БАЗЫ ДАННЫХ И ЭКСПЕРТНЫЕ СИСТЕМЫ

ФИО преподавателя: Тараканов О.В., канд. техн. наук, доцент

e-mail: tarakanov@mirea.ru

Канд. техн. наук, доцент ТАРАКАНОВ Олег Викторович

Базы данных и экспертные системы

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Учебные вопросы:

1. Декартово (прямое) произведение множеств. Отношение
2. Ключи реляционного отношения
3. Домены. Типы данных
4. Схема отношения

Декартово (прямое) произведение множеств. Отношение

Дано:

$D_1 = \{a_1, a_2, a_3\}$ - значимое свойство D_1, например, «Ф. И. О.»

$D_2 = \{b_1, b_2\}$ - значимое свойство D_2, например, «Дата рождения»

$D_3 = \{c_1, c_2, c_3\}$ - значимое свойство D_3, например, «Почтовый адрес»

Найти:

$R = D_1 \times D_2 \times D_3$

$$R = D_1 \times D_2 \times D_3 = \begin{Bmatrix} a_1 & b_1 \\ a_2 & b_1 \\ a_3 & b_1 \\ a_1 & b_2 \\ a_2 & b_2 \\ a_3 & b_2 \end{Bmatrix} \times \{c_1, c_2, c_3\} =$$

Заголовок таблицы ≡ «Схема отношения»

D_1	D_2	D_3
a_1	b_1	c_1
a_2	b_1	c_1
a_3	b_1	c_1
a_1	b_2	c_1
a_2	b_2	c_1
a_3	b_2	c_1
a_1	b_1	c_2
a_2	b_1	c_2
a_3	b_1	c_2
a_1	b_2	c_2
a_2	b_2	c_2
a_3	b_2	c_2
a_1	b_1	c_3
a_2	b_1	c_3
a_3	b_1	c_3
a_1	b_2	c_3
a_2	b_2	c_3
a_3	b_2	c_3

Кортеж ≡ «Запись»

Экземпляр объекта учета

Атрибут ≡ «Столбец»

Декартово (прямое) произведение множеств. Отношение

Виды атрибутов

$\langle \text{Атрибут} \rangle = \text{U}(\langle \text{Атрибут}_1 \rangle, \langle \text{Атрибут}_2 \rangle, \dots, \langle \text{Атрибут}_N \rangle)$ – **составной** атрибут, может быть разделен на атрибуты без потери смысла.

$\langle \text{Атрибут} \rangle$ – **атомарный** (простой) атрибут, не может быть разделен на атрибуты без потери смысла.

$\langle \text{Атрибут} \rangle$ – **именующий** (суррогатный) атрибут, абстрагирующий не существующее свойство объекта учета.

$\langle \text{Атрибут} \rangle$ – **ассоциативный** атрибут, абстрагирующий соединение отдельных экземпляров объектов учета (разрешение связи типа **M : N**).

$\langle \text{Атрибут}_1 \rangle = f(\langle \text{Атрибут}_2 \rangle)$ – **производный** атрибут, вычисляемый на основании значений других атрибутов. **Не допускается в отношениях!**

Свойства атрибутов

обязательные:

ИМЯ	должно быть уникальным в отношении и давать понятие об абстрагированном свойстве объекта учета
ДОМЕН (тип данных)	обеспечивает однородность данных исходного множества

не обязательные:

РАЗРЕШЕНИЕ ХРАНЕНИЯ NULL	допускается оставить без значения (NULL) или значение обязательно (NOT NULL)
ОГРАНИЧЕНИЕ ЦЕЛОСТНОСТИ	принадлежность к тому или иному ключу, вводящему ограничение (правило)
ПРАВИЛО ВАЛИДАЦИИ	автоматическая защита от ввода ошибочных (не корректных) значений
ПРАВИЛО DEFAULT	автоматический ввод заранее предусмотренного значения если ничего не введено пользователем

Декартово (прямое) произведение множеств. Отношение

С1. Дублирование строк таблицы не допускается.

С2. Порядок следования строк в отношении не существеннен.

К1. Каждое отношение имеет, по крайней мере, один ключ (состоящий из всех атрибутов).

К2. Значение ключа уникально идентифицирует кортеж отношения (не существует двух строк, которые имели бы равные значения атрибутов, входящих в ключ и рассматриваемых как одно целое).

К3. Никакое подмножество атрибутов ключа, которое образуется при удалении из ключа любого атрибута, не обладает свойством уникальности идентификации (свойств), что называется минимальностью ключа.

К4. Первичный ключ (один из потенциальных ключей отношения) не допускается обновлять или оставлять без значения.

Ноутбук

Артикул	Производитель	Модель
N1604A5	*ASUS*	*ROG se8*
N3211B1	*SONY*	*VAIO16*
N8210A4	*ACER*	*Aspire ONE*
~~***N8210A4***~~	~~*ACER*~~	~~*Aspire ONE*~~

тождественно

Ноутбук

Артикул	Производитель	Модель
N3211B1	*SONY*	*VAIO16*
N8210A4	*ACER*	*Aspire ONE*
N1604A5	*ASUS*	*ROG se8*

Сотрудник

Фамилия	Имя	Отчество
Семигорелов	*Иван*	*Петрович*
Семигорелов	*Иван*	*Вадимович*
Семигорелов	*Алексей*	*Вадимович*
Семигорелов	*Вадим*	*Петрович*

не достаточный ключ

не достаточный ключ

достаточный ключ (суперключ)

Сотрудник

Фамилия	Имя	Отчество
Семигорелов	*Иван*	*Петрович*
Семигорелов	*Иван*	*Вадимович*
~~***Семигорелов***~~	~~*Иван*~~	~~*Вадимович*~~
Семигорелов	*ИВАН*	*Вадимович*
Составной первичный ключ (Фамилия, Имя, Отчество)		

Не допустимое дублирование значения первичного ключа

Значения ASCII кодов атрибута первичного ключа отличаются от семантически идентичного значения

Сотрудник

Фамилия	Имя	Отчество
Семигорелов	*Иван*	*Петрович*
Семигорелов	*Иван*	*Вадимович*
Семигорелов	*Алексей*	*Вадимович*
Составной первичный ключ (Фамилия, Имя)		

Не допустимое дублирование значения первичного ключа

Сотрудник

Фамилия	Имя	Отчество
Семигорелов	*Иван*	*Петрович*
NULL	*Иван*	*Вадимович*
Семигорелов	***NULL***	*Вадимович*
Семигорелов	*Вадим*	***NULL***
Составной ключ отношения (Фамилия, Имя, Отчество)		

Не допустимое значение атрибута ключа отношения

Декартово (прямое) произведение множеств. Отношение

К5. С помощью ключей должны поддерживаться неявные связи между отношениями, отображающие семантику предметной области и обеспечивающие корректность транзакций (изменений в значениях атрибутов).

А1. Атрибуты отношений должны быть определены по типу (integer, char, BLOB, real,..) и формату представления.

А2. Диапазон области допустимых значений атрибута может быть ограничен с помощью средств языка запросов.

А3. При присвоении каждому столбцу уникального имени роли порядок столбцов также становится несущественным.

Условие связывания: Значение первичного ключа записи родительской таблицы равно значению внешнего ключа кортежа дочерней таблицы
"Производитель"."Идентификатор" = "Модель"."Идентификатор"

Производитель

Наименование	Идентификатор
Hewllet Packard	***Ap0013d***
Lenovo Corp. Ltd.	***D23-09ss***
ASUS Corp. Ltd.	***Jrt09/14a***
Потенциальный ключ	

Модель

Идентификатор	Модельный номер	Наименование
Jrt09/14a	*G752VT*	*ROG*
D23-09ss	*700-14ISK*	*YOGA*
Ap0013d	*640 G2*	*ProBook*
Ap0013d	*15-ae002ur*	*Envy*
D23-09ss	*100-15IBY*	*IdeaPad*
Внешний ключ	*Первичный ключ*	

Классификатор

Группа : Smallint

Экземпляр : Integer (4)
Каталожный номер : Smallint
Описание : LongRaw

Объект

Идентификатор : Number (8,2)
Наименование : Varchar2(40)
Группа : Smallint (FK)

Класс : Varchar2(24)
Дата поставки : Date

Задание типов данных для манипулирования и создание объектов базы данных:

```
declare
   type tinyint integer(2);
   xcount tinyint; //хост-переменная
begin
   create table counter (
        id_count tinyint notnull, //колонка таблицы
        count_name varchar2(50),
        description longraw);
   ...
end
```

Ноутбук

Артикул	Производитель	Модель
N3211B1	*SONY*	*VAIO16*
N8210A4	*ACER*	*Aspire ONE*
N1604A5	*ASUS*	*ROG se8*

тождественно

Ноутбук

Модель	Производитель	Артикул
VAIO16	*SONY*	***N3211B1***
Aspire ONE	*ACER*	***N8210A4***
ROG se8	*ASUS*	***N1604A5***

Ключи реляционного отношения

Ключ (key) для множества сущностей *E* (отношения *E*) – это множество *K*, состоящее из одного или более атрибутов множества *E*, такое, что при выборе из *E* любых двух различных сущностей (экземпляров) e_1 и e_2 последние не могут обладать одинаковыми значениями атрибутов, относящихся к множеству *K*.

$$R = D_1 \times D_2 \times D_3 = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{Bmatrix} \times D_2 \times D_3 = \begin{Bmatrix} a_1 b_i c_j \\ a_2 b_i c_j \\ a_3 b_i c_j \\ \dots \\ a_n b_i c_j \end{Bmatrix}$$

Виды ключей отношения

Потенциальный ключ (возможный ключ, *candidate key*) – атрибут или совокупность атрибутов, которые своими значениями однозначно отличают кортеж от остальных. Именно этот тип ключа однозначно идентифицирует каждый кортеж отношения.

Первичный ключ (*Primary key*) – один из потенциальных ключей отношения. Первичный ключ имеет область действия в рамках всего отношения.

Альтернативный ключ (*Alternate key*) – ключ с областью действия на уровне одного (группы атрибутов, объединенных ключом) атрибута. Отличает значение атрибута в одном кортеже от значения одноименного атрибута в другом кортеже.

Инверсионный вход (*Inversion Entry*) – ключ с областью действия только на уровне слота кортежа (ячейки таблицы базы данных). Применяется для проверки вводимых значений и сортировки кортежей отношения, отличающейся от заданной первичным ключом.

Суперключ – совокупность всех потенциальных ключей, в том числе выбранного первичного и альтернативных.

$$\boldsymbol{SK} \rightarrow \{\text{Потенциальный ключ}_{\boldsymbol{i}}\} \rightarrow \boldsymbol{PK} \rightarrow \{\boldsymbol{AK_j}\}$$

Внешний ключ (*Foreign key*) – ключ с областью действия в рамках связи между двумя отношениями, обеспечивающий согласованное ведение данных в связанных отношениях.

Ключ отношения – подмножество ключей, состоящее из первичного и всех внешних, участвующих в идентифицирующих связях.

$$\boldsymbol{K = PK + \{FK_k\}}$$

Домены. Типы данных

Домен – это область возможных значений атрибута. Может быть задан прямым перечислением, описанием, законом, функцией или любым другим способом, позволяющим однозначно задать данную область.

Домен **NUMBER**: множество всех чисел (положительных и отрицательных, целых и вещественных, рациональных и иррациональных).

Данные домена **NUMBER** занимают в памяти вычислительной системы, стандартно, 4 байта. Представляются в дополненном коде, могут быть усечены до **2** или **1** байтного представления.

Диапазон целых чисел, которые можно хранить в базе данных, обычно ограничен величинами свыше 2 млрд, как в положительном, так и отрицательном направлении оси целых чисел.

Типы данных СУБД PostgreSQL домена NUMBER

Имя типа данных	Размер, занимаемый в памяти	Описание	Диапазон
`smallint`	2 байта	целое в небольшом диапазоне	-32768 .. +32767
`integer`	4 байта	типичный выбор для целых чисел	-2147483648 .. +2147483647
`bigint`	8 байт	целое в большом диапазоне	-9223372036854775808 .. 9223372036854775807
`decimal`	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
`numeric`	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
`real`	4 байта	вещественное число с переменной точностью	точность в пределах 6 десятичных цифр
`double precision`	8 байт	вещественное число с переменной точностью	точность в пределах 15 десятичных цифр
`smallserial`	2 байта	небольшое целое с автоувеличением	1 .. 32767
`serial`	4 байта	целое с автоувеличением	1 .. 2147483647
`bigserial`	8 байт	большое целое с автоувеличением	1 .. 9223372036854775807

Домены. Типы данных

Домен **STRING**: статический или динамический массив допустимых символов, в последней ячейке содержащий 0-символ, обозначающий «конец строки». Для хранения строк при создании соответствующего атрибута должна быть указана наибольшая длина хранимой строки – размерность массива.

Выровненность строк

Обязательный параметр домена STRING – длина хранимой строки в символах ***n***. При задании домена для атрибута необходимо указать данный параметр ***n*** в явном виде. При вставке в базу данных строки, короче, чем предусмотрено параметром ***n***, в статических массивах оставшиеся позиции будут заполнены пробелами. В динамических массивах не занятые «концы» строк будут использованы для хранения других строк.

Типы данных СУБД PostgreSQL домена STRING

Имя типа данных	Размер, занимаемый в памяти	Описание	Диапазон
character varying(***n***), varchar(***n***)	2 байта на символ	строка ограниченной переменной длины	1 .. 10485760 символов
character(***n***), char(***n***)	2 байта на символ	строка фиксированной длины, дополненная пробелами	1 .. 10485760 символов
text		строка неограниченной переменной длины	
"char"	1 байт	**внутренний** однобайтный тип	1
name	64 байта	**внутренний** тип для имён объектов	

Для обеспечения возможности хранения символов в национальной раскладке клавиатуры, и в соответствие с форматом юникода UTF-8, каждый символ домена кодируется двумя октетами, следовательно, для его хранения в памяти требуется 2 байта.

Домены. Типы данных

Домен **DATETIME**: большое целое, хранящее число микросекунд (секунд), прошедших с условного начала отсчета к дате и времени, указанному в атрибуте.

Пользователю всегда представляется в отформатированном виде, как регулярная строка.
Формат представления задается шаблоном (маской), например: **'dd.mm.yyyy hh24:mi:ss'**, где:
'dd' – двухразрядное представление даты;
'mm' – двухразрядное представление порядкового номера месяца в году;
'yyyy' – четырехразрядное представление года;
'hh24' – двухразрядное представление числа часов в 24-часовом формате;
'mi' – двухразрядное представление числа минут;
'ss' – двухразрядное представление числа секунд.

Формат представления может быть задан в запросе, или используется настройка отображения дат сервера.

Типы данных домена **DATETIME**: **DATE, TIME, TIMESTAMP (with/without TIMEZONE), INTERVAL**

Домены. Типы данных

Специальные значения даты/времени

Вводимая строка	Допустимые типы	Описание
`epoch`	`date, timestamp`	1970-01-01 00:00:00+00 (точка отсчёта времени в Unix)
`infinity`	`date, timestamp`	время после максимальной допустимой даты
`-infinity`	`date, timestamp`	время до минимальной допустимой даты
`now`	`date, time, timestamp`	время начала текущей транзакции
`today`	`date, timestamp`	время начала текущих суток (00:00)
`tomorrow`	`date, timestamp`	время начала следующих суток (00:00)
`yesterday`	`date, timestamp`	время начала предыдущих суток (00:00)
`allballs`	`time`	00:00:00.00 UTC (Всемирное координированное время)

Замечание!

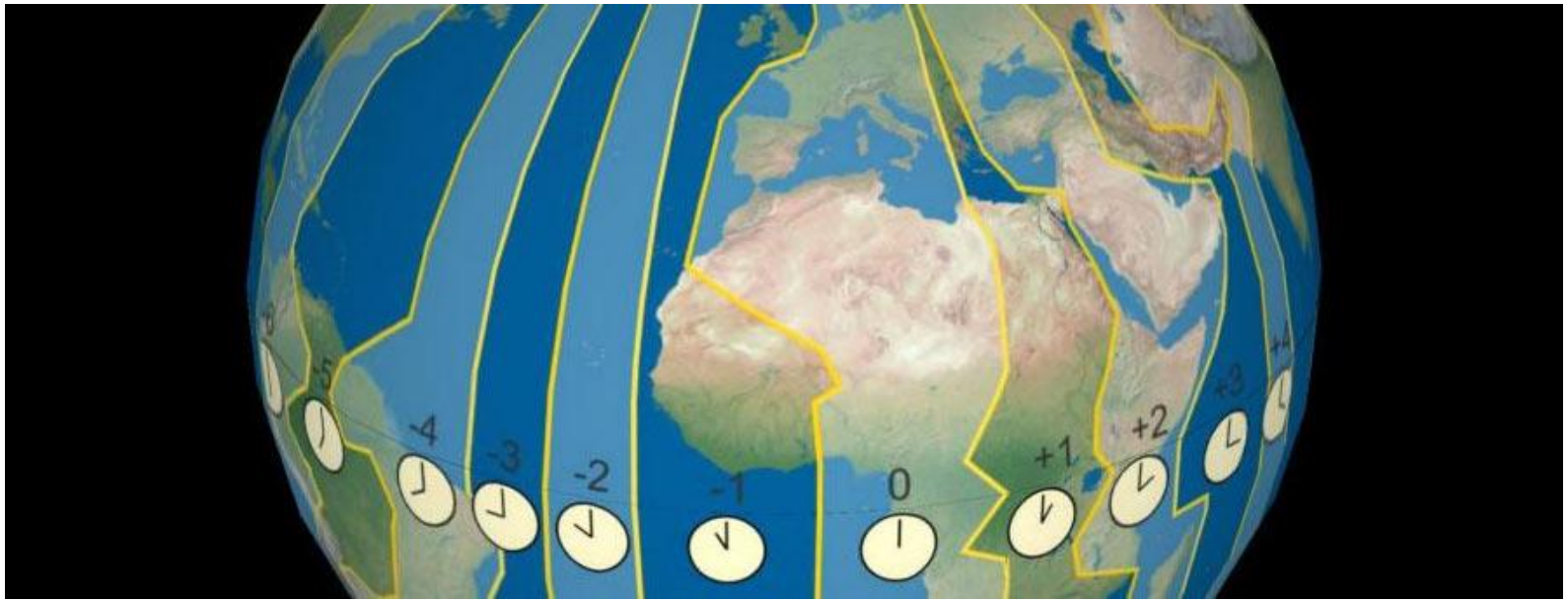
В стандарте языка SQL предусмотрены два варианта типа **`timestamp`**: без учета часового пояса – **`timestamp without time zone`** (время без часового пояса); с учетом часового пояса – **`timestamp with time zone`**. Тип данных **`timestamp with time zone`** для краткости можно записать как **`timestamptz`**. Это разрешено только в СУБД PostgreSQL.

Использование типа данных **`timestamp with time zone`** не рекомендуется, так как возможны ошибки автоматического определения смещения для времени года и текущего местоположения.

Домены. Типы данных

ISO/IEC 9075:2016 определяет типы для ведения данных о дате и времени (домен **DATETIME**):
DATETIME, DATE, TIME, TIMESTAMP

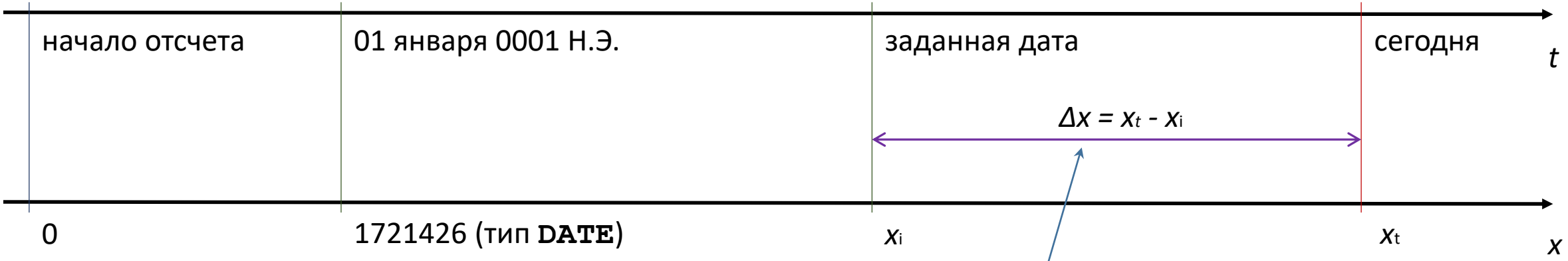
В чем разница между **DATETIME** и **TIMESTAMP**?



Структура типа данных **TIMESTAMP**: **DATE** (календарная дата), **TIME** (время), **TIME ZONE** (часовой пояс)

Структура типа данных **DATETIME**: **DATE** (календарная дата), **TIME** (время)

Домены. Типы данных



Тип данных: **DATE** – шаг 1 день (точность 1 день)
TIMESTAMP – шаг 1 секунда (точность 1 микросекунда)
INTERVAL – шаг 1 секунда (точность 1 микросекунда)

Интервал **INTERVAL**

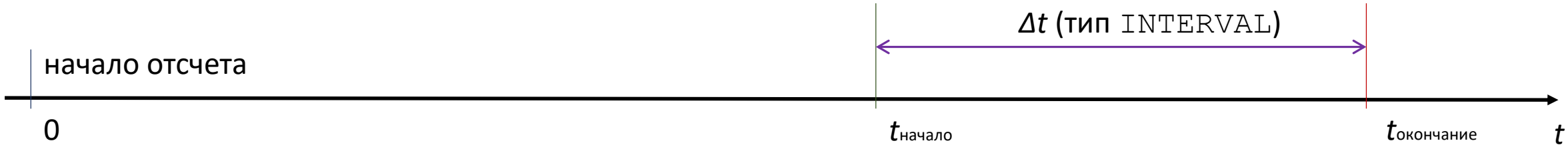
Для СУБД PostgreSQL

Начало отсчета (**0** в абсолютной шкале чисел) = **25.11.4714 г. до Н.Э.**
Объем данных типа **DATE** – 4 байта, предел учета **5874897 Н.Э.**
TIMESTAMP – 8 байт, предел учета **294276 Н.Э.**
INTERVAL – 16 байт, диапазон учета **-178000000 лет … 178000000 лет**

Домены. Типы данных

Задание интервала времени: $\Delta t = t_{\text{окончание}} - t_{\text{начало}}$

```
SELECT … WHERE <атрибут даты> <= <tокончание> AND <атрибут даты> >= <tначало>;
SELECT … WHERE <атрибут даты> BETWEEN <tначало> AND <tокончание>;
```



Формат интервала времени: `[@] количество единица [количество единица...] [направление]`

```
единица: microsecond[s], millisecond[s], second[s] | S, minute[s] | M во времени,
         hour[s] | H, day[s] | D, week[s] | W, month[es] | M в дате, year[s] | Y,
         decade[s], century[ies], millennium[s]
направление: ago (назад) | 'пусто'
```

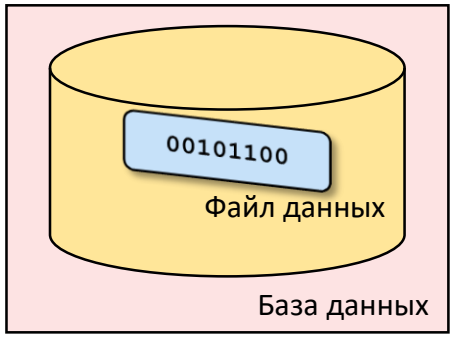
Пример: `1 day 10 hour 23 minute 36 second` **или** `1 10:23:36` **или** `P1DT10H23M36S`

`2 days 1 hour 12 minutes 26 seconds ago` **или** `-2 1:12:26` **или** `-P1DT10H23M36S`

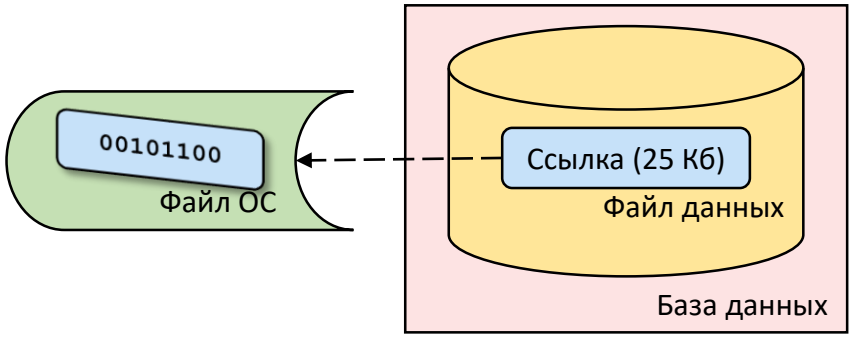
Домены. Типы данных

Домен **BLOB (Binary Large OBject)**: двоичные последовательности байтов любой природы (не являющиеся строками – поддерживаются любые непечатаемые символы, в том числе и символ <Возврат каретки>)

Способы хранения BLOB-данных



Способ **внутреннего** хранения – BLOB-данные размещаются внутри файла данных (в ячейке таблицы)



Способ **внешнего** хранения – BLOB-данные размещаются внутри файла операционной системы, а в ячейку таблицы помещается ссылка на данный файл

Типы данных СУБД PostgreSQL домена BLOB

Имя типа данных	Размер, занимаемый в памяти	Описание	Диапазон
bytea (шестнадцатиричный формат)	1 .. 4 байта + объем BLOB	Двоичные данные кодируются двумя шестнадцатеричными цифрами на байт, при этом первая цифра соответствует старшим 4 битам. К полученной строке добавляется префикс \x (чтобы она отличалась от формата спецпоследовательности).	1 байт .. предел, установленный ОС
bytea (формат спецпоследовательности)	1 .. 4 байта + объем BLOB	Двоичная строка представляется в виде последовательности ASCII-символов, а байты, непредставимые в виде ASCII-символов, передаются в виде спецпоследовательностей.	1 байт .. предел, установленный ОС

Схема отношения

Схема отношения (псевдокод)

```
<Имя_отношения>
 (<Имя_атрибута> <Имя_домена>
  [<Хранение_NULL> <Имя_ограничения_целостности> <Имя_правила_валидации> <Имя_правила_DEFAULT>],
  <Имя_атрибута> <Имя_домена>
  [<Хранение_NULL> <Имя_ограничения_целостности> <Имя_правила_валидации> <Имя_правила_DEFAULT>],
   …,
  <Имя_атрибута> <Имя_домена>
  [<Хранение_NULL> <Имя_ограничения_целостности> <Имя_правила_валидации> <Имя_правила_DEFAULT>]);
```

CREATE TABLE + схема отношения;

```
CREATE TABLE "table_1"
      ("attr_1" INTEGER NOT NULL PRIMARY KEY auto_count(), is_not_zero(),
       "attr_2" DATE NOT NULL UNIQUE current_day(),
       "attr_3" VARCHAR(500) NOT NULL CHECK,
       "attr_4" VARCHAR(1000) NOT NULL,
       "attr_5" BYTEA);
```

Схема базы данных = Схема отношения + Схема отношения + … + Схема отношения

Литература

1. **Кайт, Т., Кун, Д.** Oracle для профессионалов: архитектура и методики программирования, 3-е изд.: Пер. с англ . – Москва: ООО "ИД Вильямс", 2016. – 960 с.
2. **Гарсиа-Молина**, **Г.** Системы баз данных. Полный курс: пер. с англ. / Гарсиа-Молина – Москва : Издательский дом "Вильямс", 2003. – 1088 с.
3. **Конноли, Т.** Базы данных: проектирование, реализация и сопровождение. Теория и практика: учебное пособие / Т. Конноли, К. Бегг, А. Страчан **–** 2-е изд. : пер. с англ. : – Москва : Издательский дом "Вильямс", 2000. – 1120 с.
4. **Когаловский М. Р**. Энциклопедия технологий баз данных / М. Р. Когаловский. – Москва: Финансы и статистика, 2002. – 800 с.
5. **Мейер**, **Д.** Теория реляционных баз данных / Д. Мейер – Москва: Мир, 1987 г.
6. **Дейт, К. Дж.** Введение в системы баз данных: пер. с англ. – 7-е изд. / К. Дж. Дейт. – Москва: Издательский дом "Вильямс", 2001. – 1072 с.
7. **Райордан, Р.** Основы реляционных баз данных / Р. Райордан. : пер. с англ. – Москва : Изд-торг. Дом "Русская редакция", 2001. – 384 с.
8. **Кузнецов, С. Д.** Основы баз данных: курс лекций : учеб. пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / С. Д. Кузнецов. – Москва: Интернет-ун-т ин-форм. технологий, 2005. – 488 с.
9. **Бойко, В. В.,** Проектирование баз данных информационных систем / В. В. Бойко, В. М. Савинков. – Москва: Финансы и статистика, 1989.
10. **Тараканов, О. В., Паршенкова, Ю. А., Дементьев, А. Н., Конышев, М. Ю., Смирнов, С. В.** Системы баз данных: организация, инженерия, ведение / Под ред. О. В. Тараканова. – Москва: РТУ – МИРЭА, 2023. – 335 с.
11. **Смирнов С. Н. Задворьев И.С.** Работаем с ORACLE.: Учебное пособие/2-е изд., испр. и доп. – М: Гелиос АРВ, 2002 г. – 496 с.
12. **Фейерштейн, С., Прибыл, Б.,** Oracle PL/SQL. Для профессионалов. 6-е изд. – Санкт-Петербург: Питер, 2015. – 1024 с.
13. **Задворьев, И. С.,** Язык PL/SQL. Учебно-методическое пособие. – Москва: Онто-Принт, 2017. – 178 с.
14. **Кормен, Т.** Алгоритмы: построение и анализ / Т. Кормен, Ч. Л. Лейзерсон, Р. Ривест. – Москва: МЦНМО, 1999. – 960 с.
15. **ISO/TR 16044**: 2004 – Graphic technology – Database architecture model and control parameter coding for process control and workflow (Database AMPAC).
16. **ISO/IEC 9075**: 2018 – Structured Query Language.

Электронные ресурсы образовательного портала ACADEMY.ORACLE.COM.
Электронные ресурсы образовательного портала INTUIT.RU.
Электронные ресурсы портала HTTPS://ORACLEPLSQL.RU.
Электронные ресурсы портала POSTGRESPRO.RU.