# DDL

* CREATE DATABASE `databaseName`
* DROP DATABASE `databaseName`
* CREATE TABLE `tableName` (  
  `columnName` datatype constraints extras,  
  )
* DROP TABLE `tableName`
* ALTER TABLE `tableName` ADD `columnName` Datatype constraints extras
* ALTER TABLE `tableName` DROP `columnName`
* ALTER TABLE `tableName` CHANGE `columnName` `NewColumnName` Datatype constraints extras
* ALTER TABLE `tableName` ADD CONSTRAINT `constraintName` FOREIGN KEY(`columnName`) REFERENCES `table`(`primary`) ON UPDATE CASCADE ON DELETE RESTRICT

🡺 DML

* INSERT INTO `tableName` (`columnName1`,`columnName2`) VALUES (‘value1’,’value2’)
* UPDATE `tableName` SET `columnName` = ‘value1’ , `columnName2` = ‘value2’ WHERE condition
* DELETE FROM `tableName` WHERE condition

# DQL

* SELECT `columnName`,`columnName2` , …. FROM `tableName`
* SELECT \* FROM `tableName` WHERE condition (logical operators && comparison operators)
* SELECT \* FROM `tableName` WHERE `columnName` IN(‘value1’,’value2’)
* SELECT \* FROM `tableName` WHERE `columnName` BETWEEN miniNumber AND MaxNumber
* SELECT \* FROM `tableName` WHERE `columnName` IS NOT NULL
* Aggregate functions (MIN,MAX,COUNT,SUM,AVG,GROUP\_CONCAT)
* GROUP BY `columnName`
* HAVING condition
* ORDER BY `columnName` ASC|DESC , `columnName2` ASC|DESC
* LIMIT number,offset
* WHERE LIKE `columnName` = ‘string\_%’
* (LEFT-RIGHT-INNER) JOIN
* ON `parent`.`id` = `child`.`foreignKey`
* CREATE VIEW `viewName` AS (query)
* Read About (GROUP\_CONCAT,CONCAT,BUILT IN Functions,IF(),LIMIT value,offset, UNION , UNION ALL , Sub Query , JOIN Sub Query , Database normalization , Sequence Diagram , Use Case , UML )