BASIS

Администрирование баз данных

Ростелеком
Технологии возможностей

Администрирование баз данных

1. Типы и структура СУБД
2. Запросы, индексы и эксплейны
3. Администрирование MySQL
4. Администрирование PostgreSQL
5. Troubleshooting

Трудоемкость
8 часов

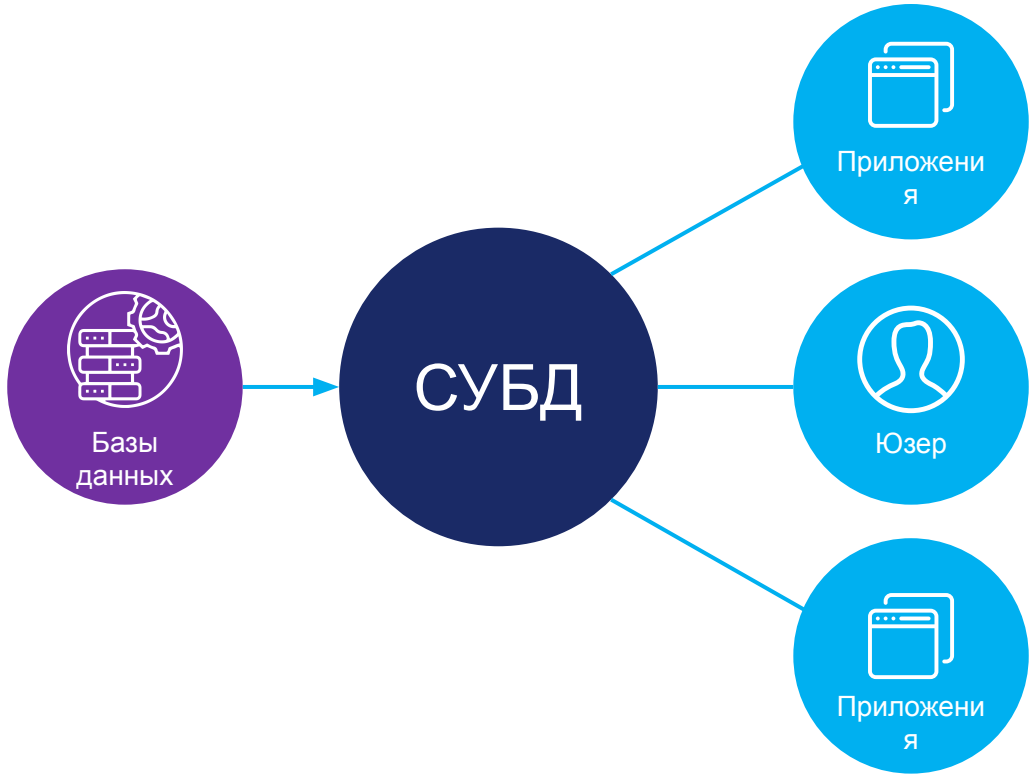
BASIS

Ростелеком

Типы и структура СУБД

Типы и структура СУБД

Типы БД, называемых также моделями БД или семействами БД, представляют собой шаблоны и структуры, используемые для организации данных в системе управления базами данных (СУБД)

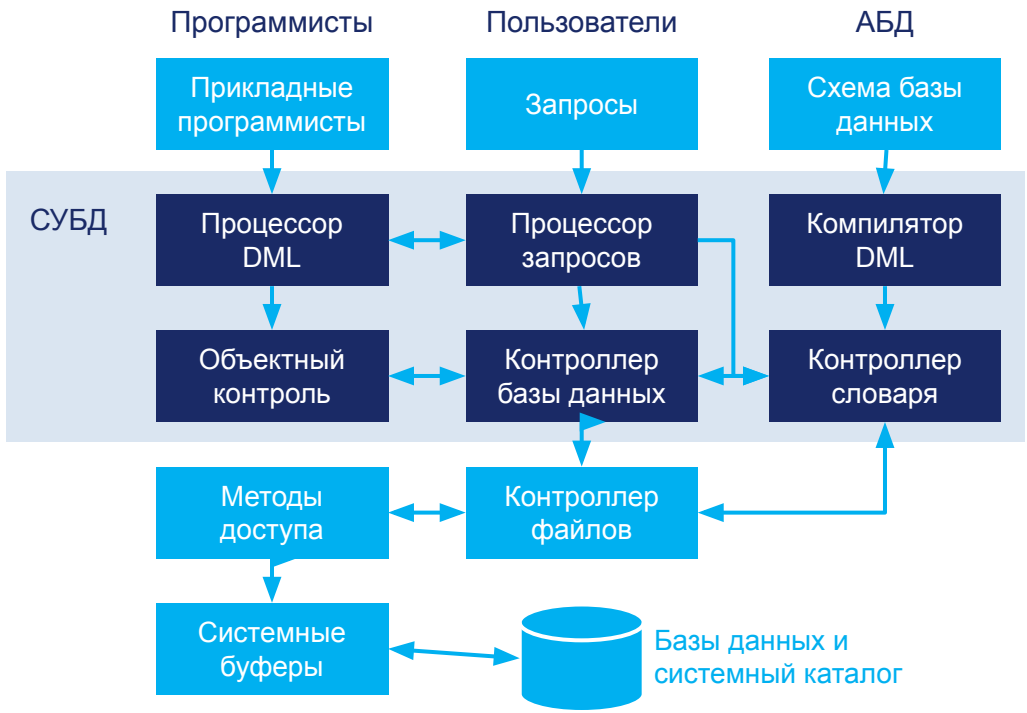


BASIS

Ростелеком

Основные функции и компоненты СУБД

- Поддержка целостности файлов
- Восстановление согласованного состояния данных после сбоев
- Обеспечение параллельной работы нескольких пользователей
- Поддержка языка манипулирования данными

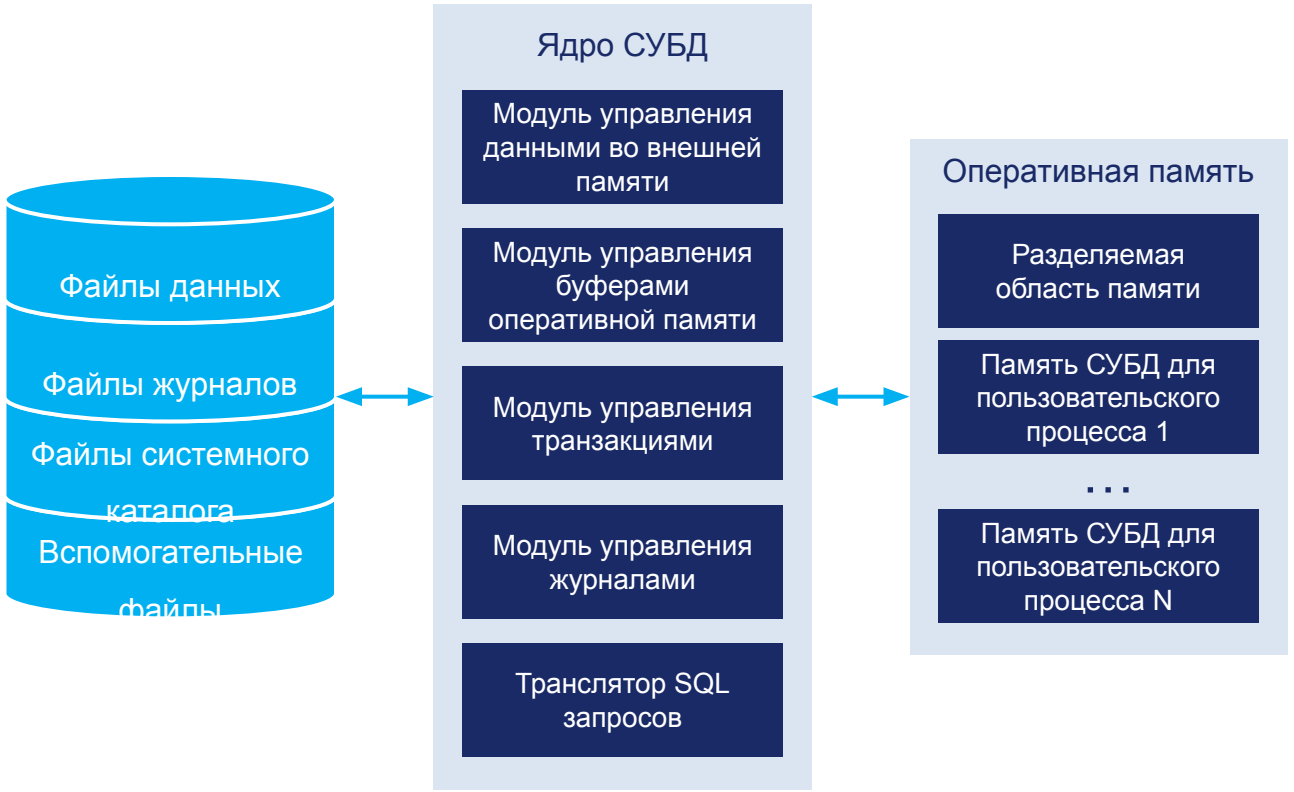


BASIS

Ростелеком

Управление данными во внешней памяти

- Для хранения данных и метаданных, входящих в БД
- Для служебных целей



BASIS

Ростелеком

Управление буферами оперативной памяти

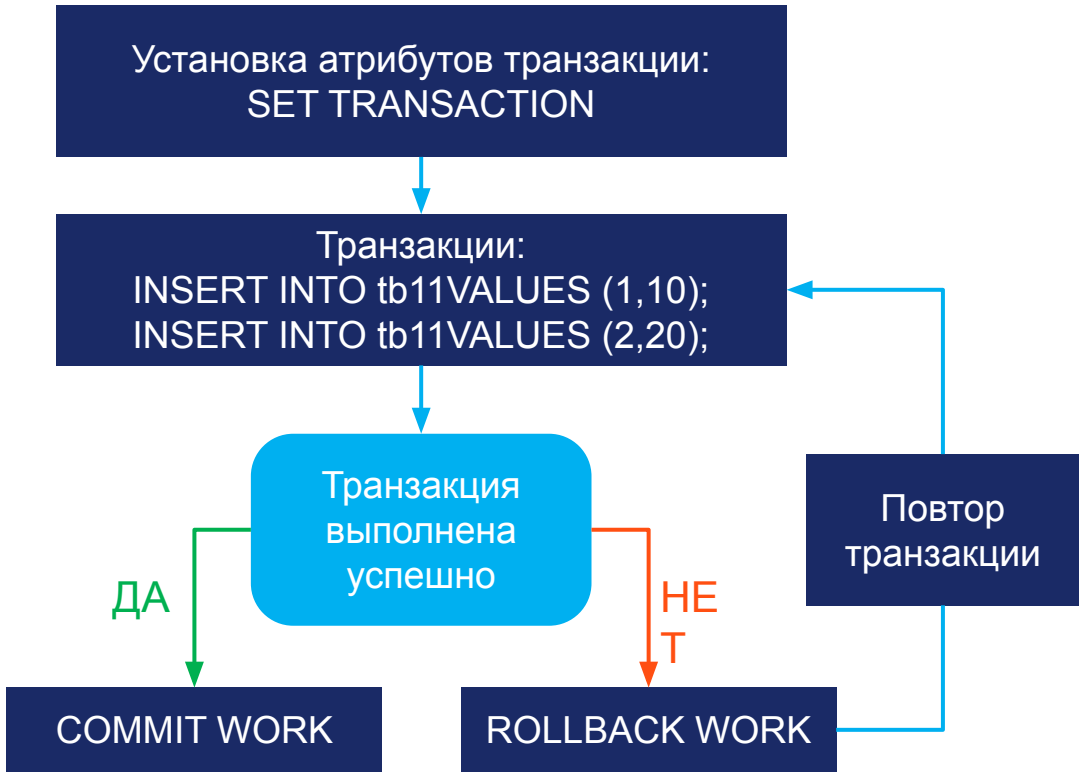
Управление буферами оперативной памяти (ОП) необходимо для увеличения скорости работы с данными.

BASIS

Ростелеком

Управление транзакциями

Транзакция — это последовательность операций над БД, рассматриваемых СУБД как единое целое: либо все операции внутри транзакции выполняются, либо ни одна не выполняется.

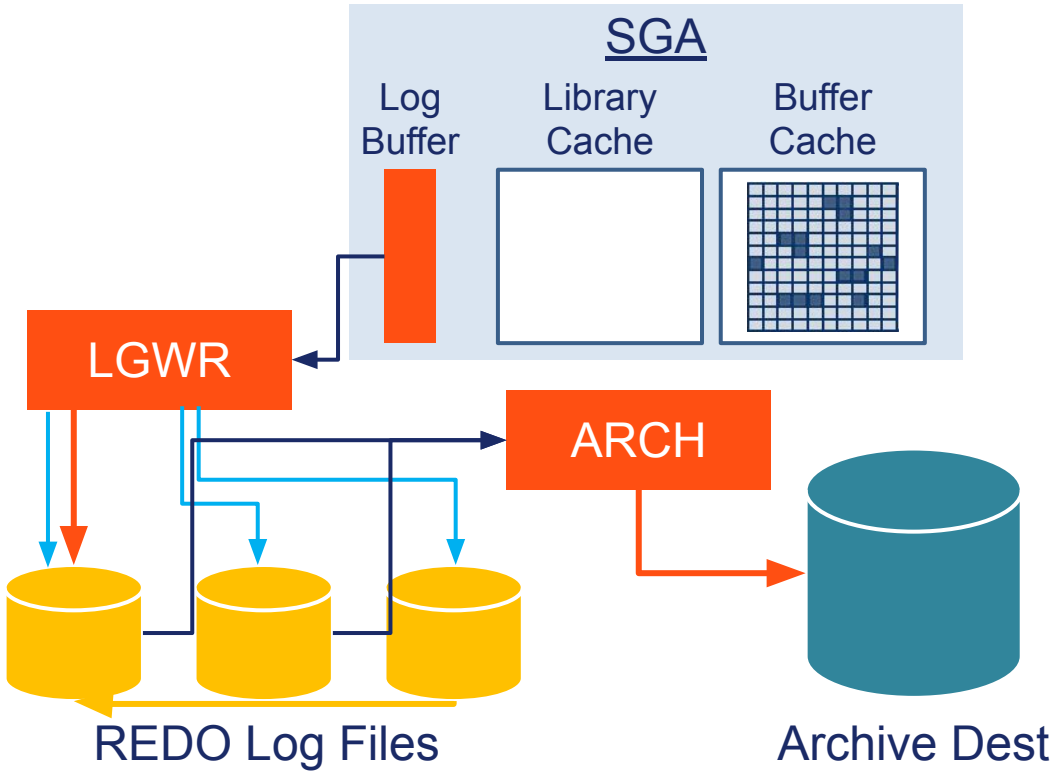


BASIS

Ростелеком

Журнализация

Журнализация необходима для восстановления БД в случае сбоев. Одним из основных требований к СУБД является надежность хранения данных во внешней памяти.



BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@s-pg13:~$ psql

Timing is on.

psql (13.3)

Type "help" for help.

postgres@postgres=# SELECT pg_current_wal_lsn();

pg_current_wal_lsn

--------------------

0/17BD9A0

(1 row)

Time: 1,602 ms
```

BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@postgres=# SELECT pg_current_wal_lsn() AS pos1 \gset

Time: 0,224 ms

postgres@postgres=# CREATE TABLE t(n integer);

CREATE TABLE

Time: 2,113 ms

postgres@postgres=# INSERT INTO t SELECT gen.id FROM generate_series(1,1000) AS gen(id);

INSERT 0 1000 Time: 2,242 ms

postgres@postgres=# SELECT pg_current_wal_lsn() AS pos2 \gset

Time: 0,179 ms

postgres@postgres=# SELECT :'pos2'::pg_lsn - :'pos1'::pg_lsn;

?column?

----------
138968 (1 row) Time: 1,193 ms
```

BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@postgres=# SELECT * FROM pg_ls_waldir();

name | size | modification

------------------------+----------+----------------------

000000010000000000000001 | 16777216 | 2022-10-15 16:28:49+03

(1 row)

Time: 8,770 ms
```

BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@postgres=# \q

postgres@s-pg13:~$ ps -o pid,command --ppid `head -n 1 $PGDATA/postmaster.pid`

PID COMMAND

24122 postgres: checkpointer

24123 postgres: background writer

24124 postgres: walwriter

24125 postgres: autovacuum launcher

24126 postgres: stats collector

24127 postgres: logical replication launcher
```

BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@s-pg13:~$ rm /home/postgres/logfile

postgres@s-pg13:~$ pg_ctl -w -l /home/postgres/logfile -D /usr/local/pgsql/data restart

waiting for server to shut down.... done

server stopped

waiting for server to start.... done

server started

postgres@s-pg13:~$ cat /home/postgres/logfile

2022-06-25 15:31:18.747 MSK [29370] LOG: starting PostgreSQL 13.3 on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit

2022-06-25 15:31:18.747 MSK [29370] LOG: listening on IPv6 address "::1", port 5432

2022-06-25 15:31:18.747 MSK [29370] LOG: listening on IPv4 address "127.0.0.1", port 5432

2022-10-15 15:31:18.747 MSK [29370] LOG: listening on Unix socket "/tmp/.s.PGSQL.5432"

2022-10-15 15:31:18.748 MSK [29372] LOG: database system was shut down at 2022-10-15 15:31:18 MSK

2022-10-15 15:31:18.749 MSK [29370] LOG: database system is ready to accept connections
```

Ростелеком

Пример WAL PostgreSQL

```
postgres@s-pg13:~$ rm /home/postgres/logfile

postgres@s-pg13:~$ pg_ctl -w -D /usr/local/pgsql/data stop -m immediate

waiting for server to shut down.... done

server stopped

postgres@s-pg13:~$ pg_ctl -w -l /home/postgres/logfile -D /usr/local/pgsql/data start

waiting for server to start.... done

server started
```

BASIS

Ростелеком

Пример WAL PostgreSQL

```
postgres@s-pg13:~$ cat /home/postgres/logfile

2022-10-15 15:32:37.988 MSK [29389] LOG: starting PostgreSQL 13.3 on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit

2022-10-15 15:32:37.988 MSK [29389] LOG: listening on IPv6 address "::1", port 5432

2022-10-15 15:32:37.988 MSK [29389] LOG: listening on IPv4 address "127.0.0.1", port 5432

2022-10-15 15:32:37.989 MSK [29389] LOG: listening on Unix socket "/tmp/.s.PGSQL.5432"

2022-10-15 15:32:37.991 MSK [29391] LOG: database system was interrupted; last known up at 2022-10-15 15:31:18 MSK

2022-10-15 15:32:38.006 MSK [29391] LOG: database system was not properly shut down; automatic recovery in progress

2022-10-15 15:32:38.009 MSK [29391] LOG: redo starts at 0/17E3938

2022-10-15 15:32:38.009 MSK [29391] LOG: invalid record length at 0/17E59D8: wanted 24, got 0

2022-10-15 15:32:38.009 MSK [29391] LOG: redo done at 0/17E59A0

2022-10-15 15:32:38.013 MSK [29389] LOG: database system is ready to accept connections
```

BASIS

Ростелеком

Типовая организация современной СУБД

Основные функции СУБД

- Управление данными во внешней памяти
- Управление буферами оперативной памяти
- Управление транзакциями
- Журнализация и восстановление БД после сбоев
- Поддержка языков БД

Логически в современной реляционной субд можно выделить

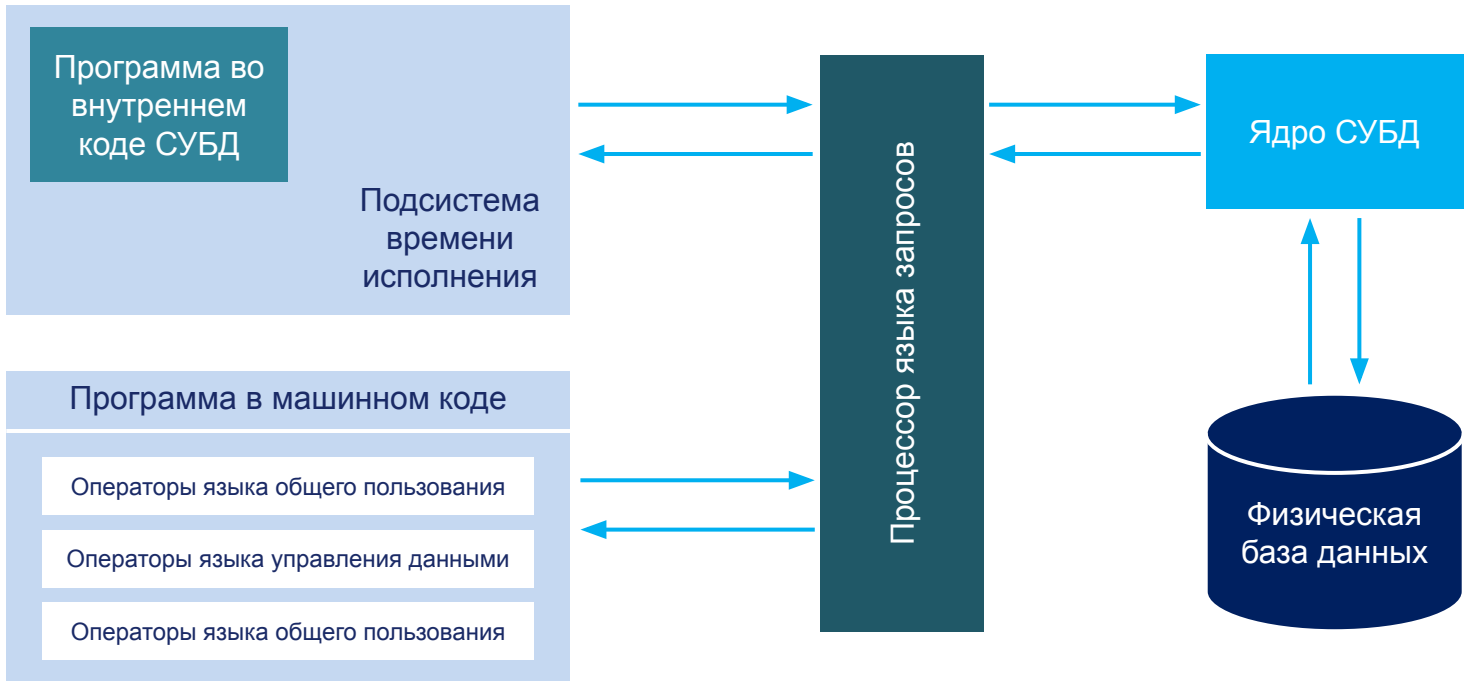
- Поддержка целостности файлов
- Наиболее внутреннюю часть – ядро СУБД (database engine)
- Компилятор языка БД (обычно SQL)
- Подсистему поддержки времени выполнения
- Набор утилит

BASIS

Ростелеком

Ядро СУБД

- Менеджер данных
- Менеджер буферов
- Менеджер транзакций
- Менеджер журнала

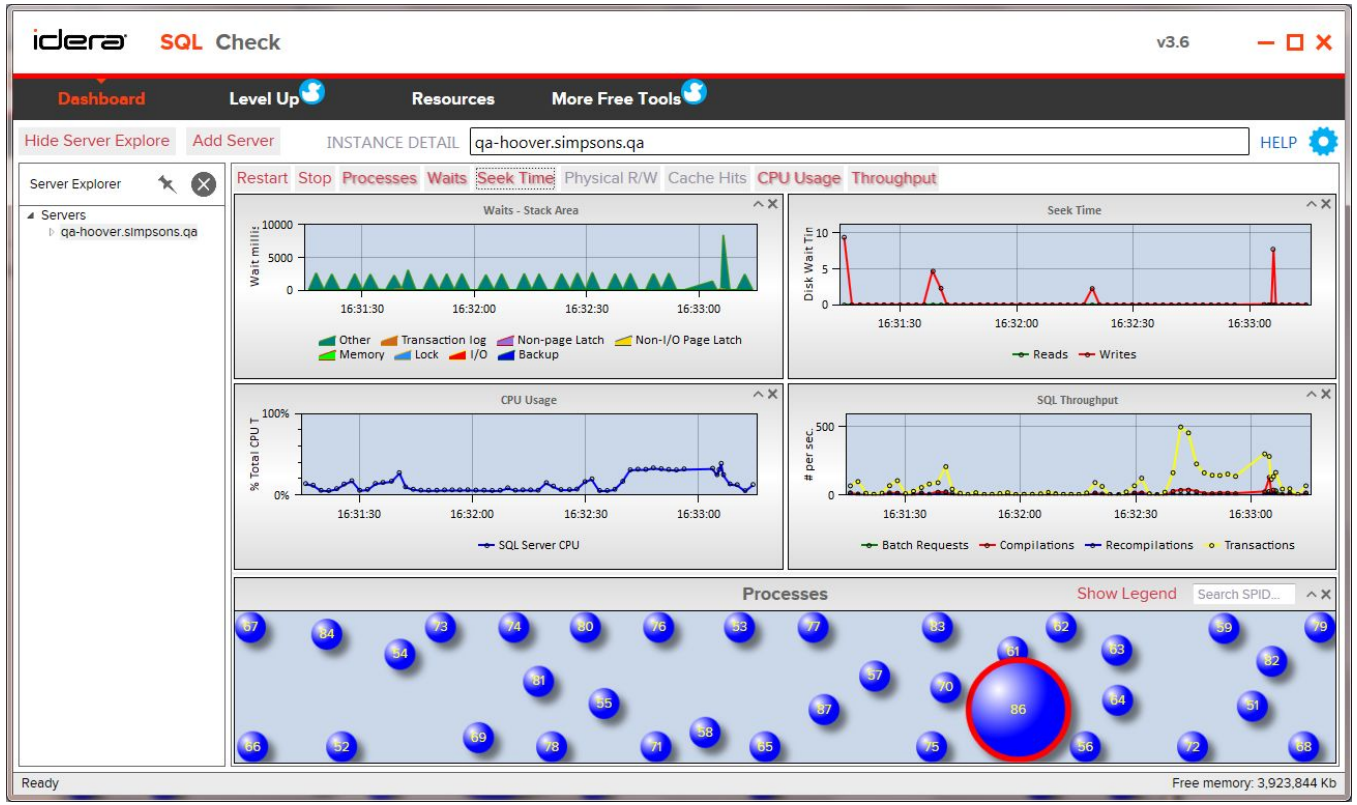


BASIS

Ростелеком

Утилиты БД

- загрузка и выгрузка БД
- сбор статистики
- глобальная проверка целостности БД



BASIS

Ростелеком

Классификация СУБД

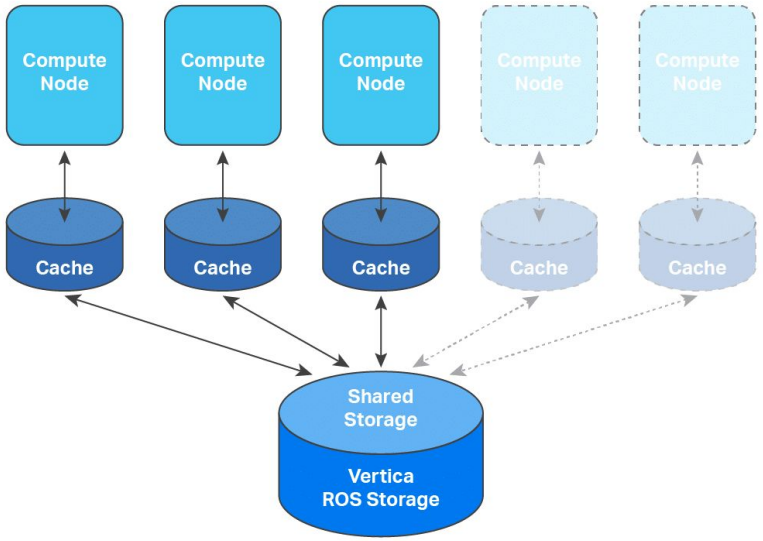
По модели данных

- Загрузка и выгрузка БД
- Сетевые
- Иерархические
- Реляционные (и sql-ориентированные)
- Объектно-ориентированные
- Xml-ориентированные и другие

BASIS

Ростелеком

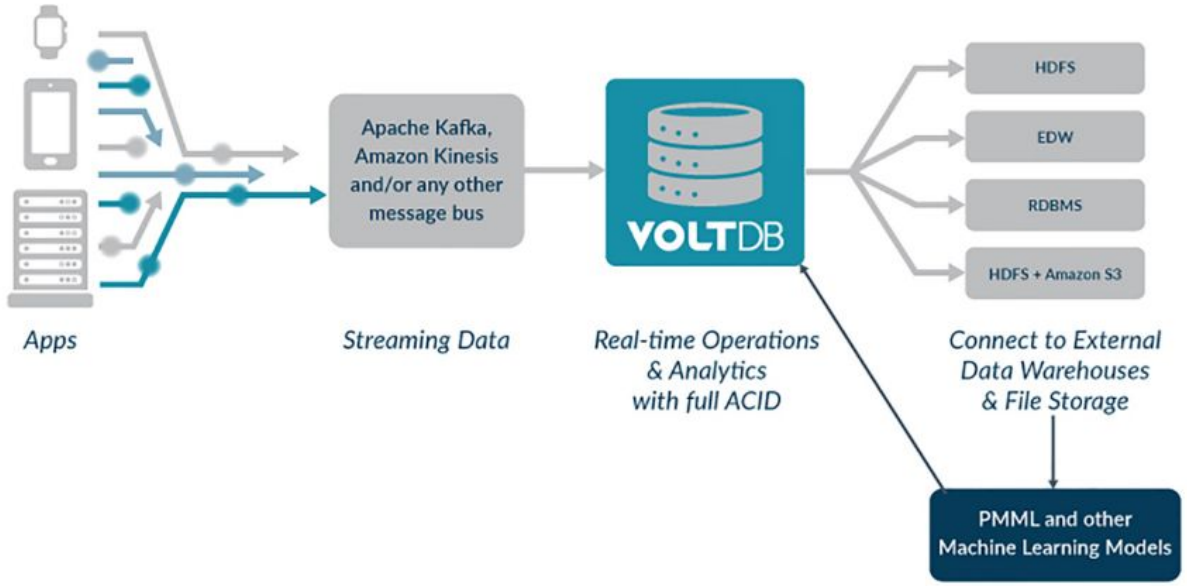
Универсальные и специализированные СУБД



DELL EMC MINIO hadoop

SCALITY PURESTORAGE

H3C NetApp



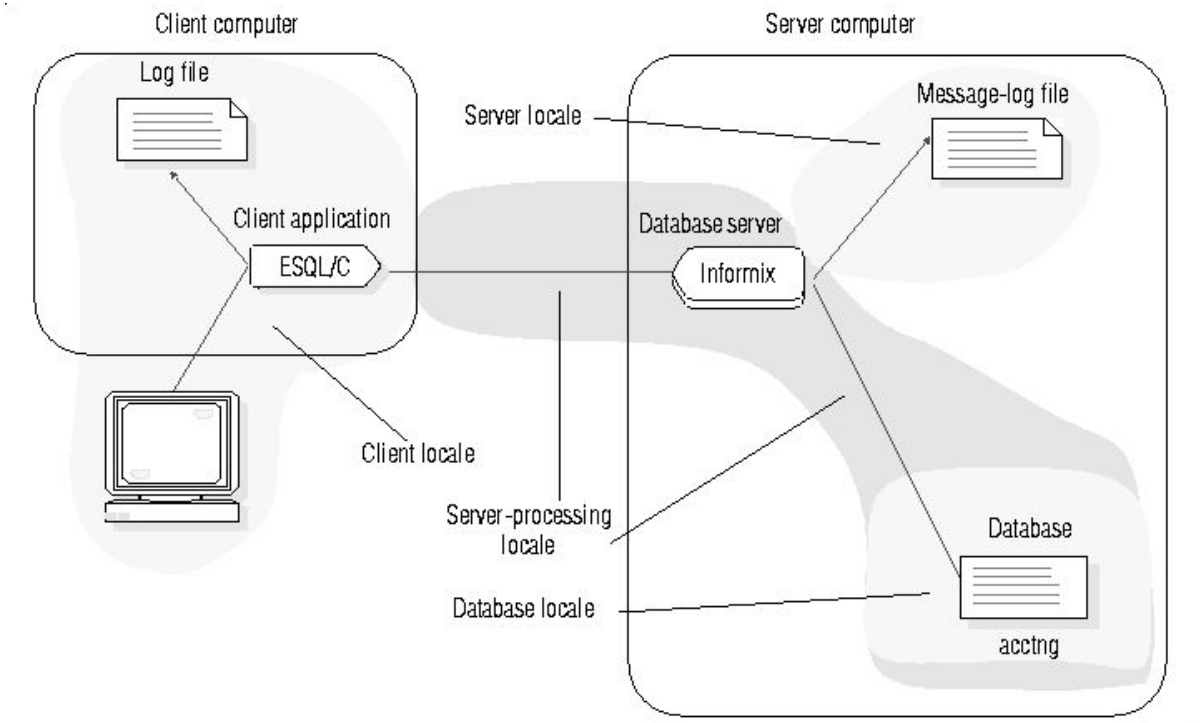
BASIS

Ростелеком

Файл/клиент-серверные и встраиваемые СУБД

СУБД

- Файл-серверные



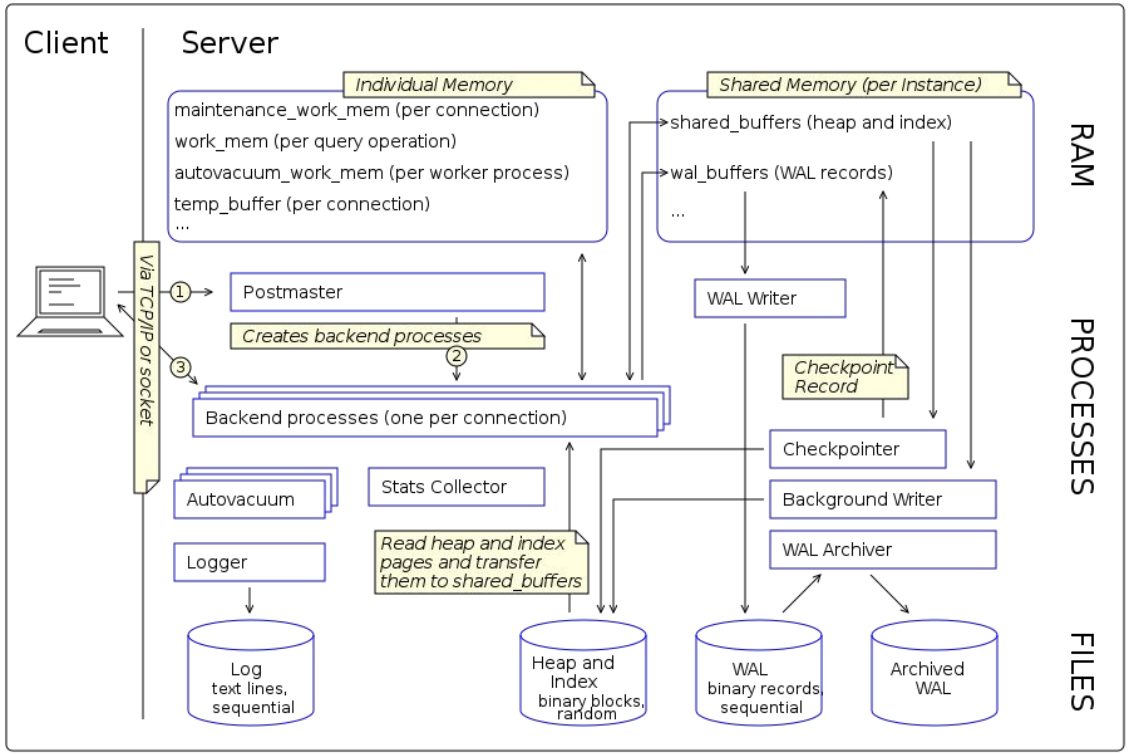
BASIS

Ростелеком

Файл/клиент-серверные и встраиваемые СУБД

СУБД

- Файл-серверные
- Клиент-серверные



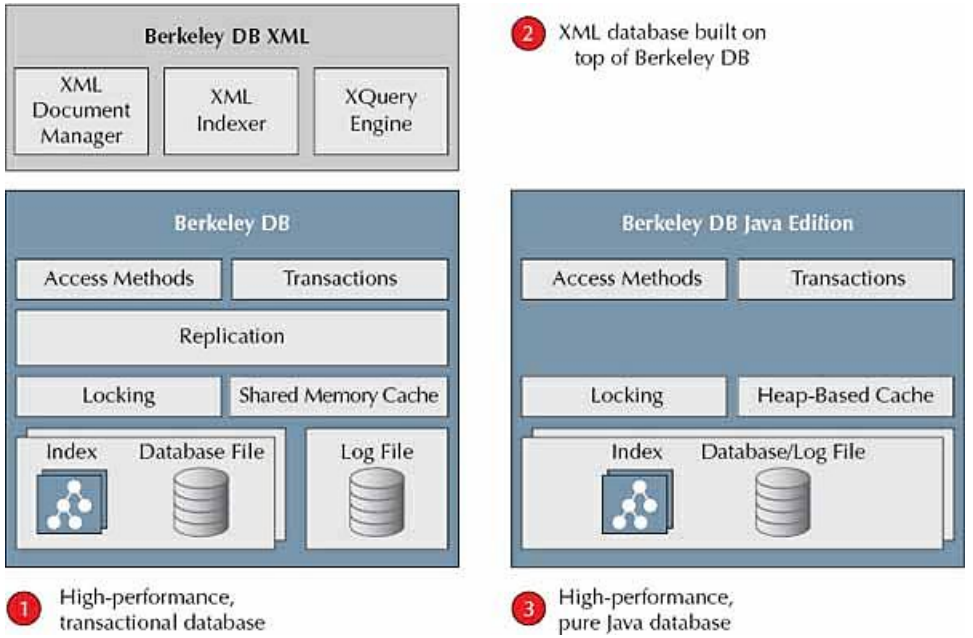
BASIS

Ростелеком

Файл/клиент-серверные и встраиваемые СУБД

СУБД

- Файл-серверные
- Клиент-серверные
- Встраиваемые

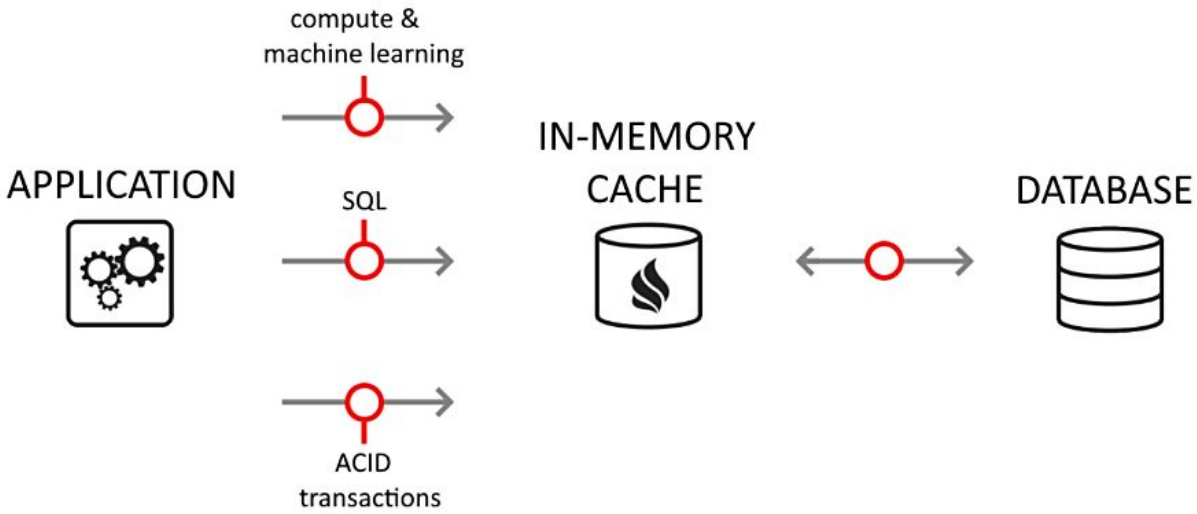


BASIS

Ростелеком

СУБД по месту хранения БД

- Внешняя память вообще не используется, а надёжность достигается за счёт хранения реплик БД в разных узлах кластерной системы
- БД хранится целиком в ОП, а журнал изменений во внешней памяти

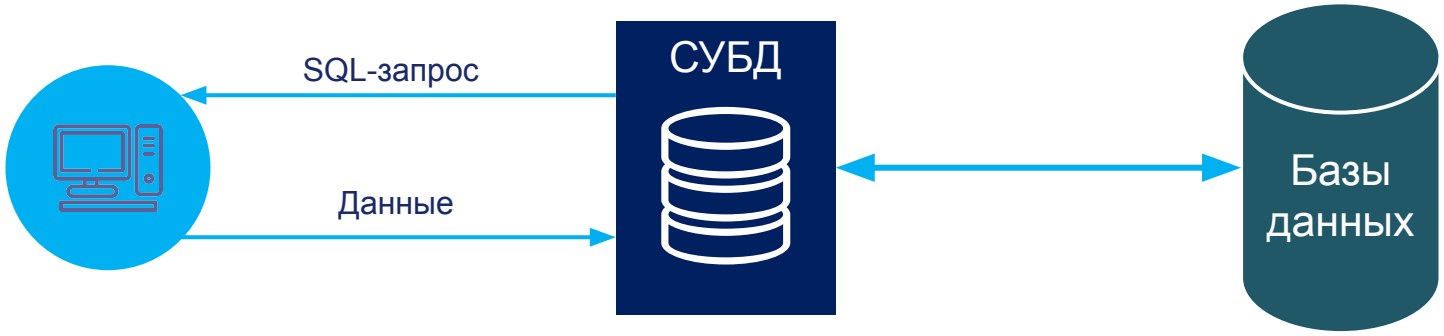


BASIS

Ростелеком

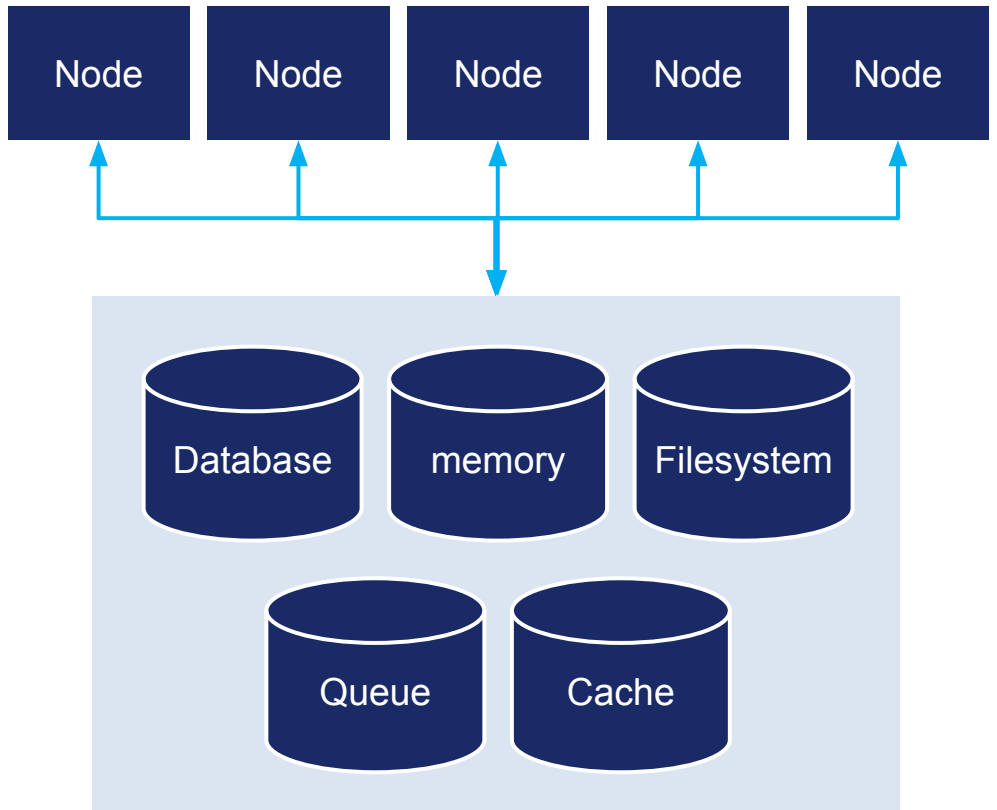
СУБД по типу параллельности

- Однопроцессорные



СУБД по типу параллельности

- Однопроцессорные
- Параллельные с общей памятью (shared-everything)

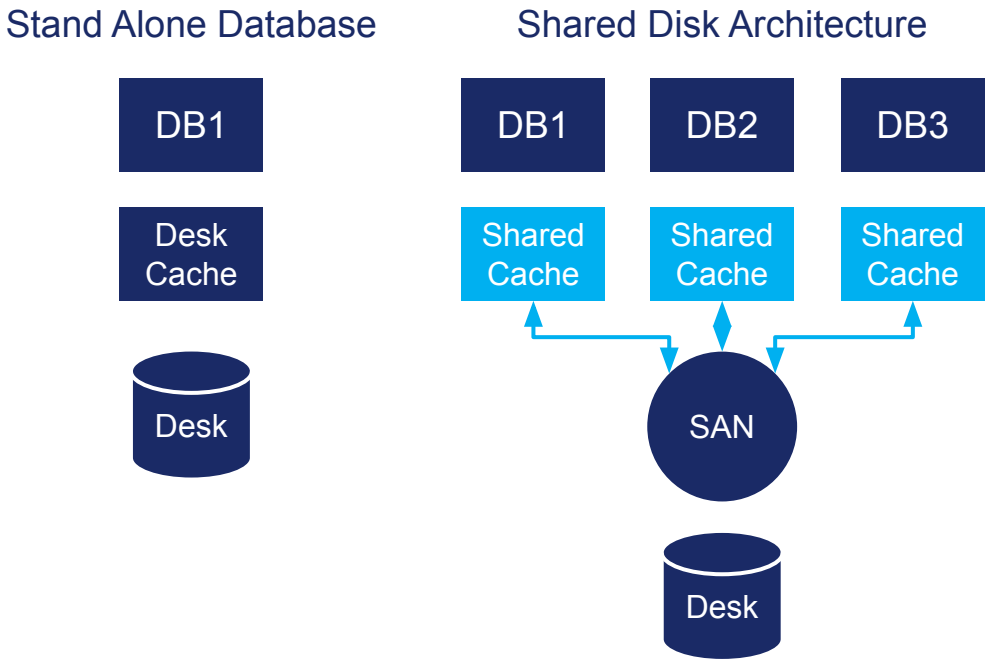


BASIS

Ростелеком

СУБД по типу параллельности

- Однопроцессорные
- Параллельные с общей памятью (shared-everything)
- Параллельные с общими дисками (shared-disks)

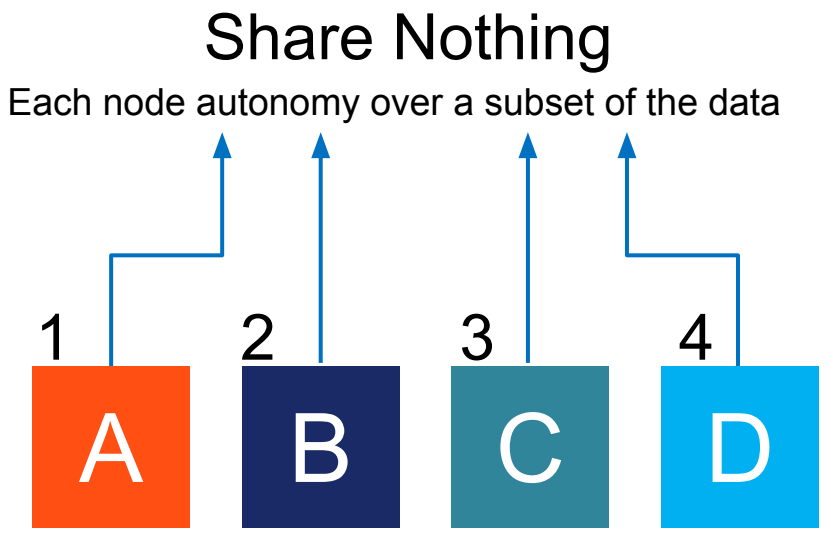


BASIS

Ростелеком

СУБД по типу параллельности

- Однопроцессорные
- Параллельные с общей памятью (shared-everything)
- Параллельные с общими дисками (shared-disks)
- Параллельные без использования общих ресурсов (shared-nothing)



BASIS

Ростелеком

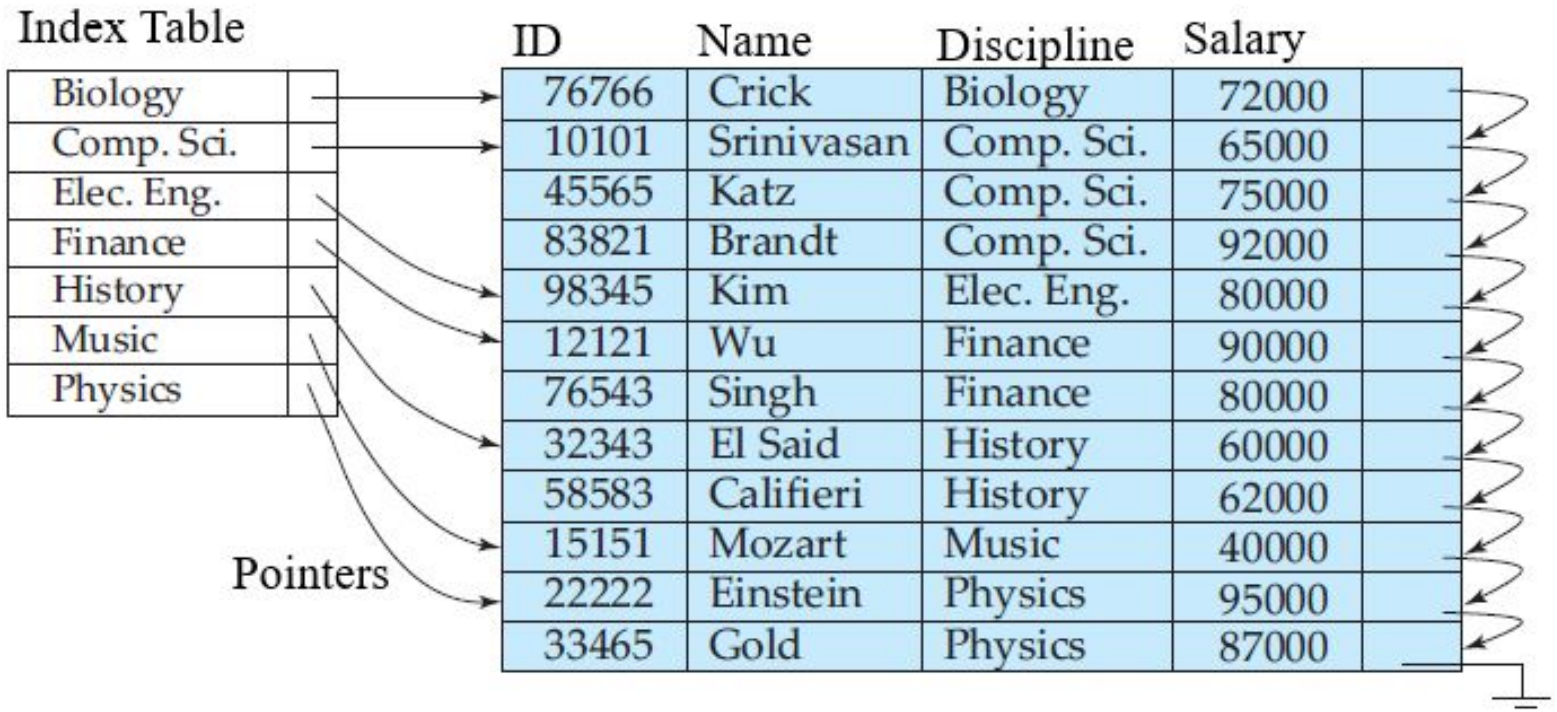
Запросы, индексы и эксплейны

BASIS

Ростелеком

Что такое индексы?

Индексы (indexes) – это особые таблицы, используемые поисковыми системами для поиска данных

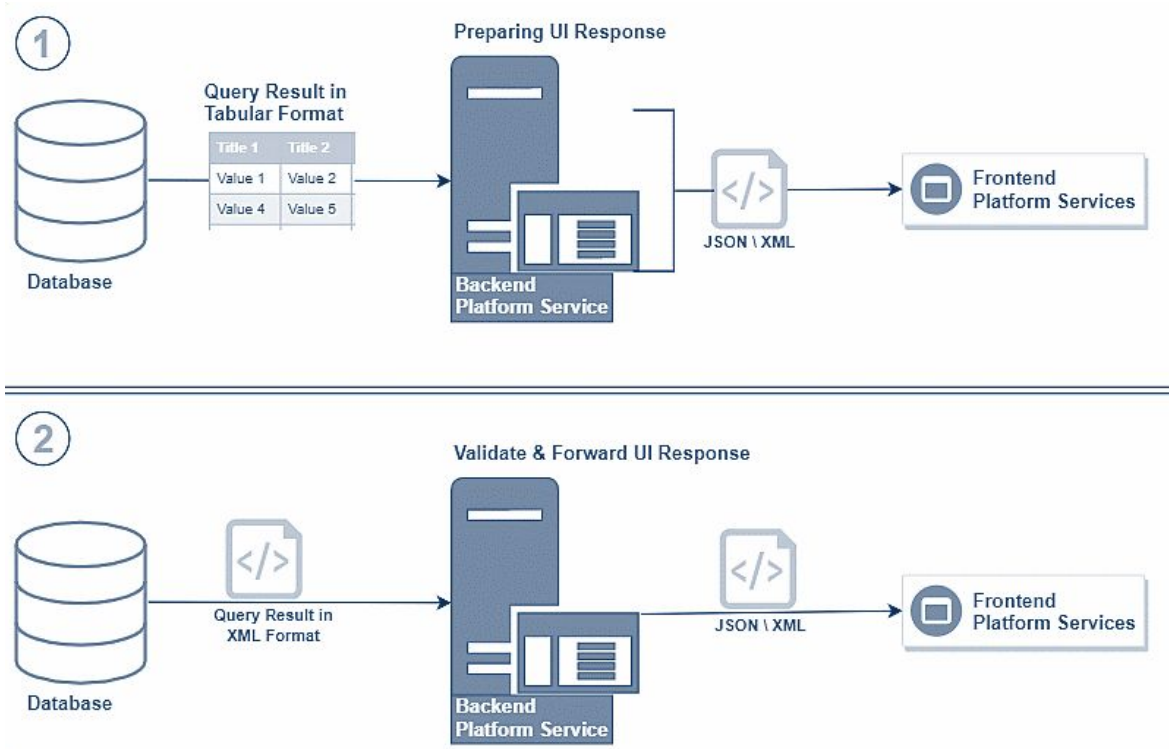


BASIS

Ростелеком

Нельзя создать индекс

- Столбцов, которые используются для хранения данных объектов, имеющих большие размеры, (LOB): TEXT, IMAGE, VARCHAR (MAX)
- Представленных в XML



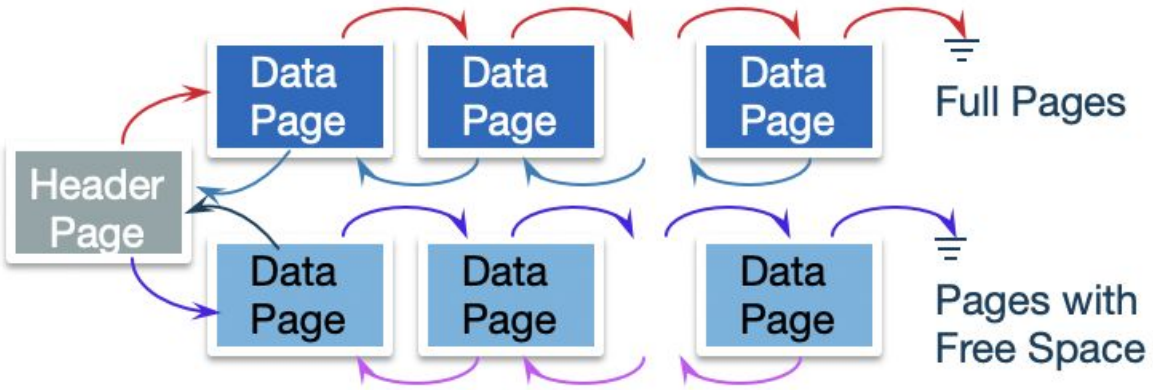
BASIS

Ростелеком

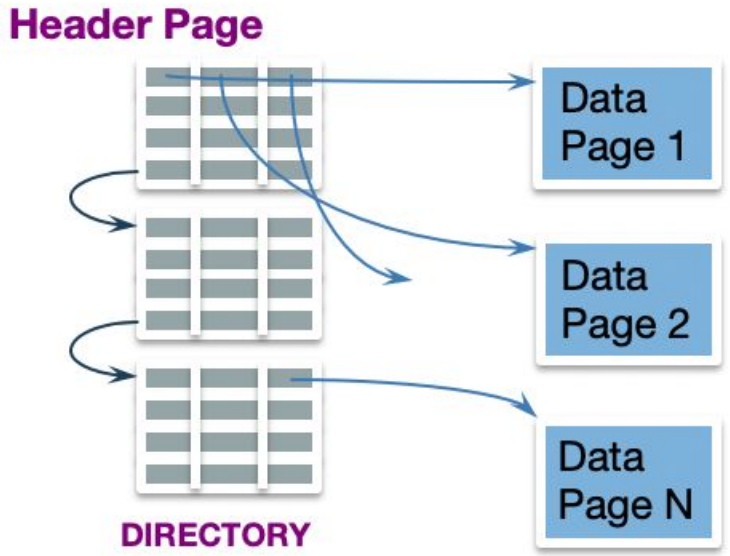
Об индексах и кучах

Как только таблица создана и в ней еще нет индексов, она выглядит как куча данных (Heap). В ней все записи хранятся хаотично, без определенного порядка. Потому их и называют «кучами»

Heap File Implemented as List



Use a Page Directory

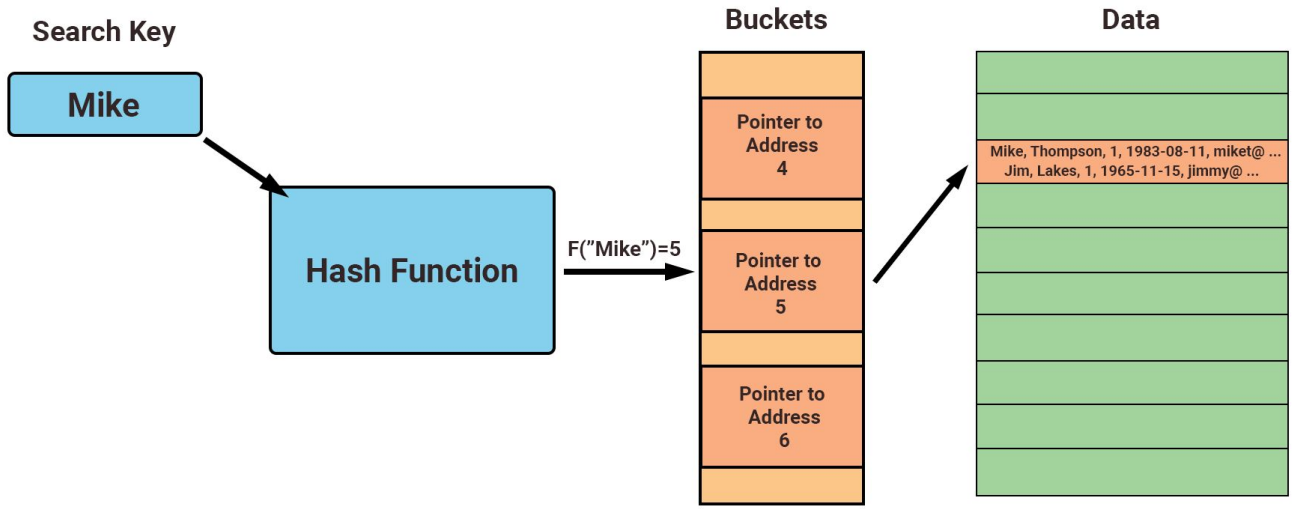


BASIS

Ростелеком

Функции индексов

- Повышение скорости поиска информации и производительности запросов
- Сохранение целостности данных через обеспечение уникальности строк таблицы

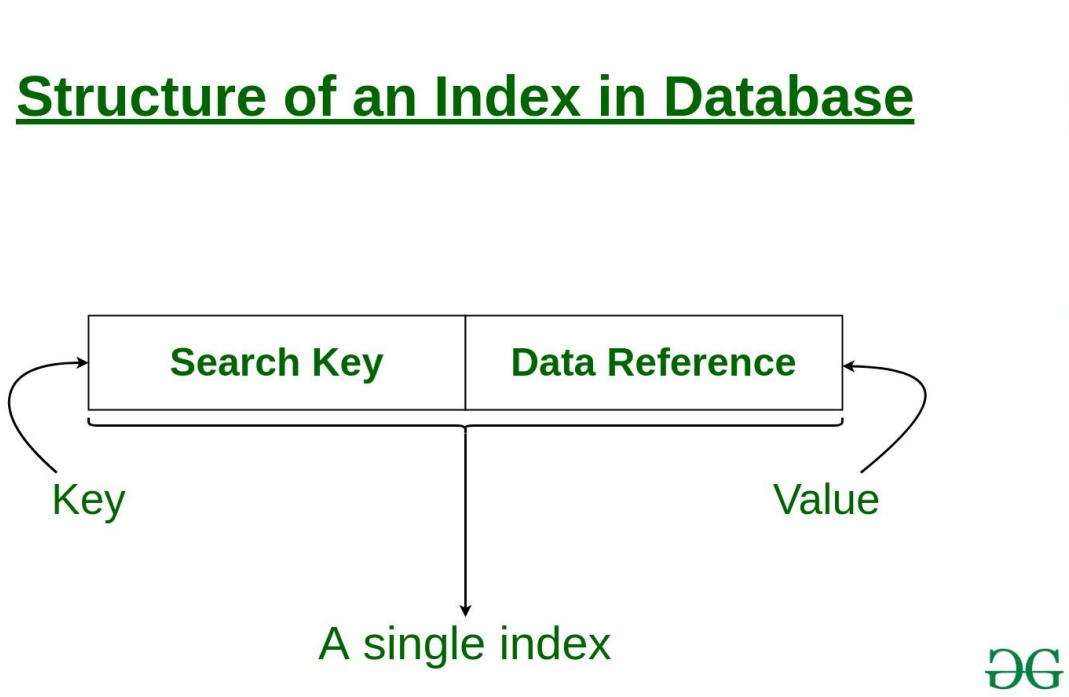


BASIS

Ростелеком

Структура индексов

- Наборов страниц
- Узлов, имеющих древовидную структуру, иерархическую по природе

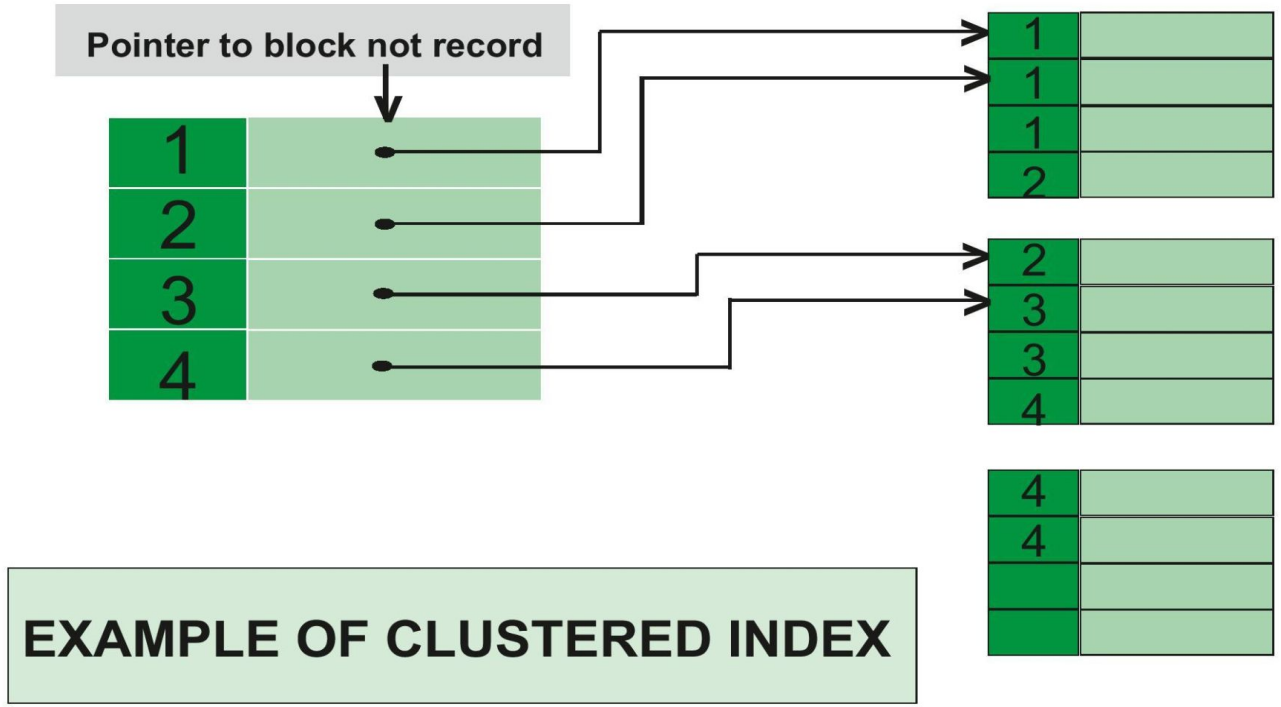


BASIS

Ростелеком

Типы индексов. Кластерный индекс

Задача — сохранение табличных данных в виде, отсортированном по значению ключа.



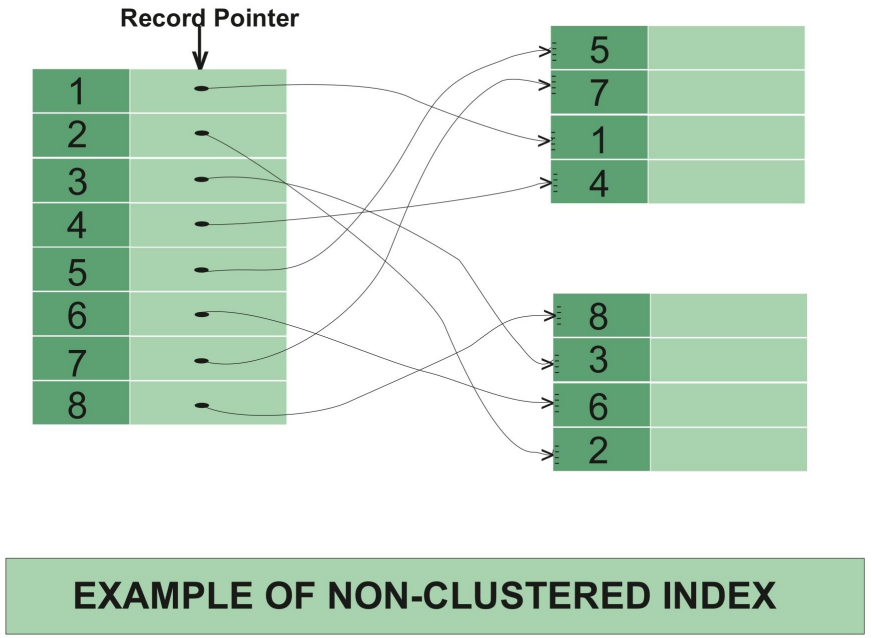
BASIS

Ростелеком

Типы индексов. Некластерный индекс

Индекс содержит

- Значения ключей – ключевые столбцы, по которым они определены
- Указатели на строки в таблице, содержащие реальные данные (значения ключа).



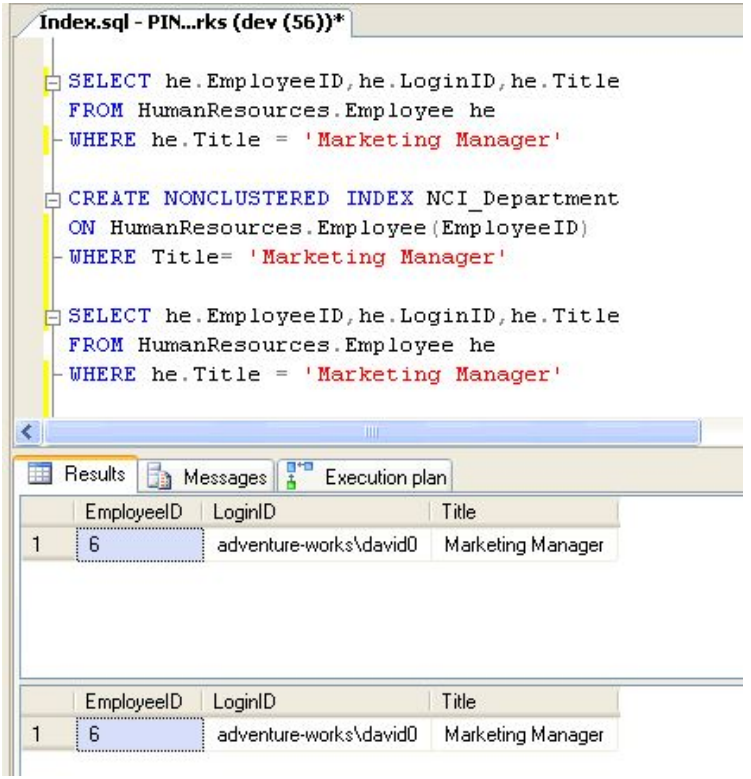
EXAMPLE OF NON-CLUSTERED INDEX

BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (Filtered)



BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)

```
1.  CREATE INDEX name_in
2.  ON dataflair_employee (name_emp);
```

BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)

```
1.  CREATE UNIQUE INDEX name_unique
2.  ON dataflair_employee (name_emp);
```

BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)
- Колоночный (columnstore)

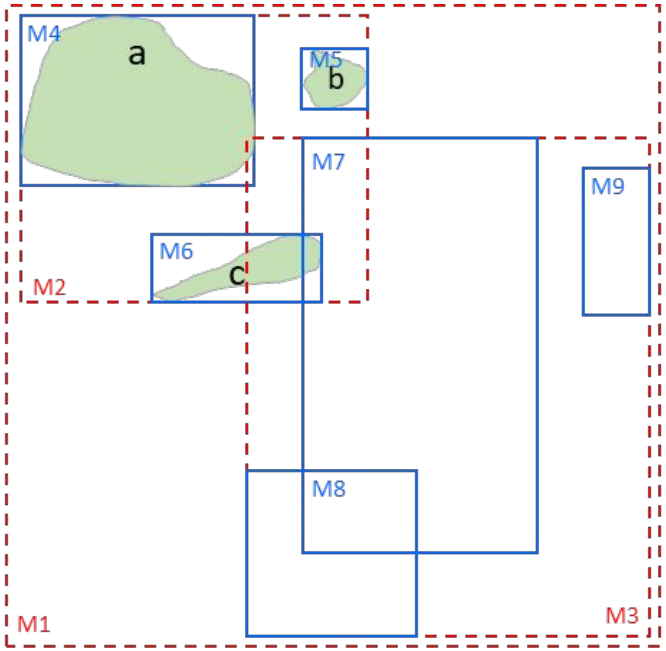
```
1.  CREATE INDEX name_multiple
2.  ON dataflair_employee (name_emp, salary);
```

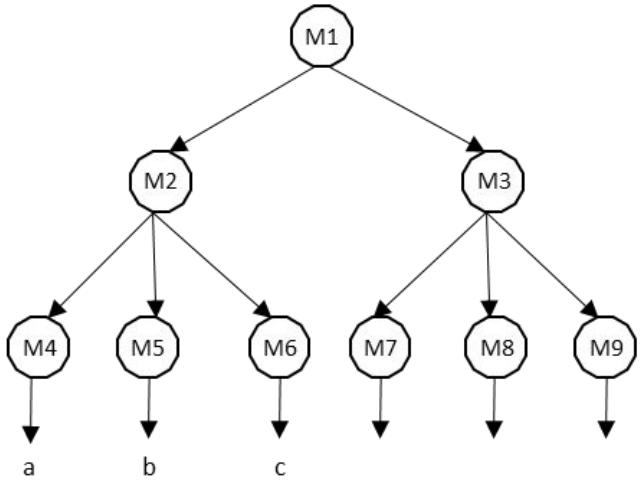
BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)
- Колоночный (columnstore)
- Пространственный (spatial)



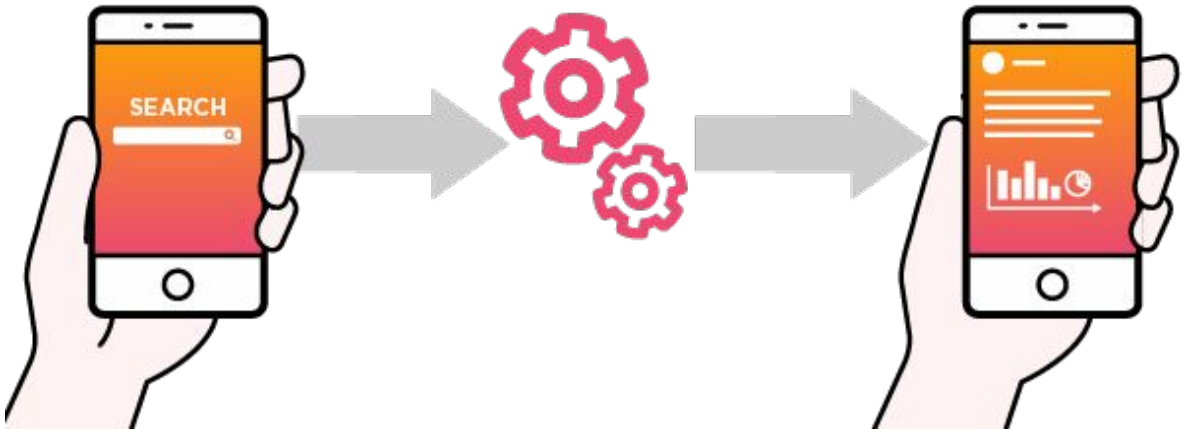


BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)
- Колоночный (columnstore)
- Пространственный (spatial)
- Полнотекстовый (full-text)

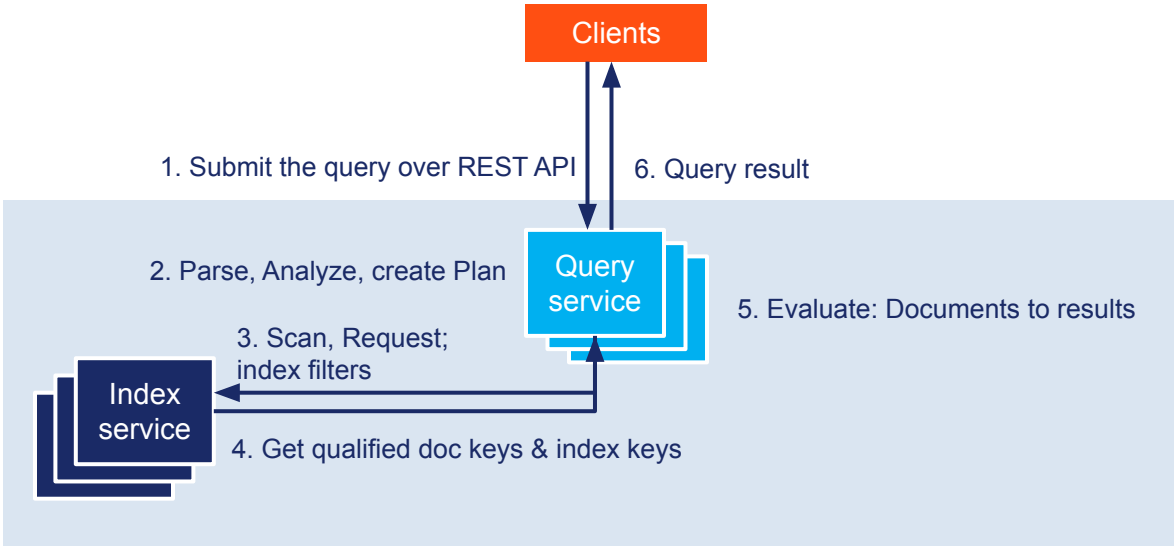


BASIS

Ростелеком

Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)
- Колоночный (columnstore)
- Пространственный (spatial)
- Полнотекстовый (full-text)
- Покрывающий (covering)

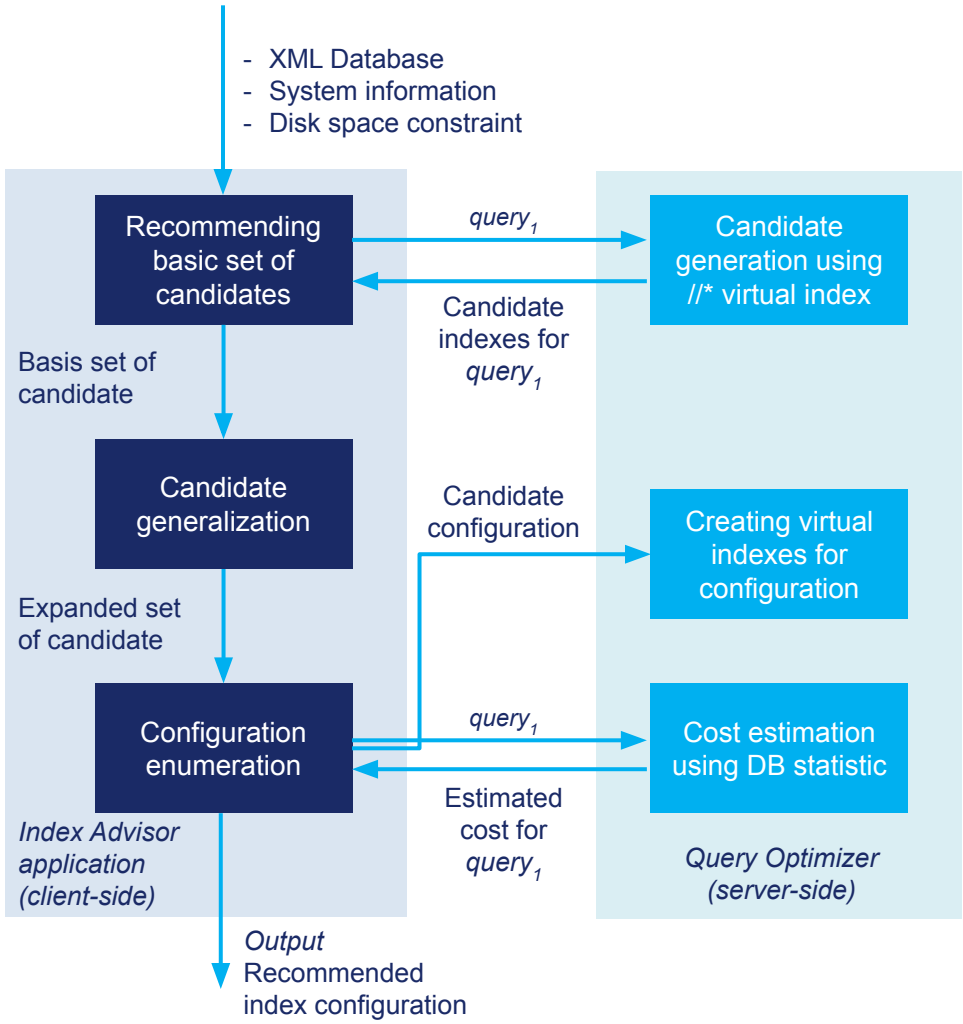


BASIS

Ростелеком

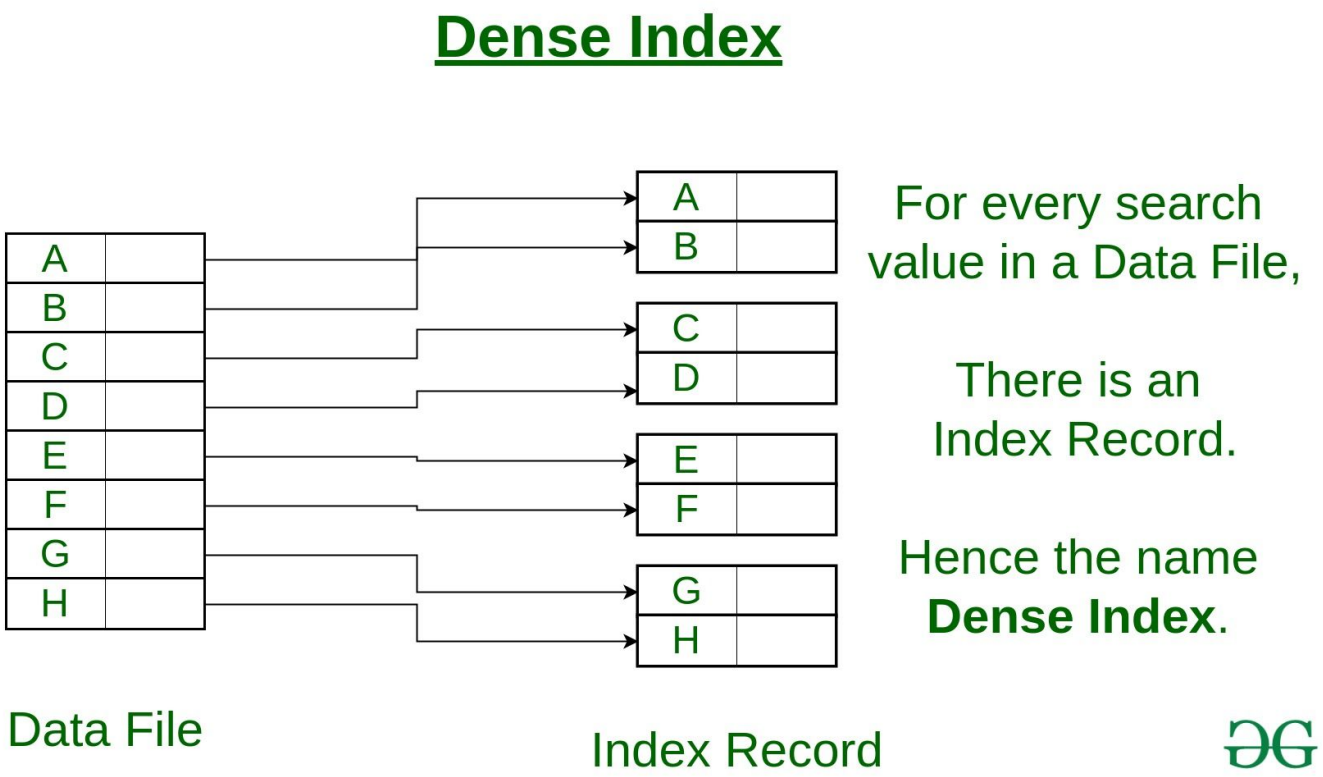
Специальные типы индексов

- Фильтруемый (filtered)
- Составной (composite)
- Уникальный (unique)
- Колоночный (columnstore)
- Пространственный (spatial)
- Полнотекстовый (full-text)
- Покрывающий (covering)
- Xml-индекс



Индексы в оптимизированных таблицах

- Оптимизированные для памяти (In-Memory OLTP)
- Nonclustered indexes



Performance database

1. Если предполагается частое обновление данных в таблице, то для нее нужно применять минимум индексов
2. Для таблицы с большим кол-вом данных можно использовать то число индексов, которое улучшит производительность запросов
3. Для Clustered indexes используйте самые короткие поля.
4. Производительность индекса зависит от того, насколько уникальны значения в столбце.
5. Если используется составной индекс, то в нем нужно учитывать порядок столбцов
6. Допускается использование индекса на вычисляемых столбцах таблицы, но лишь при условии соблюдения определенных требований

BASIS

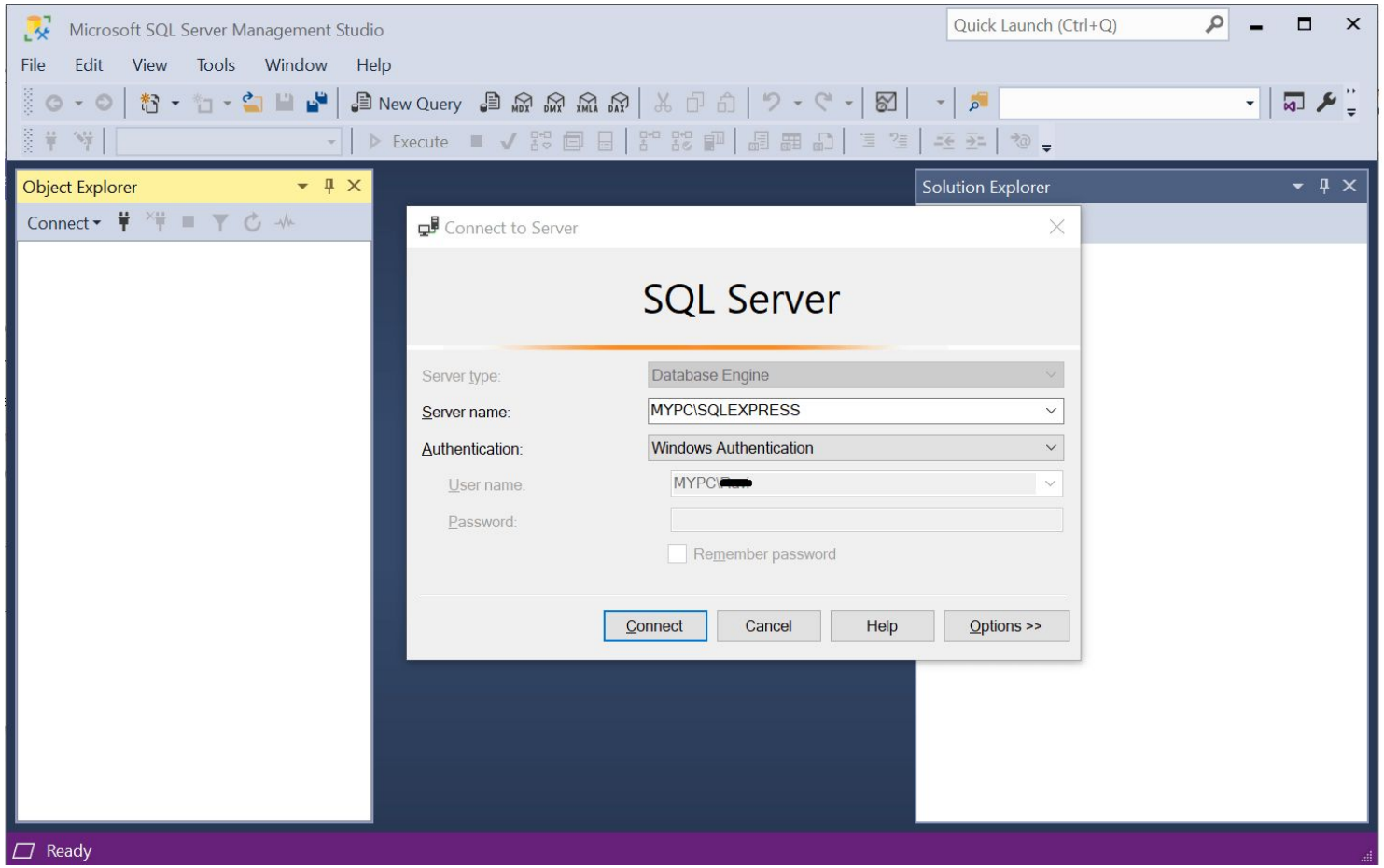
Ростелеком

Запросы к БД

- Предпочтительнее, чтобы один запрос содержал наибольшее число строк
- На столбцах, используемых в запросах с WHERE, предпочтительнее создавать Nonclustered index в качестве условия поиска и соединения в JOIN
- Следует воспользоваться возможностями индексирования столбцов

Способы создания индексов ms sql server

- SSMS (MSSQL Management Studio)
- Специальный язык Transact-SQL

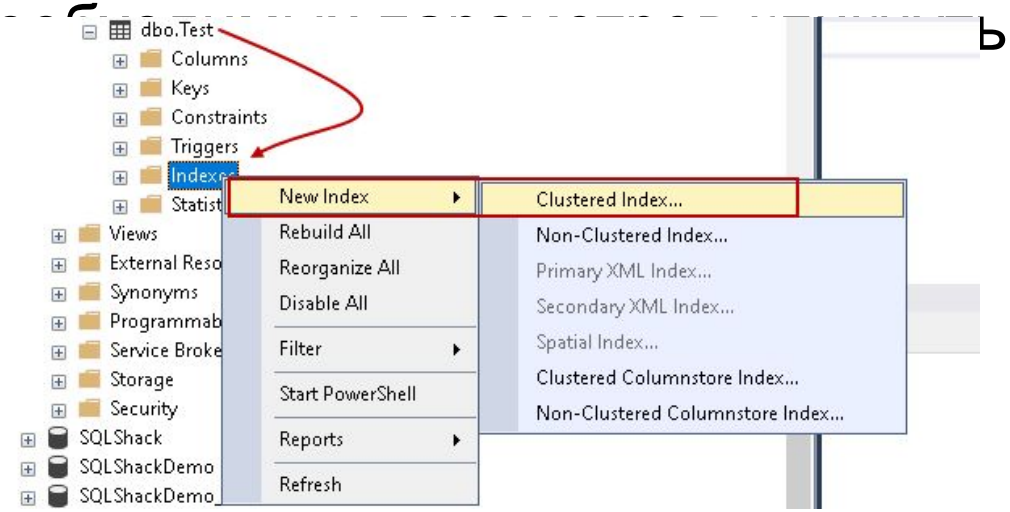


BASIS

Ростелеком

Создать кластерный индекс в Management Studio

1. Открыть SSMS
2. Выбрать соответствующую таблицу
3. Остановившись на пункте «Индексы»
4. Выбрать «Создать индекс» и выбираем «Кластеризованный»
5. В новом окне появится форма «Новый индекс»
6. Выбрать столбец, который будет являться ключом индекса и «Добавить»
7. После ввода всех [illegible] «ОК»

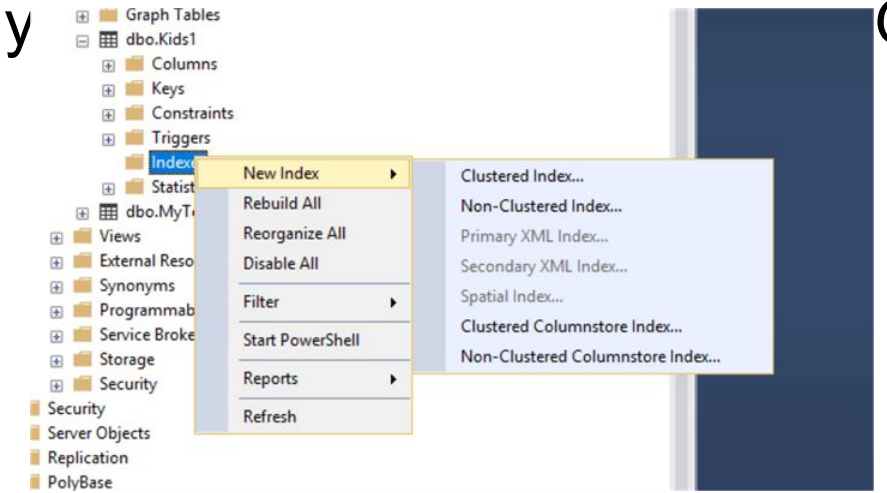


BASIS

Ростелеком

Создать некластерный индекс в Management Studio

1. Открыть SSMS
2. Выбрать требуемую таблицу и щелкнуть по пункту «Индексы»
3. Выбрать «Создать индекс», «Некластеризованный»
4. В открывшейся форме «Новый индекс» вписать наименование нового индекса, добавить один или несколько столбцов через «Добавить»
5. Перейти во вкладку «Включено столбцы». Добавить все столбцы, воспользовавшись кнопкой «Добавить».
6. Когда введены все ну ОК»

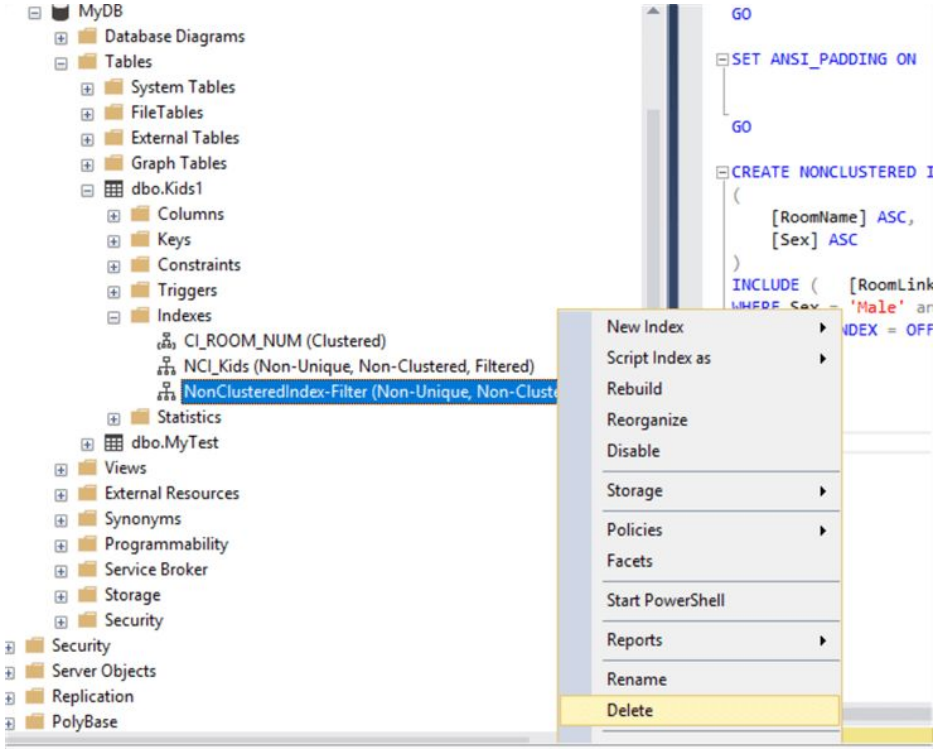


BASIS

Ростелеком

Удаление индекса в Management Studio

1. Открыть SSMS
2. Выбрать индекс, подлежащий удалению
3. Щелкнуть мышкой по нему и из списка выбрать «Удалить»
4. Выполненное действие подтвердить нажатием «ОК»



BASIS

Ростелеком

Оптимизация индексов

Выполнить запрос:

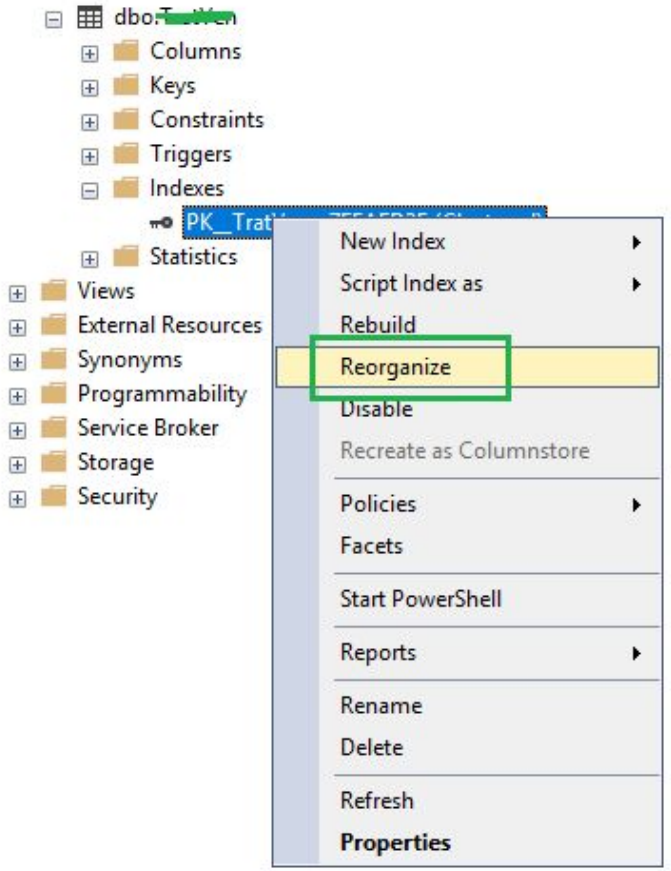
```
SELECT OBJECT_NAME(T1.object_id) AS NameTable,
        T1.index_id AS IndexId,
        T2.name AS IndexName,
        T1.avg_fragmentation_in_percent AS Fragmentation
    FROM sys.dm_db_index_physical_stats (DB_ID(), NULL, NULL,
NULL, NULL) AS T1
    LEFT JOIN sys.indexes AS T2 ON T1.object_id = T2.object_id AND
T1.index_id = T2.index_id
```

BASIS

Ростелеком

Реорганизация индексов

1. Открыть SSMS
2. На выбранном индексе следует щелкнуть мышкой, из списка выбрать и нажать «Реорганизовать»
3. Соответствующими инструкциями T-SQL



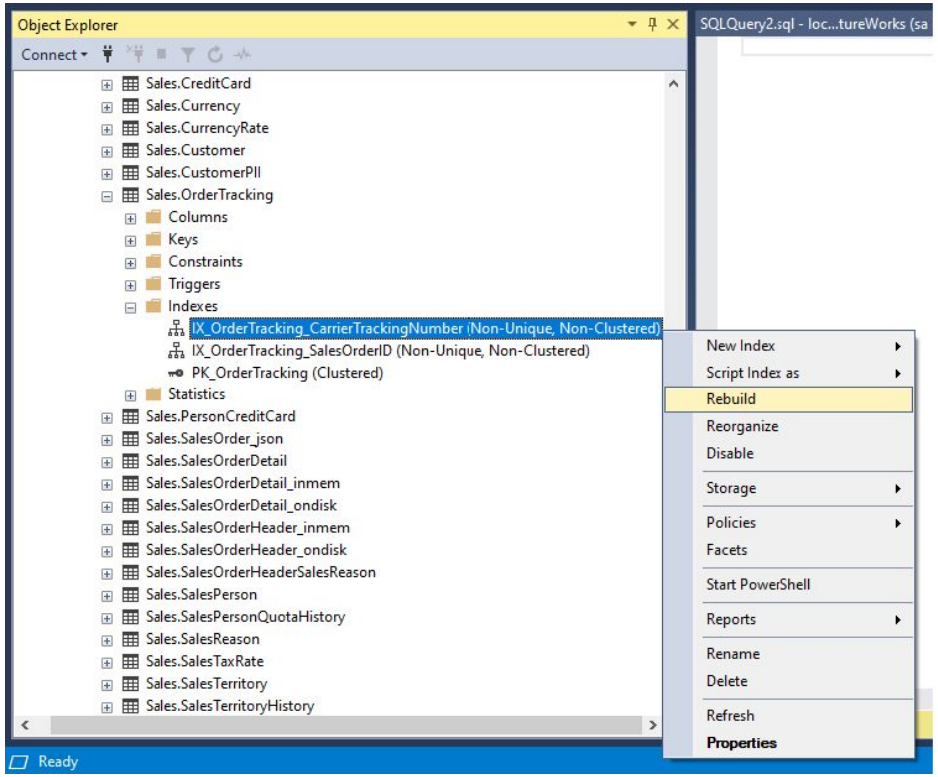
BASIS

Ростелеком

Перестроение индексов

Открыть SSMS:

- Выбрать нужный индекс, мышкой кликнуть по нему и выбрать «Перестроить»
- ALTER INDEX ix с предложением REBUILD, которая по сути является заменой инструкции DBCC DBREINDEX
- CREATE NONCLUSTERED INDEX (CREATE INDEX) с предложением DROP_EXISTING



BASIS

Ростелеком

Администрирование MySQL

BASIS

Ростелеком

База данных MySQL

MySQL является ведущей системой управления базами данных с открытым исходным кодом. Разработка MySQL началась в 1994 году шведской компанией MySQL AB.

MariaDB — это разработанный сообществом форк MySQL, предназначенный для того, чтобы оставаться свободным под GNU GPL.

BASIS

Ростелеком

Основные понятия и компоненты MySQL

Каталог данных - содержит всю информацию, которая управляется сервером *«mysqld»* (базы данных, таблицы, файлы состояния). Место расположение *«каталога данных»* можно задать при запуске сервера с помощью опции:

```
-h|--datadir=path Path to the database root.
```

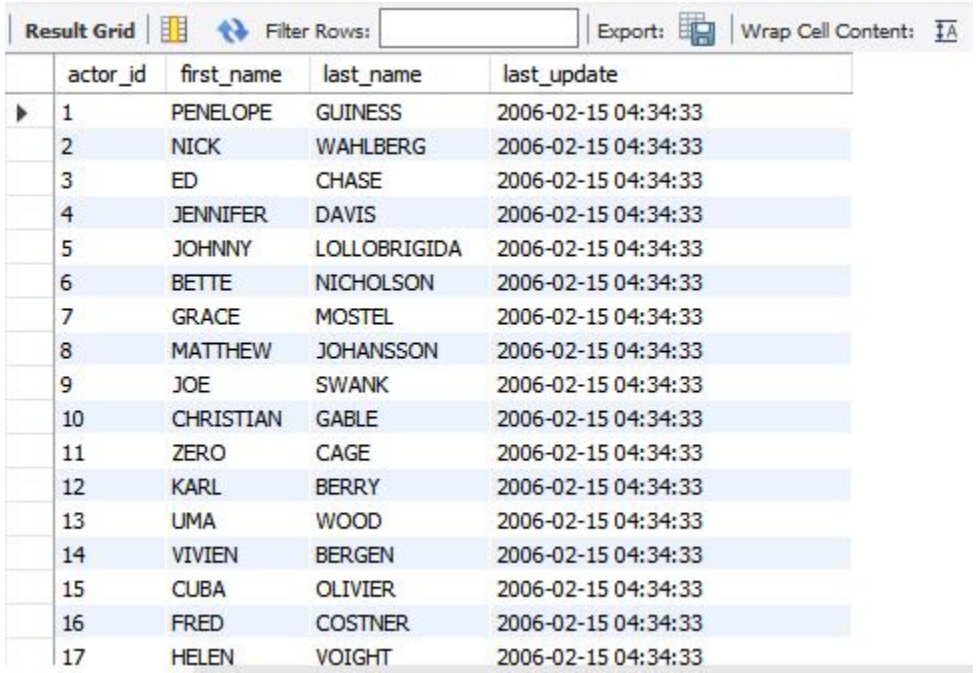
Определить текущие расположение *«каталог данных»* можно с помощью команды:

```
shell#> mysqladmin variables | grep datadir
```

Основные понятия и компоненты MySQL

База данных - каждая БД представляет собой подкаталог в каталоге *«каталога данных».*

Таблица - это три специальных файла размещенных внутри каталога «базы данных» для каждой таблицы.

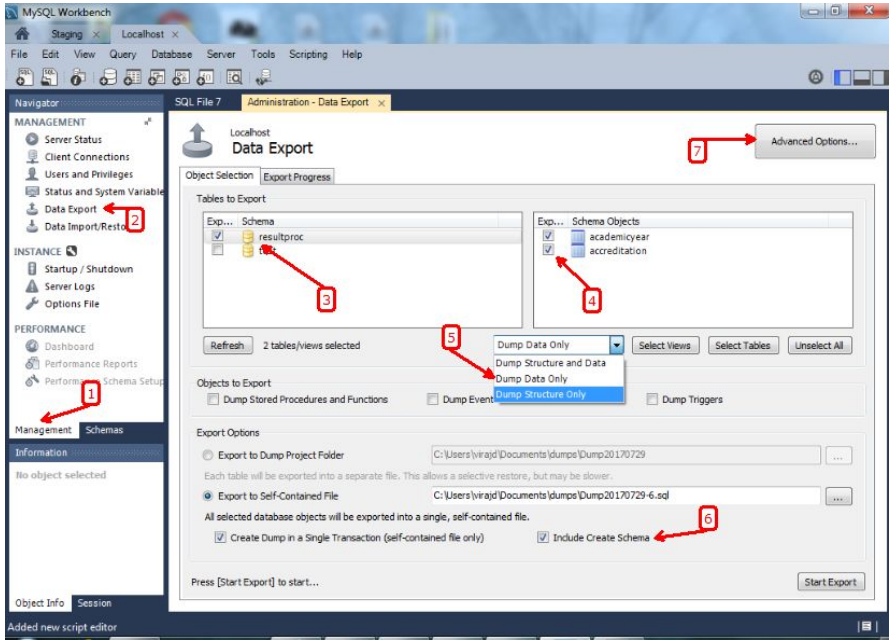


actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	ZERO	CAGE	2006-02-15 04:34:33
12	KARL	BERRY	2006-02-15 04:34:33
13	UMA	WOOD	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33
16	FRED	COSTNER	2006-02-15 04:34:33
17	HELEN	VOIGHT	2006-02-15 04:34:33

Основные понятия и компоненты MySQL

Файлы состояний MySQL

- .pid PID процесса сервера *--pid-file*
- .err журнал ошибок
- .log общий журнал *-l | --log*
- .nnn журнал обновлений *--log-bin | --log-update*



BASIS

Ростелеком

Основные программы и утилиты MySQL

mysqld	Сам сервер/демон MySQL.
mysql	клиент для работы с сервером MySQL.
mysqladmin	ПО для выполнения административных функций.
myisamchk	ПО для проверки и восстановления MyISAM таблиц.
mysqldump	Консольный клиент для создания «дампов» или резервных копии БД, таблиц и хранимых данных.
perror	По номеру ошибки выводит на экран описание этой ошибки.
mysqld_safe	Скрипт для запуска mysqld в системах UNIX.

Полезные команды/запросы клиента mysql

Подключение к серверу MySQL с БД осуществляется с помощью клиента «mysql». Синтаксис для подключения следующий:

```
shell#> mysql -h [hostname] -P [порт]
--protocol=[tcp|socket|pipe|memory] -u [username] -p[пароль]
[имя_БД]
```

BASIS

Ростелеком

Полезные команды/запросы клиента mysql

SHOW DATABASES;	Выводит список всех БД обслуживаемых сервером, аналогично «mysqlshow».
USE [имя_БД]	Делает базу данных [имя_БД] «текущей» (активной).
SHOW TABLES;	Выводит список всех таблиц в *«текущей»* БД. аналогично *«mysqlshow [имя_БД]»*.
DESCRIBE [имя_таблицы];	Выводит описание таблицы [имя_таблицы] (имена столбцов, типы данных, и т.п). Аналогично «mysqlshow [имя_БД] [имя_таблицы]»

BASIS

Ростелеком

Полезные команды/запросы клиента mysql

CREATE DATABASE [имя_БД];	Создает БД с именем [имя_бд]
SELECT DATABASE();	Выводит текущую БД
SELECT USER();	Выводит имя (username) текущего пользователя
SELECT VERSION();	Выводит информацию о версии сервера *«mysqld»*
TRUNCATE TABLE [имя_таблицы];	Удаляет из таблицы [имя_таблицы] все строки
SELECT	Выбирает и возвращает строки из заданных таблиц.

BASIS

Ростелеком

Полезные команды/запросы клиента mysql

Чтобы найти все установленные файлы какого-либо пакета, можно воспользоваться командой:

```
shell#> pkg_info -xL [имя_пакета] (для debain семейства)
```

среди этих файлов есть файлы документации:

```
/usr/local/share/doc/mysql/manual.html
/usr/local/share/doc/mysql/manual.txt
/usr/local/share/doc/mysql/manual_toc.html
```

Полезные команды/запросы клиента mysql

Чтобы найти справку по нужному оператору надо выполнить соответствующий запрос `SELECT`.

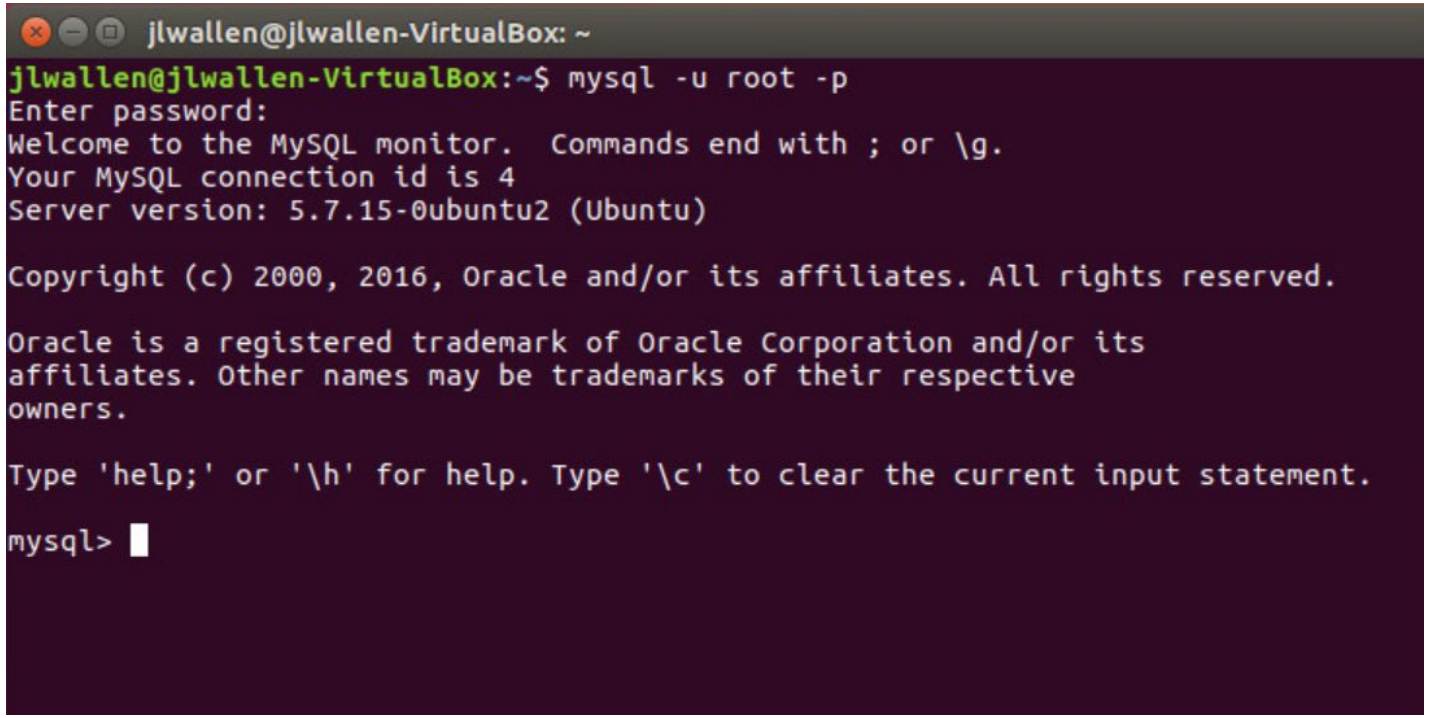
Пример:

```
mysql#> USE mysql;
mysql#> SELECT description, example FROM help_topic WHERE
name="SHOW";
```

поиск описания и примеров синтаксиса оператора SHOW.

Методы запуска сервера

- Непосредственный вызов mysqld



BASIS

Ростелеком

Методы запуска сервера

- Непосредственный вызов mysqld
- Вызов сценария safemysqld(mysqld_safe)

```
mysql> mysqld_safe --port=port_num --defaults-file=file_name
```

Instead, use the following command:

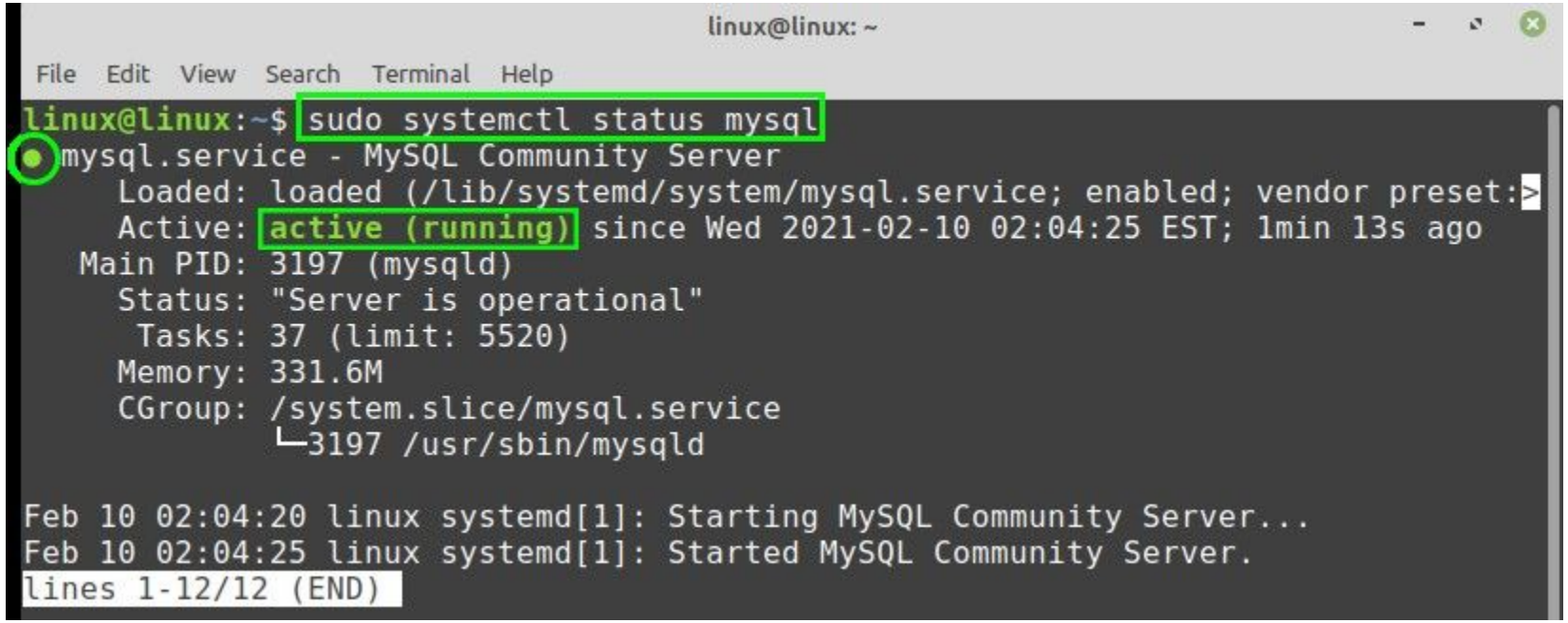
```
mysql> mysqld_safe --defaults-file=file_name --port=port_num
```

BASIS

Ростелеком

Методы запуска сервера

- Непосредственный вызов mysqld
- Вызов сценария safemysqld(mysqld_safe)
- Вызов сценария mysql.server



BASIS

Ростелеком

Определение опций запуска

Во-первых, можно изменить используемый *сценарий запуска* (safemysqld или mysql.server) и задать параметры непосредственно в строке вызова сервера.

Во-вторых, можно определить параметры собственно в конфигурационном файле.

Однако есть *информация*, которую невозможно задать в конфигурационных файлах. Для ее определения необходимо изменить *сценарий* safemysqld.

Следует помнить, что после повторной *инсталляции MySQL* (например, при обновлении версии) все внесенные в *сценарий запуска* изменения будут потеряны.

Завершение работы сервера

Для самостоятельного завершения работы сервера применяется *команда* mysqladmin:

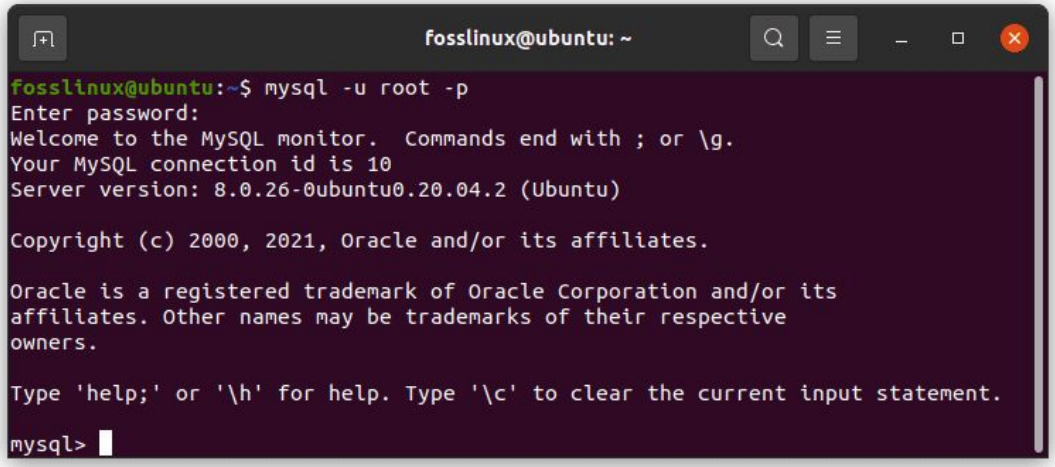
```
% mysqladmin shutdown
```

Работа с учетными записями пользователей MySQL

Идентификация и права доступа

Проверка прав доступа к данным осуществляется в два этапа:

1. Сервер проверяет, разрешено ли пользователю вообще подключаться к *«mysqld»* демону
2. Если 1-й этап прошел успешно, то сервер начинает, проверяет каждый *«запрос»* пользователя на наличие привилегий для выполнения этого *«запроса»*



BASIS

Ростелеком

Работа с учетными записями пользователей MySQL

Четыре уровня привилегий

Глобальный уровень:

Глобальные привилегии применяются ко всем БД на указанном сервере. Они хранятся в таблице «user».

Уровень базы данных:

Привилегии БД применяются ко всем таблицам указанной базы данных. Они хранятся в таблицах *«db»* и *«host»*.

Уровень таблицы:

Привилегии таблицы применяются ко всем столбцам указанной таблицы. Они хранятся в таблице *«tables_priv»*.

Уровень столбца:

Привилегии столбца применяются к отдельным столбцам указанной таблицы. Они хранятся в таблице *«columns_priv»*.

BASIS

Ростелеком

Работа с учетными записями пользователей MySQL

Два типа полей

- Поля контента
- Поля привилегий

Поля контекста определяют область действия каждой из записей в таблицах

Для таблицы «user» контекстные поля следующие: *Host,User,Password*

Для таблицы «db» контекстные поля следующие: *Host,Db,User*

Для таблицы «host» контекстные поля следующие: *Host, Db*

Для таблицы «tables_priv» контекстные поля следующие: *Host,Db,User,Table_name*

Для таблицы «columns_priv» контекстные поля следующие: *Host,Db,User,Table_name, Column_name*

BASIS

Ростелеком

Создание MySQL пользователей и назначение прав

Создавать/удалять пользователей MySQL можно используя, операторы *CREATE USER*, *DROP USER*:

```
CREATE USER user [IDENTIFIED BY [PASSWORD] 'password']
[, user [IDENTIFIED BY [PASSWORD] 'password']]

DROP USER user [, user] ...
```

BASIS

Ростелеком

Создание MySQL пользователей и назначение прав

Назначать привилегии лучше используя, оператор *GRANT*:

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)] ...]
    ON {tbl_name | * | *.* | db_name.*}
    TO user_name [IDENTIFIED BY [PASSWORD] 'password']
        [, user_name [IDENTIFIED BY 'password'] ...]
    [REQUIRE
  NONE |
        [{SSL| X509}]
        [CIPHER cipher [AND]]
        [ISSUER issuer [AND]]
        [SUBJECT subject]]
    [WITH [GRANT OPTION | MAX_QUERIES_PER_HOUR # |
                          MAX_UPDATES_PER_HOUR # |
                          MAX_CONNECTIONS_PER_HOUR #]]
```

BASIS

Ростелеком

Создание MySQL пользователей и назначение прав

Отнимать привилегии лучше используя, оператор *REVOKE*.

```
REVOKE priv_type [(column_list)] [, priv_type [(column_list)] ...]
    ON {tbl_name | * | *.* | db_name.*}
    FROM user_name [, user_name ...]
```

Пример установки привилегий:

```
mysql#>
GRANT [тип_привилегии] ON [уровень_привилегии] TO [имя_пользователя
] IN IDENTIFIED BY '[пароль]';
mysql#>
GRANT ALL ON *.* TO "newuser@%.firma.lan" IN IDENTIFIED BY 'qwe
rty';
```

BASIS

Ростелеком

Поиск разрешения прав идет следующим образом:

«use» => *«db»* & *«host»* => *«tables_priv»* => *«columns_priv»*
или на языке алгебры логики:
«user» OR («db» AND «host») OR «tables_priv» OR «columns_priv
»

Если это описать более понятным языком, то если, хоть в одной из указанных таблиц существует разрешение на привилегию для пользователя, то пользователь сможет ей воспользоваться.

BASIS

Ростелеком

Сменить пароль можно с помощью оператора SET PASSWORD

```
SET  PASSWORD  =  PASSWORD('some password')
SET  PASSWORD  FOR  user  =  PASSWORD('some password')
```

Первая строчка меняет пароль текущему пользователю, а вторая пользователю с именем *«user»*.

Пример:

```
mysql#> SET  PASSWORD  FOR  'username'@'%.loc.gov'  =
PASSWORD('новый_пароль');
```

BASIS

Ростелеком

Создание резервной копии БД

mysqldump

«mysqldump» - консольный клиент для «бэкапа», создания *«дампов»* БД MySQL. *«Дамп»* помещается в текстовый файл и выглядит как набор операторов MySQL необходимых для нового воссоздания БД.
Синтаксис в *«man mysqldump»*.

Пример запуска *«mysqldump»* со следующими опциями:

```
shell#>
mysqldump  --ignore-table=db.table  -x  -F  --opt  -A  >  /[путь_
куда_делать_дамп]/[имя_файла_дампа].sql
```

BASIS

Ростелеком

Восстановление БД из «дамп» файлов

Восстанавливать информацию из *«дампа»*:

```
shell#> cat  /<путь_до_дамп_файла>/<имя_дамп_файла>  |  mysql
```

Или когда «дамп» сделан для определенной БД:

```
shell#> mysql  db_name  <  db-backup-file.sql
```

«mysqlhothopy» - это скрипт написанный на языке perl для быстрого в режиме on-line резервного копирования БД и таблиц MySQL.

Для просмотра подробной справки по *mysqlhothopy* нужно выполнить команду:

```
shell#> perldoc  mysqlhotcopy.sh
```

BASIS

Ростелеком

Обнаружение ошибок и восстановление БД после сбоя

Процедура обнаружения и исправления ошибок состоит из этапов:

1. Проверка таблиц на наличие ошибок
2. Перед началом исправления создается копия файлов таблиц на случай негативного развития событий
3. Попытка исправления таблицы
4. Если попытка оказывается неудачной, остается лишь восстанавливать базу данных из архива («дампа») и если есть, то из *«журналов обновлений»*

BASIS

Ростелеком

Проверка таблиц на наличие ошибок

Проверять и восстанавливать MyISAM таблицы можно с помощью утилиты *«myisamchk»*, а также можно использовать операторы *CHECK* и *REPAIR*.

Синтаксис *«myisamchk»* можно посмотреть командой:

«myisamchk --help | less» вкратце это выглядит так:

```
shell#>myisamchk  [список_опций]  [имя_таблицы] ...
```

BASIS

Ростелеком

Проверка таблиц на наличие ошибок

Для определения нескольких таблиц каталога:

```
shell#> myisamchk  список_опций_проверки  *.MYI
```

Где *«список_опций_проверки»*:

-c | --check	обычная проверка (по умолчанию)
-e | --extend-check	более тщательная проверка
-m | --medium-check	детальная проверка (самая долгая)

Можно проверить все таблицы во всех базах данных, если задать шаблон вместе с путем к каталогу данных MySQL:

```
shell#> myisamchk  /path/to/datadir/*/*.MYI
```

BASIS

Ростелеком

Исправление таблиц, содержащих ошибки

Для исправления ошибок можно:
восстановление без модификации файла данных (.MYD)

```
shell> myisamchk  - -quick  [имя_таблицы]
```

Если проблема осталась нерешенной то:
может исправить большинство проблем за исключением несовпадения ключей

```
shell> myisamchk  - -recover  [имя_таблицы]
```

Если проблема осталась нерешенной то:
использует старый метод восстановления, медленней чем *«--recover»*, но может исправить некоторые случаи, в которых не помогает опция *«--recover»*

```
shell> myisamchk  - -safe-recover  [имя_таблицы]
```

BASIS

Ростелеком

Восстановление INDEX файла таблицы (*.MYI)

1. Перейти в каталог БД, содержащий файлы поврежденной таблицы.
2. Скопировать файл данных таблицы (*.MYD) в безопасное место.
3. Запустить «mysql» и выполнить следующие команды:

```
mysql#> use  [имя_БД];
mysql#> SET  AUTOCOMMIT=1;
mysql#> TRUNCATE  TABLE  [имя_восстанавливаемой_таблицы];
mysql#> quit;
```

1. Скопировать файл данных таблицы (*.MYD) обратно в каталог БД.
2. Выполнить команду:

```
shell#> myisamchk  -r  -q  [имя_таблицы]
```

1. Затем после восстановления выполнить операторы:

```
mysql#> use  [имя_БД];
mysql#> FLUSH  TABLE [имя_таблицы];
mysql#> quit;
```

Или перезапустить демон "mysqld".

BASIS

Ростелеком

Восстановление файла описания таблицы (*.frm)

Чтобы воссоздать файл описаний таблицы, его можно восстановить из архива (если архив создавался), или заново с помощью оператора «CREATE TABLE».

1. Скопировать файл данных таблицы (**.MYD*) в безопасное место
2. Восстанавливаем файл из архива или заново создать таблицу с помощью оператора *«CREATE TABLE»*
3. Снова запускаем процедуру восстановления *«myisamchk -r -q [имя_таблицы]»*

BASIS

Ростелеком

Работа с блокировками таблиц во время ремонта

Сервер MySQL использует два вида блокировок:

1. внутренняя блокировка
2. внешняя блокировка (на уровне файловой системы)

1-я применяется чтобы избежать взаимного влияния запросов клиентов (пример: не позволяет *«SELECT»* одного клиента выдать неправильные данные из-за одновременной запроса *«UPDATE»* другого клиента).

2-я не позволяет внешним программам изменять файлы таблиц, пока с ними работает сервер «mysqld».

BASIS

Ростелеком

Настройка основных параметров сервера

--bind-address=IP
--port=#
--character-sets-dir=[path]
--chroot=[path]
--datadir=[path]
--log[=file]
--pid-file=[path]
--skip-name-resolve
--skip-networking
--socket=path
--user=[user_name]

--skip-name-resolve

Эту опцию полезно использовать, когда в сети существуют «проблемы» с DNS, при включении этой опции демон «mysqld» не будет преобразовывать IP адреса в их канонические имена.

--skip-networking

Эту опцию полезно включать, если вы решили не предоставлять доступ по сети к базам данных. При включении этой опции соединиться сервером можно будет, только используя *UNIX SOCKET*.

BASIS

Ростелеком

Работа нескольких серверов mysql на ВМ

Используется утилита «mysqld_safe» указав ей соответствующий конфигурационный файл в котором можно/нужно задать основные опции.

/etc/mysqld3306.cnf

```
port            = 3306
socket          = /tmp/mysql.sock
```

/etc/mysqld3307.cnf

```
port            = 3307
socket          = /tmp/mysql3307.sock
```

И запустить «mysqld_safe» со следующими опциями:

```
shell#>
mysqld_safe  --defaults-file=/etc/mysqld3307/mysqld3307.cnf  --dat
adir=/var/db/mysql3307  --user=mysql3307  -ledir=/usr/local/libexe
c  &
```

BASIS

Ростелеком

Советы по повышению безопасности mysql

- Следить за последними обновлениями (заплатками) MySQL
- Ограничить с помощью брандмауэра, доступ по сети к серверу MySQL, разрешив доступ к серверу только с доверенных/нужных хостов
- Удалить из таблицы User «анонимного» пользователя
- Переименовать учетную запись root пользователя MySQL, во что нибудь другое и задать учетной записи root сложный пароль
- Для каждого web приложения требующего MySQL желательно создавать отдельную учетную запись
- Привилегии глобального уровня выдавать пользователям только в случае крайней необходимости
- Не оставлять паролей по умолчанию от root в любом клиенте БД

BASIS

Ростелеком

Советы по повышению безопасности mysql

- Привязать доступ пользователей MySQL к БД только заранее определенных хостов (поле host в таблице User) и исключить использование пользователями пустых паролей
- Запускать демон «mysqld» под системной учетной записью обладающую минимальными правами (под FreeBSD демон «mysqld» по умолчанию запускается с правами пользователя «mysql»)
- Запускать демон «mysqld» с опцией «--chroot» это позволит ограничить доступ к файлам, находящимися выше «chroot» директории для операторов «LOAD DATA INFILE» и «SELECT . INTO OUTFILE»

BASIS

Ростелеком

Советы по повышению безопасности mysql

- Установить для «каталога данных» и «журналов» MySQL разрешения на доступ и просмотр только для пользователя, под которым работает демон «mysqld»
- С большой осторожностью выдавайте пользователям привилегии «File_priv», «Grant_priv» и «Alter_priv»
- Включить опцию «--skip-show-database»

BASIS

Ростелеком

Администрирование PostgreSQL

BASIS

Ростелеком

База данных PostgreSQL

PostgreSQL является одной из наиболее популярных систем управления БД.

Развитие postgresql началось еще в 1986 году. Тогда он назывался POSTGRES.

В 1996 году проект был переименован в PostgreSQL, что отражало больший акцент на SQL.

8 июля 1996 года состоялся первый релиз продукта.

Официальный сайт проекта: https://www.postgresql.org.

PostgreSQL развивается как opensource. Исходный код проекта можно найти в репозитории на гитхабе по адресу https://github.com/postgres/postgres

BASIS

Ростелеком

PostgreSQL. Утилита psql

Для управления сервером баз данных PostgreSQL есть много разных инструментов, но при установке сервера по умолчанию устанавливается только утилита psql. Это консольная утилита, с помощью которой можно подключится к серверу баз данных и начать с ним работать.

```
postgres@frubuntu:/usr/lib/postgresql/9.3/bin$
postgres@frubuntu:/usr/lib/postgresql/9.3/bin$ psql test
psql (9.3.5)
Type "help" for help.

test=# \d
                List of relations
 Schema |       Name        |   Type   |   Owner
--------+-------------------+----------+-----------
 public | Graeber           | table    | geoserver
 public | Graeber_gid_seq   | sequence | geoserver
 public | geography_columns | view     | postgres
 public | geometry_columns  | view     | postgres
 public | raster_columns    | view     | postgres
 public | raster_overviews  | view     | postgres
 public | spatial_ref_sys   | table    | postgres
(7 rows)

test=# select * from Graeber;
ERROR:  relation "graeber" does not exist
LINE 1: select * from Graeber;
                      ^
test=#
```

BASIS

Ростелеком

Подключение к серверу баз данных

Подключение выполняется таким способом:

```
$ psql -d <база> -U <роль> -h <узел> -p <порт>
```

По умолчанию при подключении вы используете:

- В качестве имени базы и роли – имя пользователя ОС;
- В качестве адреса сервера – локальный сокет, который находится в каталоге /tmp/ и порт 5432.

Таким образом если вы в системе находитесь под пользователем postgres, то следующие команды будут равнозначными:

```
$ psql
$ psql -d postgres -U postgres -h /tmp/ -p 5432
```

BASIS

Ростелеком

Получение информации об узле \conninfo

```
postgres@s-pg15:~$ psql
psql (15.0)
Type "help" for help.
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/tmp" at port "5432".
postgres=# \q
postgres@s-pg15:~$ psql -d postgres -U postgres -h /tmp -p 5432
psql (15.0)
Type "help" for help.
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/tmp" at port "5432".
postgres=# \q
```

Ростелеком

PostgreSQL. Утилита psql

Все команды psql начинаются с символа обратного слеша “\”.
Можно выполнять запросы SQL, для них “\” не нужен, например SELECT.
Чтобы выйти из терминала psql можно использовать команду `\q` или `exit`.

Получение справочной информации

Получить справку о psql из ОС:

- `psql --help`
- `man psql` – если postgres был собран с поддержкой man

Получить справку в терминале psql:

- `\?` – список команд psql
- `\? variables` – переменные psql
- `\h` – список команд SQL
- `\h <команда>` – синтаксис определённой команды SQL

BASIS

Ростелеком

Файлы, которые использует psql

.psqlrc

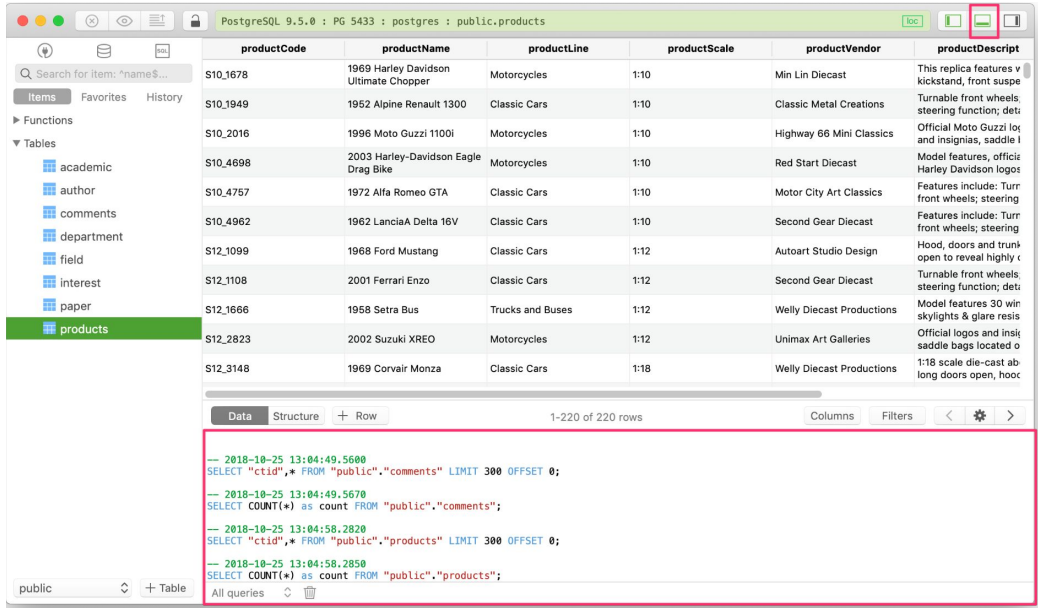
Примеры настроек, которые можно ввести в ~/.psqlrc:

- `\setenv PAGER 'less -XS'` – результат запроса, будет попадать в утилиту less;
- `\timing on` – после запроса показывать время его выполнения;
- `\set PROMPT1 '%n@%/%R%# '` – приглашение ввода команды, когда psql ждет новую команду;
- `\set PROMPT2 '%n@%/%R%# '` – приглашение ввода команды, когда psql ждет дополнительный ввод;
- `\set HISTSIZE 2000` – история команд будет хранить 2000 строк.

Файлы, которые использует psql

.psql_history

Другой полезный файл это ~/.psql_history. В нем хранится история команд введенных в терминале psql. Перемещаться по истории команд в терминале psql можно клавишами вверх и вниз. Количество хранимых команд изменяется установкой переменной HISTSIZE.



BASIS

Ростелеком

Формат выводимой информации

Настроить формат выводимой информации:

- `\a` – с выравниванием/без выравнивания
- `\t` – отображение строки заголовка и итоговой строки/без такого отображения
- `\pset fieldsep ' '` – можно задать разделитель (по умолчанию используется вертикальная черта ‘|’)
- `\x` – расширенный режим, когда нужно вывести много столбцов одной таблицы, они будут выведены в один столбец

BASIS

Ростелеком

Конфигурационный файл postgresql.conf

Главный конфиг файл для кластера PostgreSQL – postgresql.conf

По умолчанию он находится в каталоге PGDATA

Для настройки сервера существует другой файл – postgresql.auto.conf

Он были придуман для настройки сервера из консоли psql

Он читается после postgresql.conf, параметры из него имеют приоритет

Этот файл всегда находится в каталоге с данными (PGDATA)

BASIS

Ростелеком

Информация о текущих настройках сервера

В PostgreSQL есть 2 представления через которые можно посмотреть текущие настройки сервера:

- pg_file_settings – какие параметры записаны в файлах postgresql.conf и postgresql.auto.conf;
- pg_settings – текущие параметры, с которыми работает сервер.

Статистика работы PostgreSQL

Статистика PostgreSQL включается в файле postgresql.conf:

- track_counts – обращения к таблицам и индексам
- track_io_timing – статистика операций ввода/вывода
- track_functions – статистика вызовов функций и времени их выполнения. По умолчанию выключен. Значения:
  - pl – включает отслеживание функций только на процедурном языке
  - all – включает отслеживание функций на всех языках, например, SQL и C

Статистика работы PostgreSQL

Каждый backend процесс собирает статистику в процессе своей работы

Раз в полсекунды, статистика сбрасывается в каталог $PGDATA/pg_stat_tmp

При остановке сервера PostgreSQL, статистика сбрасывается в $PGDATA/pg_stat

Статистика ведется с момента первого запуска сервера, а с помощью функции pg_stat_reset() её можно сбросить

На уровне всего кластера обнулить счетчики можно с помощью функции pg_stat_reset_shared (). Аргумент может принимать значения bgwriter и archiver, с которыми обнуляются все счётчики в представлении pg_stat_bgwriter или pg_stat_archiver

BASIS

Ростелеком

Статистика работы PostgreSQL

Статистику можно смотреть в следующих представлениях:

pg_stat_all_tables	в разрезе строк и страниц для БД
pg_statio_all_tables	в разрезе 8 KB страниц для БД
pg_stat_all_indexes	по индексам для БД
pg_statio_all_indexes	по индексам для БД в разрезе страниц
pg_stat_database	глобальная статистика по БД
pg_stat_bgwriter	статистика для анализа фоновой записи

BASIS

Ростелеком

Утилита pgbench

В PostgreSQL есть специальная утилита pgbench. С помощью, которой можно произвести нагрузочное тестирование (НТ).

`pgbench -i <база данных>`

создание таблиц pgbench_accounts, pgbench_branches, pgbench_history и pgbench_tellers.

Запустить нагрузочное тестирование на 10 секунд

`pgbench -T 10 <имя базы данных>`.

BASIS

Ростелеком

Текущие активности в PostgreSQL

Инструменты текущей активности:

- Посмотреть на текущие активности сервера PostgreSQL с помощью представления pg_stat_activity
- Чтобы завершить один из обслуживающих процессов нужно использовать функцию pg_terminate_backend(<pid>)
- С помощью функции pg_blocking_pids(<pid>), можно посмотреть кого ожидает процесс с этим pid

Все эти действия можно выполнить с помощью инструментов командной строки ОС:

- Посмотреть процессы с помощью команды ps
- Завершить процесс с помощью команды kill -9 <pid>

BASIS

Ростелеком

Журнал PostgreSQL. Настройка и анализ

В журнал PostgreSQL записывает некоторые из своих действий

Настраивая журналирование мы можем задать:

- Какие действия заносить в журнал
- Насколько подробно описывать эти действия
- Сколько будут хранится файлы журнала и как переключаться на другие файлы

BASIS

Ростелеком

Журнал PostgreSQL. Настройка и анализ

Опции настройки журнала:

- log_destination = можем указать один, или через запятую несколько приёмников:
  - stderr – поток ошибок
  - csvlog – формат CSV
  - syslog – писать ошибки в syslog
  - eventlog – писать ошибки журнал событий Windows
- logging_collector = (on или off). Можно вести запись в stderr или csvlog
- log_directory и log_filename – каталог и файл журнала. Следует указывать только если log_destination = stderr

Что можем записывать в журнал?

- log_min_messages – минимальный уровень логирования. Допустимые значения: DEBUG5 – DEBUG1, INFO, NOTICE, WARNINF, ERROR, LOG, FATAL, PANIC. По умолчанию используется WARNINF
- log_min_duration_statement – время в миллисекундах. Если установить равное нулю, то абсолютно все команды будут записаны в журнал
- log_duration – (on или off) записывать время выполнения команд
- application_name – (on или off) записывать имя приложения
- log_checkpoints – (on или off) записывать информацию по контрольным точкам

BASIS

Ростелеком

Что можем записывать в журнал?

- log_(dis)connections – (on или off) записывать подключения к серверу и отключения от него;
- log_lock_waits – (on или off) записывать, если сеанс ожидает блокировку дольше, чем указано в deadlock_timeout;
- log_statement – (none, ddl, mod, all) записывать текст выполняемых команд:
  - none – отключено
  - ddl – CREATE, ALTER, DROP
  - mod – dll + INSERT, UPDATE, DELETE, TRUNCATE, COPY
  - all – все команды (кроме команд с синтаксическими ошибками)
- log_temp_files – использование временных файлов. Находится в зависимости с параметром workmem.

BASIS

Ростелеком

Ротация журналов

Настроить ротацию, если мы используем log_destination=stderr:

- log_filename – может принять не просто имя файла, а маску имени
- log_rotation_age – задает время переключения на следующий файл в минутах
- log_rotation_size – задает размер файла, при котором нужно переключиться на следующий файл
- log_truncate_on_rotation – если включить (on) то вы разрешите серверу перезаписывать уже существующие файлы. Если выключить (off) – то файл не будет перезаписываться, записи будут писаться в конец файла

Например:

- log_filename = postgres-%H.log / log_rotation_age = 1h – 24 файла в сутки
- log_filename = postgres-%a.log / log_rotation_age = 1d – 7 файлов в неделю

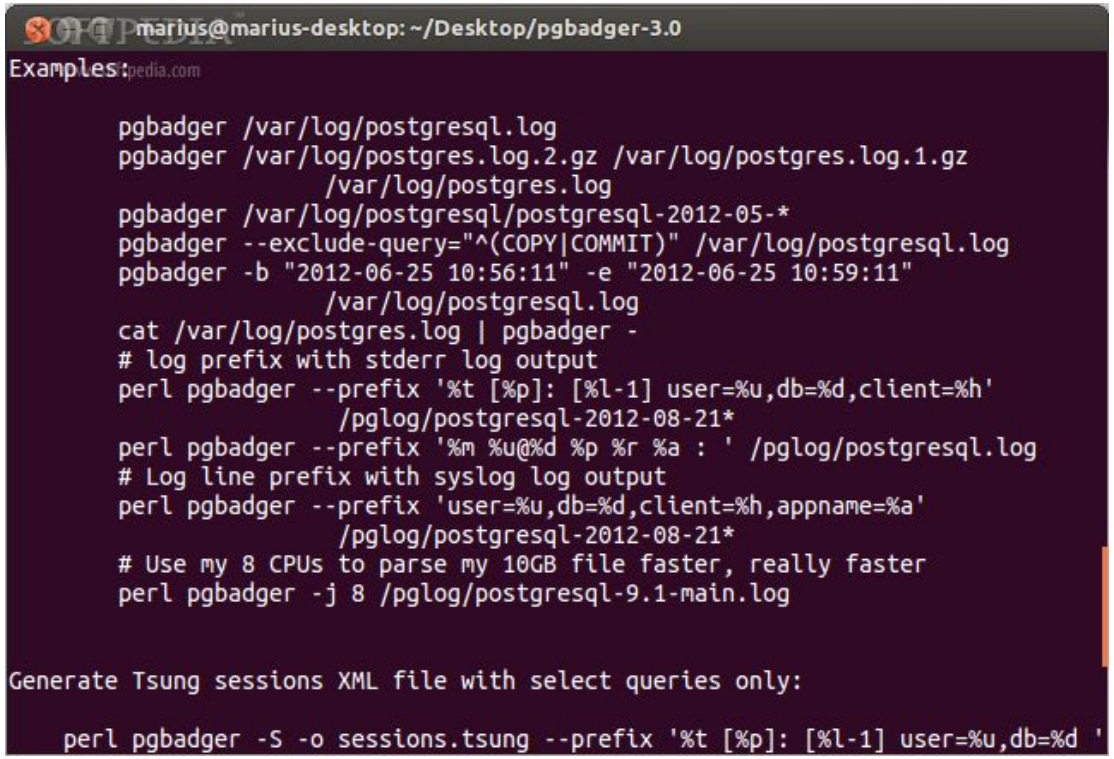
BASIS

Ростелеком

Анализ журнала

Анализировать журнал можно средствами ОС, например: grep, awk и подобными.

А также можно использовать pgBadger – это анализатор лога PostgreSQL, но он требует определённых настроек журнала.



BASIS

Ростелеком

Роли и атрибуты в PostgreSQL

В PostgreSQL пользователи и группы – это роли.

Псевдороль public неявно включает в себя все остальные роли.

Атрибуты ролей:

- LOGIN / NOLOGIN – возможность подключения;
- SUPERUSER / NOSUPERUSER – суперпользователь;
- CREATEDB / NOCREATEDB – возможность создавать базы данных;
- CREATEROLE / NOCREATEROLE – возможность создавать роли;
- REPLICATION / NOREPLICATION– использование протокола репликации.

Управление ролями в PostgreSQL

Создают роль следующим способом:

```
CREATE ROLE <роль> [WITH] <атрибуты через запятую>;
```

Если при создании роли не указать атрибуты, то роль получит запрещающие атрибуты (NOLOGIN, NOSUPERUSER) автоматом.

Для включения одной роли в другую - GRANT:

```
GRANT <групповая роль> TO <роль>;
```

А чтобы исключить роль из группы:

```
REVOKE <групповая роль> FROM <роль>;
```

BASIS

Ростелеком

Управление ролями в PostgreSQL

Право включать роли в другие роли могут:

- Роль может включить в саму себя любую другую роль
- SUPERUSER – может включать любую роль в другую любую роль
- CREATEROLE – может включать любую роль в любую групповую роль, кроме суперпользовательской

```
GRANT <групповая роль> TO <роль> WITH ADMIN OPTION;
```

```
REVOKE ADMIN OPTION FOR <групповая роль> FROM <роль>;
```

BASIS

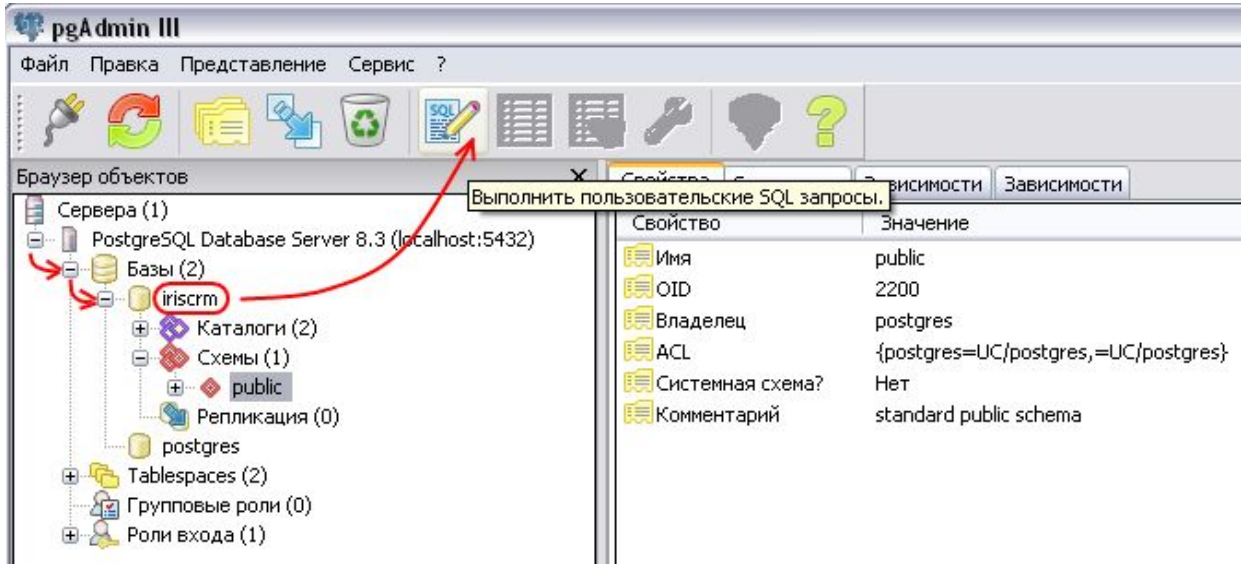
Ростелеком

Управление ролями в PostgreSQL

Владелец объекта – это роль, которая этот объект создала, а также роли включённые в неё.

Владельца можно переназначить с помощью ALTER:

```
ALTER [TABLE, VIEW] <название объекта> OWNER TO <роль>;
```



BASIS

Ростелеком

Процесс подключения

- Идентификация – определение имени роли БД.
- Аутентификация – проверка того, что пользователь тот за кого себя выдаёт.
- Авторизация – проверка прав этого пользователя.

BASIS

Ростелеком

Основные настройки аутентификации

Конфигурационный файл отвечающий за настройки аутентификации – pg_hba.conf находится в каталоге PGDATA.

Файл pg_hba.conf состоит из строк, а строки состоят из полей:

- тип подключения
- имя БД
- имя пользователя
- адрес узла
- метод аутентификации
- необязательные дополнительные параметры в виде *имя=значение*

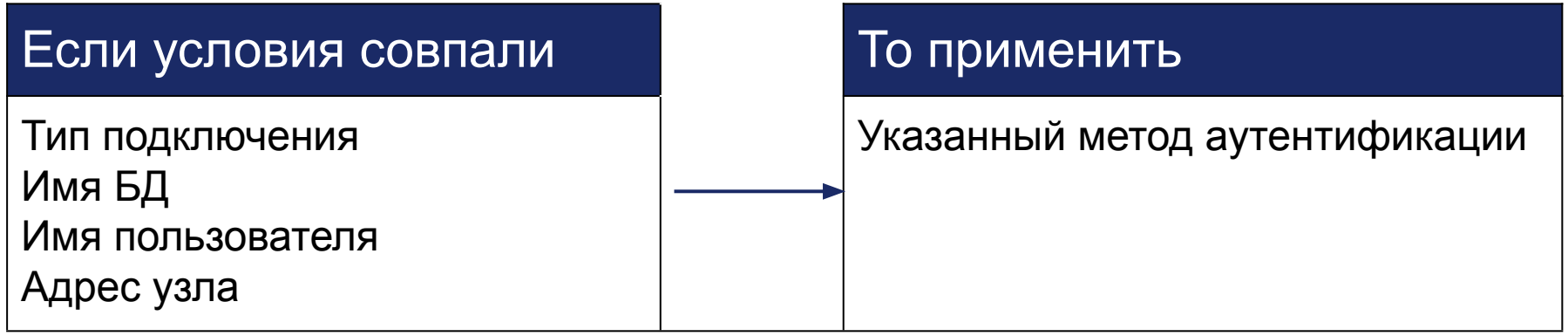
BASIS

Ростелеком

Основные настройки аутентификации

Если тип подключения, имя БД, имя пользователя и адрес сервера совпали, то применяется определённый метод аутентификации

При подключении выполняется аутентификация и проверяется привилегия CONNECT



pg_hba – если-то

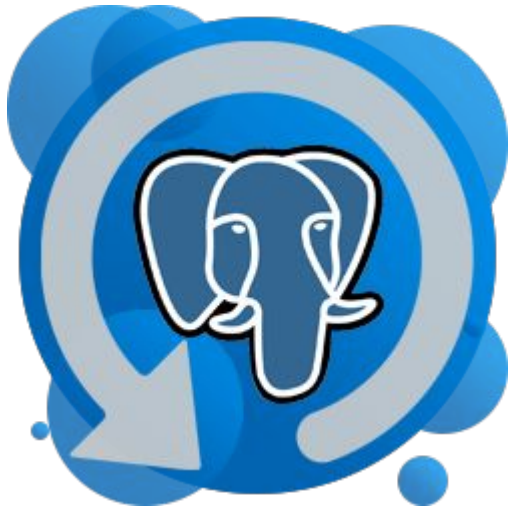
BASIS

Ростелеком

Резервирование PostgreSQL

Существует логическое и физическое резервирование PostgreSQL. Первый тип сохраняет SQL команды, выполнив которые можно восстановить объекты, например, создать БД, наполнить её таблицами, заполнить таблицы данными и т.д.

Второй тип резервирует сами данные, то есть сохраняет каталог PGDATA.



BASIS

Ростелеком

Логическое резервирование PostgreSQL

Есть 3 инструмента для логического копирования:

- COPY – команда SQL для копирования данных из таблицы в файл или наоборот из файла в таблицу.
- pg_dump – утилита postgresql для копирования всей БД. Она использует команду COPY для выгрузки данных. Можно создать дамп в другом формате, тогда при загрузке нужно использовать pg_restore. После восстановления надо выполнить сбор статистики, так как pg_dump статистику не выгружает.
- pg_dumpall – утилита postgresql для копирования всего кластера. Выгружает только в текстовом формате.

Физическое резервирование PostgreSQL

Физическое резервное копирование разделяется на:

- Холодное резервирование (при выключенном сервере) – после корректного выключения можно перенести данные на другой сервер.
- Горячее резервирование (при включенном сервере) – делается спец средствами, при этом требуются все файлы предварительной записи с момента начала копирования и до его окончания.

Для горячего резервирования используется утилита pg_basebackup

BASIS

Ростелеком

Протокол репликации

Протокол репликации – специальный протокол, который позволяет:

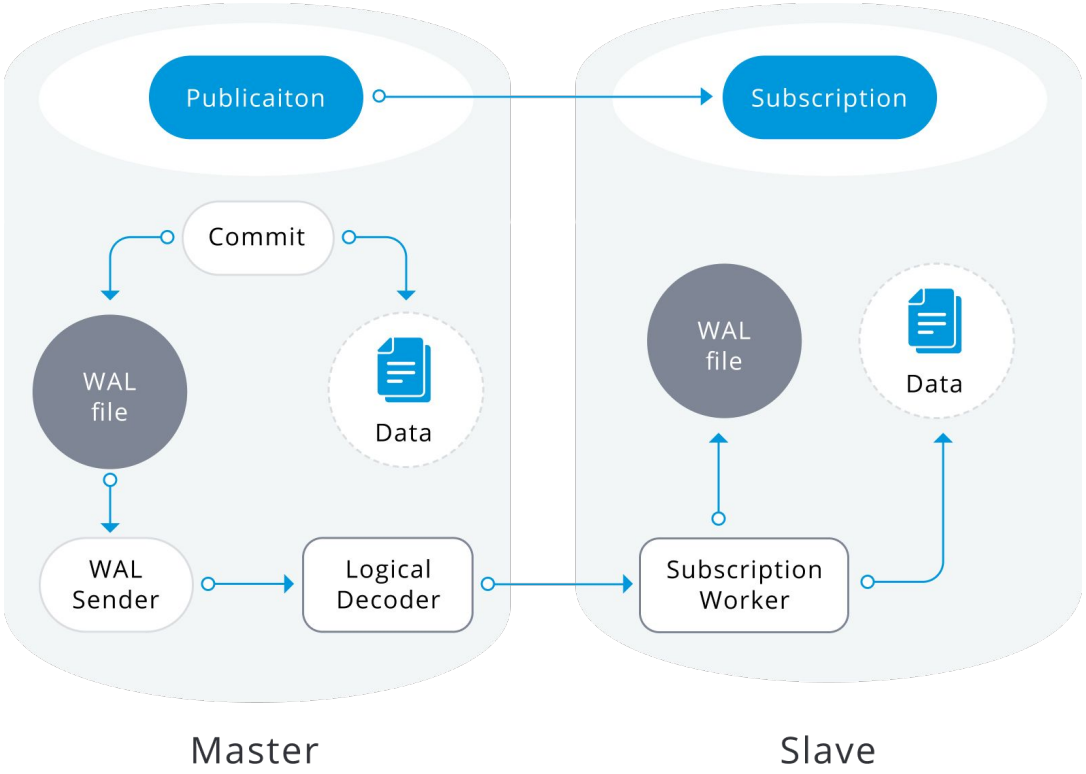
- Получать поток журнальных записей
- Выполнять команды управления резервным копированием и репликацией

Когда мы подключаемся по протоколу репликации нас начинает обслуживать процесс wal_sender

Чтобы мы могли работать по протоколу репликации нужно выставить параметр сервера: wal_level=replica

Протокол репликации

Слот репликации – механизм для резервирования wal файлов. Подключившись по протоколу репликации, мы создаём слот репликации и через этот слот передаются wal файлы.



BASIS

Ростелеком

Архив журналов. Файловый архив.

- Сегменты WAL копируются в архив по мере заполнения;
- Механизм работает под управлением сервера
- Неизбежны задержки попадания данных в архив

Чтобы запустить файловый архив нужно запустить процесс archiver

Для этого нужно настроить 3 параметра:

- archive_mode = on
- archive_command – команда shell для копирования сегмента WAL в отдельное хранилище (или скрипт);
- archive_timeout – максимальное время для переключения на новый сегмент WAL.

При заполнении сегмента WAL вызывается команда archive_command

Архив журналов. Потоковый архив.

- В архив постоянно записывается поток журнальных записей
- Требуются внешние средства
- Задержки минимальны

Для потокового архива используется утилита pg_receivewal.

Стартовая позиция – начало текущего сегмента. В отличии от файлового архивирования записи wal передаются постоянно.

При восстановлении БД, когда есть данные на определённый момент времени и архив wal файлов. Нужно создать файл $PGDATA/recovery.conf в котором указать, откуда брать wal файлы, и включить сервер.

BASIS

Ростелеком

Репликация в PostgreSQL

Репликация в PostgreSQL – это процесс синхронизации нескольких копий кластера БД на разных серверах. Она бывает логической и физической.

Задачи и виды репликации

Репликация PostgreSQL решает две задачи:

- Отказоустойчивость – если сломается один из серверов, клиенты могут продолжить работать на резервном;
- Масштабируемость – резервный сервер принимает запросы на чтение, так что некоторую нагрузку можно возложить на него.

Репликация в PostgreSQL

Физическая – основной сервер передает поток wal записей на сервер репликации. Требования:

- Одинаковые версии postgresql
- Одинаковые ОС
- Возможна репликация только всего кластера

Логическая – поставщик публикует свои изменения, а подписчик получает и применяет эти изменения у себя. Особенности:

- Оба сервера могут быть и поставщиком и подписчиком, но на разные объекты
- Репликация возможна между разными ОС
- Возможна выборочная репликация отдельных объектов кластера.

BASIS

Ростелеком

Физическая репликация PostgreSQL

Алгоритм создания репликации:

- Делаем резервную копию с помощью pg_basebackup
- Разворачиваем полученную резервную копию на сервере репликации
- Создаем специальный файл с настройками репликации:
  - В 10 версии PostgreSQL создаём файл recovery.conf, прописываем там standby_mode = on
  - Начиная с 12 версии создаём пустой файл standby.signal
- Стартуем PostgreSQL на реплике, после чего начнётся процесс репликации. Сервер начнёт процесс восстановления из потока wal записей.

Сценарии использования физической репликации

- Обычная репликация – для создания резервного сервера
- Каскадная репликация – к основному серверу подключаем реплику, а к этой реплики еще одну реплику
- Отложенная репликация – в recovery.Conf специальным параметром можно указать задержку воспроизведения. Чтобы реплика всегда отставала от основного сервера, например, на час

Логическая репликация PostgreSQL

При репликации передаются wal записи, но для работы логической репликации нужно изменить формат этих записей. Для этого нужно поменять параметр кластера wal_level = logical.

Поставщик – передаёт логические wal записи. Но передаются не все команды, а
только INSERT, UPDATE, DELETE и TRANSCATE.

Подписчик – получает wal записи и применяет изменения без разбора, трансформаций и планирования.

На поставщике работает процесс wal sender, а на подписчике logical replication worker, который получает логические wal записи и применяет их от имени суперпользователя.

BASIS

Ростелеком

Сценарии использования логической репликации

- Собираем данные на центральном кластере.
- Распространяем данные с центрального кластера.
- Можно использовать логическую репликация для обновления кластера. Затем поменять местами поставщика и подписчика. И наконец выключить, и обновить основной сервер.
- Мультимастер – кластер в котором данные могут менять несколько серверов.

BASIS

Ростелеком

Troubleshooting

Что такое Troubleshooting

Устранение неполадок сбоев базы данных и проблем с подключением - Troubleshooting

BASIS

Ростелеком

Проблемы с подключением к БД

- Подсказки из журналов приложений
- Успешно ли сервер приложений обрабатывает подключения?
- Запросы сервера приложений к базе данных

Например:

{"level":30,"time":1617808854673,"pid":96741,"hostname":"do-server-1","msg":"Server listening at http://0.0.0.0:8000"}

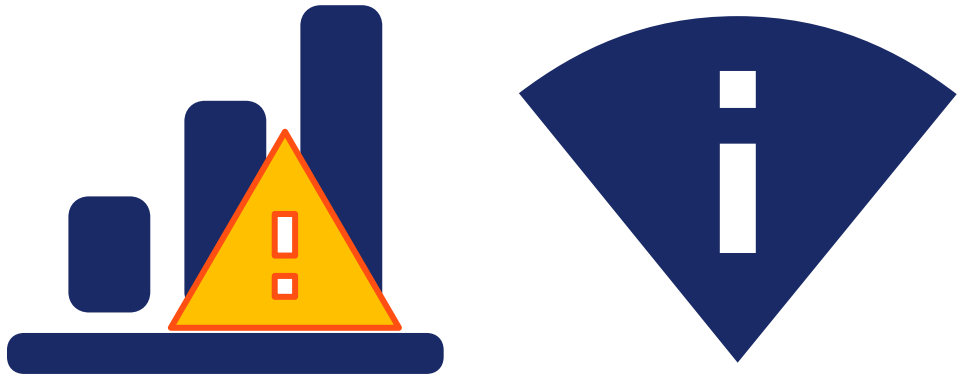
BASIS

Ростелеком

Проблемы с сетью

К числу вопросов, связанных с сетевым взаимодействием, относятся:

- Проблемы с политикой VPC и брандмауэра
- Задержка и тайм-ауты между приложением и БД



VPC

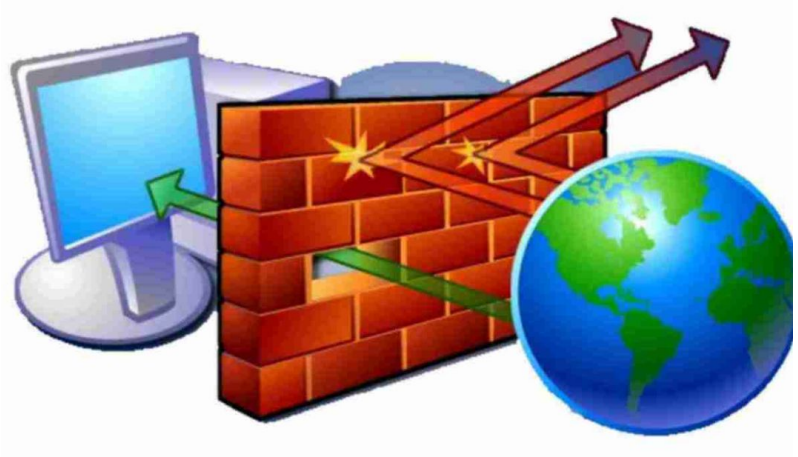
При выделении облачных ресурсов, таких как базы данных на облачных платформах, они изолированы в виртуальном частном облаке (VPC). На практике VPC служит частной сетью для ресурсов приложения и изолирован от общедоступного Интернета

BASIS

Ростелеком

Средства защиты правил брандмауэра

Лучше развертывать приложение и базу данных в одном и том же VPC и одном регионе, чтобы они взаимодействовали по частной сети. Это также предотвращает узкие места, которые могут возникнуть в общедоступных сетях.

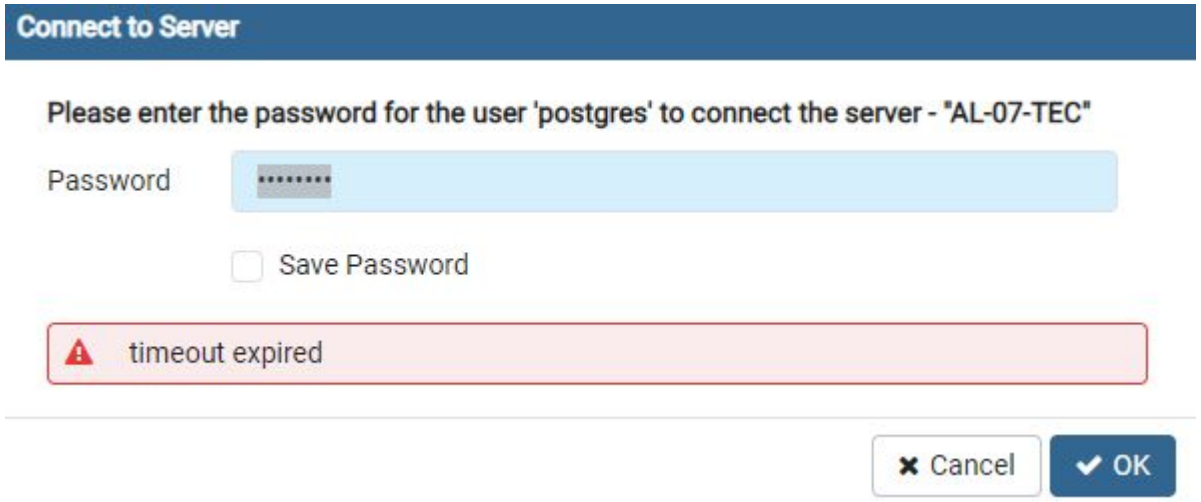


BASIS

Ростелеком

Лимит исчерпанного соединения(timeout limit)

Еще одна распространенная проблема с БД на основе подключений, такими как MySQL и PostgreSQL, заключается в том, что вы можете быстро исчерпать лимит подключения БД. БД, ориентированные на подключения, накладывают ограничение на количество открытых подключений к БД.



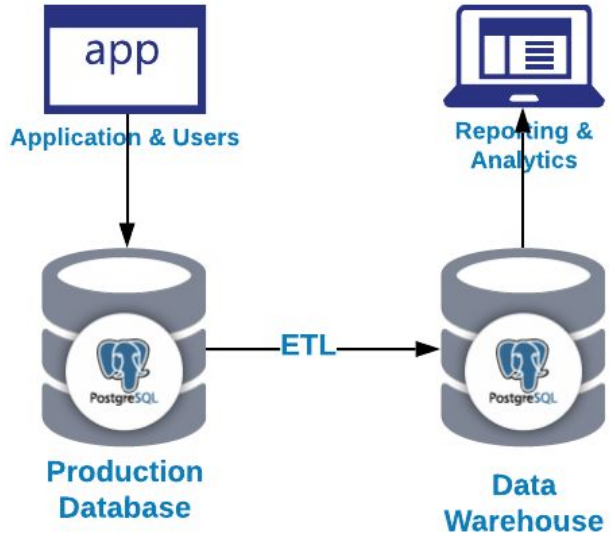
BASIS

Ростелеком

Проблемы с объемом данных

По мере роста приложения объем данных для этого приложения, скорее всего, также будет расти. Ключевым фактором как для производительности, так и для времени безотказной работы базы данных является объем данных, обрабатываемых для удовлетворения заданного запроса.

- Сервер базы данных
- Сервер приложений
- Клиент
- Сервер базы данных.

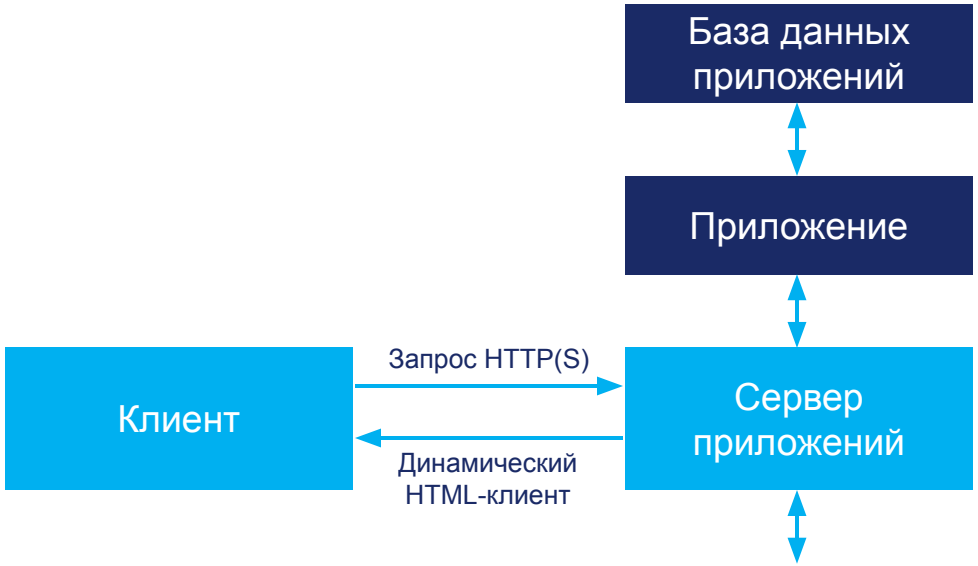


BASIS

Ростелеком

Сервер приложений

Подобно тому, как сервер БД не имеет неограниченной емкости для обработки больших объемов данных, то же самое верно и для сервера приложений.



Популярные серверы приложений:

- APPACHE
- APPACHE TOMCAT
- ORANGE WEBLOGIC
- GLASSFISH
- JBOSS

- Транзакции
- Безопасность
- Внедрение зависимости DI
- Одновременность (Concurrency)

BASIS

Ростелеком

Клиент

Клиентские приложения могут быть наиболее подвержены узким местам, вызванным большими объемами данных. В отличие от сервера приложений и сервера баз данных, где у вас может быть возможность увеличить емкость, клиентские приложения, которые выполняются в браузере или на мобильных устройствах, подвержены ограничениям браузера, операционной системы или того и другого.

BASIS

Ростелеком

Средства защиты от размера данных

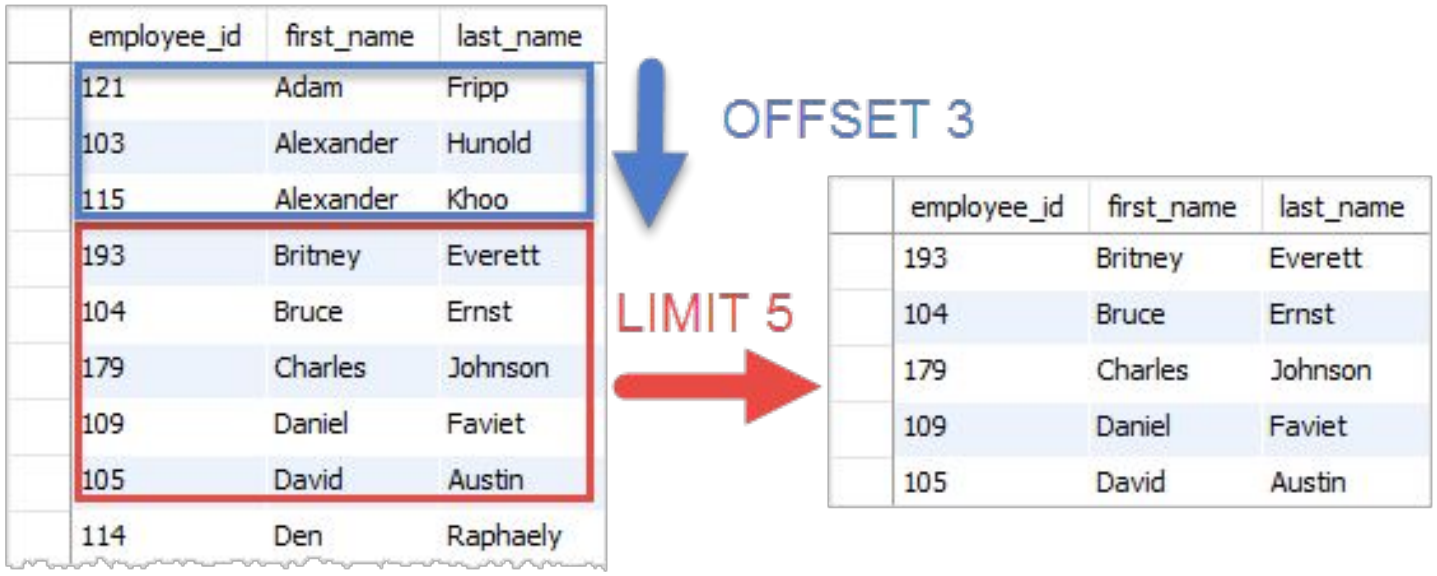
Исправление снижения производительности и простоев, вызванных проблемами с объемом данных, почти всегда заключается в ограничении объема данных, возвращаемых сервером БД. Это облегчит проблемы на сервере БД, сервере приложений и клиенте.

BASIS

Ростелеком

Разбиение данных на страницы LIMIT/OFFSET

Разбиение на страницы — это шаблон проектирования, который ограничивает общее количество записей, запрашиваемых и возвращаемых в данный момент времени.



	employee_id	first_name	last_name
	121	Adam	Fripp
	103	Alexander	Hunold
	115	Alexander	Khoo
	193	Britney	Everett
	104	Bruce	Ernst
	179	Charles	Johnson
	109	Daniel	Faviet
	105	David	Austin
	114	Den	Raphaely

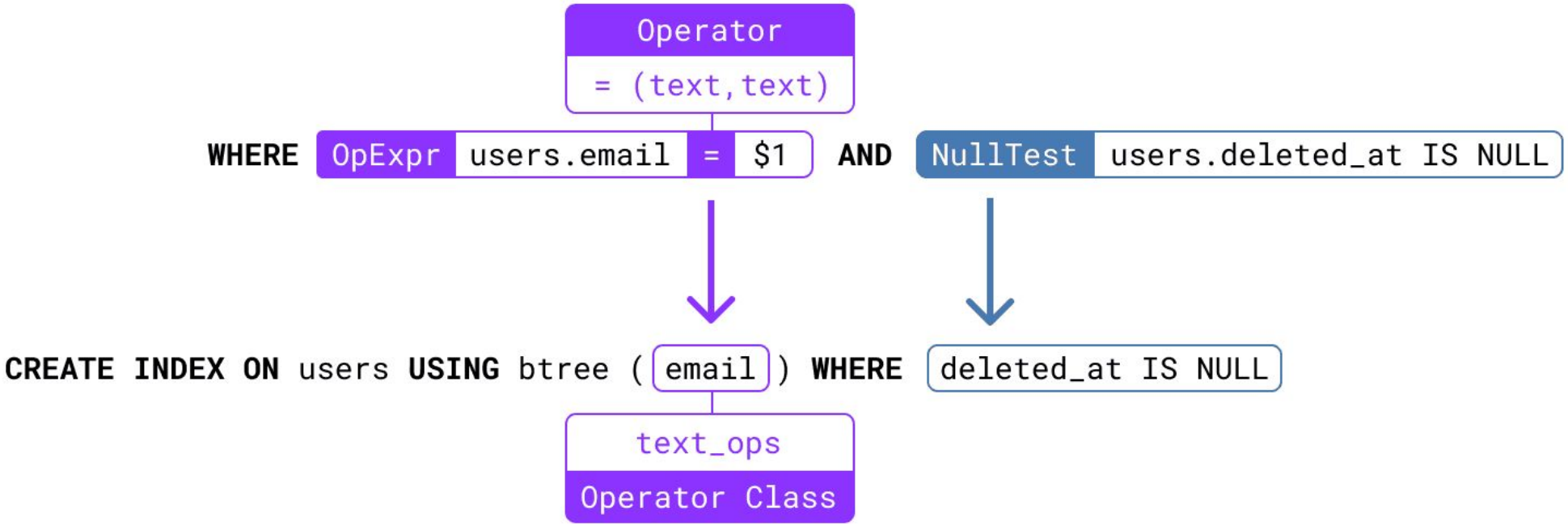
	employee_id	first_name	last_name
	193	Britney	Everett
	104	Bruce	Ernst
	179	Charles	Johnson
	109	Daniel	Faviet
	105	David	Austin

BASIS

Ростелеком

Добавление индексов

Проблемы, связанные с большими объемами данных, часто можно устранить с помощью индексов.



BASIS

Ростелеком

Взлом изменений кода

Предполагаемый сбой базы данных может быть прослежен до недавних критических изменений, внесенных в клиентский или серверный код. В этих случаях, скорее всего, не сама база данных испытывает сбой, а код, используемый для извлечения данных, их обработки и возврата клиенту, может сломаться.

BASIS

Ростелеком

Рекомендуемая литература

BASIS

Ростелеком

Полезные источники

- Using Postgres CREATE INDEX: Understanding operator classes, index types & more
- 10 способов сделать резервную копию в PostgreSQL
- 11 типов современных баз данных: краткие описания, схемы и примеры БД
- Основы администрирования СУБД MySQL
- Администрирование баз данных

BASIS

Ростелеком

Спасибо за внимание!

BASIS

Ростелеком