

EGTDC Database Course SQL Cheat-sheet
18-19th May 2004

SELECT

<i>Command</i>	<i>What it does</i>
SELECT * from <table>	get all entries
SELECT DISTINCT * from <table>	get all unique entries
SELECT <col1>, <col2> FROM <table>	get only columns col1 and col2
SELECT <col1> as "foo" FROM <table>	get col1, but call it "foo" in the results
SELECT * FROM <table> ORDER BY <col1> ASC, <col2> DESC	sort by col1 – if 2 rows have the same value here then sort these by col2 in reverse order
SELECT * FROM <table> WHERE <col1> = <col2>	a simple search condition based on equality
SELECT * FROM <table> WHERE <col1> = 5 OR (<col1> = 6 AND lower(<col2>) = 'snake')	find all rows where col1 is 5, or where col1 is 6 but also col2 contains 'snake', 'Snake', 'sNAke' etc.
SELECT * FROM <table> WHERE <col1> IS NULL	searching for NULL values with 'IS NULL'
SELECT * FROM <table> WHERE <col1> IN (3, 4, 5)	specify that col1 must be 3, 4 or 5
SELECT * FROM <table1> WHERE <col1> IN (SELECT <col2> FROM <table2>)	obtain the list of matches from col2 in table2
SELECT * FROM <table1> t1 WHERE EXISTS (SELECT * FROM <table2> t2 WHERE t2.<col2> = t1.<col1>)	same as the previous, but using the EXISTS clause. Choose whichever you find easiest!
SELECT t1.*, t2.* FROM <table1> t1 INNER JOIN <table2> t2 ON (t1.<pk_col> = t2.<fk_col>)	inner join, where tables 1 and 2 have a one-to-many relationship
SELECT t1.*, t2.* FROM <table1> t1 LEFT OUTER JOIN <table2> t2 ON (t1.<pk_col> = t2.<fk_col>)	an outer join where columns from table 1 will be displayed even if there is no matching entry in table 2
SELECT <col1> as foo, <col2> as bar FROM <table1> UNION SELECT <col1> as foo, <col3> as bar FROM <table2> ORDER BY foo	combine two SELECTS into a single result set – note the SELECT keyword appears twice, but the ORDER BY affects the whole result. Duplicate rows are not shown. Use 'UNION ALL' to show all duplicate rows
SELECT <col1> as foo, <col2> as bar FROM <table1> INTERSECT SELECT <col1> as foo, <col3> as bar FROM <table2> ORDER BY foo	Shows all rows common to two SELECT statements. 'INTERSECT ALL' will show all duplicate rows
SELECT <col1>, sum(<col2>) FROM <table1> WHERE <col3> = 1 GROUP BY <col1> HAVING count(*) < 3	group entries in table1 by col1, showing each value of col1 as well as the sum of col2 for those rows disregard rows where col3 is not 1, and show only results with 1 or 2 occurrences of that value of col1 in the table
SELECT * INTO <newtable> FROM <oldtable>	Create newtable from oldtable

DELETE, INSERT, UPDATE

<i>Command</i>	<i>What it does</i>
DELETE from <table>	delete EVERYTHING in a table
DELETE from <table> WHERE <col1> != 5	delete by some condition
INSERT INTO <table> (<col1>, <col2>) VALUES ('foo', 'bar')	insert with specific values
INSERT INTO <table2> SELECT <col1>, <col2> FROM <table1>	insert values based on a SELECT statement

Command	What it does
UPDATE <table> SET <col1> = 'foo', <col2> = <col2> * <col2> WHERE <col3> IS NOT NULL	update with WHERE condition

OPERATORS AND FUNCTIONS

Command	What it does
SELECT (<col1> / <col2>) *100 as "percentage" FROM <table>	perform some arithmetic on the columns (other mathematical operators include + and -)
SELECT sqrt(<col1>) FROM <table>	Select the square root of col1
SELECT trunc(<col1>,3) FROM <table>	Truncate the value of col3 after 3 decimal points
abs(<col1>),round(<col1>,2),ln(<col1>),sin(<col1>),tan(<col1>)	Absolute value of col1,round col1 to 2 decimal places, natural log col1,sine of col1,tangent of col1 See documentation for further mathematical functions
SELECT <col1> '-' <col2> FROM <table>	Concatenate col1 to col2 with a hyphen
SELECT substr(<col1>,2,3) FROM <table>	Select substring of col1 starting at character 2 ,3 characters in length
SELECT upper(<col1>) FROM <table>	Display col1 in upper case
SELECT lower(<col1>) FROM <table>	Display col1 in lower case
SELECT * from <table> WHERE <col1> = 'badger'	Select rows where col1 matches string 'badger'
SELECT * from <table> WHERE <col1> != 'badger'	Select rows where col1 does not match string 'badger'
SELECT * from <table> WHERE <col1> LIKE '%adg%'	Select rows where col1 contains the string 'adg' (note use of '%' wildcard character)
SELECT sum(<col1>), avg(<col1>) FROM <table>	return both sum and average of col1
SELECT count(*) FROM <table>	Count the number of rows in specified table
SELECT <col1>, sum(<col2>) FROM <table1> WHERE <col3> = 1 GROUP BY <col1> HAVING count(*) < 3	group entries in table1 by col1, showing each value of col1 as well as the sum of col2 for those rows disregard rows where col3 is not 1, and show only results with 1 or 2 occurrences of that value of col1 in the table
SELECT current_timestamp	Display the current date and time
SELECT to_char(<datecol>,'DD/MM/YYYY') FROM <table1>	Select datecol in the format DD/MM/YYYY (eg.21/04/2003)
SELECT to_date('200425thJune','YYYYDDthmonth')	Convert the string '200425thJune' to a date
SELECT <datecol>::varchar FROM <table1>	Typecast datecol to data type varchar
SELECT coalesce(<col1>,'No value') FROM <table1>	Display 'No value' is value in col1 is null
SELECT case <col1> ='snake' then 'badger' else 'mushroom' end FROM <table1>	If value in col1 is 'snake' print 'badger' otherwise print 'mushroom'

Data Types

<i>Data Type</i>	<i>Description</i>
integer,int,int4	Whole number/integer
float	Floating point number
numeric(<i>p</i> , <i>s</i>)	Exact numeric type with digits ' <i>p</i> ' and digits after decimal point ' <i>s</i> '
date	Calendar date
timestamp	Date and time
varchar(<i>n</i>)	Variable length character string of max length ' <i>n</i> '
char(<i>n</i>)	Fixed length character string of length ' <i>n</i> '
text	Variable length character of unlimited length
boolean, bool	A single true or false value (supported values: true/false,'t'/'f','true'/'false','y'/'n','yes'/'no','1','0')