



Discussion

Compare and contrast **two** of the following: INNER JOIN; subquery; set operation; single-table query. Please do not write about more than two of these query types. Illustrate your answer with T-SQL code examples.

6/28/2014 4:37:25 PM

The two queries I have chosen to elaborate on are INNER JOIN and set operation. According to Ben-Gan, Sarka, and Talmage (2013), INNER JOINS are used to match rows from tables where the relationships are formed via a predicate (Ben-Gan, Sarka, and Talmage, 2013). This is accomplished by using primary and foreign keys. For example, let's say we have two tables, Table A and Table B. Table A would contain the primary key which would be linked to a foreign key in Table B. The result would be a combined table containing the columns of both tables. I have create some tables to demonstrate the INNER JOIN.

Here are the two tables that I want to join. So, I use the INNER JOIN to tie the tables together at StatID (primary key) in the Status table, to StatID (foreign key) in the Student table.

Student			
StuID	StatID	LName	FName
1	B	Johnson	Tom
2	B	Tyler	Mary
3	A	Nelson	Jack

Status	
StatID	Status
A	Graduate
B	Undergraduate
C	Dropped

And then if you execute this query

```
SELECT Student.StatID, Status.StatID, Status, LName, FName  
FROM Student  
INNER JOIN Status  
ON Student.StatID=Status.StatID;
```

(W3Schools, n.d.)



the end result looks like this

Student Record				
StuID	StatID	Status	LName	FName
1	B	Undergraduate	Johnson	Tom
2	B	Undergraduate	Tyler	Mary
3	A	Graduate	Nelson	Jack

Now for the second query; I have selected *set operation*. Set operation is actually a set of operations that are performed on data in tables to obtain meaningful results (StudyTonight, n.d.). The set operations are UNION, UNION ALL, INTERSECT, and EXCEPT. These operations are similar to the JOIN in that they manipulate tables to combine data sets; however, they are different because they can perform more complex operations, thus returning more practical, meaningful data than just a JOIN, which just returns the entire joined column.

An example of UNION, which just combines the results of SELECT statements, would be:

Table_1		Table_2	
StudID	Name	StudID	Name
1	Eddie	2	Terry
2	Terry	3	Jack

And running this query

```
SELECT * FROM Table_1  
UNION  
SELECT * FROM Table_2
```



would result in

Table_3	
StudID	Name
1	Eddie
2	Terry
3	Jack

Note, UNION ALL is very similar to UNION, except UNION ALL would return the duplicate rows. In the above example, that means that the student Terry would show up twice.

The INTERSECT operator combines two SELECT statements; however, the INTERSECT will only return like records that are common in each of the SELECT statements. For example (using our same tables),

Table_1	
StudID	Name
1	Eddie
2	Terry

Table_2	
StudID	Name
2	Terry
3	Jack

and running this query

```
SELECT * FROM Table_1  
INTERSECT  
SELECT * FROM Table_2
```

The results would be

Table_3	
StudID	Name



2	Terry
---	-------

And, the last set operator is EXCEPT. EXCEPT also combines the results of two SELECT statements; however, the EXCEPT returns only the unique items that are in the first set.

For example,

Table_1	
StudID	Name
1	Eddie
2	Terry

Table_2	
StudID	Name
2	Terry
3	Jack

and running this query

```
SELECT * FROM Table_1  
EXCEPT  
SELECT * FROM Table_2
```

would result in

Table_3	
StudID	Name
1	Eddie

References


Ben-Gan, I., Sarka, D., & Talmage, R. (2013). *Training Kit (Exam 70-461): Querying Microsoft® SQL Server® 2012*. Microsoft Press.



StudyTonight. (n.d.). Set Operation in SQL. Retrieved from <http://www.studytonight.com/dbms/set-operation-in-sql.php>

W3Schools. (n.d.). SQL INNER JOIN Keyword. Retrieved from http://www.w3schools.com/sql/sql_join_inner.asp


RE: Unit 2 -

Set Ops and Subqueries Edward Jackson 

6/29/2014 12:03:07 PM

You have given some good details on UNION, INTERSECT, and EXCEPT set operators. In my research, I saw that UNION returns the rows but does not return duplicates. If you want to return the duplicate rows, you have to use the UNION ALL set operator. Something else that I found interesting is that, originally, I saw the MINUS instead of EXCEPT operator (I guess it has been replaced). I'm wondering if the professor can tell us the difference. In your research, did you come across single-table queries? I could not find them in the book.

RE:

Initial Post Edward Jackson 

6/30/2014 9:15:06 AM

You have given some excellent examples of inner join and subqueries. I do have a few examples of subqueries in general. In the following example, JOINS are used to return certain titles (app titles) that are being purchased under the game title. You will notice that the titles (app titles) table (app sales) are joined in the query.

```
SELECT DISTINCT ap.application_name as [App Name]
```

```
FROM dbo.applications ap
```

```
JOIN dbo.sales sa ON ap.application_id = sa.app_id
```

```
JOIN dbo.titles t ON t.title_id = sa.title_id
```



```
WHERE t.type = 'game'
```

This returns apps that have the type "game."

Why would it be better to use a subquery such as this:

```
SELECT application_name as AppName
```

```
FROM dbo.applications
```

```
WHERE app_id IN
```

```
    (SELECT app_id
```

```
      FROM dbo.titles
```

```
      WHERE type = 'game')
```

Does it process faster? Or is it just best practice to use subqueries over joins when in our book, it states that a subquery can be self-contained (page 118). I'm wondering if you use subqueries over joins is because subqueries offer higher functionality when dealing with large datasets in tables.

My Notes

EXAM TIP

Often, when joining tables, you join them based on a foreign key–unique key relationship. For example, there's a foreign key defined on the supplierid column in the Production.Products table (the referencing table), referencing the primary key column supplierid in the Production.Suppliers table (the referenced table). It's also important to note that when you define a primary key or unique constraint, SQL Server creates a unique index on the constraint columns to enforce the constraint's uniqueness property. But when you define a foreign key, SQL Server doesn't create any indexes on the foreign key columns. Such indexes could improve the performance of joins based on those relationships. Because SQL Server doesn't create such indexes automatically, it's your responsibility to identify the cases where they can be



useful and create them. So when working on index tuning, one interesting area to examine is foreign key columns, and evaluating the benefits of creating indexes on those.

Quick Check

1. What is the difference between the old and new syntax for cross joins?
2. What are the different types of outer joins?

Quick Check Answer

1. The new syntax has the CROSS JOIN keywords between the table names and the old syntax has a comma.
2. Left, right, and full.

Lesson Summary

- Cross joins return a Cartesian product of the rows from both sides.
- Inner joins match rows based on a predicate and return only matches.
- Outer joins match rows based on a predicate and return both matches and nonmatches from the tables marked as preserved.
- Multi-join queries involve multiple joins. They can have a mix of different join types. You can control the logical join ordering by using parentheses or by repositioning the ON clauses.

Quick Check

1. What is the difference between self-contained and correlated subqueries?
2. What is the difference between the APPLY and JOIN operators?

Quick Check Answer

1. Self-contained subqueries are independent of the outer query, whereas correlated subqueries have a reference to an element from the table in the outer query.
2. With a JOIN operator, both inputs represent static relations. With APPLY, the left side is a static relation, but the right side can be a table expression with correlations to elements from the left table.

Lesson Summary



- With subqueries, you can nest queries within queries. You can use self-contained subqueries as well as correlated ones. You can use subqueries that return single-valued results, multi-valued results, and table-valued results.
- T-SQL supports four kinds of table expressions, which are named query expressions. Derived tables and CTEs are types of table expressions that are visible only in the scope of the statement that defined them. Views and inline table-valued functions are reusable table expressions whose definitions are stored as objects in the database. Views do not support input parameters, whereas inline table-valued functions do.
- The APPLY operator operates on two table expressions as input. It applies the right table expression to each row from the left side. The inner query in the right table expression can be correlated to elements from the left table. The APPLY operator has two versions; the CROSS APPLY version doesn't return left rows that get an empty set back from the right side. The OUTER APPLY operator preserves the left side, and therefore, does return left rows when the right side returns an empty set. NULLs are used as placeholders in the attributes from the right side in the outer rows.

Quick Check

1. Which set operators does T-SQL support?
2. Name two requirements for the queries involved in a set operator.

Quick Check Answer

1. The UNION, INTERSECT, and EXCEPT set operators, as well as the UNION ALL multiset operator.
2. The number of columns in the two queries needs to be the same, and corresponding columns need to have compatible types.

Lesson Summary

- Set operators compare complete rows between the result sets of two queries.
- The UNION operator unifies the input sets, returning distinct rows.
- The UNION ALL operator unifies the inputs without eliminating duplicates.
- The INTERSECT operator returns only rows that appear in both input sets, returning distinct rows.
- The EXCEPT operator returns the rows that appear in the first set but not the second,



returning distinct rows.

Combining Sets

In this unit, you will study how to combine result sets using UNION, UNION ALL, INTERSECT, and EXCEPT. You will also study how to retrieve information from multiple tables using join queries and subqueries.

Outcomes

After completing this unit, you should be able to:

- Write queries that combine result sets.
- Write a two-table INNER JOIN.
- Write a three-table INNER JOIN.
- Write a subquery.
- Write an outer join.

Course outcome(s) practiced in this unit:

IT526-1: Create queries to retrieve data.

What do you have to do in this unit?

- Complete the assigned Reading.
- Participate in Seminar or complete the alternative assignment.
- Participate in the Discussion.
- Complete the unit Assignment.

Reading

The Reading for this unit will be from Chapter 4 and from supplementary material posted to Doc Sharing. It is important for your learning to read the material in the suggested order.

First, read Chapter 4 Lesson 3, pages 136–144. This lesson covers the UNION, UNION ALL, INTERSECT, and EXCEPT operators, that are used to combine two or more result sets into one result set.

Next, read the document "How to Write an INNER JOIN Query" in Doc Sharing. After reading that, read Chapter 4 Lesson 1 (pages 101–108), which covers Cross Joins and INNER JOINS, stopping before reading about Outer Joins.

Read the "Subqueries" document in Doc Sharing.

Finally, read the "Outer Joins" document in Doc Sharing. Then read Chapter 4 Lesson 1 (pages 108–117), which covers Outer Joins and Multi-join Queries.

Seminar



Attending live Seminars is important to your academic success, and attendance is highly recommended. The Seminar allows you to review the important concepts presented in each unit, discuss work issues in your lives that pertain to these concepts, ask your instructor questions, and allow you to come together in real time with your fellow classmates. There will be a graded Seminar in Units 1, 2, 3, 5, and 6 in this course. You must either attend the live Seminar or you must complete the Seminar alternative assignment in order to earn points for this part of the class.

Option 1

INNER JOINS, outer joins, and subqueries

The Unit 2 Seminar will cover:

- INNER JOIN queries
- Self-contained subqueries
- Multi-table INNER JOIN queries
- Correlated subqueries
- Outer join queries

The Seminar will emphasize topics covered in the supplementary documents in Doc Sharing. Bring your questions about that material and material from the textbook.

Option 2- Alternative Assignment:

You will benefit most from attending the graded Seminar as an active participant. However, if you are unable to attend you have the opportunity to make up the points by completing the alternative assignment.

Write a 200-word summary of the Seminar. Include at least one T-SQL code example. Your paper should be in APA format and cite all references used. Submit to the Seminar Dropbox.

Unit 2 Assignment

Course outcomes addressed in this activity:

IT526-1: Create queries to retrieve data.

Project Instructions:

Single-line-spacing is preferred to APA format.

Put your name, course, unit number, and date at the beginning of your Assignment Word document.

Before you begin, run the following two INSERT statements in TSQL2012:

-- This is the same as the insert on pg. 102 of the text.

-- Do not run this insert if you already have inserted this row into Suppliers.



```
INSERT Production.Suppliers
VALUES (N'Supplier XYZ', N'Jiru', N'Head of Security'
, N'42 Sekimai Mushashino-shi', N'Tokyo', NULL
, N'01759', N'Japan', N'(02)0 4311-2609', NULL );
GO
```

-- This inserts an unordered product into the Products table.

```
INSERT Production.Products
VALUES (N'Love Potion Number 9', 18, 2, 29.95, 0);
GO
```

Part 1. An INNER JOIN. (40 points; 4 points each part except 8 points for d)

Your boss wants to know which orders (TSQL2012 database) have shipcity (where the order is shipped to) *not* the same as the customer's city. He wants the report to show the orderid, orderdate, custid, company name, shipcity, and customer's city.

Please letter your answers to correspond to the letter of the question you are answering.

- a. What columns are needed for this query?
- b. What tables are the needed columns in?
- c. Are the tables directly connected? If so, what is the join column?
- d. Write the FROM ... JOIN ... ON ... clause of this query.
- e. Write the WHERE clause of this query.
- f. Write the entire query, including the SELECT clause.
- g. Run the query. Show the column headers and at least the first row of the result set. How many rows are returned?
- h. What is the same for all rows returned by the query?
- i. What single-table query would you write to determine whether there are any orders that have a different shipcity for that customer? Are there any?

Part 2. Queries. [Use the TSQL2012 database unless otherwise stated.] (60 points; 6 points each for a, b, c, e, and g; 7 points each for d, f, and h; 9 points for i)

Label each query with its letter. Copy the query from the SQL Server Management Studio query window into your Assignment Word document. Then copy the results into the Word document. If the result set returns more than 10 rows, copy the first five rows of the result set into your Word document and then state how many rows were returned.

- a. What cities are common to Suppliers and shipped orders? The result set should have column title City.



- b. The TSQL company wants to generate a mail merge database to send holiday cards to all customer and supplier contacts. Create a list giving contact name, company name, address, city, region, postal code, and country for customers and suppliers.
- c. In which city is the supplier of product Love Potion Number 9 located? Use a subquery.
- d. List the empid, last and first names, and title of employees who made sales in May 2008. Use a join query. List each employee only once.
- e. List the empid, last and first names, and title of employees who made sales in May 2008. Use a subquery.
- f. List empid, last and first names, orderid, orderdate, custid, and companyname for orders placed in May 2008. Use a join query.
- g. List the productid and productname of products that have never been ordered. Use a subquery.
- h. List the productid and productname of products that have never been ordered. Use an outer join.
- i. List the productid and productname of products that were not ordered in May 2008.

**Review the grading rubric below before beginning this activity.
100-point Assignment grading rubric**

Project Requirements / criterion	Points Possible	Points earned by student
1.INNER JOIN: a, b, c, e, f, g, h, i are 4 points each d is 8 points Lack of clarity in labeling answers (up to -5 points)	0-40	
2.Queries: a, b, c, e, g – 6 points each d, f, h – 7 points each i – 9 points Each result set is worth up to 2 points	0-60	
Total (Sum of all points)		
Points deducted for spelling, grammar, and/or APA errors.		
Adjusted total points		