33:136:370 MIS

SQL Queries - Quick Reference Guide

SELECT

SELECT expression(s) FROM table	Select data from a table.	SELECT LName, FName FROM Persons
SELECT * FROM table	Select data from all columns from a table.	SELECT * FROM Persons
SELECT DISTINCT expression(s) FROM table	Select only distinct (different) data from a table.	SELECT DISTINCT LName, FName FROM Persons
SELECT expression(s) FROM table(s) WHERE condition(s) *Note: an expression can be a field, calculation (e.g. field 1+field2), or a function*	Select only certain data from a table. **Operators** *Operator* — *Description* = — Equal <> — Not equal >, <, >=, <= — Inequalities BETWEEN … AND … — Between an inclusive range IN(.. , .. , ..) — Checks if a field is in a list of values IS NULL — Checks if a field is empty AND, OR, NOT — Logical operators LIKE — Search for a pattern. A "*" means: match any number of occurrences of any character; A "?" sign means: match one occurrence of any character.	SELECT * FROM Persons WHERE sex='female' SELECT * FROM Persons WHERE Year>1970 SELECT * FROM Persons WHERE Date>#1/1/1970# SELECT * FROM Persons WHERE LName IN ('Hansen','Pettersen') SELECT * FROM Persons WHERE (FName='Tove' OR FName='Stephen') AND LName='Svendson' SELECT * FROM Persons WHERE LName LIKE 'C*' SELECT * FROM Books WHERE Year LIKE '18??'
SELECT expression(s) FROM table(s) ORDER BY expression1, expression2 DESC ,...	Select data from a table sorting the rows. Note: ASC (ascend) is a alphabetical and numerical order DESC (descend) is a reverse alphabetical and numerical	SELECT * FROM Persons ORDER BY LName SELECT FName, LName FROM Persons ORDER BY LName DESC. Fname DESC

<table>
<tr><td>SELECT expression(s), aggregate_functions(…)
FROM table(s)
GROUP BY expression(s)</td><td>Groups the data and calculates aggregate functions for each group
Some aggregate functions<table><tr><th>Function</th><th>Description</th></tr><tr><td>COUNT()</td><td>counts the non-blank values (non NULLs)</td></tr><tr><td>MAX(), MIN(), SUM(), AVG()</td><td>Highest, lowest, sum and average values</td></tr></table></td><td>SELECT Company, SUM(Amount)
FROM Sales
GROUP BY Company</td></tr>
<tr><td>SELECT column(s), aggregate_functions(…)
FROM table(s)
GROUP BY expression(s)
HAVING condition(s)</td><td>Filters groups</td><td>SELECT Company, SUM(Amount)
FROM Sales
GROUP BY Company
HAVING SUM(Amount)>10000</td></tr>
<tr><td colspan="3">Alias</td></tr>
<tr><td>SELECT expression AS column_alias
FROM table(s)</td><td>Column name alias: just for displaying purposes</td><td>SELECT LName AS Family, FName AS Given
FROM Persons</td></tr>
<tr><td>SELECT table_alias.column
FROM table AS table_alias</td><td>Table name alias, that can be used in other clauses of the query (since FROM is the first to be read)</td><td>SELECT P.LName, P.FName
FROM Persons AS P</td></tr>
<tr><td colspan="3">Join</td></tr>
<tr><td>SELECT expression(s)
FROM first_table INNER JOIN second_table ON condition(s)</td><td>The INNER JOIN returns only rows from both tables where the condition(s) hold.</td><td>SELECT Employees.Name, Orders.Product
FROM Employees INNER JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID</td></tr>
<tr><td>SELECT expression(s)
FROM first_table LEFT JOIN second_table ON condition(s)</td><td>The LEFT JOIN returns all the rows from the first table, even if there are no matches in the second table.</td><td>SELECT Employees.Name, Orders.Product
FROM Employees LEFT JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID</td></tr>
<tr><td>SELECT expression(s)
FROM first_table RIGHT JOIN second_table ON condition(s)</td><td>The RIGHT JOIN returns all the rows from the second table, even if there are no matches in the first table.</td><td>SELECT Employees.Name, Orders.Product
FROM Employees RIGHT JOIN Orders ON Employees.EmployeeID=Orders.EmployeeID</td></tr>
<tr><td colspan="3">Adapted from: http://www.cheat-sheets.org/sites/sql.su/</td></tr>
</table>