

ГАПОУ "Камский строительный колледж
имени Е.Н.Батенчука"

Учебная дисциплина
СИСТЕМЫ УПРАВЛЕНИЯ БАЗЫ
ДАННЫХ

Лекция 7. PostgreSQL.
Вводная лекция



Руководитель занятия:
преподаватель
специализации

Еремина И.И.

Произношение

PostgreSQL произносится как ***post-gress-Q-L***

В разговоре часто употребляется ***postgres***
(***пост-гресс***)

Также, употребляется сокращение
pgsql (***пи-джи-эс-ку-эл*** или ***п-г-эс-ку-эль***)

PostgreSQL – это...

- ...самая развитая СУБД с открытым кодом
- ...надежность и устойчивость на очень больших нагрузках
- ...кроссплатформенность (FreeBSD, Linux, Solaris, Windows, OS X)
- ...высокий уровень соответствия ISO/ANSI SQL 92, 99, 2003
- ...интерфейсы для Tcl, Perl, C, C++, PHP, ODBC, JDBC, Embedded SQL in C, Python, Ruby, ...
- ...расширяемость
- ...быстродействие

Место в мире СУБД

PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире, являющаяся реальной альтернативой коммерческим СУБД.

Происхождение

- 1985 г. – **Post Ingres** (v1 – v4.2)
Майкл Стоунбрейкер
Калифорнийский университет в Беркли
POSTQUEL
- 1994 г. – **Postgres95** v5
Эндрю Ю и Джолли Чен
Калифорнийский университет в Беркли
SQL
- 1996 г. – **PostgreSQL** (v6 – v8)
Развивается сообществом
Стабильная версия на 21.10.2009 – v8.4.1

Функциональность

Название	DB2	FireBird	InterBase	MS SQL	MySQL	Oracle	PostgreSQL
Лицензия	\$\$\$	IPL	\$\$\$	\$\$\$	GPL/\$\$\$	\$\$\$	BSD
ACID	Yes	Yes	Yes	Yes	Depends	Yes	Yes
Referential integrity	Yes	Yes	Yes	Yes	Depends	Yes	Yes
Transaction	Yes	Yes	Yes	Yes	Depends	Yes	Yes
Unicode	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Schema	Yes	Yes	Yes	No	No	Yes	Yes
Temporary table	Yes	No	Yes	Yes	Yes	Yes	Yes
View	Yes	Yes	Yes	Yes	No	Yes	Yes
Materialized view	Yes	No	No	No	No	Yes	No
Expression index	No	No	No	No	No	Yes	Yes
Partial index	No	No	No	No	No	Yes	Yes
Inverted index	No	No	No	No	Yes	Yes	Yes
Bitmap index	Yes	No	No	No	No	Yes	No
Domain	No	Yes	Yes	No	No	Yes	Yes
Cursor	Yes	Yes	Yes	Yes	No	Yes	Yes
User Defined Functions	Yes	Yes	Yes	Yes	No	Yes	Yes
Trigger	Yes	Yes	Yes	Yes	No	Yes	Yes
Stored procedure	Yes	Yes	Yes	Yes	No	Yes	Yes
Tablespace	Yes	No	?	No	No	Yes	Yes

Ограничения

- Размер БД — нет ограничения
- Размер таблицы — до 32 TB
- Размер строки — до 1.6 TB
- Размер поля — до 1 GB
- Количество строк в таблице — нет ограничения
- Количество столбцов в таблице — до 250..1600
- Количество индексов — нет ограничения
- Длина идентификатора — до 63 байт включительно

Модель «клиент/сервер»

PostgreSQLсессия - это совместная работа двух процессов:

1. серверный процесс (postgres)
2. клиентский процесс (psql, GUI приложение, веб сервер, утилиты обслуживания и т. д.)

Концепция транзакций

Основной принцип работы системы — ACID:

- **Atomicity (атомарность)** - выполнение или не выполнение всех DML команд входящих в транзакцию;
- **Consistency (согласованность БД)** - завершение транзакции не должно нарушать целостность БД;
- **Isolation (изоляция)** - можно отображать либо исходные данные, которые были до начала транзакции, либо новые данные после выполнения транзакции;
- **Durability (сохранность данных)** - если пользователю пришло подтверждение выполнения транзакции, то его изменения не будут отменены по каким-либо причинам.

MVCC

Multiversion Concurrency Control –
многоверсионный контроль конкурентных
транзакций

- Способ реализации принципа изоляции, вместо классических блокировок
- Чтение и запись не конфликтуют
- Конфликт только при конкурентных операциях записи

Объекты в БД

- Таблицы
- Индексы
- Пользователи и группы (роли)
- Языки (для создания функций)
- Функции (FUNCTION)
- Триггеры (TRIGGER)
- Правила (RULE)
- Представления (VIEW)
- Правила преобразования типов (CAST)
- Типы данных (TYPE)
- Последовательности (SEQUENCE)

Типы данных

Поддерживаемые встроенные типы данных:

1. Числовые
2. Символьные
3. Бинарные (двоичные)
4. Логические
5. Дата и время
6. Геометрические
7. Типы для адресов компьютерных сетей
8. Битовые строки
9. Типы для поиска текста
10. UUID
11. XML

Числовые типы

smallint	короткое 2-х байтовое целое
integer	обычное 4-х байтовое целое
bigint	большое 8-байтовое целое
decimal	дробное с фиксированной точкой
numeric	дробное с фиксированной точкой
real	дробное с плавающей точкой
double precision	дробное с плавающей точкой двойной точности
serial	целое с автоувеличением
bigserial	большое целое с автоувеличением

СИМВОЛЬНЫЕ ТИПЫ

character varying(n), varchar(n)	строка переменной длины с ограничением
character(n), char(n)	строка фиксированной длины
text	строка переменной неограниченной длины

Бинарные (двоичные) типы

bytea	бинарная строка переменной длины
bit [(n)]	битовая строка фиксированной длины
bit varying [(n)]	битовая строка переменной длины

Логический тип

boolean	TRUE или FALSE
----------------	----------------

Дата и время

timestamp [(p)] [без часового пояса]	дата и время
timestamp [(p)] с часовым поясом	дата и время с часовым поясом
interval [(p)]	интервал времени
date	только дата
time [(p)] [без часового пояса]	только время
time [(p)] с часовым поясом	только время с часовым поясом

Геометрические типы

point	Точка на плоскости (x,y)
line	Невидимая линия (не полностью реализовано)
lseg	Видимый отрезок ((x1,y1),(x2,y2))
box	Четырёхугольник ((x1,y1),(x2,y2))
path	Замкнутый многоугольник (похож на полигон) ((x1,y1),...)
path	Ломаная линия [(x1,y1),...]
polygon	Полигон (похож на замкнутый многоугольник) ((x1,y1),...)
circle	Круг (x,y),r (центр и радиус)

Остальные типы

cidr	IPv4 или IPv6 сеть
inet	IPv4 или IPv6 хост и сеть
macaddr	MAC адрес
tsquery	запрос на поиск текста
tsvector	список для поиска текста
uuid	универсальный уникальный идентификатор
xml	данные XML

Составные типы данных

- Массивы простых типов
- Составные типы
- Массивы составных типов
- Перечисления (ENUM)

Пользовательские типы

CREATE TYPE name AS
(attribute_name data_type [, ...])

CREATE TYPE name AS ENUM
('label' [, ...])

CREATE TYPE name (
 INPUT = input_function,
 OUTPUT = output_function
 [, RECEIVE = receive_function]
 [, SEND = send_function]
 [, TYPMOD_IN = type_modifier_input_function]
 [, TYPMOD_OUT = type_modifier_output_function]
 [, ANALYZE = analyze_function]
 [, INTERNALLENGTH = { internallength | VARIABLE }]
 [, PASSEDBYVALUE]
 [, ALIGNMENT = alignment]
 [, STORAGE = storage]
 [, DEFAULT = default]
 [, ELEMENT = element]
 [, DELIMITER = delimiter]
)

Ограничения целостности

- NOT NULL - не NULL
- UNIQUE - уникальность
- PRIMARY KEY - первичный ключ
- FOREIGN KEY/REFERENCES - внешний ключ, ссылки
- CHECK - проверка

Операторы SQL

1. Операторы описания данных
(DDL, Data Definition Language):
CREATE ..., ALTER ..., DROP ... и пр.

Примечание: в PostgreSQL DDL
операторы можно использовать в транзакциях.

2. Операторы манипулирования данными (DML):
 1. SELECT – выборка данных
 2. INSERT – вставка данных
 3. UPDATE – обновления данных
 4. DELETE – удаление данных

Примеры DDL и DML

```
CREATE TABLE person(  
    name VARCHAR(32) NOT NULL,  
    age INTEGER,  
    CHECK(age >= 0 AND age < 200)  
);
```

```
INSERT INTO person (name, age)  
VALUES ('Вольдемар', 22);
```

```
UPDATE person SET age = 17 WHERE name = 'Вольдемар';
```

```
DELETE FROM person WHERE age < 18;
```


Индексы

PostgreSQL предлагает четыре типа индексов:

1. B-tree
2. Hash
3. GiST
4. GIN

Индексы могут быть:

составными

функциональными

частичными

```
CREATE INDEX person_lower_name_idx ON person (lower(name))  
    USING btree;
```

Функции и хранимые процедуры

- SQL
- PL/pgSQL
- PL/Perl
- PL/Tcl
- PL/Python
- PL/R
- PL/Java
- plPHP
- plRuby
- всегда можно добавить что-то своё

Анализ производительности

Порядок обработки запроса:

парсер → планировщик → исполнитель

EXPLAIN – показывает план запроса с оценочными характеристиками

EXPLAIN ANALYZE – выполняет запрос и показывает реальное время

Пример функции

SQL:

```
CREATE FUNCTION add(integer, integer) RETURNS integer
AS 'select $1 + $2;'
LANGUAGE SQL IMMUTABLE RETURNS NULL ON NULL INPUT;
```

pl/pgSQL:

```
CREATE OR REPLACE FUNCTION increment(i integer) RETURNS
integer
AS $$
    BEGIN RETURN i + 1;
    END;
$$ LANGUAGE plpgsql;
```

Триггеры

Триггеры могут быть назначены для операторов:

1. INSERT
2. UPDATE
3. DELETE
4. TRUNCATE

```
CREATE TRIGGER name { BEFORE | AFTER } { event [ OR ... ] }  
ON table [ FOR [ EACH ] { ROW | STATEMENT } ] EXECUTE  
PROCEDURE funcname ( arguments )
```

Роли и привилегии

Концепция ролей заменила концепцию пользователей и групп

```
CREATE ROLE student WITH LOGIN PASSWORD 'jw8s0F4' VALID  
UNTIL '2005-01-01';
```

```
GRANT EXECUTE ON FUNCTION add(integer, integer) TO student;
```

Первые шаги после установки

1. Создание кластера БД (initdb)
2. Начальная конфигурация postgresql.conf и pg_hba.conf
3. Запуск сервера PostgreSQL

Начальная настройка

Всего 2 файла:

postgresql.conf (основные настройки)

listen_addresses (сетевые интерфейсы)

shared_buffers (общая память)

work_mem (память для сортировок и т.п.)

pg_hba.conf (настройки доступа)

host all all 127.0.0.1/32 md5

Основные программы/утилиты

- postgres — запуск сервера в однопользовательском режиме
- initdb — инициализация кластера
- pg_ctl — утилита для контроля (запуск/останов)
- pg_dump — создание дампа (резервной копии) БД
- psql — интерактивный терминал
- createdb, createuser, dropdb, dropuser, reindexdb и т.д. — утилиты работы с БД из shell

psql

psql — развитая терминальная программа, огромное количество возможностей

Всегда помнить:

- psql --help (в shell)

- \?

- \h

- автодополнение (по нажатию на TAB)

- \h ANY-SQL-COMMAND

Полезные источники информации

- <http://postgresql.org>
 - Английский язык
 - Самые свежие дистрибутивы и документация
- <http://postgresql.ru.net>
 - Документация на русском языке
- <http://google.ru>
 - Хороший поисковик