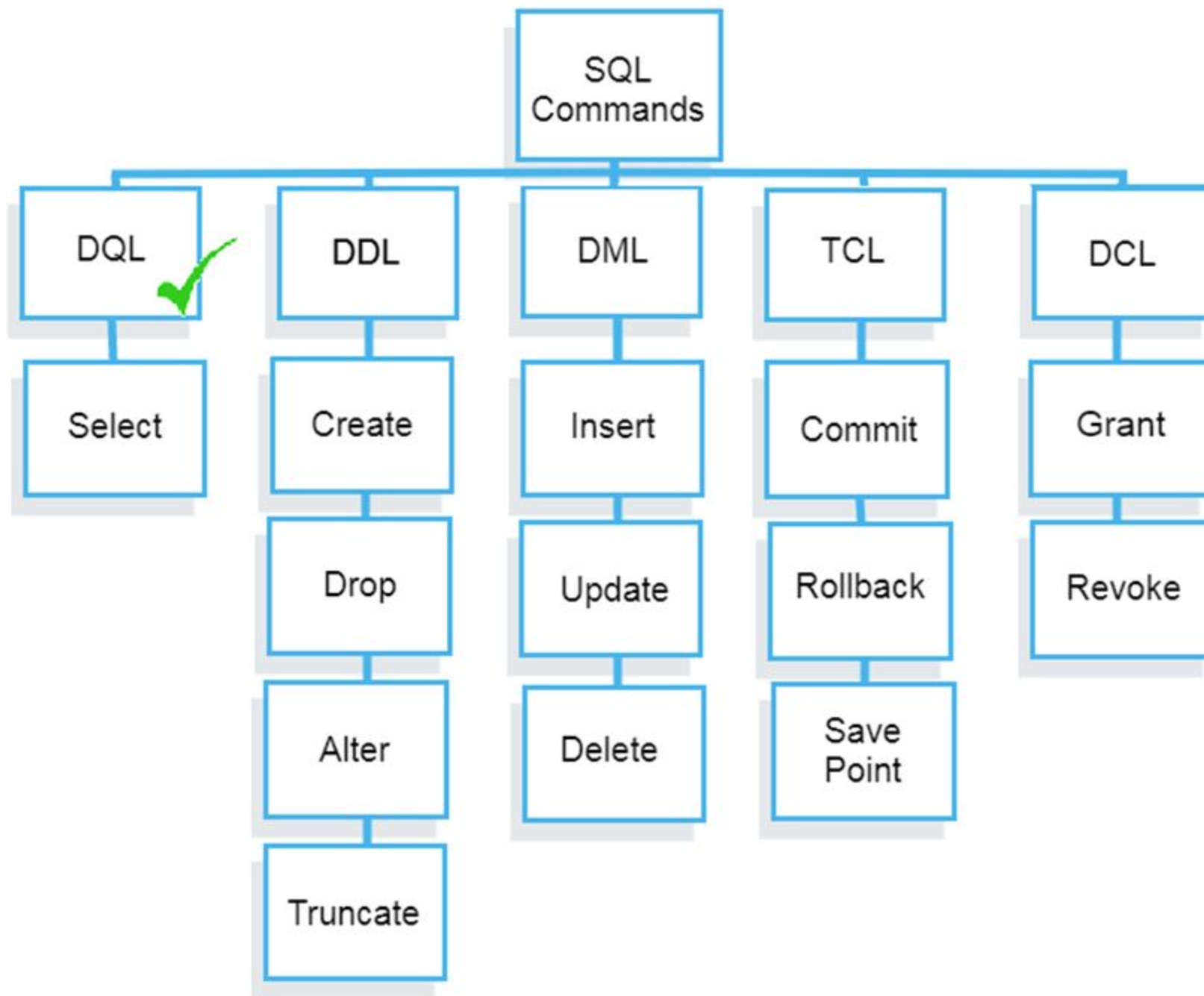


Создание объектов базы данных в PostgreSQL

Цели

- Создание простых и сложных представлений
- Получение информации из представлений
- Создание, изменение и использование последовательностей
- Создание индексов



Объекты базы данных

Объект	Описание
Таблица	Базовая единица хранения информации, содержит набор строк
Представление	Представляет подмножества данных из одной или нескольких таблиц
Последовательность	Генератор числовых значений
Индекс	Увеличивает производительность некоторых запросов

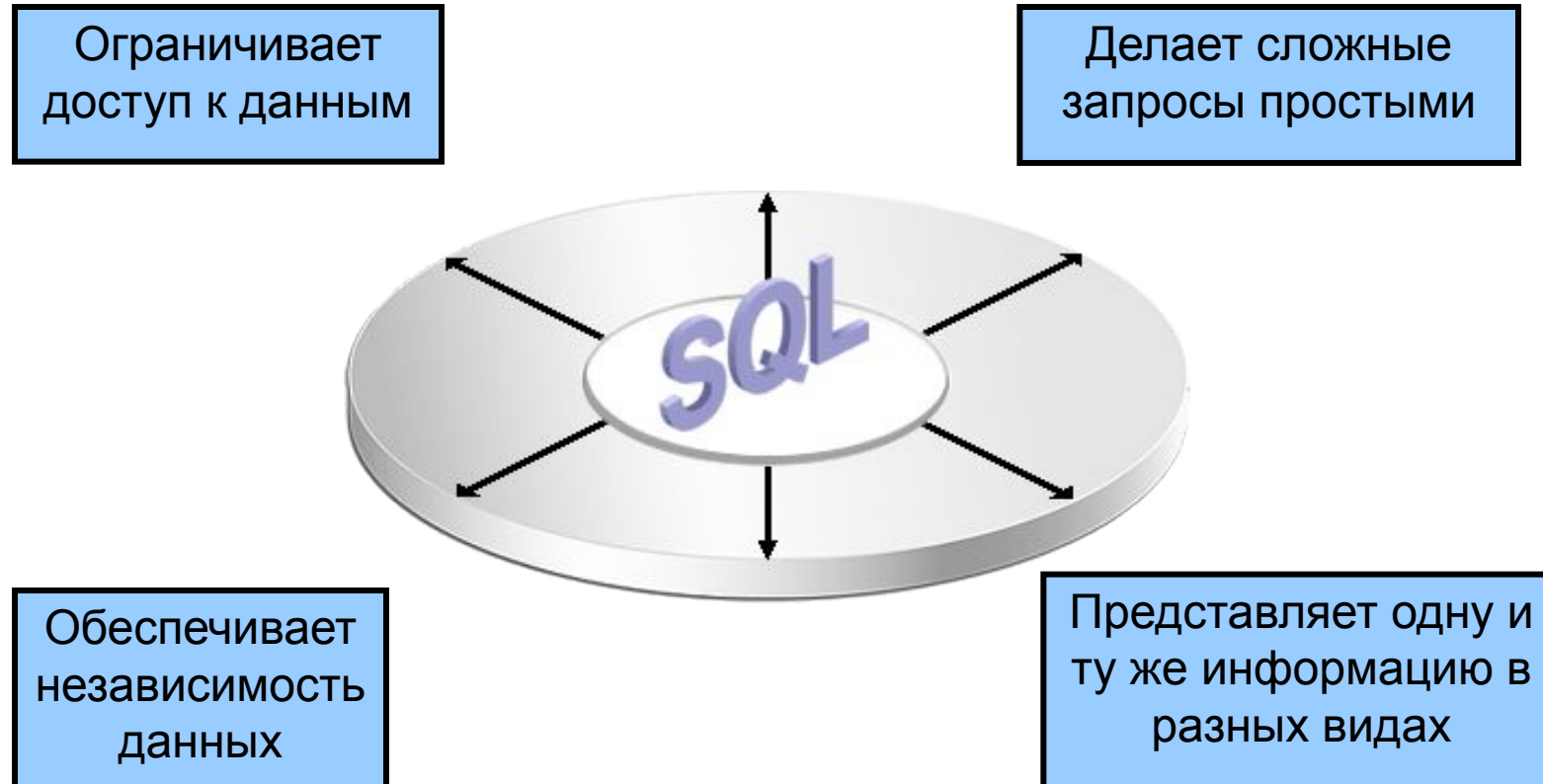
Что такое представление?

EMPLOYEES table

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALA
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_FRES	2401
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	1701
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	1701
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	901
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	601
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-98	IT_PROG	421
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	581
141	Trenna	Rae	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	351
142	Curtis	Denes	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	311
143	Randall	Mateo	RMATEO	650.121.2074	15-MAR-98	ST_CLERK	261
149	Zlotkey				JUL-96	ST_CLERK	251
174	Abel				JAN-00	SA_MAN	1051
176	Taylor				MAY-96	SA_REP	1101
177	Kimberly	Grant	KGRANT	611.44.1044.429203	MAR-98	SA_REP	861
178	Kimberly	Grant	KGRANT	611.44.1044.429203	24-MAY-99	SA_REP	701
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	441
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	1301
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	601
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	1201
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94	AC_ACCOUNT	831

20 rows selected.

Преимущества представлений



Изменяемые представления

- Список FROM в запросе, определяющем представлении, должен содержать ровно один элемент, и это должна быть таблица или другое изменяемое представление.
- Определение представления не должно содержать предложения WITH, DISTINCT, GROUP BY, HAVING, LIMIT и OFFSET на верхнем уровне запроса.
- Определение представления не должно содержать операции с множествами (UNION, INTERSECT и EXCEPT) на верхнем уровне запроса.
- Список выборки в запросе не должен содержать агрегатные и оконные функции, а также функции, возвращающие множества.
- Автоматически обновляемое представление может содержать как изменяемые, так и не изменяемые столбцы.

Простые представления

```
CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ] [ RECURSIVE ] VIEW имя  
    [ ( имя_столбца [, ...] ) ]  
    [ WITH (имя_параметра_представления  
            [=значение_параметра_представления] [, ... ] ) ]  
    AS запрос  
    [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]
```

Создание представлений

```
CREATE VIEW comedies AS
  SELECT *
  FROM films
  WHERE kind = 'Comedy';
```

```
CREATE VIEW universal_comedies AS
  SELECT *
  FROM comedies
  WHERE classification = 'U'
  WITH LOCAL CHECK OPTION;
```

```
CREATE VIEW universal_comedies (id, comedy_name) AS
  SELECT id, name
  FROM comedies
  WHERE classification = 'U'
  WITH LOCAL CHECK OPTION;
```

Удаление представлений

- Выполнить эту команду может только владелец представления.

```
DROP VIEW [ IF EXISTS ] имя [, ...] [ CASCADE | RESTRICT ]
```

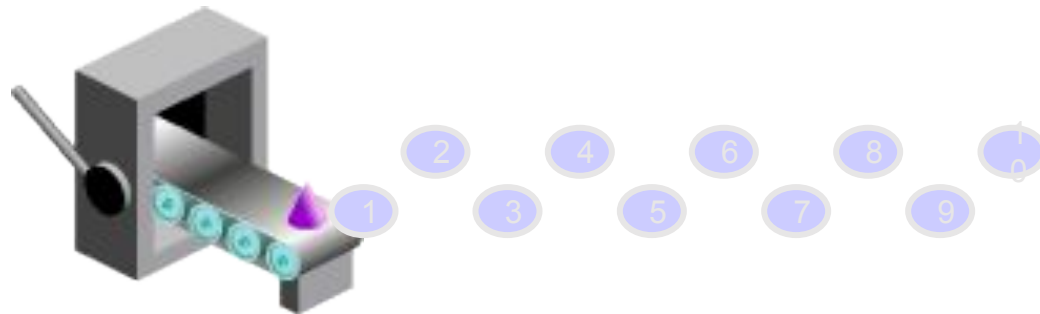
```
DROP VIEW kinds;
```

Последовательности

Объект	Описание
Таблица	Базовая единица хранения информации, содержит набор строк
Представление	Представляет подмножества данных из одной или нескольких таблиц
Последовательность	Генератор числовых значений
Индекс	Увеличивает производительность некоторых запросов

Последовательности

- Генерирует уникальные числовые значения
- Может быть использована для генерирования значений первичного ключа
- Упрощает логику приложения
- Ускоряет эффективность доступа к значениям последовательностей при кэшировании в памяти



Создание последовательности

```
CREATE [ TEMPORARY | TEMP ] SEQUENCE [ IF NOT EXISTS ] имя
    [ AS тип_данных ]
    [ INCREMENT [ BY ] шаг ]
    [ MINVALUE мин_значение | NO MINVALUE ]
    [ MAXVALUE макс_значение | NO MAXVALUE ]
    [ START [ WITH ] начало ]
    [ CACHE кеш ]
    [ [ NO ] CYCLE ]
    [ OWNED BY { имя_таблицы.имя_столбца | NONE } ]
```

Создание последовательности

```
CREATE SEQUENCE IF NOT EXISTS serial
    AS bigint
    INCREMENT BY 1
    MINVALUE 1
    NO MAXVALUE
    START WITH 1
    CACHE 1
    NO CYCLE;
```

*последовательности основаны на арифметике `bigint`, так что их значения не могут выходить за диапазон восьмибайтовых целых (-9223372036854775808 .. 9223372036854775807)

Функции для работы с последовательностями

- `nextval(regclass)` – продвигает объект последовательности к следующему значению и возвращает это значение.

```
SELECT nextval('myseq');
```

- `currval(regclass)` - возвращает значение, выданное при последнем вызове `nextval` для этой последовательности в текущем сеансе.

```
SELECT setval('myseq', 42);
```

- `setval(regclass)` – Устанавливает для объекта последовательности текущее значение и может также установить флаг `is_called`.

```
SELECT currval('myseq');
```

- `lastval()` - Возвращает значение, выданное при последнем вызове `nextval` в текущем сеансе.

```
SELECT lastval();
```

Использование последовательности

```
SELECT nextval('serial');
```

```
nextval
```

```
-----
```

```
1
```

```
INSERT INTO distributors VALUES (nextval('serial'), 'nothing');
```

Изменение последовательности

```
ALTER SEQUENCE [ IF EXISTS ] имя
    [ AS тип_данных ]
    [ INCREMENT [ BY ] шаг ]
    [ MINVALUE мин_значение | NO MINVALUE ]
    [ MAXVALUE макс_значение | NO MAXVALUE ]
    [ START [ WITH ] начало ]
    [ RESTART [ [ WITH ] перезапуск ] ]
    [ CACHE кеш ] [ [ NO ] CYCLE ]
    [ OWNED BY { имя_таблицы.имя_столбца | NONE } ]

ALTER SEQUENCE [ IF EXISTS ] имя RENAME TO новое_имя;
```

Изменение последовательности

- Вы должны быть владельцем последовательности или суперпользователем.
- Изменения вступят в силу только после того, как будут израсходованы все кэшированные значения.

```
ALTER SEQUENCE serial RESTART WITH 105;
```

```
DROP SEQUENCE [ IF EXISTS ] имя [, ...] [ CASCADE | RESTRICT ]
```

- Для удаления последовательности используйте команду DROP:

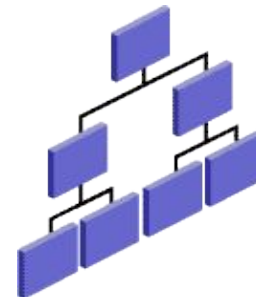
```
DROP SEQUENCE serial;
```

Индексы

Объект	Описание
Таблица	Базовая единица хранения информации, содержит набор строк
Представление	Представляет подмножества данных из одной или нескольких таблиц
Последовательность	Генератор числовых значений
Индекс	Увеличивает производительность некоторых запросов

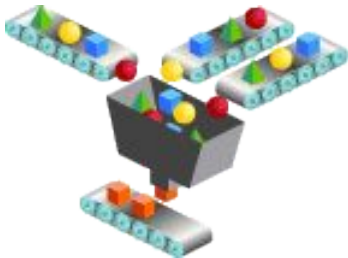
Индексы

- Является объектом схемы.
- Индексы применяются в первую очередь для оптимизации производительности базы данных.
- Сокращает количество операций дискового ввода.
- Может использоваться при выполнении команд UPDATE, DELETE, запросах с JOIN.
- Используется и поддерживается автоматически сервером баз данных.



Создание индекса

- Автоматически: Уникальный индекс создается автоматически при объявлении ограничений `PRIMARY KEY` или `UNIQUE` в описании таблицы.
- Вручную: Пользователь может создавать неуникальный индекс на колонках для ускорения выполняемых запросов.
- PostgreSQL поддерживает несколько типов индексов:
 - B-дерево
 - Хеш
 - GiST
 - SP-GiST
 - GIN
 - BRIN
- По умолчанию команда `CREATE INDEX` создаёт индексы-B-деревья, эффективные в большинстве случаев



В-дерево

- В-деревья могут работать в условиях на равенство и в проверках диапазонов с данными (<, <=, =, >=, >)
- При обработке конструкций, представимых как сочетание нескольких операторов, например BETWEEN и IN.
- Может использоваться в условиях IS NULL и IS NOT NULL.
- Может использоваться в запросах с операторами сравнения по шаблону LIKE и ~, если этот шаблон определяется константой и он привязан к началу строки — например, `col LIKE 'foo%'` или `col ~ '^foo'`, но не `col LIKE '%bar'`.

```
CREATE INDEX имя ON таблица (столбец) ;
```

Хеш-индекс

- Хеш-индексы хранят 32-битный хеш-код.
- Хеш-индексы работают только с простыми условиями равенства.

```
CREATE INDEX имя ON таблица USING HASH (столбец) ;
```

GiST, SP-GiST индекс

- GiST — сокращение от «generalized search tree»
- Предназначен для работы с геодами, массивами и т.д.

GIN-индексы

- GIN-индексы представляют собой «инвертированные индексы», в которых могут содержаться значения с несколькими ключами, например массивы или другие структуры (ex. JSON)
- Инвертированный индекс содержит отдельный элемент для значения каждого компонента, и может эффективно работать в запросах, проверяющих присутствие определённых значений компонентов.
- Позволяют применять индексы в запросах с операторами (<@, @>, =, &&)

BRIN-индексы

- Сокращение от Block Range INdexes, Индексы зон блоков.
- Хранят обобщённые сведения о значениях, находящихся в физически последовательно расположенных блоках таблицы.
- Поддерживает запросы с операторами (<, <=, =, >=, >)

Создание индекса

```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ]  
    [ [ IF NOT EXISTS ] имя ] ON имя_таблицы [ USING метод ]  
    ( { имя_столбца | ( выражение ) }  
      [ COLLATE правило_сортировки ]  
      [ класс_операторов [ ( параметр_класса_оп = значение [, ... ] ) ] ]  
      [ ASC | DESC ] [ NULLS { FIRST | LAST } ] [, ...] )  
    [ INCLUDE ( имя_столбца [, ...] ) ]  
    [ WITH ( параметр_хранения [= значение] [, ... ] ) ]  
    [ TABLESPACE табл_пространство ]  
    [ WHERE предикат ]
```

Параметры индекса

- `UNIQUE` - Указывает, что система должна контролировать повторяющиеся значения в таблице.
- `CONCURRENTLY` - PostgreSQL построит индекс, не устанавливая никаких блокировок, которые бы предотвращали DML.
- `INCLUDE` – Определяет неключевые столбцы.
- `ASC | DESC` - указывает порядок сортировки.
- `NULLS {FIRST | LAST}` - определяет положение NULL-значений после сортировки

Рекомендации По Созданию Индекса

Создавайте индекс, когда:	
☒	Колонка содержит большой диапазон значений
☒	Колонка содержит большое количество null-значений
☒	Одна или более колонок, часто используемых вместе в условии WHERE или условии соединения таблиц
☒	Таблица большая и большинство запросов выполняются для выборки от 2% до 4% строк
Не создавайте индекс, когда:	
☒	Столбцы не часто используются в качестве условия в запросе
☒	Таблица невелика или большинство запросов извлекут более 2-4% строк в таблице
☒	Таблица часто обновляется
☒	На индексируемые столбцы ссылаются как на часть выражения

Изменение индекса

- Изменение индекса определяется следующей командой:

```
ALTER INDEX [ IF EXISTS ] имя RENAME TO новое_имя;
```

- Перестроение индекса

```
REINDEX [ ( параметр [, ...] ) ]  
{ INDEX | TABLE | SCHEMA | DATABASE | SYSTEM } [ CONCURRENTLY ] имя
```

Здесь допускается параметр:

```
CONCURRENTLY [ логическое_значение ]  
TABLESPACE новое_табл_пространство  
VERBOSE [ логическое_значение ]
```

Удаление индекса

- DROP INDEX удаляет существующий индекс из базы данных.
- Выполнить эту команду может только владелец индекса.

```
DROP INDEX [ CONCURRENTLY ] [ IF EXISTS ] имя [, ...]  
          [ CASCADE | RESTRICT ]
```