ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное образовательное учреждение города Москвы

«Московский государственный колледж электромеханики и информационных технологий»

**Практическая работа №1. Работа с официальной документации различных СУБД**

[УП.11 Разработка, администрирование и защита баз данных](https://classroom.google.com/c/NDI2NDY2OTEwNTAw)

Выполнил студент группы: 3ИП-11-19

Терехин Михаил Александрович

Преподаватель: Басыров Сергей Амирович

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc88051261)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc88051262)

[1 Анализ назначения СУБД. 4](#_Toc88051263)

[2 Анализ основных возможностей СУБД. 4](#_Toc88051264)

[3 Анализ языка запроса в СУБД. 5](#_Toc88051265)

[4 Анализ типов данных в СУБД. 6](#_Toc88051266)

[5 Анализ синтаксиса SQL в СУБД. 7](#_Toc88051267)

[ЗАКЛЮЧЕНИЕ 10](#_Toc88051268)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 11](#_Toc88051269)

# ВВЕДЕНИЕ

Цель – научиться работать с официальной документацией различных производителей.

Актуальность – заключается в разнообразие СУБД.

Задачи:

1) анализ назначения СУБД;

2) анализ основных возможностей СУБД;

3) анализ языка запроса в СУБД;

4) анализ типов данных в СУБД;

5) анализ синтаксиса SQL в СУБД.

Предмет исследования – СУБД.

Объект исследования – СУБД: Redis, MySQL, MongoDB.

# ОСНОВНАЯ ЧАСТЬ

# 1 Анализ назначения СУБД.

Redis (Remote Dictionary Server) — система управления базами данных, основанных не на SQL (Structured Query Language). Redis - это хранилище структур данных в памяти с открытым исходным кодом (под лицензией BSD), используемое в качестве базы данных, кеша и брокера сообщений. Redis предоставляет такие структуры данных, как строки, хэши, списки, наборы, отсортированные наборы с запросами диапазона, растровые изображения, гиперлоггеры, геопространственные индексы и потоки. Redis имеет встроенную репликацию, сценарии Lua, удаление LRU, транзакции и различные уровни сохраняемости на диске, а также обеспечивает высокую доступность с помощью Redis Sentinel и автоматическое разбиение с помощью Redis Cluster.

MySQL — свободная реляционная система управления базами данных. Продукт распространяется как под [GNU General Public License](https://ru.wikipedia.org/wiki/GNU_General_Public_License), так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм [репликации](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)).

MongoDB — документоориентированная система управления базами данных, не требующая описания схемы таблиц. Система поддерживает [ad-hoc](https://ru.wikipedia.org/wiki/Ad_hoc" \o "Ad hoc)-запросы: они могут возвращать конкретные поля документов и пользовательские [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript)-функции. Поддерживается поиск по регулярным выражениям. Также можно настроить запрос на возвращение случайного набора результатов

# 2 Анализ основных возможностей СУБД.

Redis —  программа работает с кэшем данных, а не со значениями напрямую. Это повышает производительность при работе с базами данных (БД) в несколько раз. Информация, с которой работает Redis, хранится в виде «словаря». Каждому ключу соответствует единственное значение.

MySQL — обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

MongoDB —  самая продвинутая служба облачных баз данных на рынке с непревзойденным распределением данных и мобильностью в AWS, Azure и Google Cloud, встроенной автоматизацией для оптимизации ресурсов и рабочих нагрузок и многим другим.

# 3 Анализ языка запроса в СУБД.

Redis позволяет писать такой же сложный код с меньшим количеством простых строк. Redis позволяет писать меньше строк для хранения, использования данных и организации доступа к данным в приложениях. Разница в том, что, в отличие от языков запросов традиционных баз данных, с Redis разработчики могут использовать простую структуру команд. Разработчикам под Redis доступны более ста клиентов с открытым исходным кодом. Поддерживаемые языки программирования включают Java, Python, PHP, C, C++, C#, JavaScript, Node.js, Ruby, R, Go и многие другие.

MySQL имеет API и коннекторы для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль, Tcl и Lua, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC. MyODBC представляет собой драйвер ODBC (2.50) уровня 0 (с некоторыми возможностями уровней 1 и 2) для подсоединения совместимого с ODBC приложения к MySQL. MyODBC работает на всех системах Microsoft Windows и на большинстве платформ Unix.

MongoDB поддерживает динамические запросы к документам, используя основанный на документе язык запросов, почти такой же мощный, как SQL.

# 4 Анализ типов данных в СУБД.

Redis поддерживает пять типов данных: String (строка), Hash (хэш), список (список), комплект (набор) и zset (отсортированный набор: упорядоченный набор).

Redis string - это самый простой тип, вы можете понять тот же тип, что и с Memcached, ключ, соответствующий значению. Redis хэш является типом строка поля и значения таблицы отображения, хэш особенно подходит для хранения объектов. Список Redis представляет собой простой список строк, отсортированных порядок добавления. Вы можете добавить элемент в начало списка (слева) или сзади (справа). Redis представляет собой неупорядоченный набор Set типа строки. Redis zset задан как набор элементов типа string и не допускает дублирования элементов.

MySQL поддерживает несколько типов столбцов, которые можно разделить на три категории: числовые типы данных (TINYINT, BOOL, TINYINT UNSIGNED…), типы данных для хранения даты и времени (DATE, TIME, DATETIME…) и символьные (строковые) типы данных (CHAR, VARCHAR, TINYTEXT).

В отличие от реляционных баз данных MongoDB не использует табличное устройство с четко заданным количеством столбцов и типов данных. MongoDB является документо-ориентированной системой, в которой центральным понятием является документ. Наиболее используемые типы данных:

Integer Используется для хранения целочисленных значений. В зависимости от сервера может быть как 32-битным, так и 64-битным. Double Используется для хранения значений с плавающей точкой Boolean Используется для хранения логических (true / false) значений. String Используется для хранения символьных строк. В MongoDB используется кодировка UTF-8. Arrays Данный тип данных используется для хранения массивов значений по одному ключу. Object Используется для встроенных документов. Symbol Используется также, как и String, но, обычно, резервируется для языков, которые используют специальные символы. Null Используется для хранения значения Null. Timestamp Используется для хранения даты и времени Min / Max Используется для сравнения значений с наибольшим и наименьшим BSON (Binary JSON) элементом. Object ID Используется для хранения ID документа. Regular Expression Используется для хранения  регулярных выражений. Code Используется для хранения JavaScript кода в документе. Binary data Данный тип данных позволяет хранить бинарные данные. Date Используется для хранения текущей даты или времени в UNIX формате.

# 5 Анализ синтаксиса SQL в СУБД.

Redis

Получение данных из хранилища (Таблица 1).

Таблица 1 – Исходный код получение данных из хранилища.

|  |
| --- |
| public function getValueFromCache(string $key)  {      return $this->getRedis()->rawCommand('GET', $key);  } |

Удаление данных по массиву ключей (Таблица 2).

Таблица 2 – Исходный код удаление данных.

|  |
| --- |
| public function dropValuesFromCache(array $keys)  {      $this->getRedis()->rawCommand('MDEL', ...$keys);  } |

MySQL

Выведет количество всех записей в таблице (Таблица 3).

Таблица 3 – Исходный код вывод всех записей.

|  |
| --- |
| SELECT count(\*) FROM table\_name; |

Выведет все записи таблицы person в которых значения поля age будет равно 12 или 15 или 18 (Таблица 4).

Таблица 4 – Исходный код вывода некоторых записей.

|  |
| --- |
| SELECT \* from person where age in (12,15,18); |

Выводит все уникальные фамилии людей (last\_name), которые живут в городе с названием на букву L. (предполагаем, что в таблице address есть поля address\_no, city), см. Таблица 5.

Таблица 5 – Исходный код вывода некоторых записей (сложный запрос).

|  |
| --- |
| SELECT DISTINCT last\_name FROM person p, address adr WHERE p.adress\_no = adr.address\_no AND city LIKE 'L%'; |

MongoDB

Для вывода документов в более удобном наглядном представлении мы можем добавить вызов метода pretty(), смотреть Рисунок 1 – Вывода документов



Рисунок 1 – Вывода документов

MongoDB предоставляет замечательную возможность, создавать запросы, используя язык JavaScript. Например, создадим запрос, возвращающий те документы, в которых name=Tom. Для этого сначала объявляется функция, смотреть Рисунок 2 – Возвращение документов.



Рисунок 2 – Возвращение документов.

Еще одной замечательной возможностью при построении запросов является использование регулярных выражений. Например, найдем все документы, в которых значение ключа name начинается с буквы B, смотреть Рисунок 3 – Регулярные выражения.

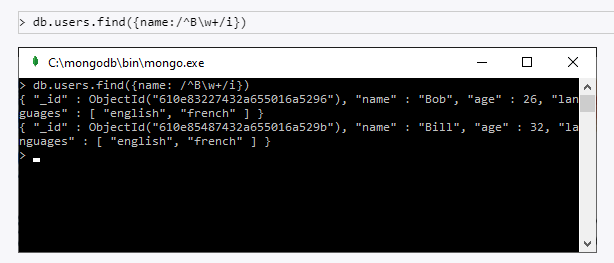


Рисунок 3 – Регулярные выражения.

# ЗАКЛЮЧЕНИЕ

Если потребуется много запросов , то лучше воспользоваться Redis, где данные могут храниться в разнообразных специализированных структурах данных, настроенных для каждого типа объекта для эффективности. В MongoDB одни и те же запросы могут быть проще, потому что структура более согласована между данными. С другой стороны, в Redis **максимальная скорость** ответа на эти запросы - это выигрыш для дополнительной работы с различными структурами, с которыми могут храниться ваши данные. MongoDB предлагает простоту, гораздо более короткую кривую обучения для разработчиков с традиционным опытом работы с базами данных и SQL. Однако нетрадиционный подход Redis требует больше усилий для изучения, но большей гибкости. MySQL гораздо более гибок при запросе различных типов результатов в разных сценах.Redis IS быстрее, чем MySQL. Это простая система ключ-значение, поэтому она превысит производительность MySQL.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1) https://itglobal.com/ru-ru/company/glossary/redis/

2) https://www.mysql.com/why-mysql/

3) https://www.mongodb.com/atlas/database

4) <https://ru.wikipedia.org/wiki/MySQL>

5) <https://aws.amazon.com/ru/redis/>

6) http://www.w3big.com/redis/redis-data-types.html