**SELECT** \* **FROM** Table1 a **JOIN** Table2 b **ON** a.Col1 = b.ColN **WHERE** Col2 = value (number vs ‘string’)

Replace \* with Col1 as [New Name],Col2,count(Col3)

**GROUP BY** Col1,Col2 **HAVING** Col2 > 100 **ORDER BY** Col1, Col3 **desc**

desc: most recent, largest, or ZYX first (opposite is **asc** which is the default)

Wildcards: % (multiple), \_ (single)

WHERE \_\_\_\_\_ **AND** \_\_\_\_\_ WHERE \_\_\_\_\_ **OR** \_\_\_\_\_ WHERE \_\_\_\_\_ **XOR** \_\_\_\_\_

WHERE \_\_\_\_\_ **= (SELECT…)**

Col **=** value Col **<>** value vs. Col **!=** value

Col **>** value Col **>=** value

Col **<** value Col **<=** value

Col **BETWEEN** value **AND** value (***inclusive***)

Col **IN** (‘value’,’value’,’value’)

Col **NOT** **IN** (‘value’,’value’,’value’)

Col **LIKE** ‘%lue’

Col **NOT LIKE** ‘val\_e’

**COUNT**(), **MAX**(), **MIN**(), **SUM**(), **AVG**() *Summaries exclude NULL? (Like count)*

**DISTINCT**

UPPER(), LOWER(), ROUND(Col,2), LEFT(Col,1), CONCAT()

LENGTH() can be used to look for greater than 0 to eliminate blanks

ALL(), ANY(), EXISTS(), NOT EXISTS(),

FIRST(),LAST()

**IS NULL**

**IS NOT NULL**

**CASE**

**WHEN** condition1 **THEN** value1

**WHEN** condition2 **THEN** value2

**ELSE**

**END**

**COALESCE**()

**SINCE NULL is dropped better to specify <> OR IS NULL**

Need to finish with ; for non T-SQL

Math: /,\*2

Numerical rounding occurs in math operations? Divide by decimal to get decimal

LIMIT 5 to reduce output similar to HEAD(5). This is for MySQL, Oracle uses ROWNUM<=5, and TOP(5) is used in TSQL

**INNER JOIN** default join results in records that have matching values in each table

**LEFT JOIN** return all the records from the left table and matching from the right or null if no match from right (outer)

**RIGHT JOIN** reverse of left join, return all the records from the right table and matching from the left or null if no match from left (outer)

**FULL OUTER JOIN** all the record from both tables with nulls if no corresponding matches (outer)

***SELF JOIN***

***CROSS JOIN (performed when the where clause is omitted and results in a cartesian product)***

COUNT of ROWs = COUNT(\*)

UNION

UNION ALL

MINUS

INTERSECT

ROW\_NUMBER

RANK

DENSE\_RANK

*Check for Null values:*

WHERE Col IS NULL

Count Null values in a column

select count(\*)-count(Col)as missing

from fortune500;

*Check for Blank Values:*

WHERE LENGTH(Col)<1

*Substitute Null Values:*

ISNULL(Col,’value’) AS NewCol

ISNULL(Col3,Col2) AS NewCol

ORACLE is NVL()