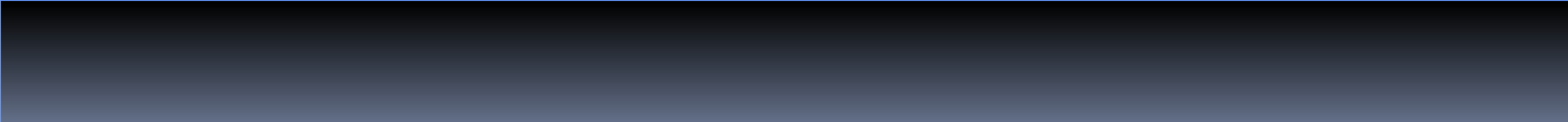
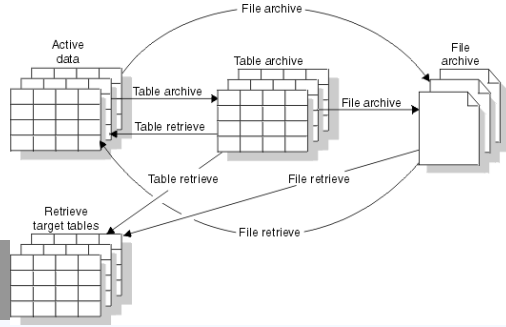
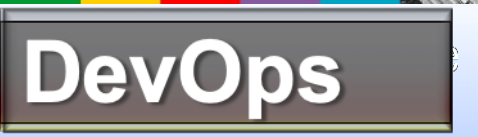


IBM Software Group

IBM Developer for z Systems – for ISPF Developers

Module 8 – Using the Data Source Explorer





Jon Sayles, IBM - jsayles@us.ibm.com

IBM Trademarks and Copyrights

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

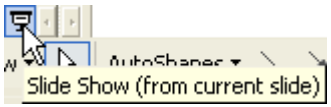
IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM Rational products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

Learning DB2 and SQL

- Many (in the thousands of) books exist that do an excellent job teaching SQL.
- Additionally, sites exist on the Internet (GOOGLE: "SQL tutorials" – or "Learn SQL") for online (and typically free) education.

- IBM Also supplies *excellent* SQL and DB2 documentation:
  - DB2 Documentation
  - SQL Getting Started
  - SQL Reference Manual
  - Message (error code) Reference. Cached pdf version of full guide.
  - DB2 Application Development Guide with example embedded SQL programs.
  - Triggers in DB2
  - Constraints in DB2
- And there are plenty of non-IBM sites to learn about SQL:
  - Http://en.wikipedia.org/wiki/SQL

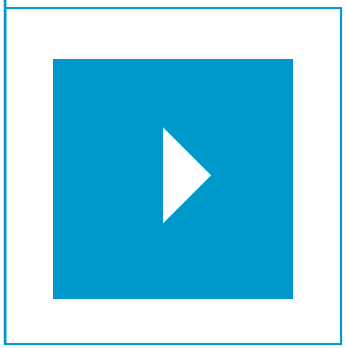
- Note: In order for you to get the above links to work, view the PowerPoint in Slide Show mode.



- Here is an example of COBOL database access:
  - http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=/com.ibm.db29.doc.apsg/db2z_samplecoboldrdathreepartnames.htm

UNIT

The IDz Workbench



Topics:

- The Data Perspective and connecting to DB2
- Understanding your DB2 objects
- Editing and managing DB2 table data
- **Coding and testing SQL**
- Extract/Load and Managing Test Data

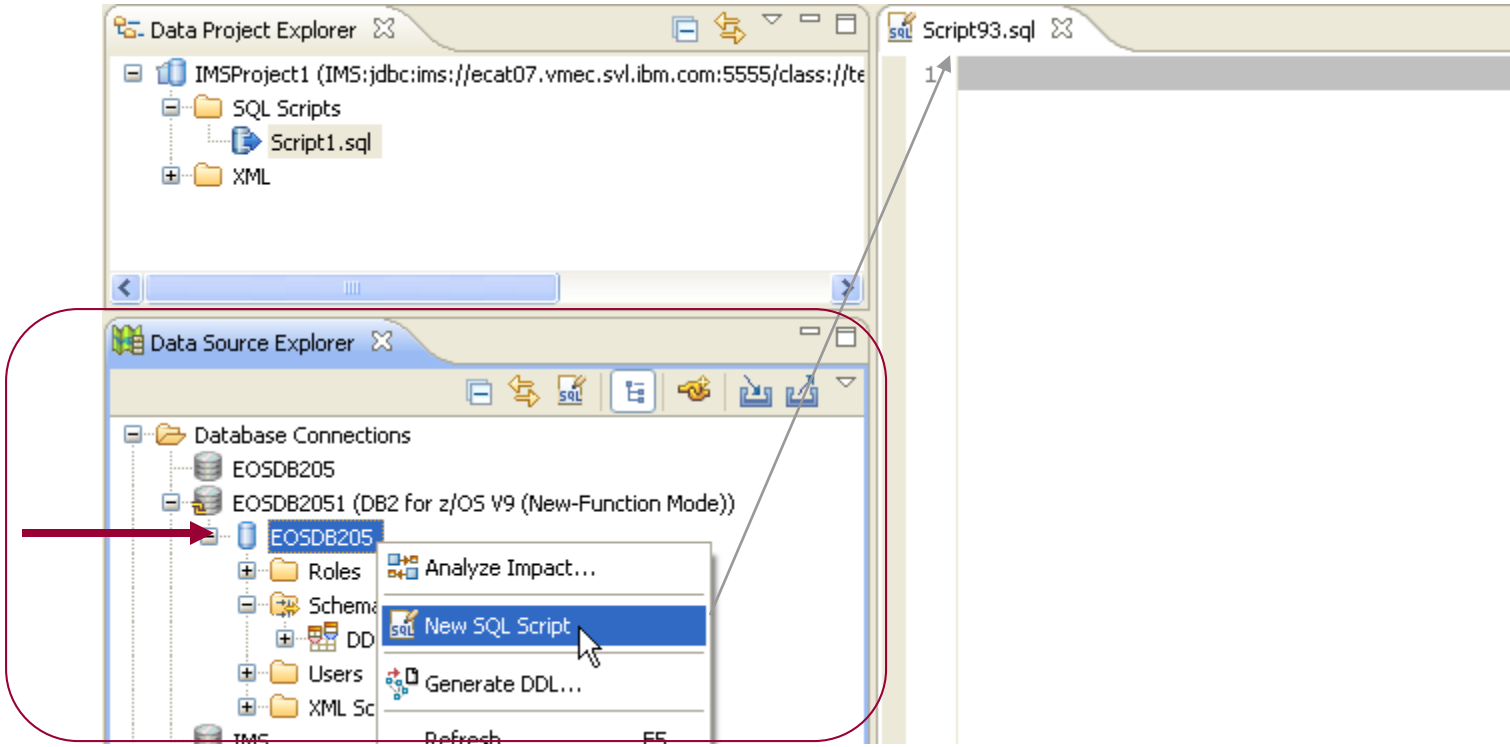
Code Interactive SQL Statements – SQL Script

Similar to SPUFI – you use a SQL Script file to edit and test your SQL. A SQL Script is a text file that contains interactive SQL statements – with or without host variables

These statements can be tested in the Data Perspective before embedding them in COBOL data access functions. For complex SQL logic, this is recommended “best practice”.

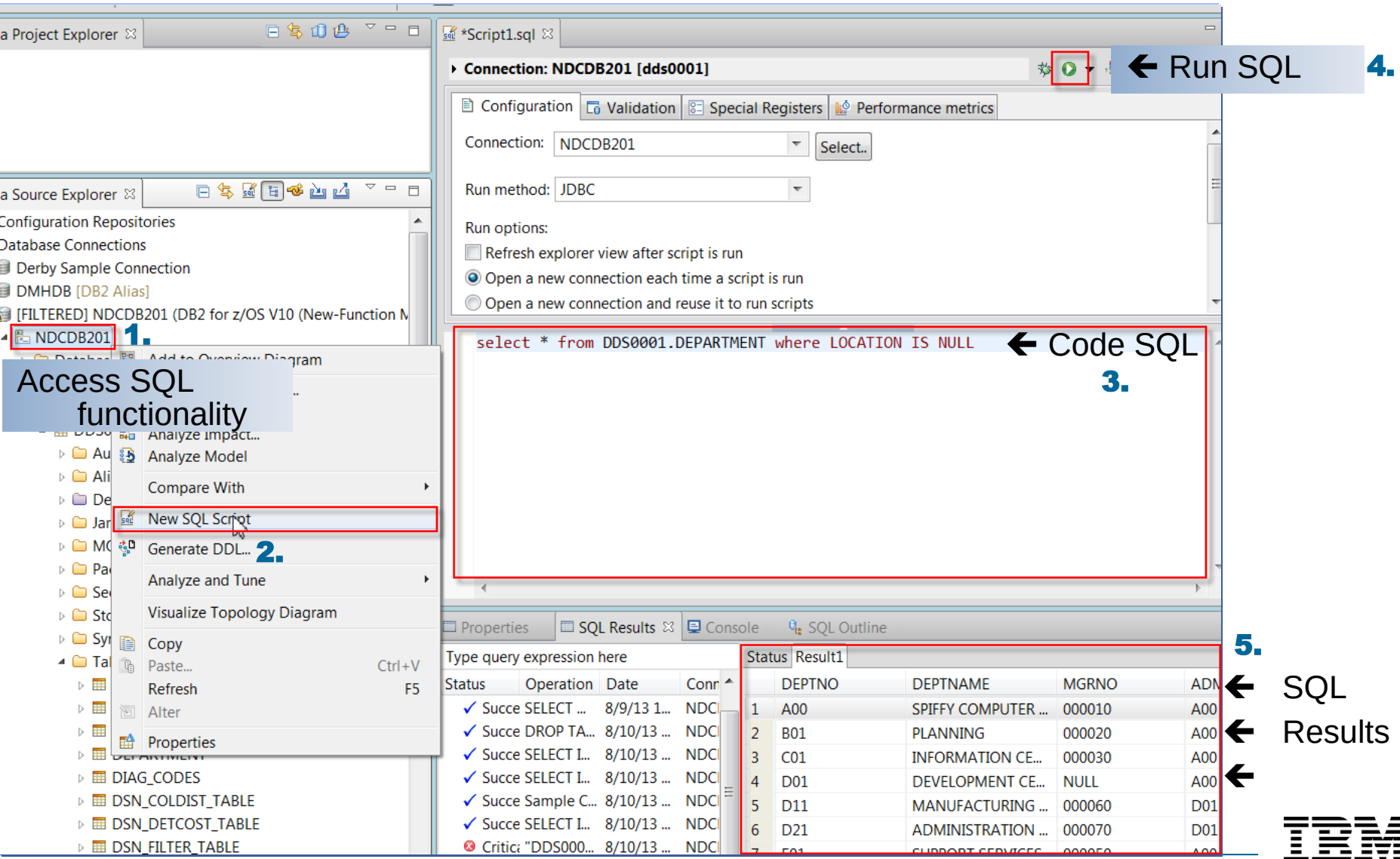
To open and work with a SQL Script:

- From the Data Source Explorer
- Right-click over the DB2 system icon shown and select: **New SQL Script**



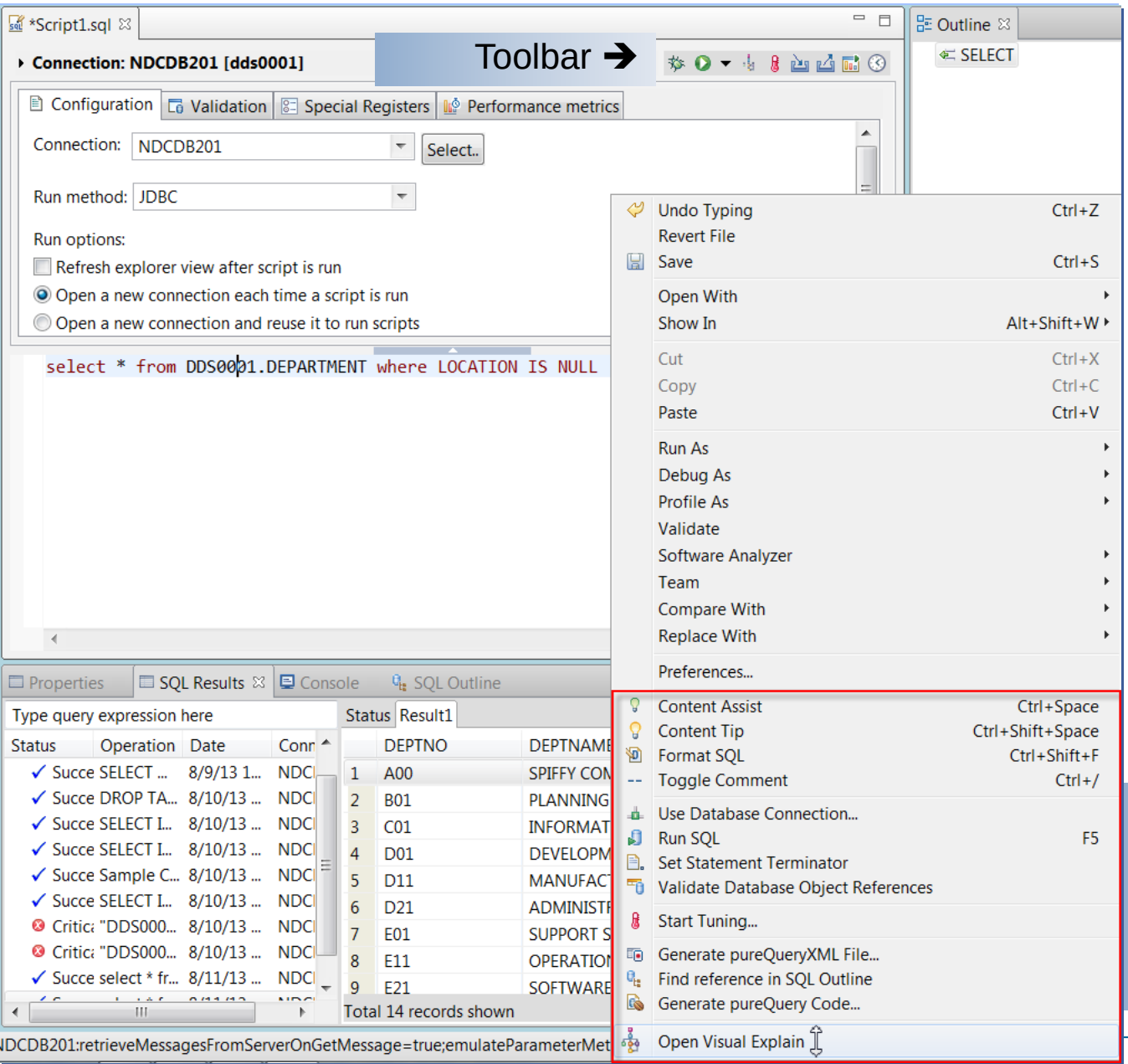
Code Interactive SQL Statements – SQL Script – w/Data Studio

Data Studio provides richer functionality, and there are many additional Context Menu options (next slide)



Code Interactive SQL Statements – SQL Script – w/Data Studio

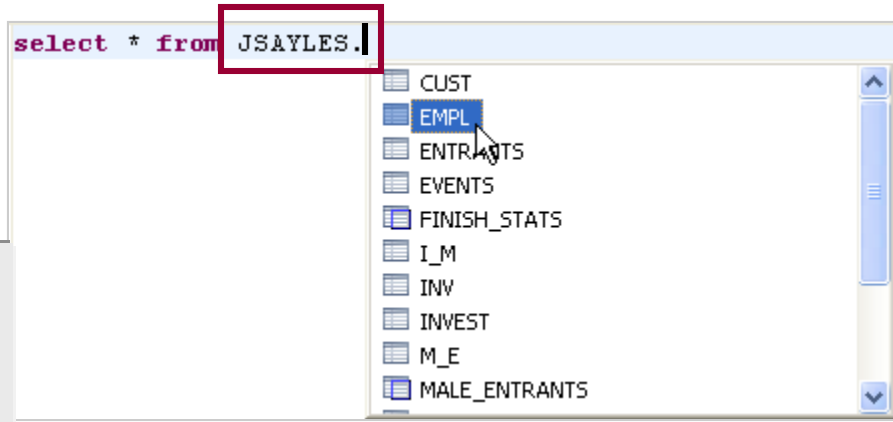
Additional SQL coding/testing/tuning options in Data Studio



Coding,
Formatting,
Testing
Tuning options

Create and Run a SQL Script Statement

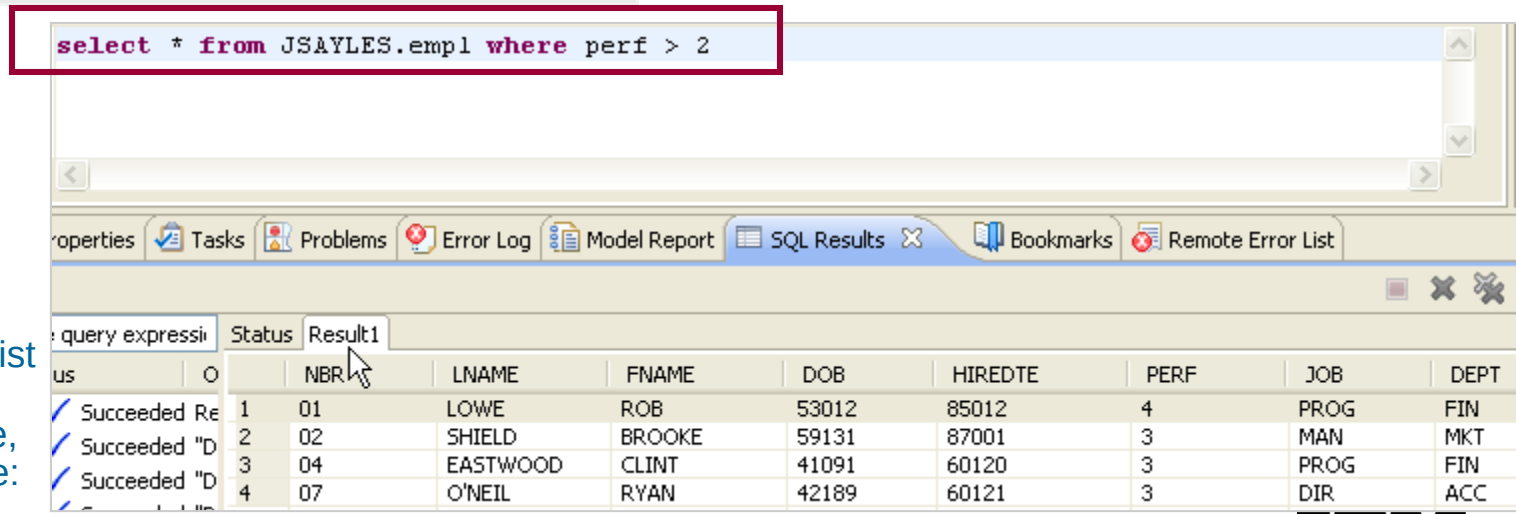
You can code your SQL statements using Content Assist – to get the table names and SQL keywords (Ctrl/Spacebar)



After you're done coding, right-click (context Menu) over the statement

Select Execute Current Text (or Execute All)

View the SQL Results



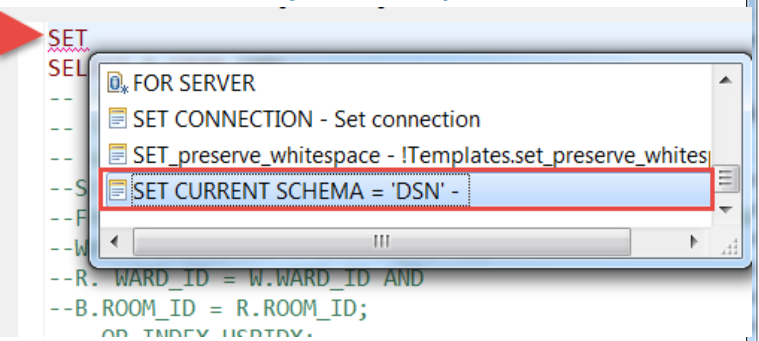
Note: For Content Assist to work, you must qualify the DB2 table, view, synonym name:

Schema.Tablename

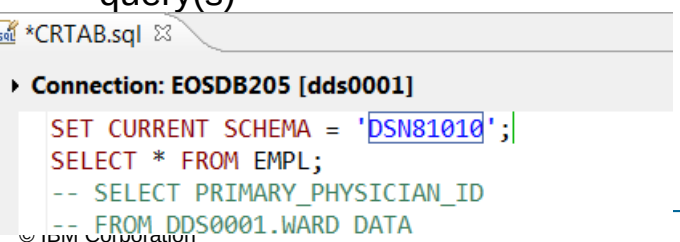
Set Current DB2 Schema

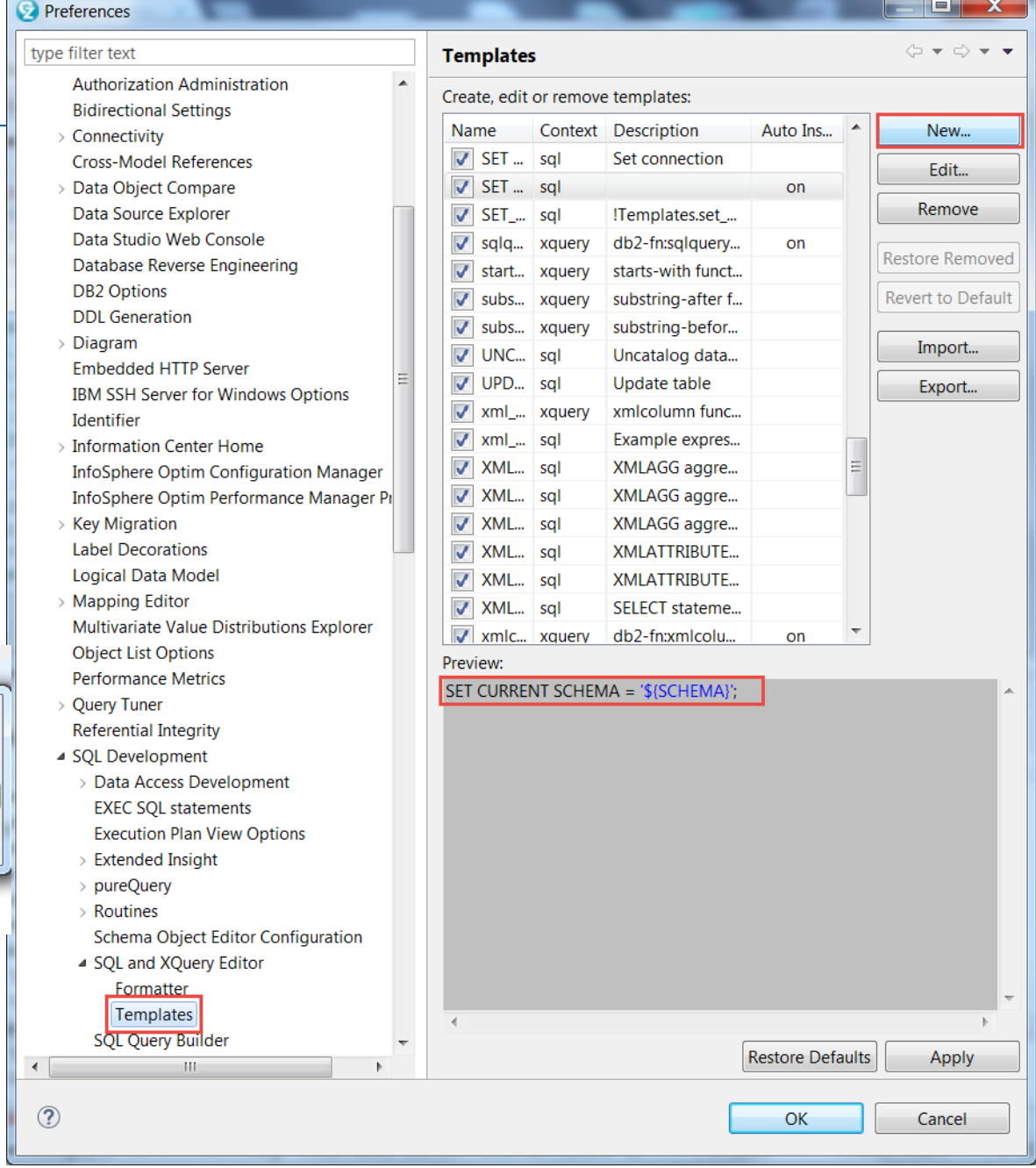
By default, IDz will assign your connection TSO ID as the Schema for your dynamic SQL statements. If you need a different schema name...

1. From Window ➔ Preferences, create a new SQL Editor Template to: **SET CURRENT SCHEMA = '${SCHEMA}';**
2. From a SQL Script editing session
   - Type: set
   - Press Ctrl+Spacebar
   - Select your Template



3. From the SQL Script editor type the Schema Name you want and run the query(s)





Create New SQL Statement Template

Not every SQL operator/expression is available as a Template

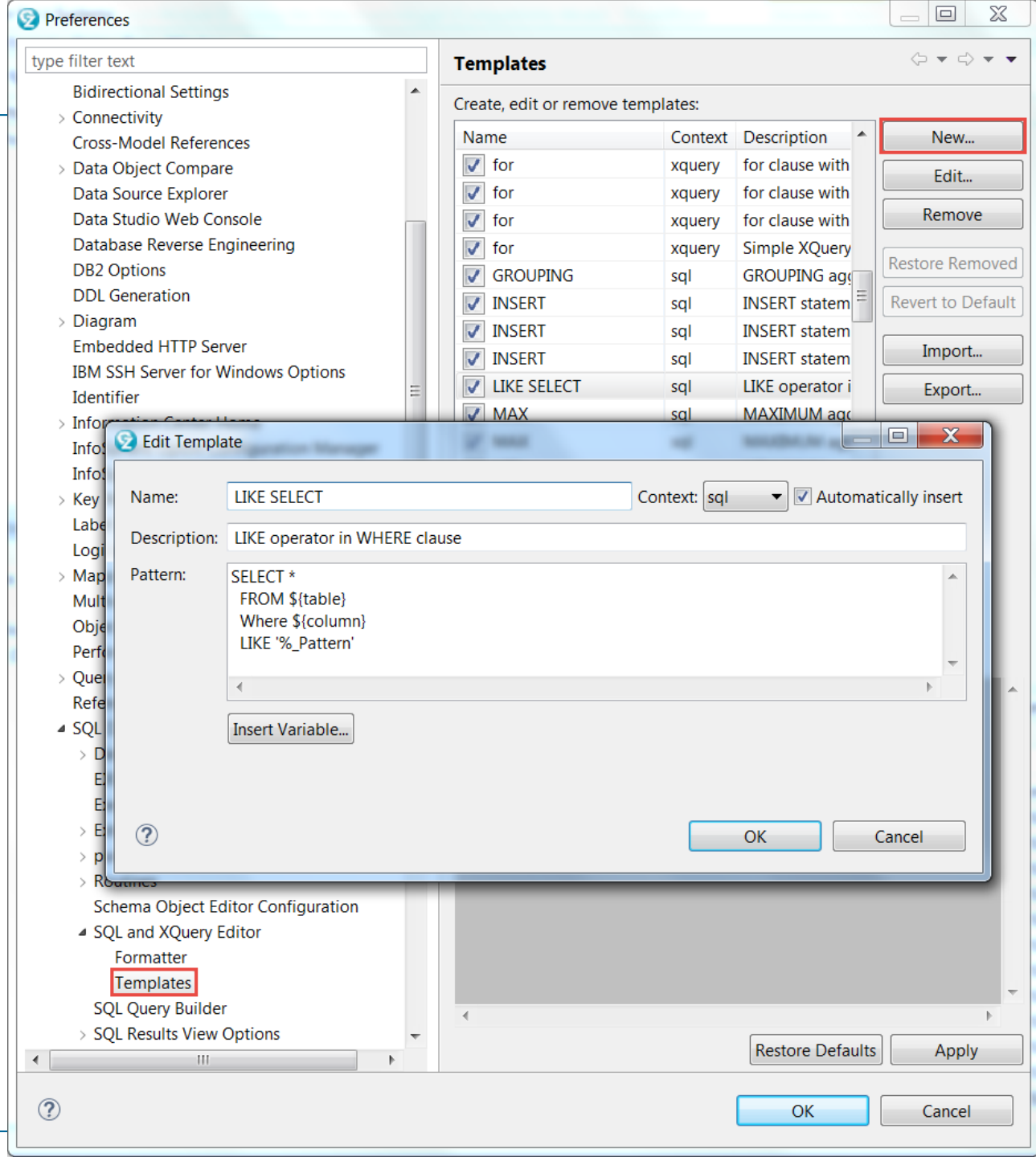
To create your own Template:

From Preferences ➔
Data Management ➔
SQL Development ➔
SQL and XQuery Editor ➔
Templates

1. Click New
2. Name the Template
3. Add a Description
4. Code the Pattern

${....} - becomes a fill-in-the-blanks variable during edit

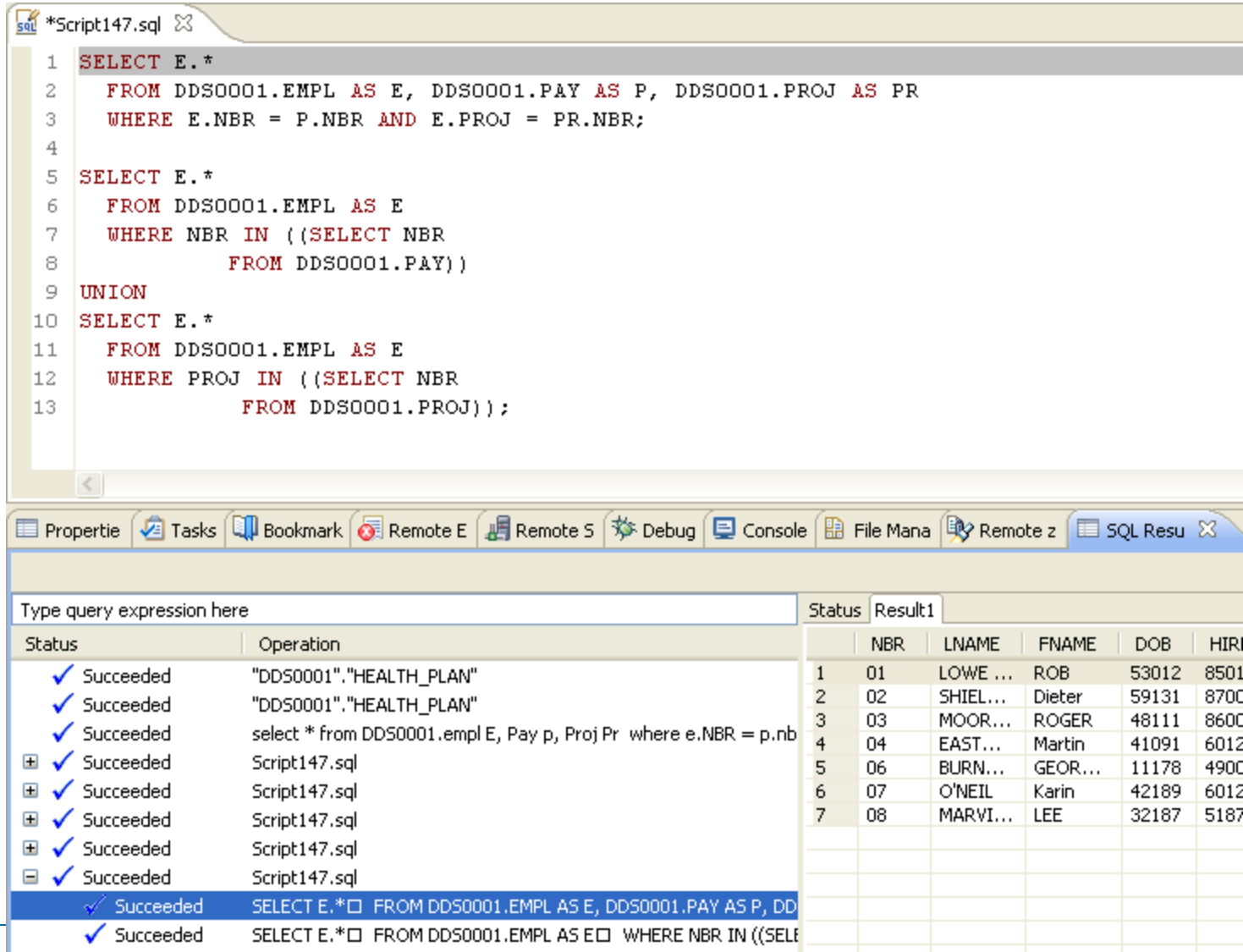
Content Assist (Ctrl+Spacebar) invokes the templates



SQL Query - **Editing and Running Multiple Statements** (Similar to SPUFI)

You can code and multiple SQL statements by ending each statement with a semi-colon (see screen capture):

- Expand the statement in SQL results.
- Each query will have its own results window



Notes:

Each query is treated as a separate unit of work.

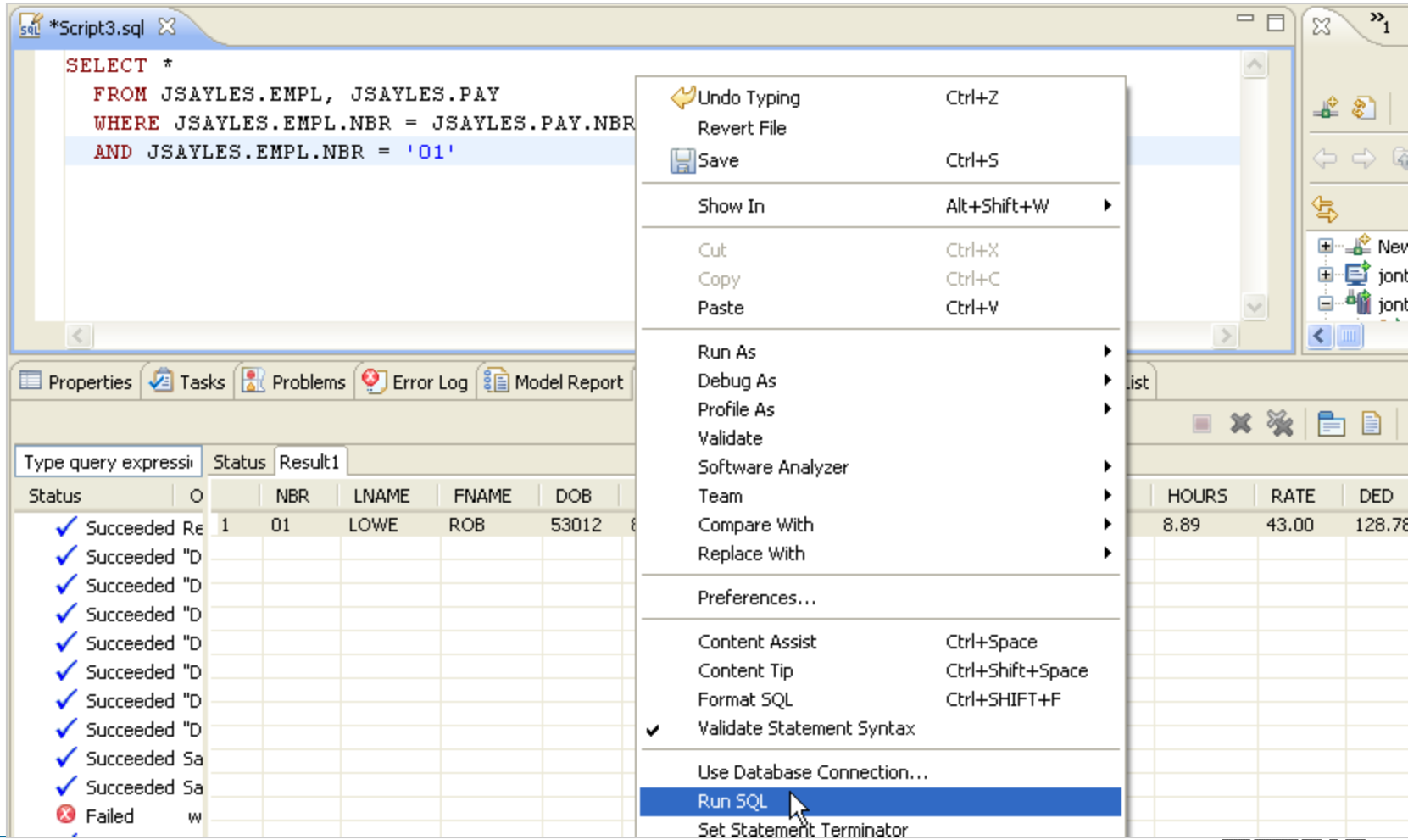
Successful updates are committed between statements

New SQL Script – Run SQL

When you are finished coding

1. **Right-click over the script area**
2. **Select Run SQL**
3. **Verify your results**

   Note that the Status will show details on your statement's execution



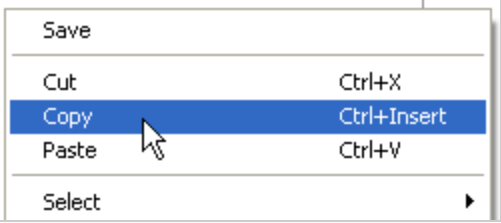
Working with Embedded SQL Statements - 1 of 3

You can use a SQL Query to test your COBOL/SQL statements out prior to testing them at the COBOL procedural logic level (note this saves both time and CPU resources)

- Steps:
  - From IDz, click back over to the **z/OS Perspective** (but do NOT close the Data Perspective)
  - Open: **cursravg.cbl**
  - Find the **100-DECLARE-CURSOR-RTN** paragraph shown below
  - Copy the `SELECT…` ➔ `GROUP BY DEPT` clauses as shown

Select and copy this

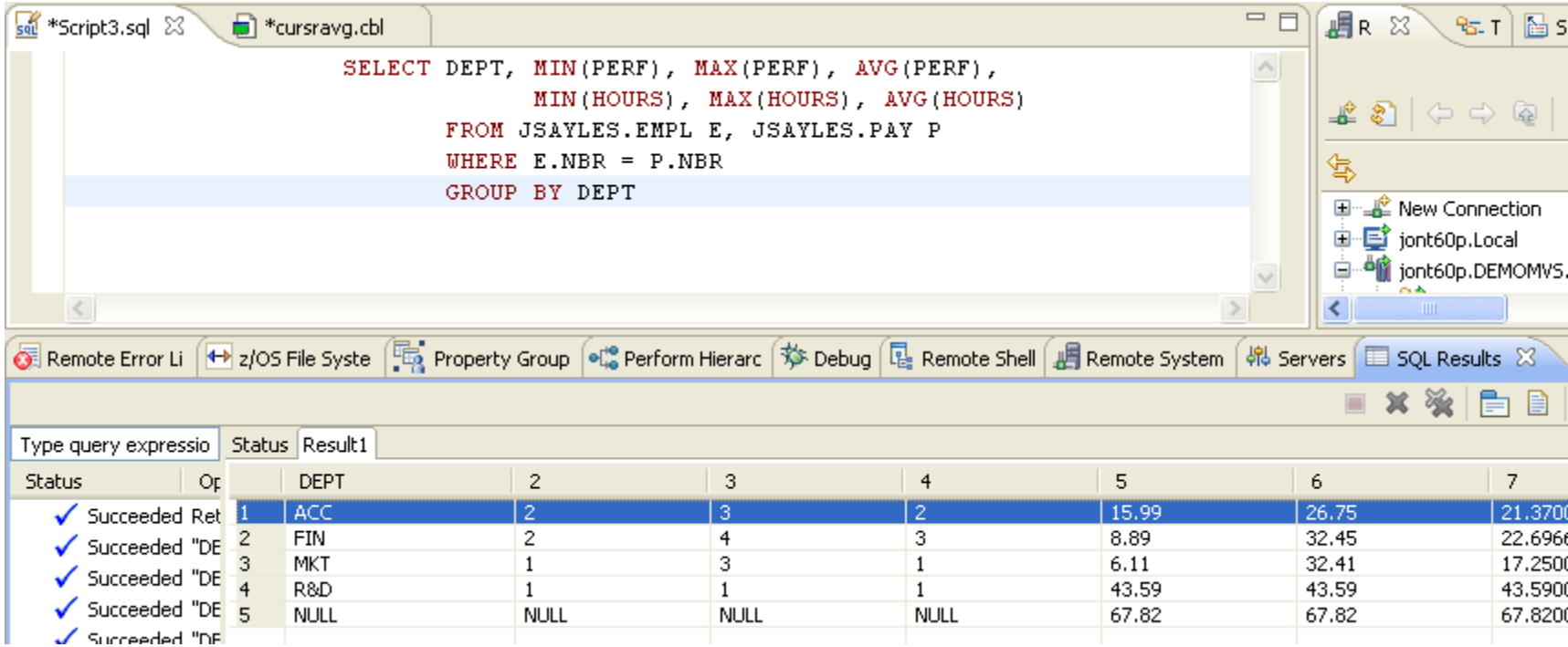
```
100-DECLARE-CURSOR-RTN.

*  THIS STATEMENT CREATES AN "ACTIVE SET", A GROUP OF ROWS
*  THAT WOULD BE THE OUTPUT FROM THE EXECUTION OF THE STATEMENT
*  IF YOU EXECUTED IT INTERACTIVELY.


* = = = > CODE THE SQL STATEMENT TO JOIN THE EMPL AND PAY TABLES
* = = = > GROUP THEM BE EMPL.DEPT AND DISPLAY THE DEPT AND:
* = = = > AVERAGE, MINIMUM AND MAXIMUM - HOURS AND PERF BY DEPT

    EXEC SQL
        DECLARE C1 CURSOR FOR
        SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PERF),
                     MIN(HOURS), MAX(HOURS), AVG(HOURS)
               FROM JSAYLES.EMPL E, JSAYLES.PAY P
               WHERE E.NBR = P.NBR
               GROUP BY DEPT
    END-EXEC.

100-EXIT.
    EXIT.

150-OPEN-CURSOR-RTN.
```



- You will test only the **interactive SQL** portion of your COBOL cursor declaration

Working with Embedded SQL Statements - 2 of 3

- From IDz, click back to the **Data Perspective**
- If you still have your other SQL Script page open, select and delete the existing statement, then Paste in the copied SQL cursor code
- Right-click and **Run the SQL** Statement and view results

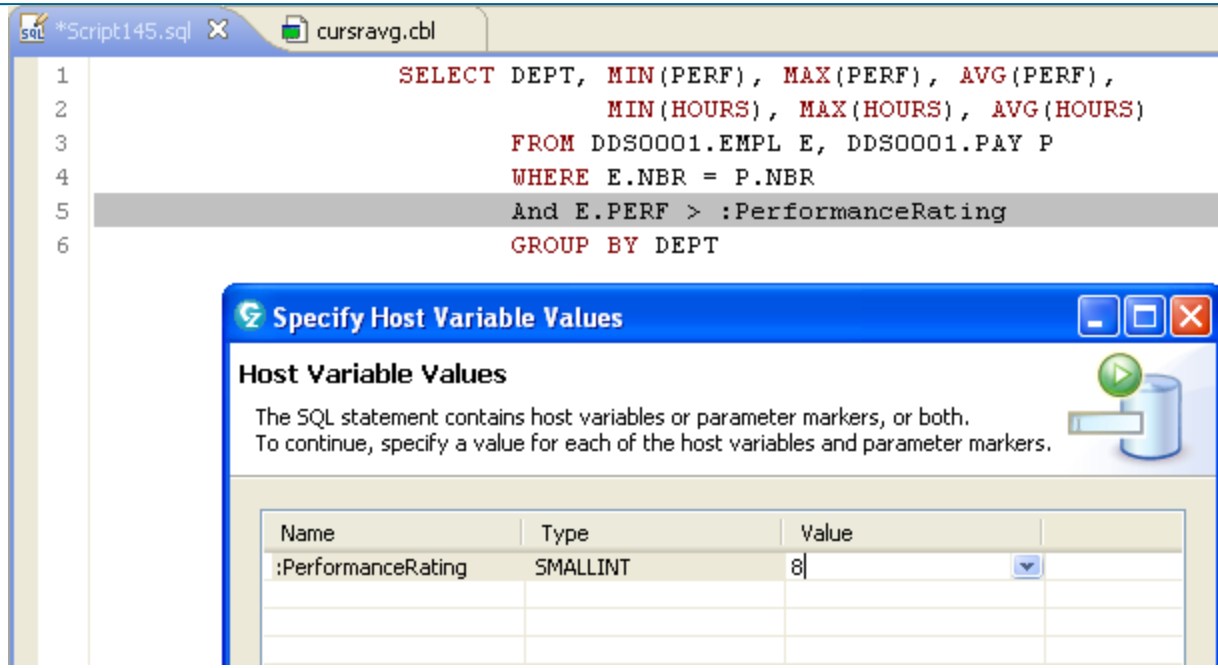


Note: As you did with the sample table contents, you can Right-click over the result rows, and Export the result data for data compare, reporting, etc.. This can be useful during DB2 application testing/debugging.

Working with Embedded SQL Statements – 3 of 3 (Host Variable Values)

- Most of the SQL statements embedded in your COBOL and PL/I programs will have host-variables
- In order to test such statements, you can:
  - "Stub the host variables out" – replacing the SQL statement operands with literals
  - Use the Host Variable Values wizard (see disclaimer for COBOL) –

    Specifying the Host Variable values as shown on the right



Disclaimer for COBOL Host Variable Values

- Host Variables with a dash in their name are not supported by the wizard.

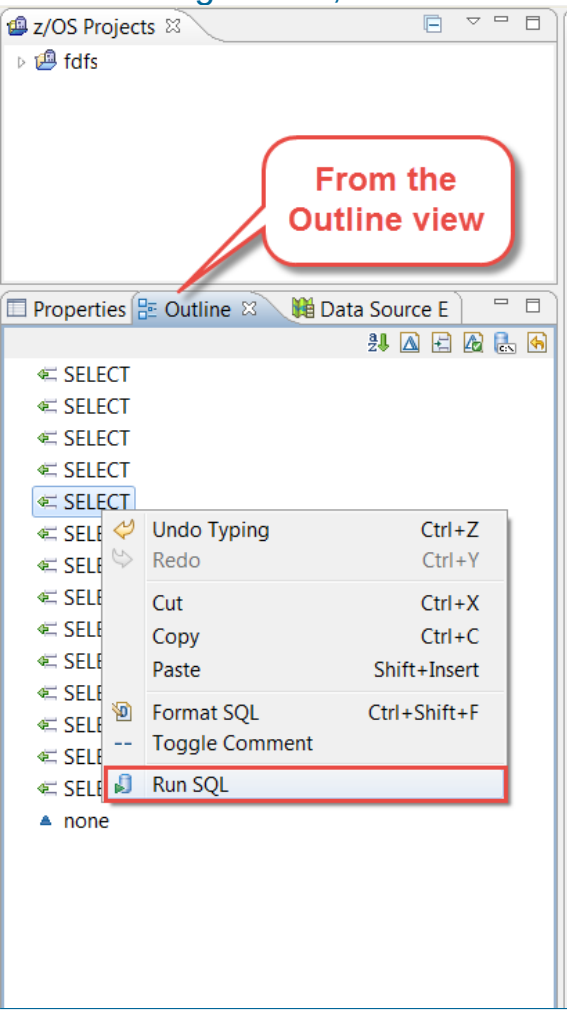
Example: :PERF-RATING

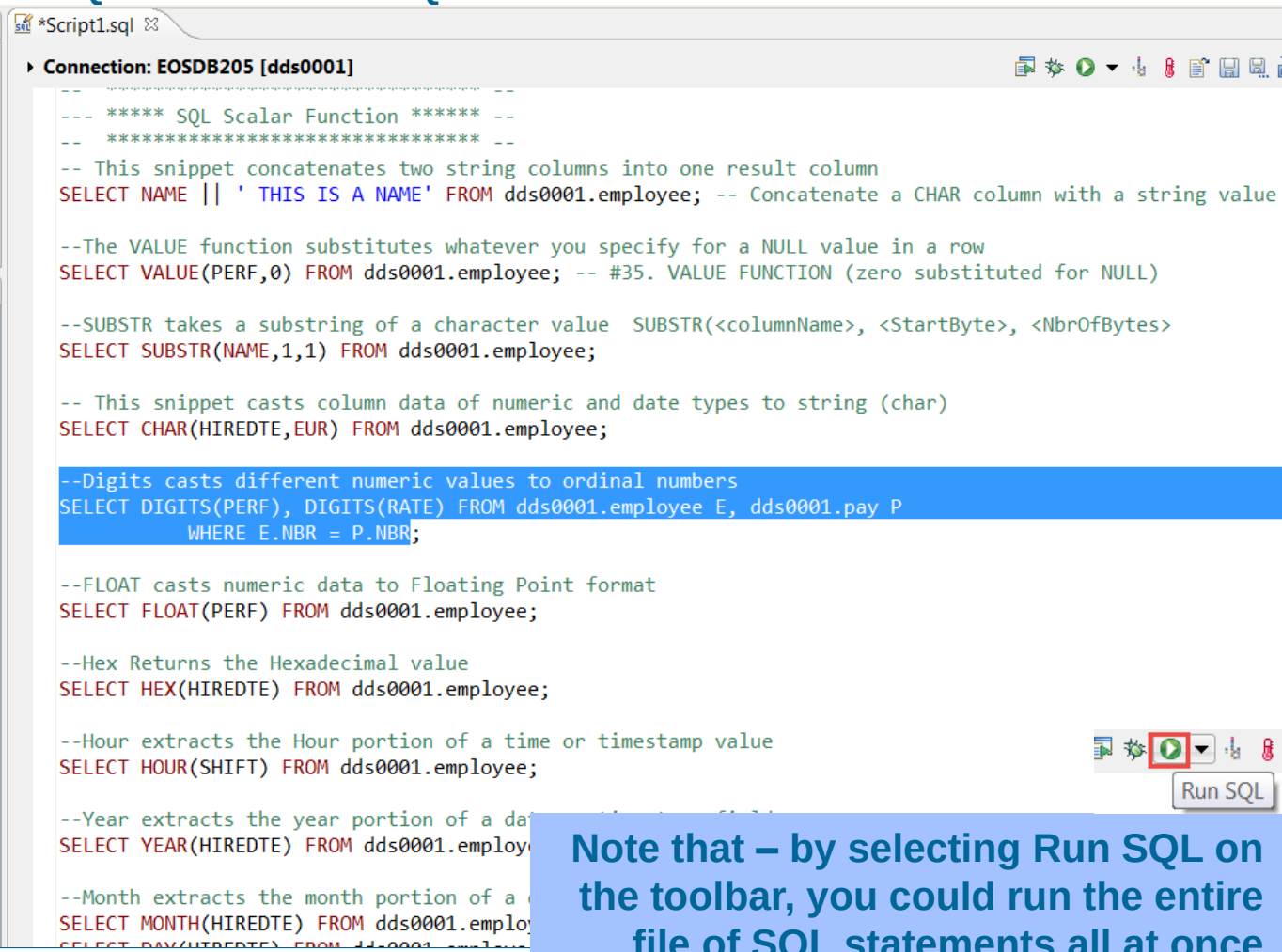
Qualified Host Variables such as: **:DCL.PERF are supported.**

PL/I Variables (separated by under-scores) are also supported

Running SQL From the Outline View

- If you have a file of SPUFI statements that you'd like to run with IDz:
  - Open the SPUFI file in a SQL editor
  - Open the **Outline View –** every single SQL statement will be listed in the view
  - **Click a statement** – to select a single statement from your file
  - Right-click, and select **Run SQL** – Check out the **SQL Results View**



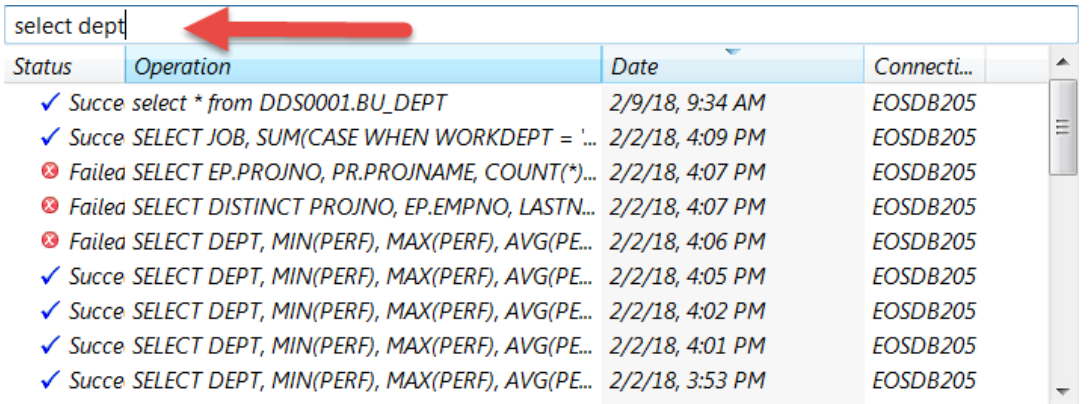


Note that – by selecting Run SQL on the toolbar, you could run the entire file of SQL statements all at once

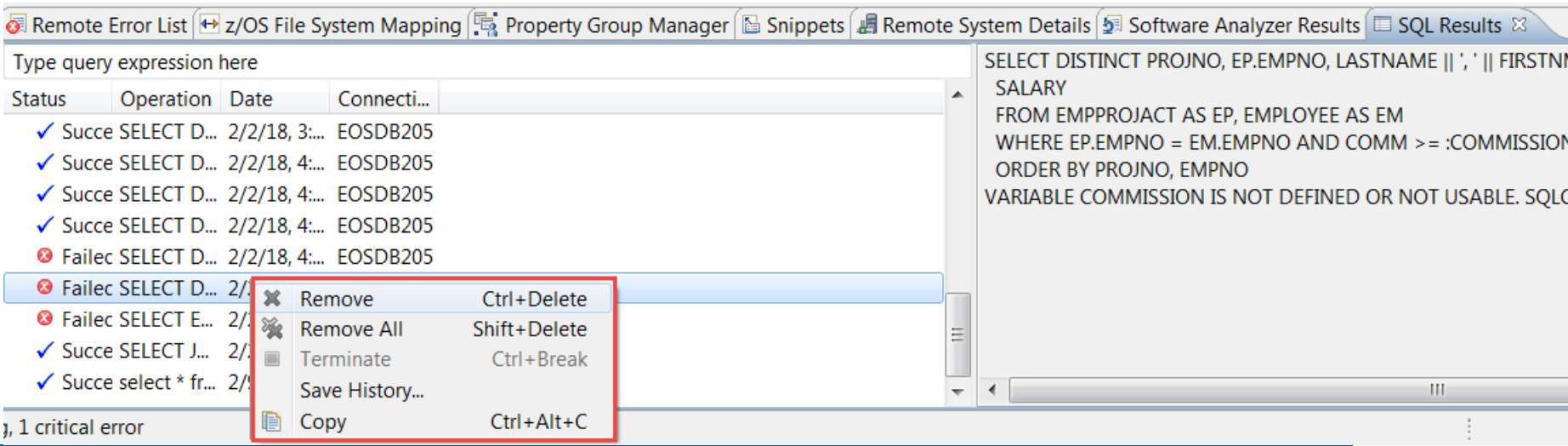
Cleaning up the scripts in SQL Results view

The SQL Results view has a number of useful actions for development/test

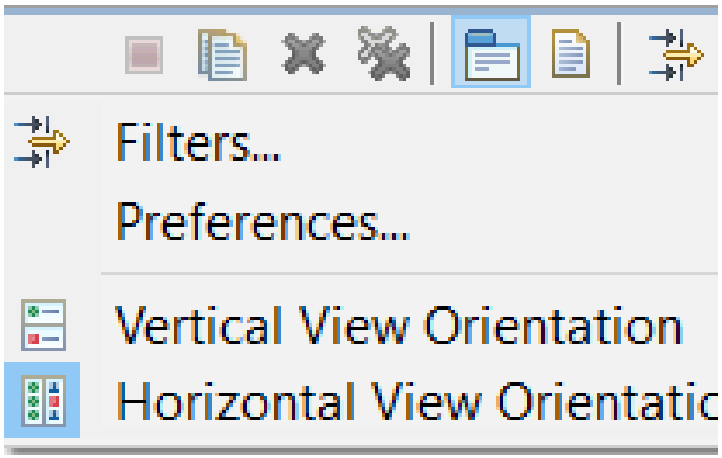
- Any statement and its results can be viewed by selecting it
- You can Filter the statement list



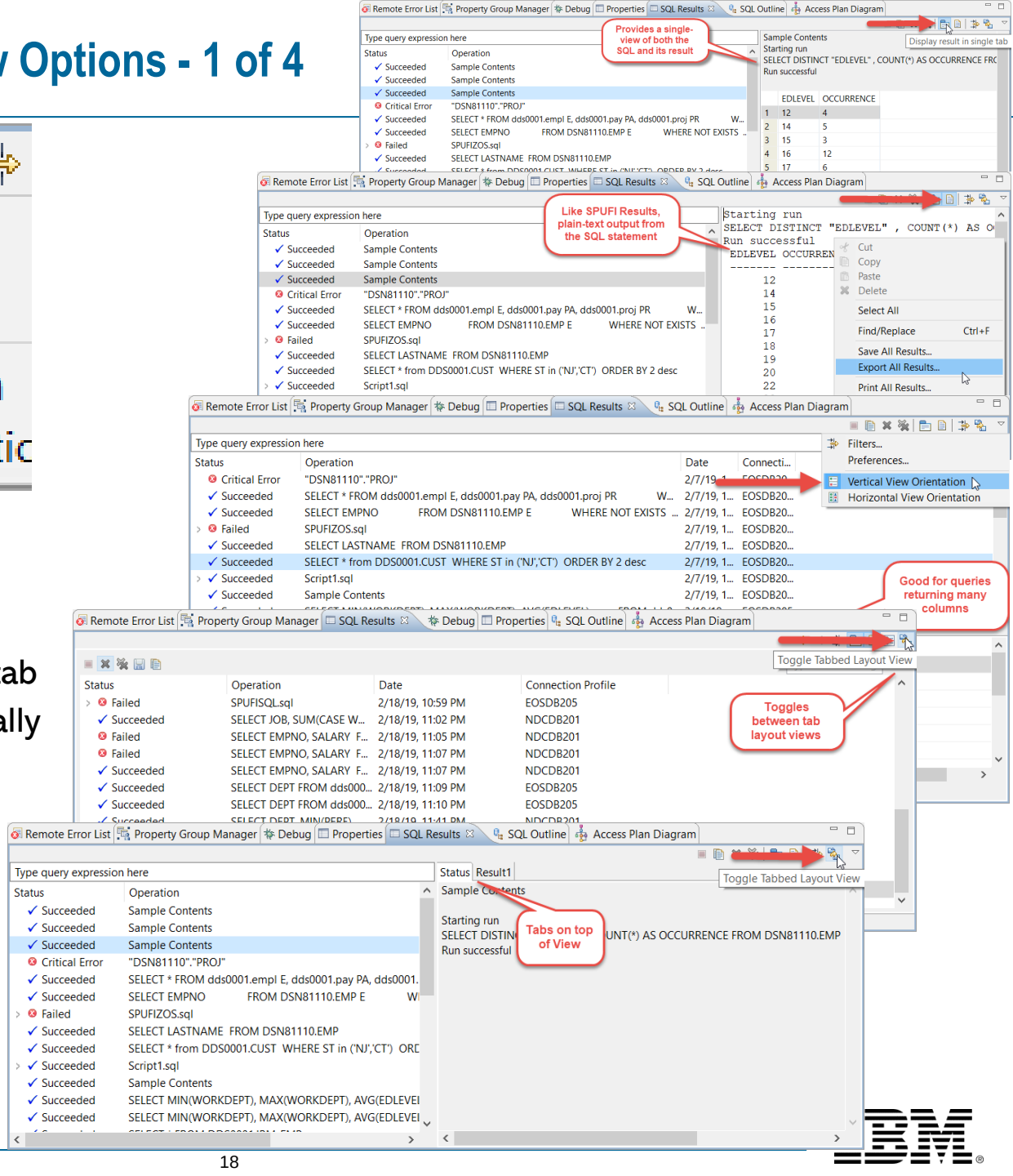
- You can Sort the list – by clicking on a column header (E.G. click on: Date)
- You can clean up (Remove), Copy & Paste the queries and Save History Query+Results to an external PC file



Additional SQL Results View Options - 1 of 4

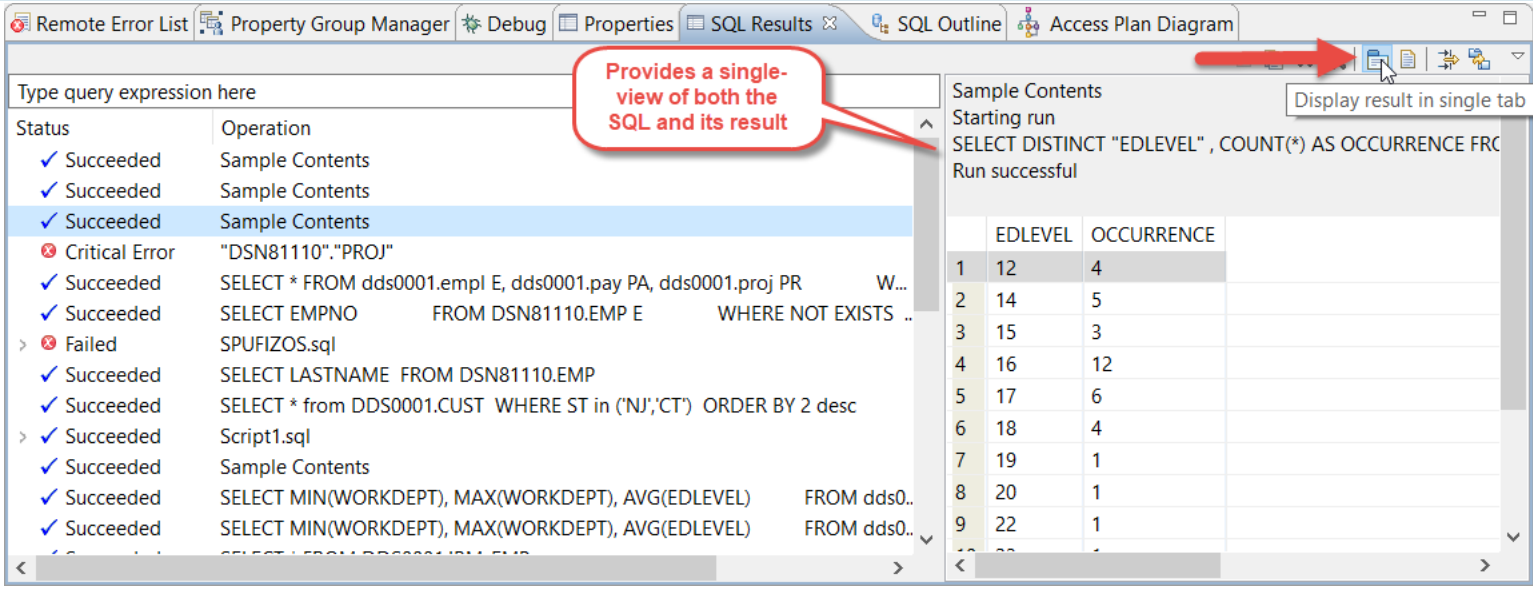


- Copy to clipboard
- Remove selected entry
- Remove ALL entries
- Display SQL & Results in single tab
- Display rows Vertically/Horizontally
- Display in Text Mode



Additional SQL Results View Options – 2 of 4

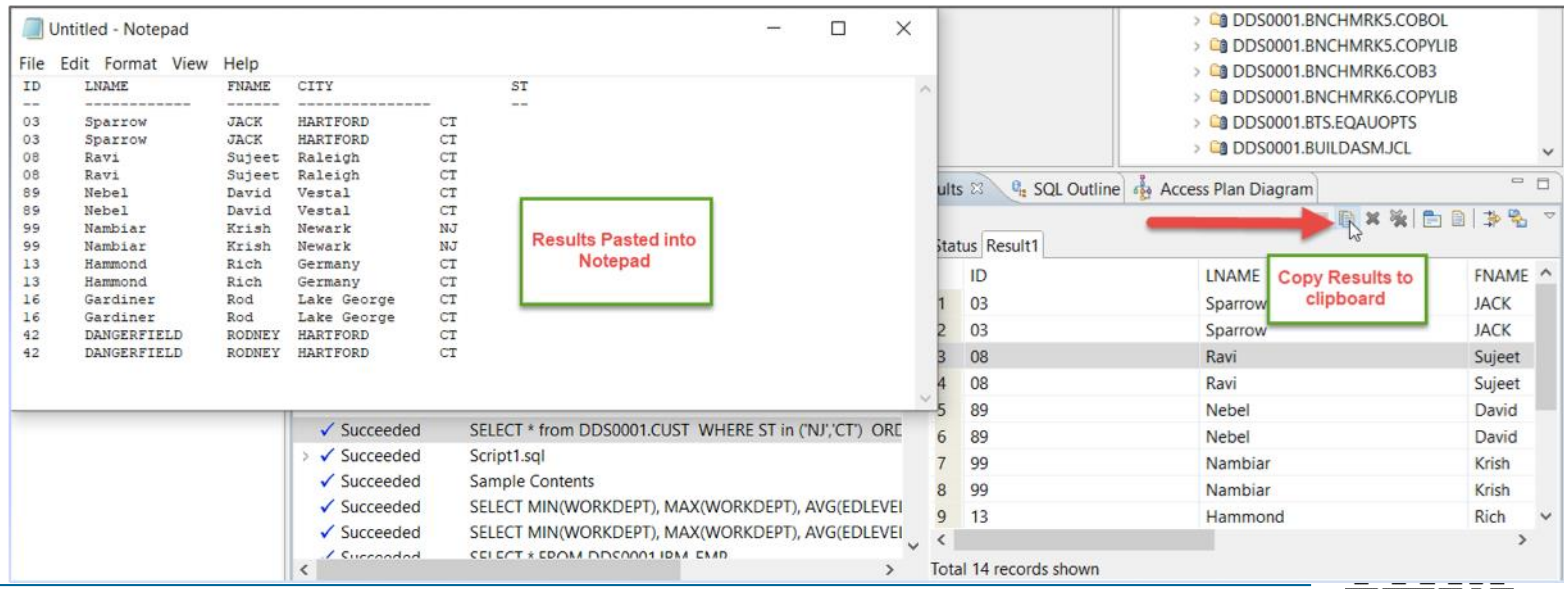
Display result in single tab…



Copy Results to Clipboard

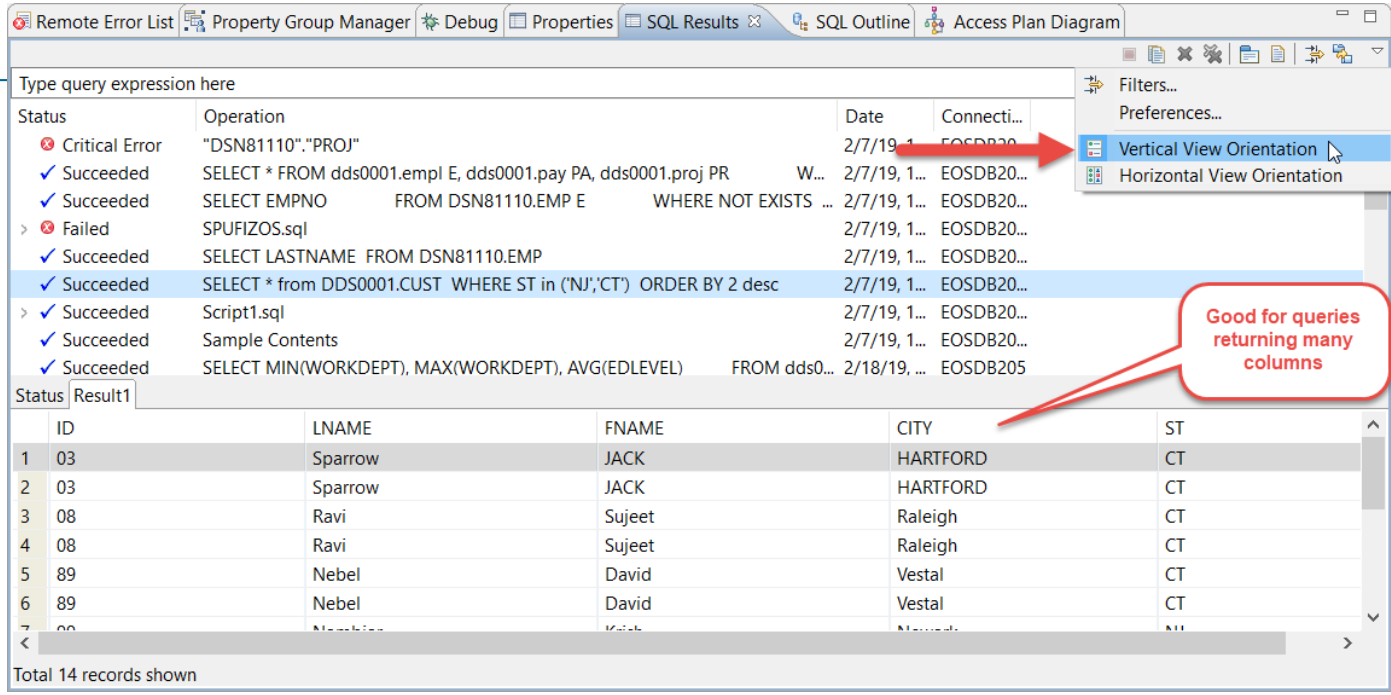
Paste into Windows file

- .txt
- .csv
- .doc
- etc.



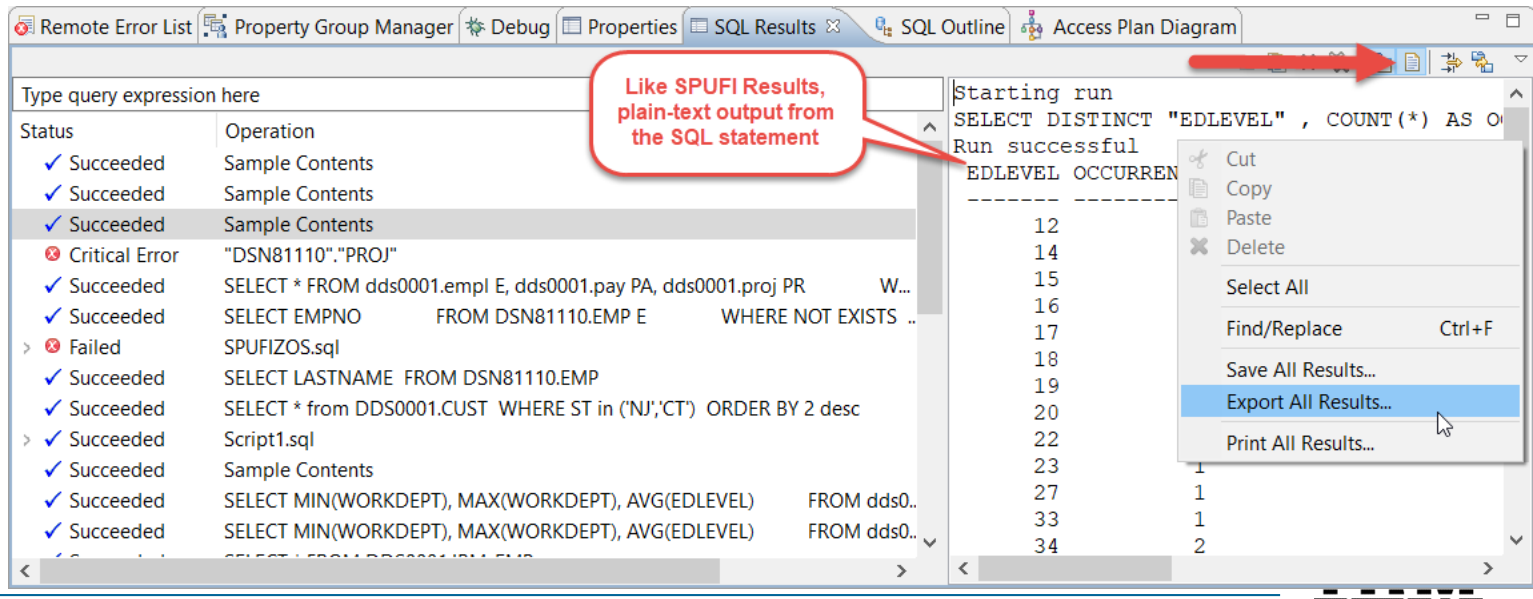
Additional SQL Results View Options – 3 of 4

Vertical Orientation



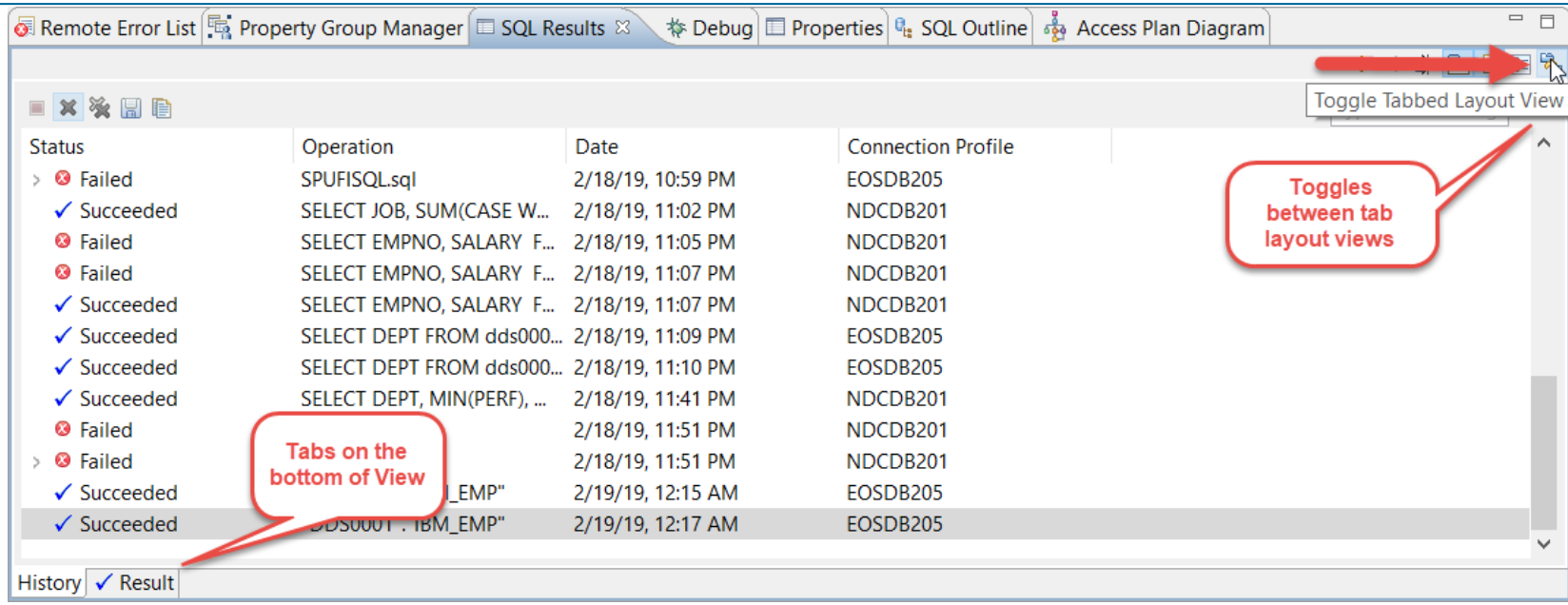
Display result in text mode…

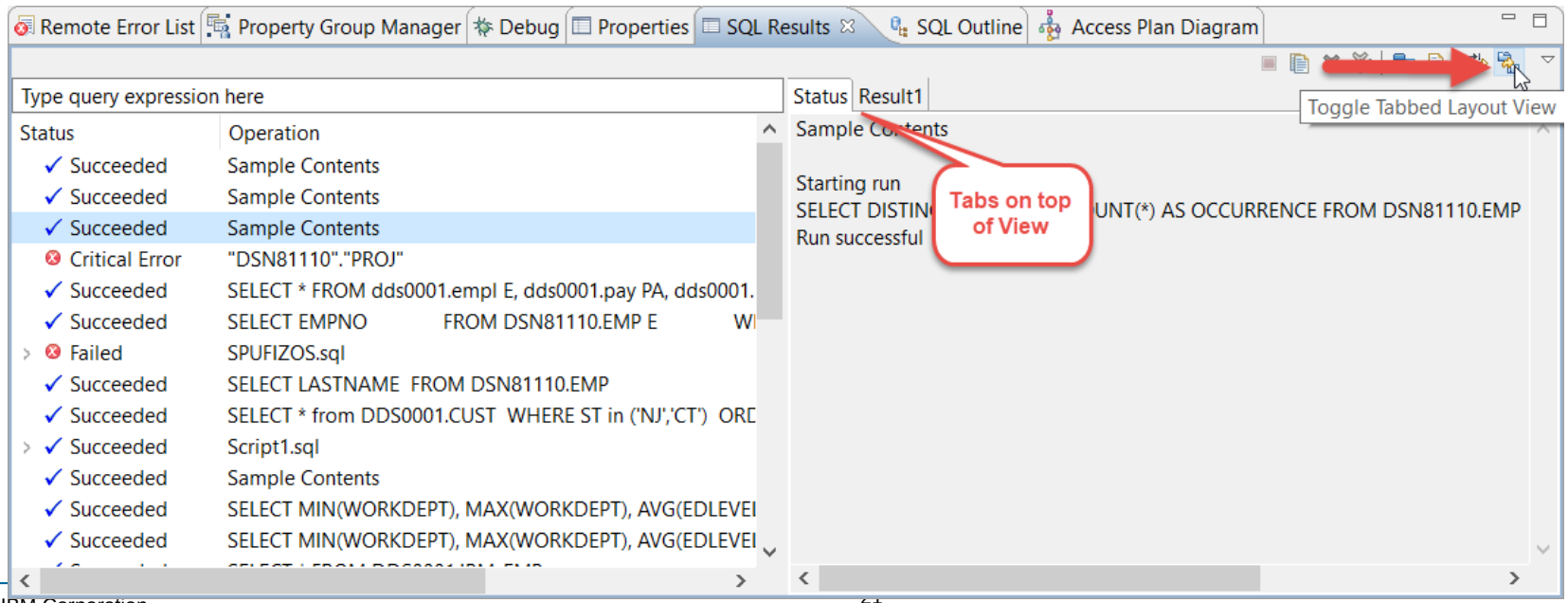
Note Context Menu **Export** and other actions available



Additional SQL Results View Options – 4 of 4

Toggle Tabbed Layout View





Review – "How do I" (Using IDz, be sure you know how to do the following…)

- Access the data tools in the IDz Workbench (how do you open the Data Perspective)?
- Create a connection to a DB2 region?
- Filter out Schemas you don't want to see in the Data Source Explorer?
- View sample rows from a table, view or synonym?
- Edit a DB2 table?
- Save changes to my row/column edits?
- List indexes for a DB2 table
- Generate the table definition (DDL) for a DB2 table or view?
- View the different values in a DB2 table column, and see the number of rows each value has in the table?
- Understand the relationships among a set of DB2 tables – connected with Primary/Foreign key constraints?
  - **i.e. Create an "Entity-Relationship Diagram"**
- List the primary or foreign key (constraints) for a DB2 table?
- Understand how an index on a table is designed (to see if it will help make my SQL query faster)?
- List the columns in a table or view?
- Change the default maximum number of rows displayed or edited in a table?
- Code a dynamic SQL statement?
- Invoke (use) Content Assist to help you code a SQL statement (what two keys??)
- Run a SQL statement?
- Code and test (run) multiple SQL statements in one batch (like SPUFI)?
- Perform (relative) SQL statement efficiency benchmarks?
- Export rows from SQL results to a: web page, XML file, Excel spreadsheet?
- "Explain" the plan DB2 will use to access the table?
- DCLGEN a table?