# Joins and Set Operators

Exercise Handout

Learn. To Change.

## Contents

## Overview

In this set of exercises, you will first work with the Suppliers, Customers and Employees tables from the Northwind database to produce a company-wide telephone directory.

You will then work with the data from the QATSQLPLUS database to run various delegates queries.

### Objectives

At the end of this lab, you will be able to write queries that combine the results of multiple other queries using the set operators UNION, UNION ALL, INTERSECT and EXCEPT.

### Setup: Launch SQL Server Management Studio (if necessary)

1. Launch the virtual machine.
2. Launch SQL Server Management Studio.
3. Connect to the server.

## Exercise 1: Create a telephone directory

Northwind Traders want to create a centralised telephone directory for all suppliers, customers, and employees.

In this exercise, you will use the UNION operator to combine results.

The main tasks for this exercise are as follows:

1. Write a query that retrieves the company name, contact name and telephone number of all customers.
2. Write a query that retrieves the same columns for all suppliers.
3. Write a query that retrieves the full name and extension number for all employees.
4. Combine the results using a UNION operator.

### Task 1: Write a query that retrieves Customer contact details

1. Create a new query and save it with a name of "ContactDirectory.sql".
2. Write a query that uses the Northwind database and selects the CompanyName, ContactName and Phone columns from the Customers table.
3. Execute the query and verify that it returns 91 rows.

### Task 2: Write a query that retrieves Supplier contact details

1. In the same script file, add another query that selects the same three columns from the Suppliers table.
2. Select only that part of the script and execute the query. Verify that 29 rows are returned.

### Task 3: Write a query that retrieves Employee contact details

1. In the same script file, add another query that selects rows from the Employees table.
2. In the select list for the query, add a column that concatenates the FirstName column to a space character and the LastName column (building up a string containing the employee's full name).
3. Add the Extension column as well.
4. Execute the query and verify that you retrieve 9 rows of two columns – one containing the full name and one containing the extension.

## Task 4: Combine the results using the UNION operator

1. In the same script file, between the customers query and the suppliers query, add the UNION keyword.
2. Select both parts of the script and execute. Verify that you get 120 rows back. Scroll through the results and note that they are sorted alphabetically by company name.
3. Now add the UNION keyword between the suppliers query and the employees query and attempt to run the whole script.
4. You get an error, because the employees query only has two columns in it whilst the other two have three columns.
5. You need to add a company name column to the employee's part of the query. Now, the Employees table doesn't have a company name column but that's not a problem because we actually know what company all of the employees work for – Northwind!
6. At the start of the select list for employees, add the string 'Northwind Traders'.
7. Run the whole query again and verify that 129 rows are returned. Scroll through the list and note that the Northwind Traders employees are all floating around in the middle of the results.
8. Change the UNIONs to UNION ALL and rerun the query. This time all the employees are right at the very end. Also, although you probably won't have noticed, the query has run considerably quicker than before as it hasn't tried to remove duplicates by sorting first.
9. Feel free to add an order by clause if you want to.

## Exercise 2: Course Overlap

In this exercise, you will work with the QATSQLPLUS database to find all delegates who have attended both the QATSQL and QATSQLPLUS courses.

The main tasks for this exercise are as follows:

1. Create a query that returns the list of DelegateID who have attended the QATSQL course.
2. Create a query that returns the list of DelegateID who have attended the QATSQLPLUS course.
3. Find the list of DelegateID which appear in both of the original queries.

### Task 1: Create the QATSQL delegates query

1. Write a query to join the DelegateAttendance, CourseRun and Course tables. The only columns returned should be DelegateID where the CourseName = QATSQL.
2. Test the query. This should return 11 rows.
3. Keep the query window open for the following tasks.

### Task 2: Create the QATSQLPLUS delegates query

1. Write a query to join the DelegateAttendance, CourseRun and Course tables. The only columns returned should be DelegateID where the CourseName = QATSQLPLUS.
2. Test the query. This should return 7 rows.
3. Keep the query window open for the following tasks.

### Task 3: Create a query showing the overlap

1. Using the queries from tasks 1 and 2, create query to return the list DelegateID appearing in both sets.
2. Test the query. This should return 6 rows.
3. Keep the query window open for the following tasks.

## Exercise 3: Course Disjoint

In this exercise, you will work with the QATSQLPLUS database to find all delegates who have attended the QATSQL course but have not attended the QATSQLPLUS courses.

The main tasks for this exercise are as follows:

1. Find the list of DelegateID which have attended QATSQL but not QATSQLPLUS.
2. Find the list of DelegateID which have attended QATSQLPLUS but not QATSQL.

### Task 1: Create a query showing the disjoint

1. Using the query from exercise 2 task 3, update the query to return the list DelegateID attending QATSQL but not QATSQLPLUS.
2. Test the query. This should return 4 rows.
3. Update the query to return the list DelegateID attending QATSQLPLUS but not QATSQL.
4. Test the query. This should return 1 row.

QA

Learn. To Change.

QA.com