**Compare And Contrast JOINS and SUB-QUERIES**

Use following Two Tables For Refrence:-

Customer

|  |  |  |
| --- | --- | --- |
| Customer\_ID | Name | City |
| 1 | A K Ltd | Delhi |
| 2 | V K Associate | Mumbai |
| 3 | R K India | Banglore |
| 4 | R S P Ltd | Kolkata |
| 5 | A T Ltd | Delhi |
| 6 | D T Info | Delhi |

Products

|  |  |  |  |
| --- | --- | --- | --- |
| Product\_ID | Prod\_Detail | Customer\_ID | Price |
| 111 | Monitor | 1 | 7000 |
| 112 | Processor | 2 | 11000 |
| 113 | Keyboard | 2 | 1200 |
| 114 | Mouse | 3 | 500 |
| 115 | CPU | 5 | 15500 |

When to Use Joins and Subqueries

Use a join or a subquery any time when you reference information from multiple tables. Joins and subqueries are often used together in the same query. In many cases, you can solve a data retrieval problem by using a join, a subquery, or both. Here are some guidelines for using joins and queries.

* If your report needs data that is from more than one table, then you must perform a join. Whenever multiple tables (or views) are listed in the FROM clause, those tables become joined.
* If you need to combine related information from different rows within a table, then you can join the table with itself.
* Use subqueries when the result that you want requires more than one query and each subquery provides a subset of the table involved in the query.
* If a membership question is asked, then a subquery is usually used. If the query requires a NOT EXISTS condition, then you must use a subquery because NOT EXISTS operates only in a subquery; the same principle holds true for the EXISTS condition.
* Many queries can be formulated as joins or subqueries. Although the PROC SQL query optimizer changes some subqueries to joins, a join is generally more efficient to process.

**From:-** [**http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/viewer.htm#a001361223.htm**](http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/viewer.htm#a001361223.htm)

**DIFFERENCE BETWEEN SUB-QUERIES AND JOINS**

**1).In most cases JOINS are faster than SUB QUERIES and it is very rare for a sub-query to be faster**

From:- <http://stackoverflow.com/questions/2577174/join-vs-subquery>

Reason:- In case of JOINS RDBMS can create an execution plan that is better for your query and can predict what data should be loaded to be processed and saves time, unlike the sub-query where it will run all the queries and load all their data to do the processing.

**subquery will be executing two select statements while the join will only be selecting one select statement, so by this attribute a join statement will always be faster.**

Query to get Customer Names Who Purchased Monitor.

Join:- Select c.Name

From Customer c

JOIN Products p ON (c.Customer\_id=p.Customer\_id);

WHERE p.Prod\_Detail=’MONITOR’; **//Only One Select Statement**

SubQuery:- Select Name from Customer WHERE Customer\_ID=

(Select Customer\_id from Products

WHERE Prod\_Detail=’MONITOR’); **//Two Select Statements**

|  |
| --- |
| Name |
| A K Ltd |

**2). Sub-queries are more readable than JOINS.**

**Ex-**

Query to get list of Customers that are available in Products table:-

By Join:-

Select \*

From Customer c

JOIN Products p ON (c.Customer\_id=p.Customer\_id);

By SubQuery:-

Select \* from Customer WHERE Customer\_ID IN

(Select Customer\_id from Products);

|  |  |  |
| --- | --- | --- |
| Customer\_ID | Name | City |
| 1 | A K Ltd | Delhi |
| 2 | V K Associate | Mumbai |
| 3 | R K India | Banglore |
| 5 | A T Ltd | Delhi |

**3).Sub-queries are the logically correct way to solve problems of the form, "Get facts from Table1, conditional on facts from Table2".**

In such instances, it makes more logical sense to stick B in a sub-query than to do a join.

Ex-

Query for getting the list of customers that are not available in Product tables.

SELECT \* FROM Customer

WHERE Customer\_Id NOT IN

(SELECT Customer\_id FROM Products);

|  |  |  |
| --- | --- | --- |
| Customer\_ID | Name | City |
| 4 | R S P Ltd | Kolkata |
| 6 | D T Info | Delhi |

**4).** **Subquery first takes separate temp table and then check’s condition**

Besides join checks conditions first and then put it into table and displays as we mentioned.  
If our tables has got big amount of data than, during the execution, subquery takes more load and hence more time as well.  
So, its more convenient to use join instead of subquery.

**Subquery Advantages**

Subqueries are advantageous over joins when we have to calculate an aggregate value on the fly and use it in the outer query for comparison.

**Example:**To Get the Customer\_Id and Name of the Customer who have purchased the Product having Least Cost. We could write the query

Select Customer\_id,Name from Customer

WHERE Customer\_id=(

Select Customer\_id FROM Products

Where price=(

Select min(price) FROM Products)

|  |  |
| --- | --- |
| Customer\_id | Name |
| 3 | R K India |

**Join Advantages**

Joins are advantageous over sub-queries if the SELECT query contains columns from more than one table.

Query to get Customer\_id, Names , product\_id , Product detail,Price who purchased products having cost more than 7000 .

Select c.Customer\_id,c.Name as Customer\_Name,p.Product\_id,p.Prod\_Detail as Product\_Details,p.Price

From Customer c

JOIN Products p ON (c.Customer\_id=p.Customer\_id);

WHERE p.Price>7000;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer\_ID | Customer\_Name | Product\_id | Product\_Details | Price |
| 2 | V K Associate | 112 | Processor | 11000 |
| 5 | A T Ltd | 115 | CPU | 15500 |