Oracle DB – Cheat Sheet on Joins

ANSI Syntax

1. Inner Join: **Natural Join**
    a. Matching rows only
    b. Join on fields in common (matching names)
    c. Must have matching values in all common fields
    d. SELECT col_a, col_b, col_c,… col_n
       FROM table1 NATURAL JOIN table2
       WHERE [additional conditions if any]
2. Inner Join
    a. Specify cols to be used when joining
    b. **JOIN USING**
        i. Simple
        ii. Both tables have same col name
        iii. SELECT col_a, col_b, col_c,… col_n
             FROM table1 INNER JOIN table2
             USING (col_d)
             WHERE [additional conditions if any]
    c. JOIN ON
        i. Complex
        ii. tables have different col names or you want to explicitly specify matching cols
        iii. Two tables:
             SELECT t2.col_a, t2.col_b, t2.col_c,… t1.col_n
             FROM table1 t1 INNER JOIN table2 t2
             ON (t1.col_d=t2.col_e)
             WHERE [additional conditions if any]
        iv. Three tables:
             SELECT t2.col_a, t2.col_b, t3.col_c,… t1.col_n
             FROM table1 t1 INNER JOIN table2 t2
             ON (t1.col_d=t2.col_e)
             JOIN table3 t3
             ON (t2.col_f=t3.col_g)
             WHERE [additional conditions if any]
3. Outer Join
    a. return all rows from one table or both tables regardless of matching condition
    b. Left Outer Join
        i. Use general JOIN ON statement but change JOIN to LEFT OUTER JOIN
        ii. returns NULL values on right if necessary

c. **Right Outer Join**
    i. Use general JOIN ON statement but change JOIN to RIGHT OUTER JOIN
    ii. returns NULL values on left if necessary
d. **Full Outer Join**
    i. Use general JOIN ON statement but change JOIN to FULL OUTER JOIN
    ii. unmatched join attributes from either side are paired with null values on the other side
    iii. you will probably not have to use this with most well-designed databases

4. **Cross Join**
    a. Use general JOIN ON statement but change JOIN to CROSS JOIN
    b. Avoid – normally an error
    c. joins everything
    d. Does not usually produce useful information
    e. Same as Cartesian Product in Oracle Proprietary Syntax
5. **Self Join**
    a. Inner or Outer Join between two attributes
    b. SELECT emp.employee_id, emp.last_name,
       emp.manager_id, mgr.last_name
       FROM employees emp LEFT OUTER JOIN employees mgr
       ON emp.manager_id = mgr.employee_id
       ORDER BY emp.employee_id

Oracle proprietary joins

1. Inner Join: **Equijoin**
    a. Join is based on equality
    b. Rows are joined if an exact match exists between table1 and table2
    c. Select only rows that match
    d. SELECT t1.col_a, t1.col_b, t2.col_c,… col_n
       FROM table1 t1, table2 t2,… tablen tn
       WHERE t1.col_e = t2.col_f [additional conditions]
2. Inner Join: **Non-Equijoin**
    a. Join is based on logical expression
    b. Rows are joined if the expression is True
    c. Select only rows with the True expression
    d. Confusing; can usually be done with an Equijoin and additional conditions in WHERE
    e. SELECT t1.col_a, t1.col_b, t2.col_c,… col_n
       FROM table1 t1, table2 t2
       WHERE t1.col_e conditional_operator t2.col_f [additional conditions]

3. Outer Join
    a. Join rows based on data in WHERE clause but include rows that do not have a match
    b. Place + on side of missing data
    c. **Right Outer Join**:
        i. (+) on left
        ii. List all rows from right side of join condition and the matches from the left (or null for no match)
    d. **Left Outer Join**:
        i. (+) on right
        ii. List all rows from left side of join condition and the matches from the right (or null for no match)
4. **Cartesian Product**
    a. Avoid – normally an error
    b. Missing WHERE clause so it joins everything
    c. Number of rows returned = #rows_table1 * #rows_table2
    d. Does not usually produce useful information
5. Compare result:
    a. Inner Join: Equijoin (Matching rows only)
```
SELECT id, first_name, last_name, order_number
FROM f_customers c, f_orders o
WHERE c.id=o.cust_id
```
    b. Right Outer Join – OK to show orders without customers (this is the NULL set for our example)
```
SELECT id, first_name, last_name, order_number
FROM f_customers c, f_orders o
WHERE c.id(+)=o.cust_id
```
    c. Left Outer Join – OK to show customers without orders
```
SELECT id, first_name, last_name, order_number
FROM f_customers c, f_orders o
WHERE c.id=o.cust_id(+)
```
6. **Self Join**
    a. recursive join
    b. same table referenced twice with alias in FROM clause
    c. SELECT t1.col_a, t1.col_b, t2.col_c,… col_n
```
SELECT t1.col_a, t1.col_b, t2.col_c,… col_n
FROM table1 t1, table2 t2
WHERE t1.col_a = t2.col_a
```
    d. Example
```
SELECT emp.employee_id, emp.last_name,
            emp.manager_id, mgr.last_name
FROM employees emp, employees mgr
WHERE emp.manager_id = mgr.employee_id(+)
ORDER BY emp.employee_id
```