	ACTION	COMMAND	EXAMPLE
REFINE SELECTION	DISTINCT-gives unique values	**SELECT DISTINCT <columnname> FROM <tablename>;**	
	Sorting Data with ORDER BY	**SELECT <columnname> FROM <tablename> ORDER BY <columnname>;** **SELECT <columnname> FROM <tablename> ORDER BY <columnname> DESC;**	sorting ASECDING by default. DESC for descending order
		SELECT <columnname1>,<columnname2>,<columnname3> **FROM <tablename>** **ORDER BY 2;** **SELECT <columnname1>,<columnname2>,<columnname3>** **FROM <tablename>** **ORDER BY 2 DESC;**	ORDER BY #n : choose the nth column in SELECT for sorting
		SELECT <columnname1>,<columnname2> **FROM books** **ORDER BY <columnname1>,<columnname2>;**	ORDER BY x,y : first sort wrt x and then the result is sorted on y
	LIMIT	**SELECT <columnname> FROM <tablename> LIMIT <#number>;**	LIMIT comes @ the end of statement.
		SELECT <columnname> FROM <tablename> **ORDER BY <columnname>** **DESC** **LIMIT <#number>;**	SELECT title FROM books LIMIT 3;
		SELECT <columnname> FROM <tablename> **LIMIT x,y;** **x - starting position** **y - how many entries from x**	SELECT title, released_year FROM books ORDER BY released_year DESC LIMIT 0,5;
		SELECT * FROM <tablename> **LIMIT x,18446744073709551615;** **x - starting position** **18446744073709551615- to get all the values till end**	
	LIKE	**SELECT <columnname>** **FROM <tablename>** **WHERE <columnname>** **LIKE '%abc%';**	Better searching - % and _ are Wildcards % = means anything can come ie. none to many _ = means only 1 chara
		SELECT <columnname> **FROM <tablename>** **WHERE <columnname>** **LIKE '%\%%'** **SELECT <columnname>** **FROM <tablename>** **WHERE <columnname>** **LIKE '%_%'**	To match % and _ in a text. Then use backslash before each
	Replace parts of a string	**SELECT REPLACE(<text>, '<to be replaced>', '<replace with>');** **SELECT REPLACE(<columname>, '<to be replaced>', '<replace with>') FROM <tablename>;**	The REPLACE() function, as well as the other string functions, only change the query output, they don't affect the actual data in the database.
	Reverse string	**SELECT REVERSE('<text>');** **SELECT REVERSE(<columname>) FROM <tablename>;**	
	Length of string	**SELECT CHAR_LENGTH('<text>');** **SELECT CHAR_LENGTH(<columnname>) FROM <tablename>;**	
	UPPER case LOWER case	**SELECT UPPER('<text>');** **SELECT UPPER(<columnname>) FROM <tablename>;** **SELECT LOWER('<text>');** **SELECT LOWER(<columnname>) FROM <tablename>;**	only takes one argument