**Полно математических функций см сайт, пример:**

**(/ \* + - % )** – стандарт

**Div** – целочисленное деление

**Mod** – остаток от целочисленного деления

**Abs(…)** – модуль

**Rand()** – рандомчик

**Sleep(…)** – пауза

**Greatest(…, …)** – наибольшее

**Least(…, …)** – наименьшее

**Concat(… , …)** – конкатенация строк

**Cast(… as …)** – преобразования типов

**where** – уточнение ( =, >, <, <>) (<> - не равно, при сравнении с null выдаст null) (<=> - не равно, при сравнении с null выдаст 0 )

**order by** – порядок

**group by…** - объединяет значения в столбиках по группам, у которых одинаковое заданное значение

**having …** - дополнительное условие

**default** – по умолчанию

**offset …** – смещение на …

**current\_timestamp** – текущее время

**on update** – при обновлении

**localtime** - текущее время

**localtimestamp** - текущее время

**current\_date** – текущая дата

**id in ($...)** – фильтр где id принадлежат массиву $...

**not null**  - не null

**is true** - проверка

**is false** - проверка

**is not null** – проверка

**and**  - и ( && ) умножение

**or** – или ( || ) суммирование

**between … and …** - условие на нахождение между … и …

**... in (…, …, …)** – проверка на наличие

**show databases;** - показать все базы данных

**create database … ;** - создать базу данных с именем …

**grant all privileges on имя базы .\* to ‘имя пользователя’@’localhost’ identified by ‘пароль’;** - создание нового пользователя базы данных

**show grants for ‘имя пользователя’@’localhost’;** - показать привилегии пользователя

**CRUD**

**create table subjects (** - создание таблицы

**-> id int(11) not null auto\_increment,** - поле id на 11 значений типа int без null и автоматическим инкрементом

**-> menu\_name varchar(255) not null,** - меню с заполнением типа varchar без null

**-> content text,** - тип текст

**-> position int(3) not null,** - позиция типом int на 3-х значное число без null

**-> visible tinyint(1) not null,** - булев тип без null

**-> primary key (id)** – установка ключевого для поиска столбца

**-> index (…)** – ссылки на другие таблицы

**-> );inset**

**insert into … (… , … , … ) values (‘…’, …, …);** - заполнение базы ( в первых душках колонки, во вторых значения идентично названиям )

**update имя set … = …, … = … where … = … limit … ;** - обновление параметров

**delete from имя where … = … limit …;**  - удаление строк

**select уточнение колонки from имя таблицы where название столбца=… order by название столбца … ;** - вывод таблицы **where** – уточнение ( =, >, <) **order by** – порядок ( в уточнении можно поставить \* что скажет все колонки )

**drop database … ;** - удалить базу данных

**show tables;** - показать таблицы

**show columns from …;** - выводит настройки таблицы

**Types**

**NUMERIC:** - целы числа

**TINYINT(…)** – 1 байт диапазон со знаками (-128 ; 127) или без (0 ; 255)

**SMALINT(…)** – 2 байта диапазон со знаками (-32768 ; 32767)

**MEDIUMINT(…)** – 3 байта диапазон со знаками (-8388608 ; 8388607) или без (0 ; 16777215)

**INT(…)** – 4 байта диапазон со знаками (-2147483648 ; 2147483647) или без (0 ; 4294967295)

**BIGINT(…)** – 8 байта диапазон со знаками (-9223372036854775808 ; 9223372036854775807) или без

(0 ; 18446744073709551615)

**DECIMAL:** - дробные числа

**DECIMAL(… , …)** – создание дробного (всего знаков, из них после комы)

**FLOAT:** - дробные с плавающей точкой

**Double(… , …)** – создание дробного не точного (всего знаков, из них после комы)

**Float(… , …)**  – создание дробного не точного (всего знаков, из них после комы)

**BIT:** - в битах

**Bit(…)** – создание битовой колонки (0 ; 64)

Переменная записывается **b’…’** – внутри единицы и нули

Чтение переменной через типизированние **bin(…)** – бинарный тип **oct(…)** – восьмеричный тип **hex(…)** – шестнадцатеричный тип

**STRING:** - строки, наборы бит

**Char(…)** – до 255 символов, запись в знаках

**Varchar(…)** - до 65535 символов, запись в знаках

**Binary(…)** – до 255 символов, запись в байтах

**Varbinary(…)** – до 6535 символов, запись в байтах

**Tinyblob** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в двуичном

**Blob** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в двуичном

**Mediumblob** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в двуичном

**Longblob** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в двуичном

**Tinytext** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в симвоольном

**Text** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в симвоольном

**Mediumtext** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в симвоольном

**Longtext** – хранение больших записей, обработка идет на диске, а не в оперативе, запись в симвоольном

**DATE:** - дата и время

**Date** –‘ YYYY-MM-DD’ от 1000-01-01 до 9999-12-31 - дата

**Datetime(…)** – ‘YYYY-MM-DD HH:MM:SS’ от 1000-01-01 00:00:00 до 9999-12-31 23:59:59 – дата и время, в дужках количество цифр для миллисекунд

**Timestamp(…)** – автоматически записывает текущее время при выполнении команд insert update ( default current\_timestamp on update current\_timestamp ) – для установки текущего времени при обновлении, в дужках количество цифр для миллисекунд

**Time(…)** – HH:MM:SS от -838:59:59 до 838:59:59 – время, в дужках количество цифр для миллисекунд

**Year** – YYYY – год

**Functions**

<https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html>

если ко всем результатам функций прибавить (+ 0), то выйдет числовой формат

**curtime(…)** – текущее время, в дужках количество цифр для миллисекунд

**now(…)** – текущая дата и время, время с начала выполнения функции

**sysdate(…)** – время по завершению функции

**current\_timestamp(…)** – текущее время

**localtime(…)** - текущее время

**localtimestamp(…)** - текущее время

**curdate(…)** – текущая дата

**current\_date(…)** – текущая дата

**utc\_date(…)** – текущая дата

**utc\_time(…)** – текущее время

**Date(‘… …’)** – вырезает дату из переменной с датой и временем

**Date\_add(‘… …’ , interval … second/day/year/hour)** – в переменную со временем добавляет время

**Date\_sub(‘… …’ , interval … second/day/year/hour)** – из переменно со временем вычитает время

**date\_format('2016-04-06 10:12:40', '%W %M %Y %H %MIN %S')** – меняет способ записи времени из цифр в слова ( расшифровка в табл. )

**datediff(‘… …’, ‘… …’)** – возвращает разницу между временами

**time(‘…’)** – вырезает время

**time\_format(‘…’, ‘%H’)** – меняет способ записи времени из цифр в слова ( расшифровка в табл. )

**time\_to\_sec('…’)** – конвертирует в секунды

**timediff(‘…’, ‘…’)** – разница во времени

**timestamp(‘…’, ‘…’)** – прибавляет время

**timestampdiff(… , ‘…’, ‘…’)** – в первом параметре уточнить тип, получает разницу во времени

**timestampadd(… , … , ‘…’)** – добавляет время (тип, количество, куда)

**from\_unixtime(…)** – перевод из типа unix , можно добавить шаблоны (‘%H, %M, %S, … ‘)

**unix\_timestamp()** – текущее время в unix , если вписать время, то конвертирует в unix

**maketime(…, …, …)** – конвертировать во временной тип

**year(‘…’)** – вырезает год из даты

**yearweek(‘…’)** – возвращает год с неделей

**month(‘…’)** – выделяет месяц

**monthname(‘…’)** – название месяца

**week(‘…’)** – возвращает неделю

**weekday(‘… …’)** – день недели

**weekofyear(‘…’)** – календарная неделя

**dayname(‘…’)** – текстовый день недели

**day(’…’)** – возвращает день из даты

**dayofmonth(‘…’)** – день месяца

**dayofweek(‘…’)** – день недели

**from\_days(…)** – из количество дней делает дату

**microsecond(‘…’)** – возвращает микросекунды из даты

**minute(‘…’)** – возвращает минуты из даты