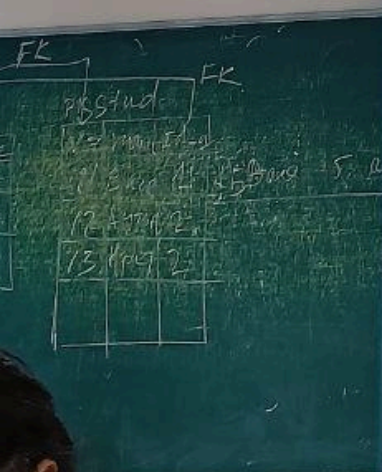


Структурой таблицы называется множество пар
<поле, тип поля>.
Имя этого множества – *имя таблицы*.

FK

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

Нулевое правило Е.Ф.Кодда: «Любая система, объявленная как система управления реляционной базой данных, должна управлять этой базой данных, опираясь только на ее реляционные свойства»



Типы данных в таблицах

- `char(n)` – имеет фиксированную длину. 1 байт на символ.

До 8000 символов.

Пример. Если `char(10)` то "abc" хранится как "abc ".

- `nchar(n)` – аналогично, но кодировка Unicode а не ANSI.

Типы данных в таблицах

- **varchar(n)** – кодировка ANSI, но в отличие от **char(n)** может содержать различное количество символов в пределах указанной максимальной длины.

Пример. Если **varchar(10)** то "abc" и хранится как "abc".

- **nvarchar(n)** – отличается от **varchar(n)** использованием формата Unicode.

Типы данных в таблицах

- **text** – хранит данные с длиной более 8000 символов.
- **ntext** – Unicode – версия text. Вместо них рекомендуется использовать `varchar(max)` и `nvarchar(max)` соответственно.

Типы данных в таблицах

- `varchar(max)` – символьные данные до $(2^{31}-1)$ символов.
- `nvarchar(max)` – символьные данные Unicode до $(2^{30}-1)$.
- `varbinary(max)` – бинарные данные переменной длины до $(2^{31}-1)$ байтов.



Типы данных в таблицах

Целочисленные

- **int** - диапазон от -2 147 483 648 до 2 147 483 647. (2^{31})
- **bigint** - диапазон от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807. (2^{63})

Типы данных в таблицах

Целочисленные

- `smallint` - диапазон от -32 768 до 32 767. (2^{15})
- `tinyint` - диапазон от 0 до 255. (2^8)
- `bit` - 0, 1, NULL.

Типы данных в таблицах

Десятичные числа

- `decimal/numeric(m,n)` - диапазон от $-10^{38}+1$ до $10^{38}-1$.

Замечание. Хранить можно только 38 цифр и меньше.

Типы данных в таблицах

Приближенные числовые

- `float` – для чисел с плавающей десятичной точкой.
Диапазон $-1.79 \cdot 10^{308}$ до $1.79 \cdot 10^{308}$.
- `real` - Диапазон $-3.4 \cdot 10^{38}$ до $3.4 \cdot 10^{38}$.

Типы данных в таблицах Денежные

`money` – тип данных для хранения числовых значений с точностью до четырех десятичных разрядов после точки от -922 337 203 685 477.5808 до 922 337 203 685 477.5807 не содержит знака \$.

`smallmoney` – от -214 748.3648 до 214 748.3647

Типы данных в таблицах

Денежные

money – тип данных для хранения числовых значений с точностью до четырех десятичных разрядов после точки от -922 337 203 685 477.5808 до 922 337 203 685 477.5807 не содержит знака \$.

smallmoney – от -214 748.3648 до 214 748.3647

Типы данных в таблицах

Дата и время

- **date** – новый тип данных от 1 января 1 г .н.э. до 31 декабря 9999 г. в формате YYYY-MM-DD.
- **datetime** - от 1 января 1753 г до 31 декабря 9999 г. в формате YYYY-MM-DD hh:mm:ss. Если указать только дату то по умолчанию сохранится время 12:00:00.

Типы данных в таблицах

Дата и время

- **datetime2** - от от 1 января 1 г.н.э. до 31 декабря 9999 г. в формате YYYY-MM-DD hh:mm:ss[.nnnnnn].
- **smalldatetime** - от 1 января 1900 г. до 6 июня 2079 г.

Типы данных в таблицах

Дата и время

- **time** - новый тип в формате hh:mm:ss[.nnnnnn].
- **datetimeoffset** – данные в UTC (Coordinated Universal Time) в формате YYYY-MM-DD hh:mm:ss[.nnnnnn][+|-]hh:mm.

Типы данных в таблицах

- **binary(n)** – бинарные данные фиксированной длины до 8000 байтов.
- **varbinary(n)** – бинарные данные переменной длины до 8000 байтов. Размер данных в байтах + 2- байтовый указатель (реально введенных данных).
- **varbinary(max)** – бинарные данные переменной длины до $(2^{31}-1)$ байтов.

Типы данных в таблицах

- **hierarchyid** – составной тип данных на основе CLR (Common Language Runtime).
- **geometry** – планарный тип данных на основе CLR, который позволяет хранить географическую информацию на плоскости.
- **geography** – тип данных на основе CLR, который позволяет хранить географическую информацию “на шаре”.

Типы данных в таблицах

- **rowversion** – тип данных, который генерирует SQL Server при модификации записи. *Timestamp* в предыдущих версиях.
- **uniqueidentifier** – тип данных, который генерирует SQL Server и содержит GUID (Globally Unique Identifier) Создается на основе MAC, идентификатора процессора, даты и времени.

Типы данных в таблицах

- `xml` – для хранения данных XML.
- `cursor` – для хранения определенной структуры данных.
- `table` – для хранения определенной структуры данных.
- `sql_variant` – для хранения нескольких различных типов данных.

Хранение в SQL изображений и больших текстов

При необходимости следует хранить большие (< 1Мбайт) объемы текста вне SQL Server на жестком диске на сервере. Для хранения изображений или больших объемов двоичных объектов (BLOB Binary Large Object) в таблице при определении столбца как **varbinary(max)** можно хранить $2^{31}-1$ байтов данных или \approx 2Гбайт.

Хранение в SQL изображений и больших текстов

Для хранения данных вне таблицы необходимо дополнить этот тип данных параметром **FILESTREAM**. В базе данных поток данных должен быть активирован хранимой процедурой `sp_filestream_configure`.

Использование параметра **FILESTREAM** допускает более быстрое чтение данных по сравнению с чтением данных, хранящихся в таблице.

Отсутствующее значение NULL

Константа NULL вместо значения любого типа обозначает отсутствующее значение.

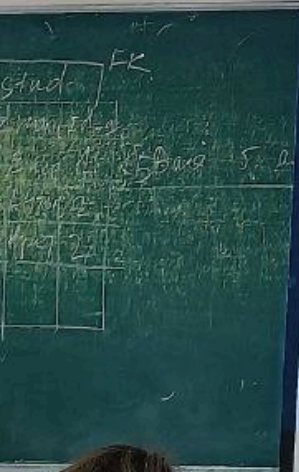
Семантика использования NULL такова:

- $NULL + x = NULL$
- $NULL - x = NULL$
- $SUBSTRING(NULL, 3) = NULL$

Отсутствующее значение NULL

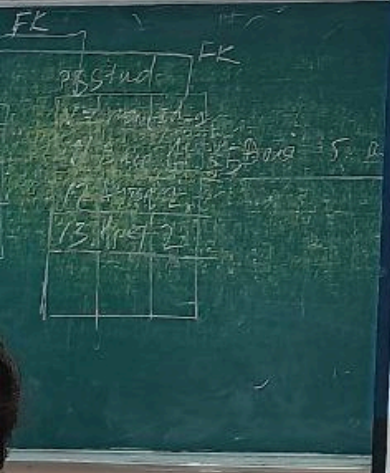
- $NULL < x = NULL$
- $NULL \text{ and } \text{ИСТИНА} = NULL$
- $NULL \text{ and } \text{ЛОЖЬ} = \text{ЛОЖЬ}$
- $NULL \text{ or } \text{ИСТИНА} = \text{ИСТИНА}$
- $NULL \text{ or } \text{ЛОЖЬ} = NULL$

Обеспечение целостности
данных



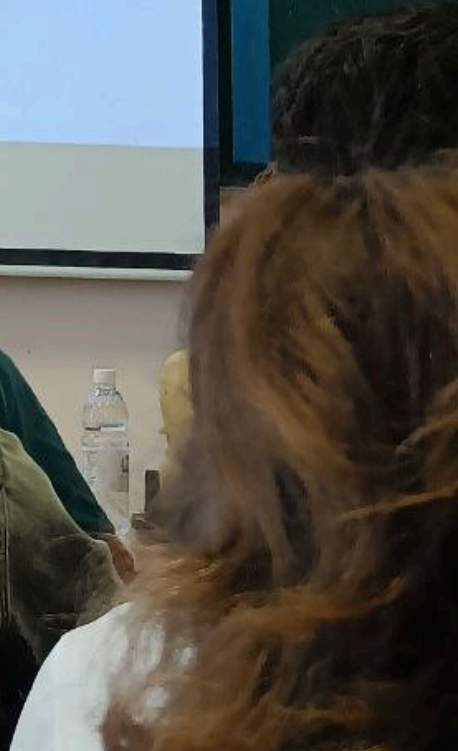
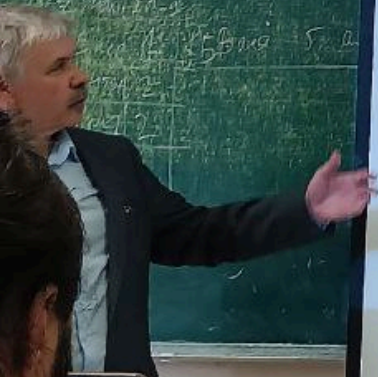
Обеспечение целостности данных

- Ограничения PRIMARY KEY
- Ограничения FOREIGN KEY
- Ограничения UNIQUE
- Ограничения CHECK
- Определения DEFAULT
- Разрешение значений NULL



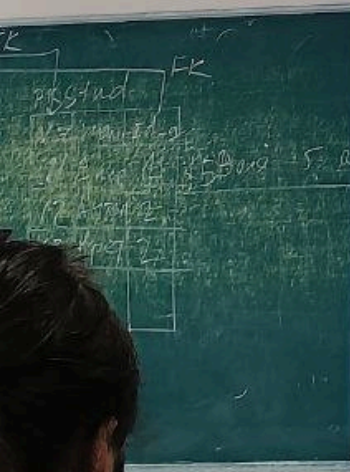
Ограничения PRIMARY KEY

При задании для таблицы ограничения PRIMARY KEY компонент Database Engine гарантирует уникальность данных созданием уникального индекса для столбцов первичных ключей. Этот индекс также обеспечивает быстрый доступ к данным при использовании первичного ключа в запросах. Таким образом, выбранные первичные ключи должны соответствовать правилам создания уникальных индексов.



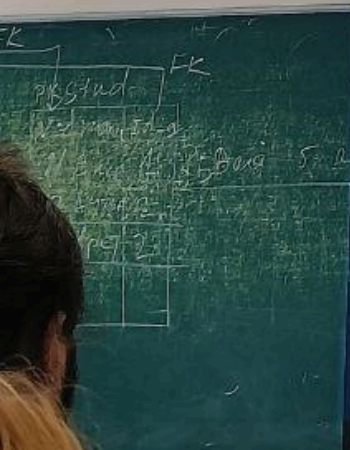
Ограничения FOREIGN KEY

Внешний ключ (FK) — это столбец или сочетание столбцов, которое применяется для принудительного установления связи между данными в двух таблицах.
Ограничение FOREIGN KEY не обязательно должно быть связано с ограничением PRIMARY KEY в другой таблице.

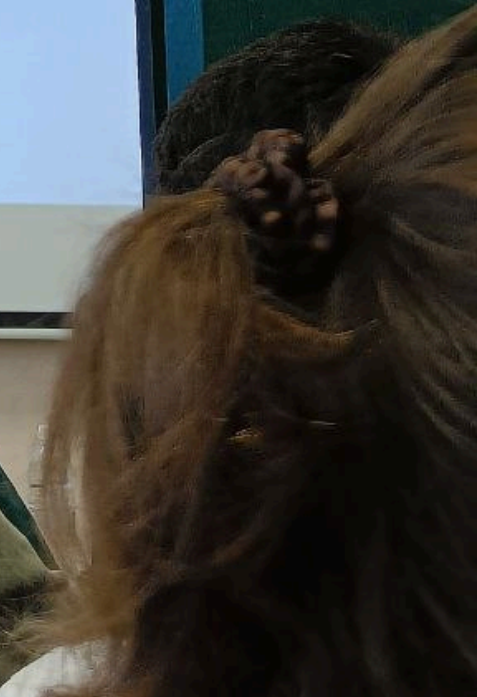
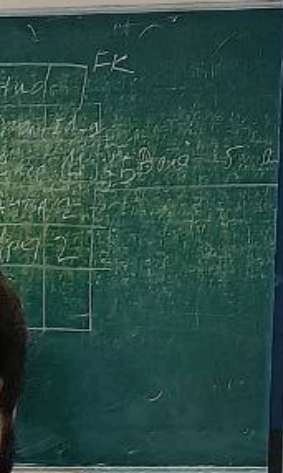


Ограничения FOREIGN KEY

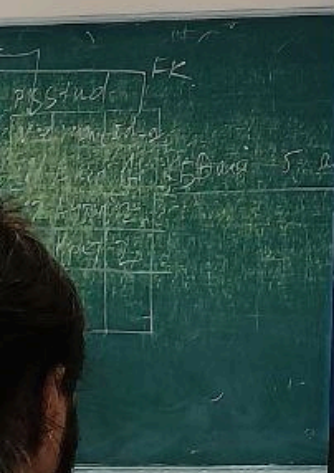
Кроме того, с помощью этого ограничения могут указываться столбцы ограничения UNIQUE в другой таблице. Ограничение FOREIGN KEY может содержать значения NULL.



Главная задача ограничения FOREIGN KEY — управление данными, которые могут быть сохранены в таблице внешнего ключа. Ограничение также контролирует изменение данных в таблице первичного ключа.



Ограничение запрещает изменение данных в таблице
первичного ключа, если такие изменения сделают
недопустимой ссылку в таблице внешнего ключа.



Если при попытке удалить строку в таблице первичного ключа или изменить значение этого ключа окажется, что удаленному или измененному значению первичного ключа соответствует значение в ограничении FOREIGN KEY в другой таблице, то действие выполнено не будет.

Ограничения UNIQUE

Ограничения UNIQUE можно использовать, чтобы убедиться, что в отдельные столбцы, не включенные в первичный ключ, не вводятся повторяющиеся значения. Для таблицы можно задать несколько ограничений UNIQUE, и только одно ограничение PRIMARY KEY.

Ограничения CHECK

Проверочные ограничения обеспечивают доменную целостность, ограничивая значения, которые может принимать столбец. Аналогично ограничениям внешнего ключа, они управляют значениями, которые присваиваются столбцу.

Ограничения CHECK

Однако они по-разному определяют допустимые значения: ограничения внешнего ключа получают список допустимых значений из другой таблицы, а проверочные ограничения определяют допустимые значения по логическому выражению, которое не основано на данных в другом столбце.

Определения DEFAULT

Каждый столбец записи должен содержать значение, даже если это значение равно NULL. Возможны случаи, когда необходимо поместить данные в таблицу, однако значение какого-либо столбца неизвестно, или же это значение не существует.

Если столбец допускает значения NULL, то можно поместить в него значение NULL. Поскольку иногда помещать в столбцы значения NULL нежелательно, лучшим решением может оказаться определение для столбца значения по умолчанию (DEFAULT). Например для числовых столбцов обычно в качестве значения по умолчанию указывается ноль, а для символьных — значение «н/д».

Разрешение значений NULL

Допустимость значения NULL в столбце определяет, могут ли строки таблицы содержать значения NULL для этого столбца. Значение NULL обозначает, что поле не было заполнено. Присутствие NULL, как правило, подразумевает, что значение неизвестно или неопределенно.

Рекомендуется избегать разрешать использование значений NULL, поскольку они усложняют выполнение запросов и процедур обновления. Кроме того, для столбцов со значением NULL не могут быть использованы некоторые параметры, например ограничения PRIMARY KEY.

Рекомендуется избегать разрешать использование значений NULL, поскольку они усложняют выполнение запросов и процедур обновления. Кроме того, для столбцов со значением NULL не могут быть использованы некоторые параметры, например ограничения PRIMARY KEY.

CONSTRAINT

Handwritten notes on a chalkboard to the left of the projector screen, including the letters "FK" and some illegible scribbles.

Handwritten notes on a chalkboard to the right of the projector screen, mostly illegible.

