|  |  |
| --- | --- |
| **My SQL Commands** | |
| **Database Commands** | |
| Create | creates a database.  Create Database database\_name; |
| Drop/Delete | Drop or delete an existing database, Once we deleted a database, all the tables and views will also be deleted.  Drop Database database\_name; |
| Rename/Alter | Used to rename the existing database.  Rename Database old\_database\_name To new\_database\_name; |
| Rename/Alter | In My SQL we need to select the database before executing any query.  Use Database database\_name; |
|  | |
| **Table Commands** | |
| Create | To create a table we should name the table and define it’s column and each column’s data type.  Create Table table\_name (  “column1” “datatype”,  “column2” “datatype”,  ...  “columnN” “datatype”); |
| Drop | Deletes the table definition and all data from a table.  Drop Table table\_name; |
| Delete | Deletes the rows from a table based on the condition defined by the WHERE clause or deletes all the rows from the table when condition is not specified.  Delete From table\_name Where condition;  Delete From table\_name; |
| Truncate | Delete all the tows from a table and free the containing space  Truncate Table table\_name; |
| Rename | Changes the name of the table.  Rename old\_table\_name To new-table\_name;  optionally we can use Alter Table also to rename the table. |
| Alter | It is used to add, modify or delete columns in an existing table. It also used to rename the table.  *--Add single column to the table*  Alter Table table\_name Add column\_name datatype  *--Add multiple columns to the existing table*  Alter Table table\_nameAdd  (“column1” “datatype”,  “column2” “datatype”  ...  “columnN” “datatype”)  *--Change the datatype of a column in a table*  Alter Table table\_name  Modify Column column\_name datatype  --*Delete a column from an existing table*  Alter Table table\_name  Drop Column column\_name;  --*Change the column name using Alter*  Alter Table table\_name  Rename Column old\_column\_name To new\_column\_name;  --*Change the table name using Alter*  Alter Table old\_table\_name To new\_table\_name; |
|  | |
| **Insert** | |
| Insert Into | It is used to insert single or multiple records to the table  Inset Into table\_name (column1, column2... columnN)  Values (value1, value2... valueN);  --*We no need to specify column name if we are adding all the values*  *of a record in columns order*  Insert Into table\_name  Values(value1, value2... valueN); |
|  | |
| **Update** | |
| Update | It is used to update the values of single or multiple columns of a row  Update table\_name  Set column\_name = value  Where condition;    Update table\_name  Set column1 = value, column2 = value... columnN = value  Where condition; |
|  | |
| **Delete** | |
| Delete | It is used to delete data from a row  Delete From table\_name Where condition; |
|  | |
| **Select** | |
| Select | It is used to retrieve data from a table in a database  --*retrieve data from column1 and column2 of table*  Select column1, column2 From table\_name;  --*retrieve all the data from table*  Select \* From table\_name; |
| Select Distinct | It is used to retrieve only distinct (different) values.  Select Distinct column1, column2 From table\_name; |
|  | |
| **Clauses & Operators** | |
| Where | It is used to provide a condition when using Select, Update and Delete statements  --*Select statement with Where clause*  Select column\_name  From table\_name  Where condition;  *--Update statement with Where clause*  Update table\_name  Set column\_name = value  Where condition;  --*Delete statement with Where clause*  Delete From table\_name  Where condition; |
| And | The And operator is used to check the multiple conditions in an SQL statement’s Where clause. It displays a record if all conditions separated by And are True.  Select column1, column2... columnN  From table\_name  Where condition1 And condition2 And... conditionN; |
| Or | The Or operator is used to check the multiple conditions in an SQL statement’s Where clause. It displays a record if any of the conditions separated by Or are True.  Select column1, column2... columnN  From table\_name  Where condition1 Or condition2 Or... conditionN; |
| Not | The Not operator displays the record if the condition of Where clause is Not True.  Select column1, column2... columnN  From table\_name  Where Not condition; |
| Limit (MySQL)  Top (SQL)  RowNum (Oracle) | The Limit (MySQL) clause is used to specify the number of records to return.  Select column\_name  From table\_name  Where condition1  Limit number; |
| Like | It is used in Where clause to compare a value to similar values using wildcard operators.  -> % sign represents the zero, one or more characters.  -> \_ sign represent a single number or character.  Select column\_name  From table\_name  Where condition Like pattern; |
| Order By | It is used to sort the data in ascending or descending order.  Select column\_name  From table\_name  Where condition  Order By column1, column2... columnN Asc | Desc; |
| Group By | It is used with aggregate functions (count, max, min, sum, avg) to group the result set by one or more columns.  Select column\_name  From table\_name  Where condition  Group By column\_names  Order By column\_name; |
| Having | The Having clause places conditions on groups created by Group By clause.  Select column\_name  From table\_name  Where condition  Group By column\_names  Having condition  Order By column\_name; |
| In | It allows you to specify multiple values in Where clause. It is shorthand form of multiple Or conditions.  Select column\_name  From table\_name  Where column\_name In (value1, value2... valueN); |
| Between | The Between operator selects values within a given range. The values can be numbers, text or dates.  Select column\_name  From table\_name  Where column\_name Between value1 And value2; |
|  | |
| **Constraints** | |
| Not Null | Ensures that column can not have Null value. |
| Unique | Ensures that all values in a column are different. |
| Primary Key | Uniquely identifies each row/ record in a database table. |
| Foreign Key | Uniquely identifies a row/ record in any of the given database table. |
| Default | Provides a default value for a column when none is specified. |
| Check | Ensures that all the values in a column satisfies certain conditions. |
| Auto\_Increment | It allows a unique number to be generated automatically when a new record is inserted into a table. |
| Index | Used to create or retrieve data from the database very quickly. |
| Ex:  Create Table Orders(  OrderId int Not Null Auto\_Increment,  OrderNumber int Not Null Unique,  PersonId int Not Null,  OrderDate date Default GetDate(),  Primary Key(OrderId)  Foreign Key(Person Id) References Persons(PersonId)  ); | |
|  | |
| **Joins** | |
| Inner Join | Returns records that have matching values in both the tables.  Select table1.column1, table2.collumn2....  From table1  Inner Join table2  On table1.common-field = table2.common-field; |
| Left (Outer) Join | Returns all records from left table, and matched records from right table.  Select table1.column1, table2.column2....  From table1  Left Join table2  On table1.common-field = table2.common-field; |
| Right (Outer) Join | Returns all records form right table, and matched records from left table.  Select table1.column1, table2.collumn2....  From table1  Right Join table2  On table1.common-field = table2.common-field; |
| Full (Outer) Join | Returns all records when there is a match in either left or right table.  Select table1.column1, table2.collumn2....  From table1  Full Join table2  On table1.common-field = table2.common-field; |
| Self Join | It is used to join a table to itself as if the table were two tables; temporarily renaming at lease one table in SQL statement  Select a.column1, b.collumn2....  From table1 a, table1 b  Where a.common-field = b.common-field; |
| Cross Join | The Cartesian Join or Cross Join returns the Cartesian product of the sets of records from two or more joined tables.  Select table1.column1, table2.column2...  From table1, table2...; |
|  | |
| **Functions** | |
| Count() | The Count aggregate functions is used to count the number of rows in a database table.  Select Count(\*)  From table\_name  Where Condition; –-optional where condition |
| Max() | The Max aggregate function is used to find out the record with highest (maximum) value for a certain column or record set.  Select Max(column\_name)  From table\_name; |
| Min() | The Min aggregate function is used to find out the record with lowest (minimum) value for a certain column or record set.  Select Min(column\_name)  From table\_name; |
| Avg() | The Avg aggregate function is used to find out the average value for a certain column or record set.  Select Avg(column\_name)  From table\_name; |
| Sum() | The Sum aggregate function is used to find out the sum of the numeric column.  Select Sum(column\_name)  From table\_name; |
| Sqrt() | The Sqrt aggregate function is used to find out the square root of any numeric record.  Select Sqrt(column\_name)  From table\_name;  --returns the Square root of every record inside the given column\_name. |
| Concat() | It is used to combine two string records.  Select Concat(column\_name1, column\_name2)  From table\_name; |