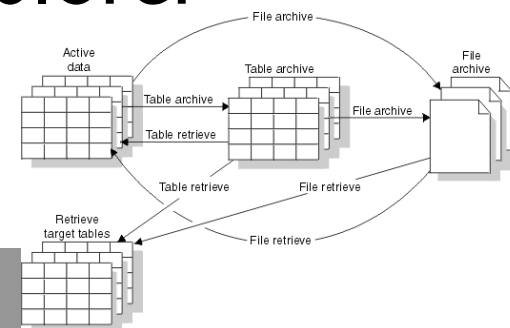




IBM Software Group

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

© Copyright IBM Corporation 2008 through 2019

All rights reserved by IBM – including the right to use these materials for in-house IDz technical instruction (please contact jsayles@us.ibm.com for permission)

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.

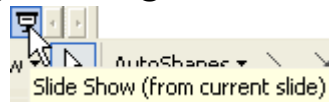
IBM, the IBM logo, the on-demand business logo, Rational, the Rational logo, and other IBM Rational products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names 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](http://en.wikipedia.org/wiki/SQL)

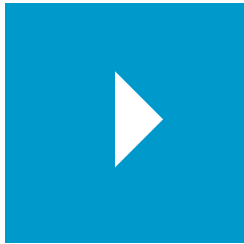
- Note: In order for you to get the above links to work, view the PowerPoint in Slide Show mode.



- 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_samplecobolrdathreepartnames.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

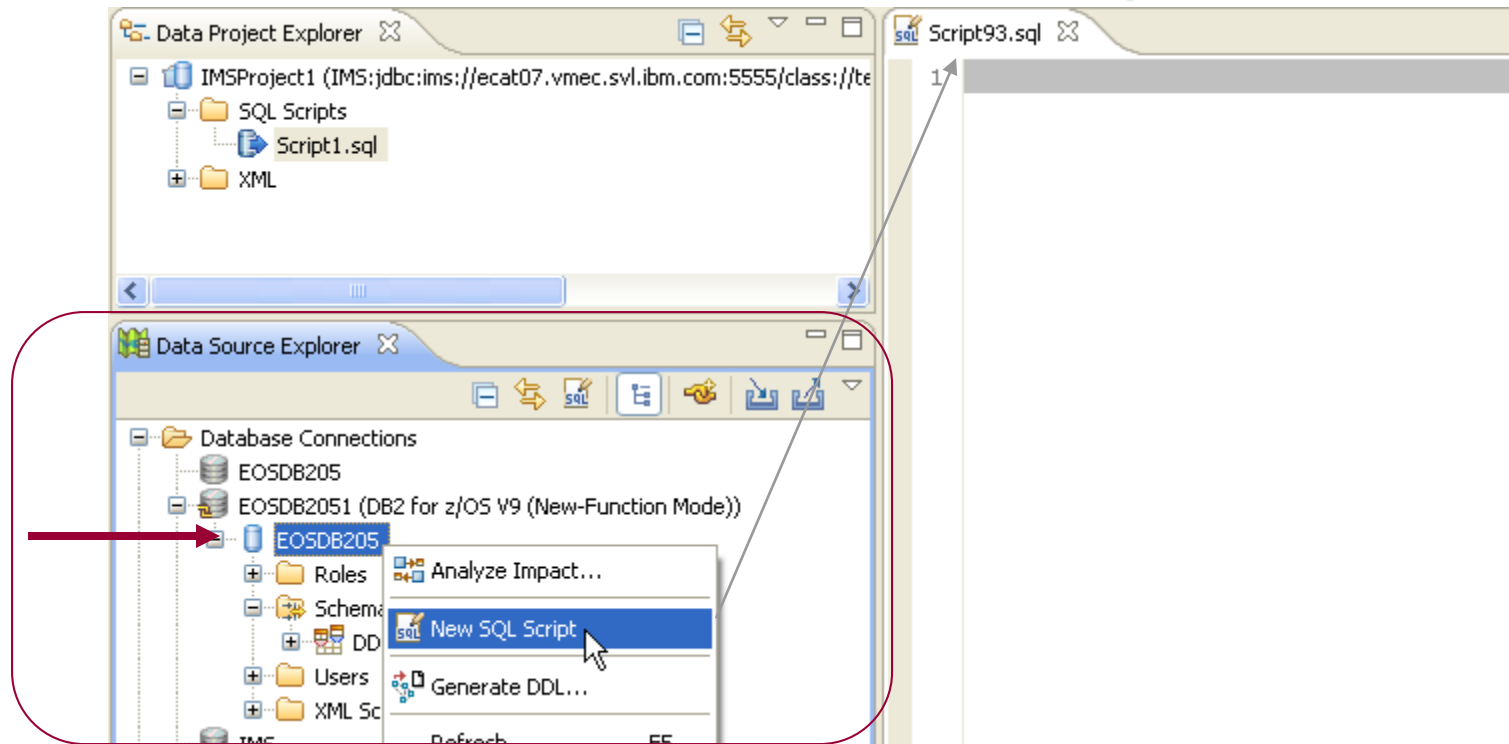
Code Interactive SQL Statements – SQL Script

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)

1. Access SQL functionality

2. New SQL Script

3. Code SQL

4. Run SQL

5. SQL Results

Status	Operation	Date	Conn	DEPTNO	DEPTNAME	MGRNO	ADM
✓	Succes SELECT ...	8/9/13 1...	NDC	1 A00	SPIFFY COMPUTER ...	000010	A00
✓	Succes DROP TA...	8/10/13 ...	NDC	2 B01	PLANNING	000020	A00
✓	Succes SELECT I...	8/10/13 ...	NDC	3 C01	INFORMATION CE...	000030	A00
✓	Succes SELECT I...	8/10/13 ...	NDC	4 D01	DEVELOPMENT CE...	NULL	A00
✓	Succes Sample C...	8/10/13 ...	NDC	5 D11	MANUFACTURING ...	000060	D01
✓	Succes SELECT I...	8/10/13 ...	NDC	6 D21	ADMINISTRATION ...	000070	D01
✗	Critic: "DDS000...	8/10/13 ...	NDC	7 F01	SUPPORT SERVICES	000050	A00

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

Additional SQL
coding/testing/tuning
options in Data Studio

Toolbar →

Connection: NDCDB201 [dds0001] Select..

Run method: JDBC

Run options:

- ☐ Refresh explorer view after script is run
- ☒ Open a new connection each time a script is run
- ☐ Open a new connection and reuse it to run scripts

```
select * from DDS001.DEPARTMENT where LOCATION IS NULL
```

Properties | SQL Results | Console | SQL Outline

Type query expression here	Status	Operation	Date	Conn	DEPTNO	DEPTNAME
✓ Succes SELECT ...	8/9/13 1...	NDC	1	A00	SPIFFY COM	
✓ Succes DROP TA...	8/10/13 ...	NDC	2	B01	PLANNING	
✓ Succes SELECT I...	8/10/13 ...	NDC	3	C01	INFORMAT	
✓ Succes SELECT I...	8/10/13 ...	NDC	4	D01	DEVELOPM	
✓ Succes Sample C...	8/10/13 ...	NDC	5	D11	MANUFAC	
✓ Succes SELECT I...	8/10/13 ...	NDC	6	D21	ADMINISTR	
✗ Critic: "DDS000...	8/10/13 ...	NDC	7	E01	SUPPORT S	
✗ Critic: "DDS000...	8/10/13 ...	NDC	8	E11	OPERATION	
✓ Succes select * fr...	8/11/13 ...	NDC	9	E21	SOFTWARE	

Total 14 records shown

NDCDB201:retrieveMessagesFromServerOnGetMessage=true;emulateParameterMet

Context Menu:

- Undo Typing (Ctrl+Z)
- Revert File
- Save (Ctrl+S)
- Open With
- Show In (Alt+Shift+W)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Run As
- Debug As
- Profile As
- Validate
- Software Analyzer
- Team
- Compare With
- Replace With
- Preferences...
- Content Assist (Ctrl+Space)
- Content Tip (Ctrl+Shift+Space)
- Format SQL (Ctrl+Shift+F)
- Toggle Comment (Ctrl+)
- Use Database Connection...
- Run SQL (F5)
- Set Statement Terminator
- Validate Database Object References
- Start Tuning...
- Generate pureQueryXML File...
- Find reference in SQL Outline
- Generate pureQuery Code...
- Open Visual Explain

Coding,
Formatting,
Testing
Tuning options

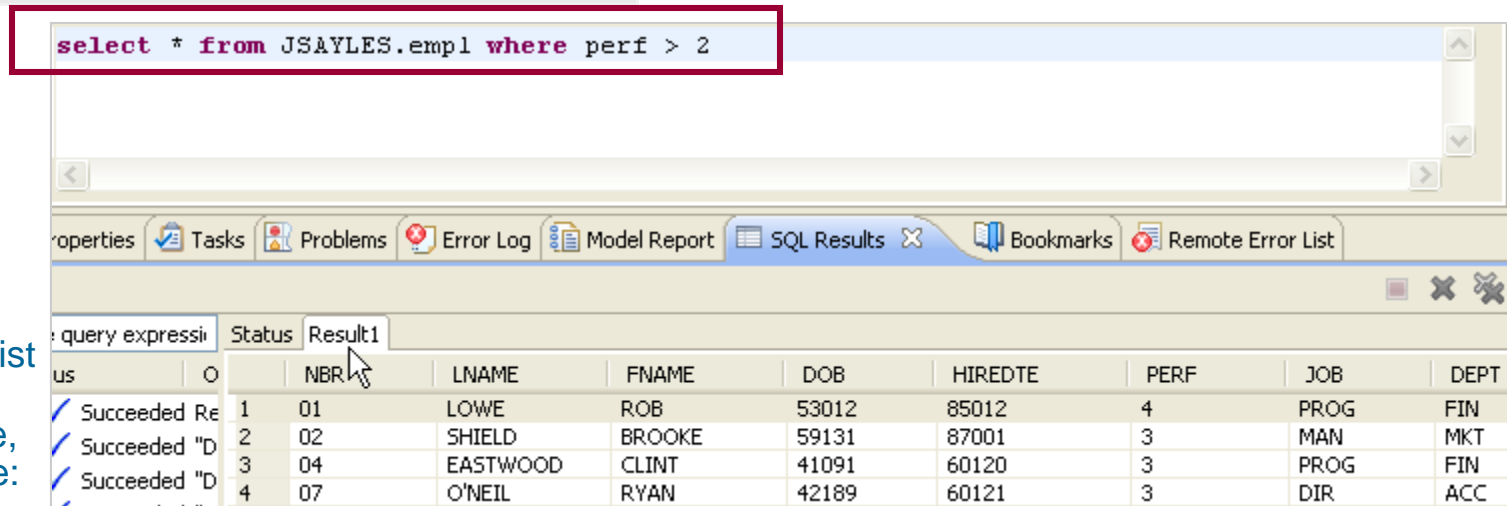
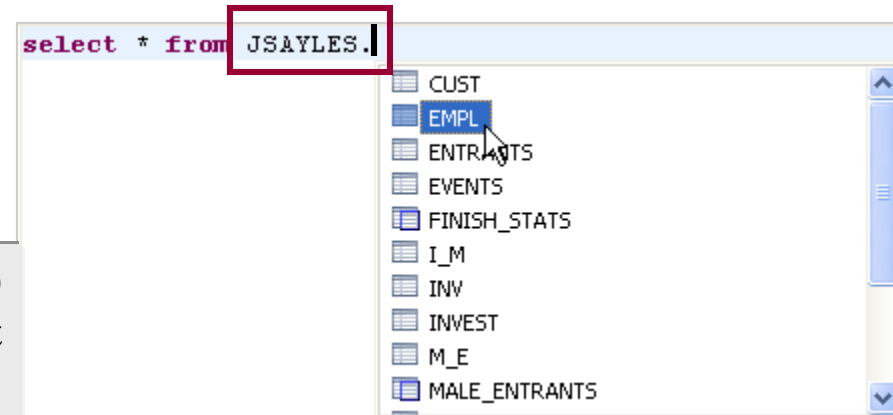
Create and Run a SQL Script Statement

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

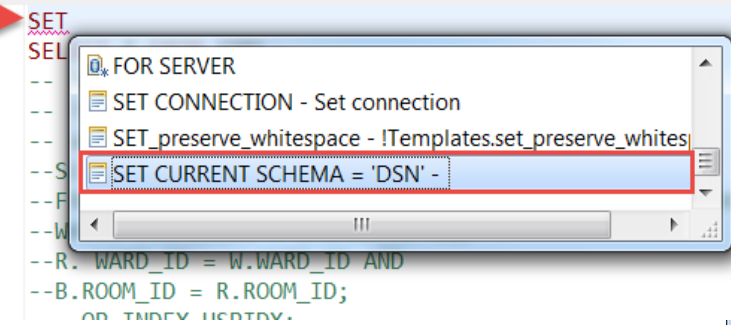


Note: For Content Assist to work, you must qualify the DB2 table, view, synonym name: **Schema.Tablename**

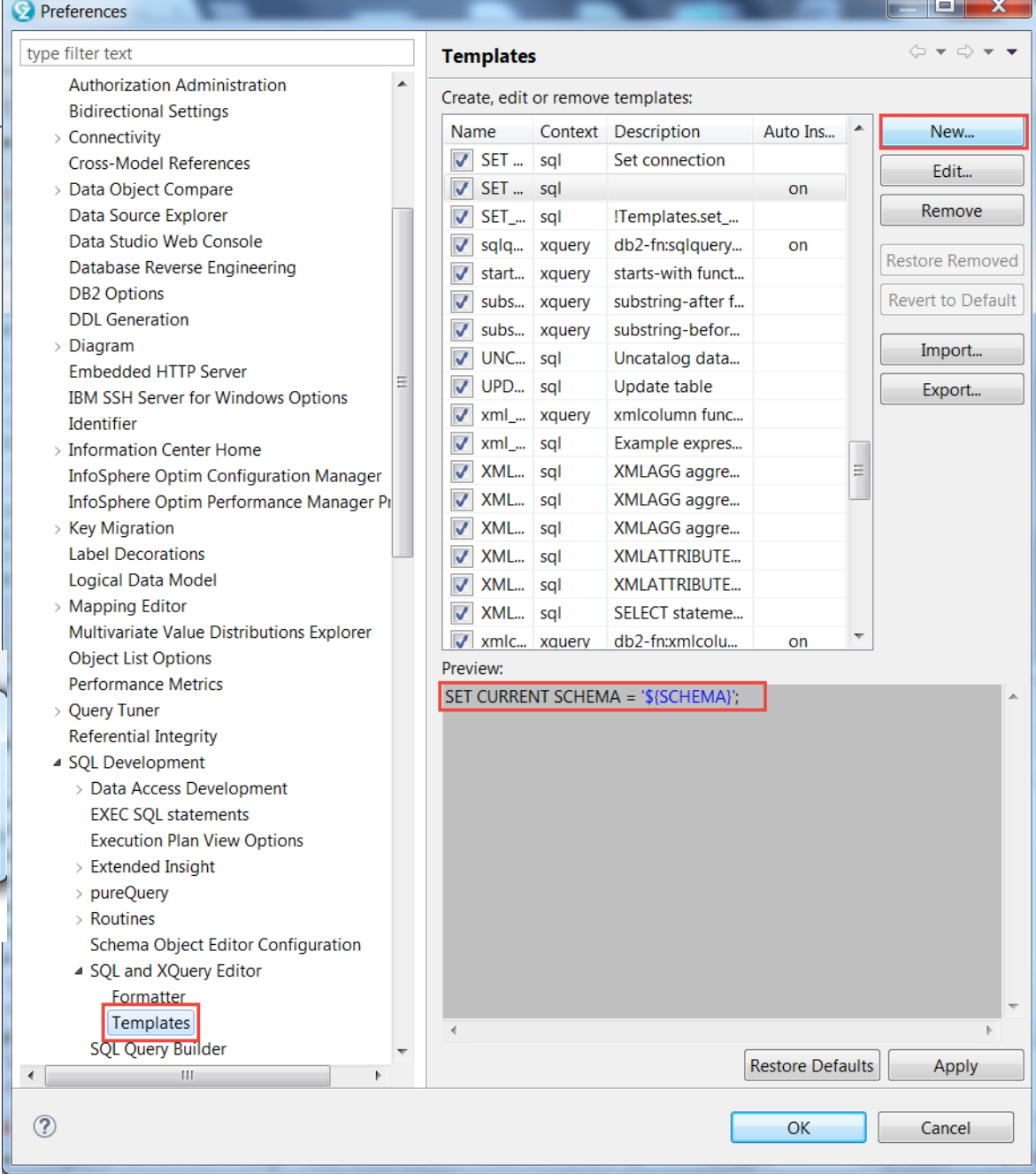
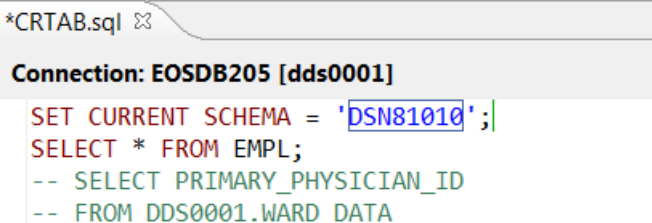
Set Current DB2 Schema

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 →

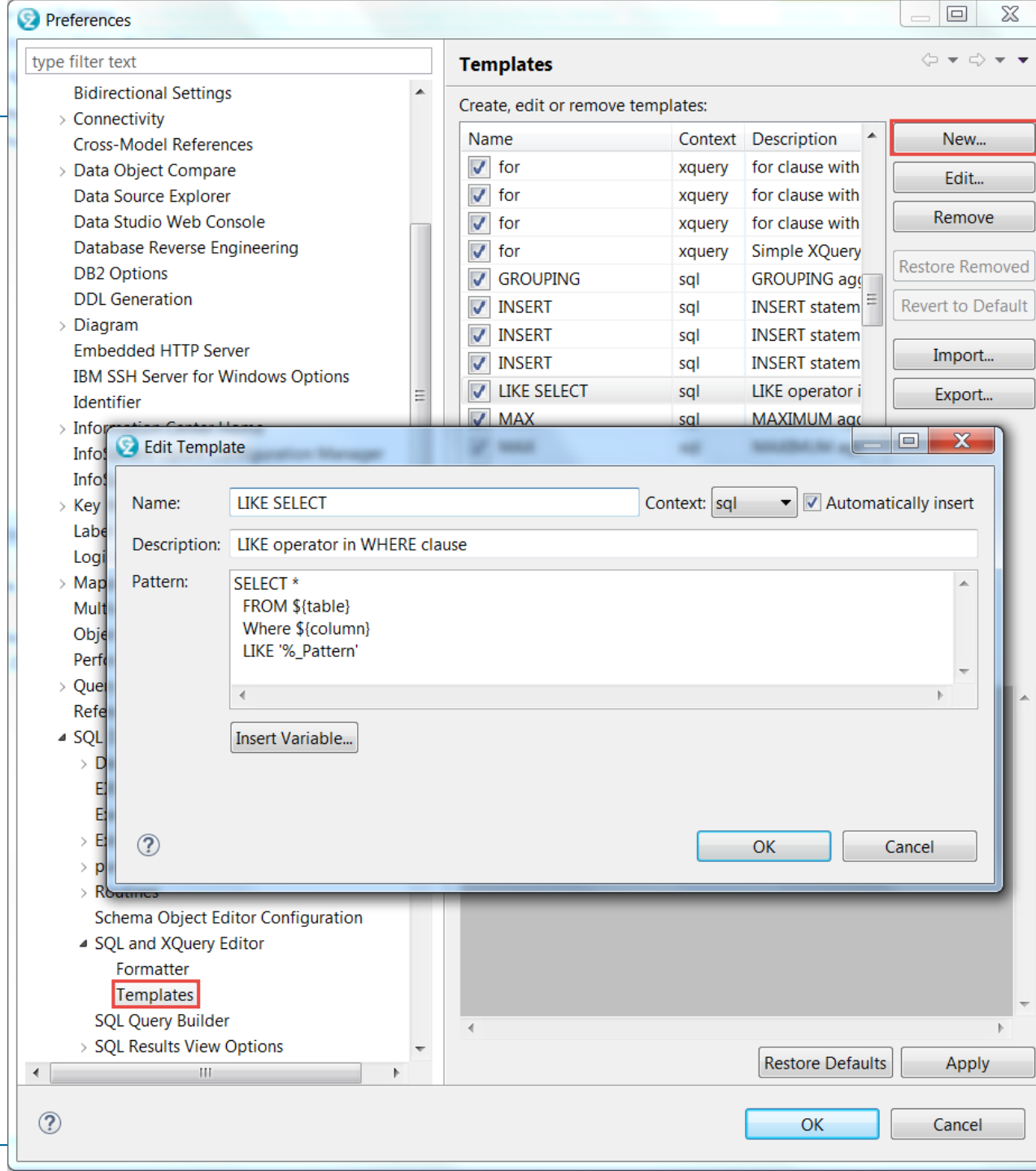
SQL and XQuery Editor →

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

The screenshot displays an SQL editor window titled '*Script147.sql' containing three SQL statements:

```
1 SELECT E.*
2 FROM DDS0001.EMPL AS E, DDS0001.PAY AS P, DDS0001.PROJ AS PR
3 WHERE E.NBR = P.NBR AND E.PROJ = PR.NBR;
4
5 SELECT E.*
6 FROM DDS0001.EMPL AS E
7 WHERE NBR IN ((SELECT NBR
8 FROM DDS0001.PAY))
9 UNION
10 SELECT E.*
11 FROM DDS0001.EMPL AS E
12 WHERE PROJ IN ((SELECT NBR
13 FROM DDS0001.PROJ));
```

Below the editor is a toolbar with icons for Properties, Tasks, Bookmark, Remote E, Remote S, Debug, Console, File Manager, Remote z, and SQL Results. The SQL Results window is open, showing a log of operations and a table of results.

Status	Operation
✓ Succeeded	"DDS0001"."HEALTH_PLAN"
✓ Succeeded	"DDS0001"."HEALTH_PLAN"
✓ Succeeded	select * from DDS0001.empl E, Pay p, Proj Pr where e.NBR = p.nb
✓ Succeeded	Script147.sql
✓ Succeeded	Script147.sql
✓ Succeeded	Script147.sql
✓ Succeeded	Script147.sql
✓ Succeeded	Script147.sql
✓ Succeeded	Script147.sql
✓ Succeeded	SELECT E.* FROM DDS0001.EMPL AS E, DDS0001.PAY AS P, DD
✓ Succeeded	SELECT E.* FROM DDS0001.EMPL AS E WHERE NBR IN ((SEL

	NBR	LNAME	FNAME	DOB	HIR
1	01	LOWE ...	ROB	53012	8501
2	02	SHIEL...	Dieter	59131	8700
3	03	MOOR...	ROGER	48111	8600
4	04	EAST...	Martin	41091	6012
5	06	BURN...	GEOR...	11178	4900
6	07	O'NEIL	Karin	42189	6012
7	08	MARVI...	LEE	32187	5187

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

The screenshot displays the IBM SQL Developer interface. The top pane shows a SQL script in a file named *Script3.sql. The script is:

```
SELECT *  
FROM JSAYLES.EMPL, JSAYLES.PAY  
WHERE JSAYLES.EMPL.NBR = JSAYLES.PAY.NBR  
AND JSAYLES.EMPL.NBR = '01'
```

A right-click context menu is open over the script area, with the 'Run SQL' option highlighted. The menu includes options like 'Undo Typing', 'Save', 'Cut', 'Copy', 'Paste', 'Run As', 'Debug As', 'Profile As', 'Validate', 'Software Analyzer', 'Team', 'Compare With', 'Replace With', 'Preferences...', 'Content Assist', 'Content Tip', 'Format SQL', 'Validate Statement Syntax', 'Use Database Connection...', and 'Set Statement Terminator'.

The bottom pane shows the 'Results' tab with a table of execution results. The table has columns for 'Status', 'NBR', 'LNAME', 'FNAME', and 'DOB'. The first row shows a successful execution for NBR 01, LNAME LOWE, FNAME ROB, and DOB 53012.

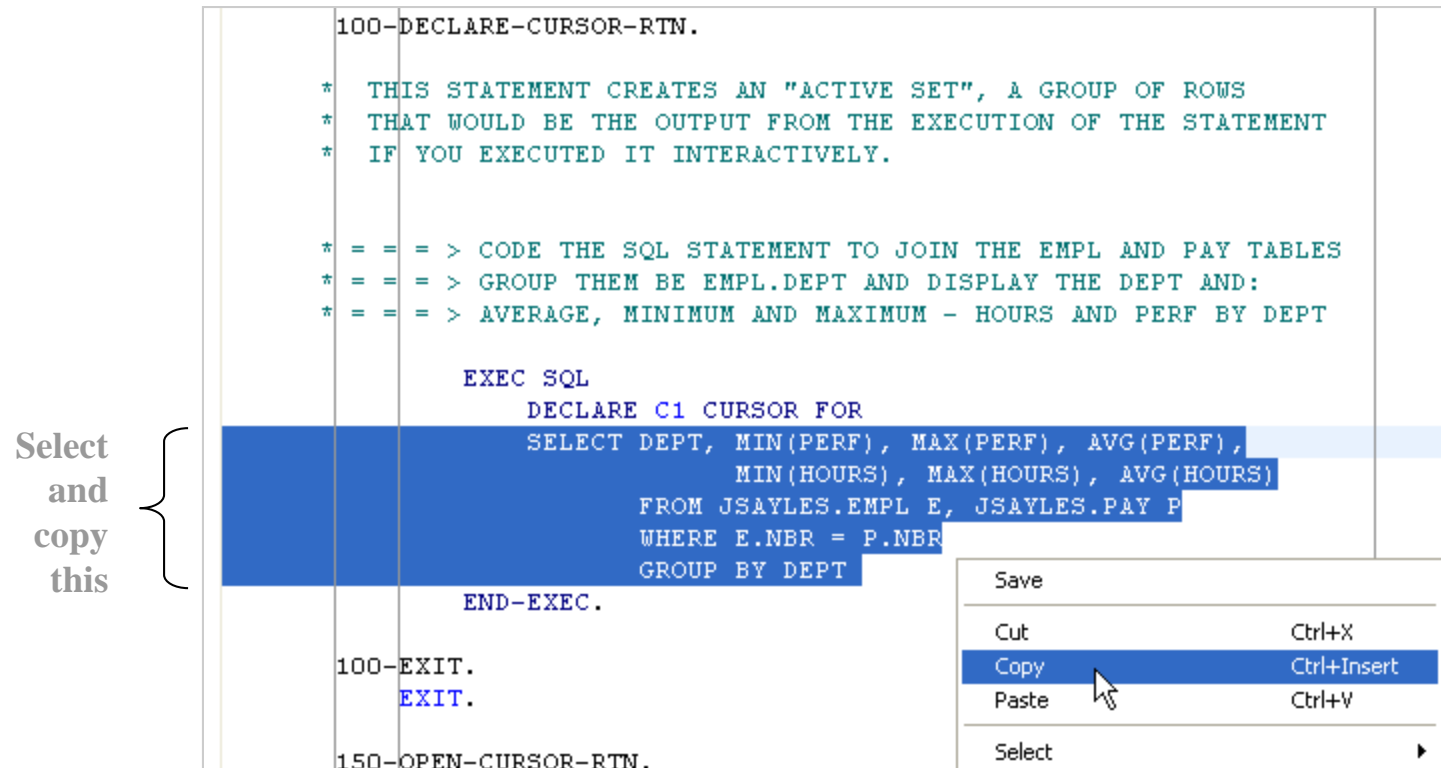
Status	NBR	LNAME	FNAME	DOB
✓ Succeeded	01	LOWE	ROB	53012

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



The screenshot shows a COBOL program with embedded SQL. The paragraph **100-DECLARE-CURSOR-RTN.** contains several comments and an SQL statement. The SQL statement is highlighted in blue, and a context menu is open over it, with the **Copy** option selected. A bracket on the left indicates that the SQL statement should be copied.

```
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.
```

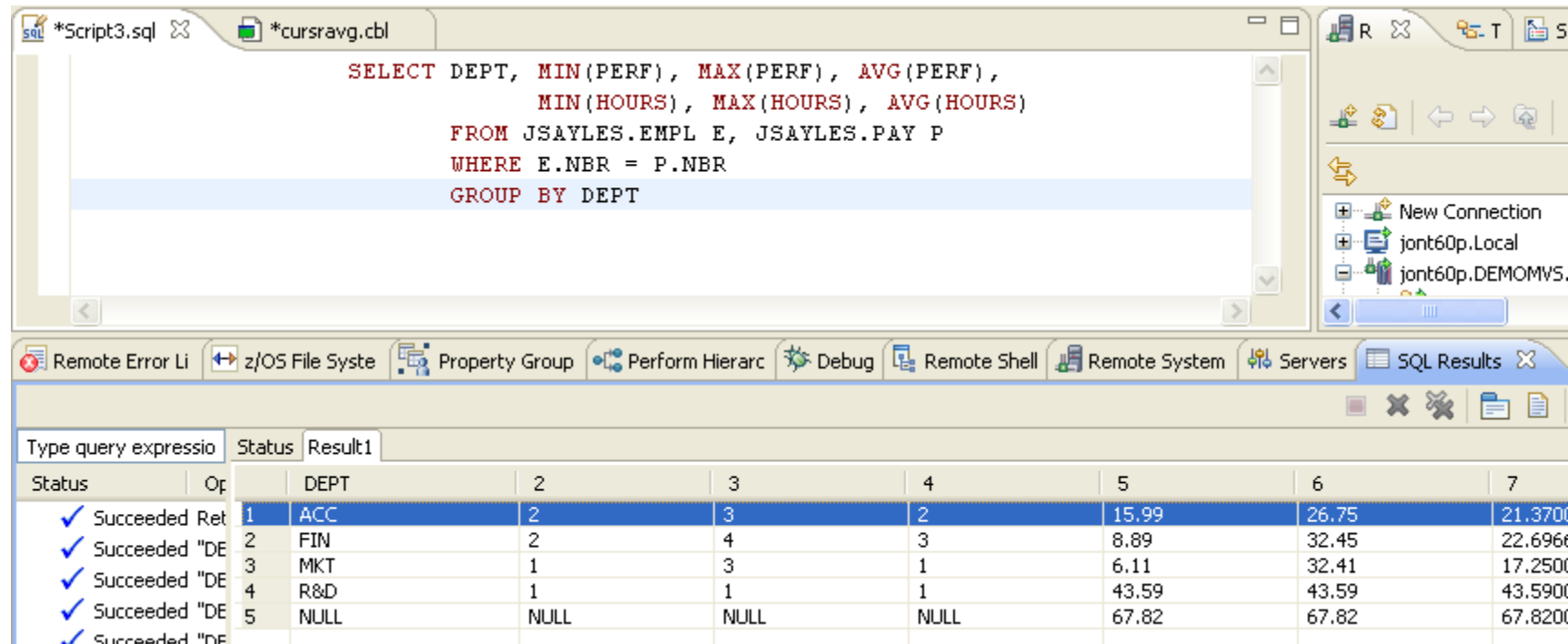
Select and copy this

Save
Cut Ctrl+X
Copy Ctrl+Insert
Paste Ctrl+V
Select

 - 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



The screenshot shows the IBM Data Studio interface. The top pane displays an SQL script in a file named *Script3.sql. The script is a SELECT statement with aggregate functions (MIN, MAX, AVG) grouped by department. The bottom pane shows the results of the query in a table format. The table has columns for Status, Op, DEPT, and seven numerical columns representing the aggregated values. The results show data for departments ACC, FIN, MKT, R&D, and NULL.

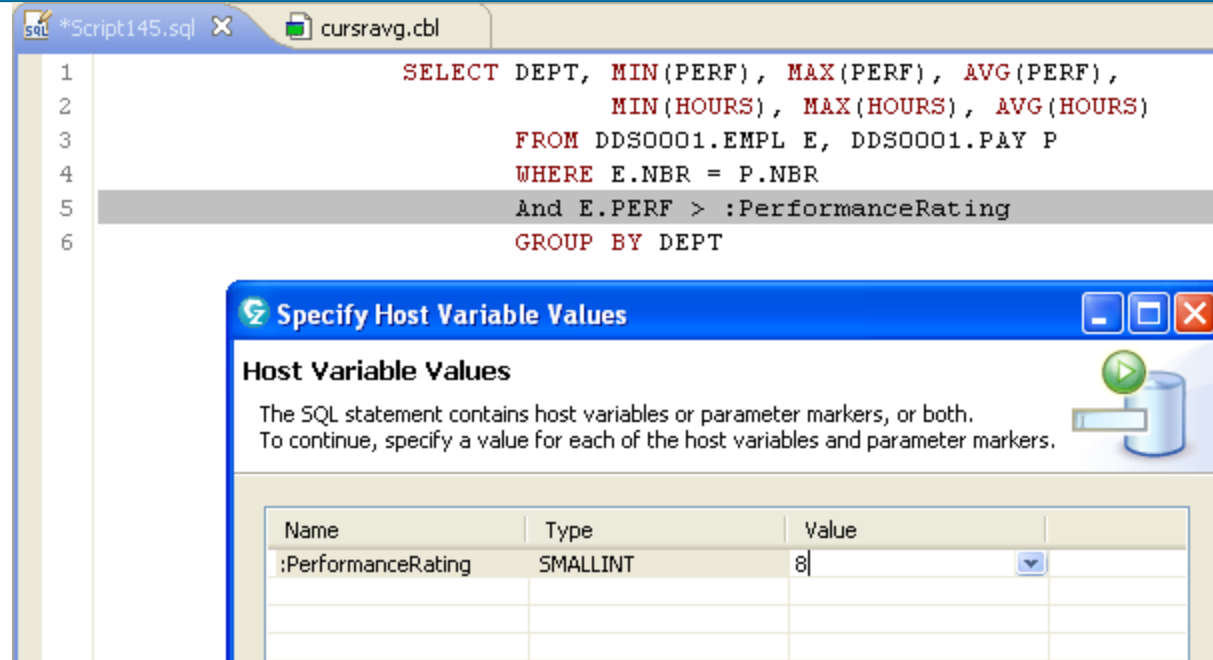
```
SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PERF),  
       MIN(HOURS), MAX(HOURS), AVG(HOURS)  
FROM JSAYLES.EMPL E, JSAYLES.PAY P  
WHERE E.NBR = P.NBR  
GROUP BY DEPT
```

Status	Op	DEPT	2	3	4	5	6	7
✓ Succeeded Ret	1	ACC	2	3	2	15.99	26.75	21.3700
✓ Succeeded "DE	2	FIN	2	4	3	8.89	32.45	22.6966
✓ Succeeded "DE	3	MKT	1	3	1	6.11	32.41	17.2500
✓ Succeeded "DE	4	R&D	1	1	1	43.59	43.59	43.5900
✓ Succeeded "DE	5	NULL	NULL	NULL	NULL	67.82	67.82	67.8200

 **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**

The screenshot displays the IBM z/OS Projects environment. On the left, the 'Outline View' is open, showing a list of SQL statements. A red callout bubble points to the 'Outline View' tab with the text 'From the Outline view'. A right-click context menu is open over the 'SELECT' statements, with the 'Run SQL' option highlighted. On the right, the 'SQL Editor' window is open, showing a file named '*Script1.sql'. The editor contains several SQL statements, including scalar functions and data retrieval queries. A blue highlight is visible over a portion of the SQL code. The toolbar at the bottom right of the editor includes a 'Run SQL' button, which is highlighted with a red box.

From the Outline view

Connection: EOSDB205 [dds0001]

```
-- ***** SQL Scalar Function ***** --
-- *****
-- This snippet concatenates two string columns into one result column
SELECT NAME || ' THIS IS A NAME' FROM dds0001.employee; -- Concatenate a CHAR column with a string value

--The VALUE function substitutes whatever you specify for a NULL value in a row
SELECT VALUE(PERF,0) FROM dds0001.employee; -- #35. VALUE FUNCTION (zero substituted for NULL)

--SUBSTR takes a substring of a character value SUBSTR(<columnName>, <StartByte>, <NbrOfBytes>)
SELECT SUBSTR(NAME,1,1) FROM dds0001.employee;

-- This snippet casts column data of numeric and date types to string (char)
SELECT CHAR(HIREDTE,EUR) FROM dds0001.employee;

--Digits casts different numeric values to ordinal numbers
SELECT DIGITS(PERF), DIGITS(RATE) FROM dds0001.employee E, dds0001.pay P
WHERE E.NBR = P.NBR;

--FLOAT casts numeric data to Floating Point format
SELECT FLOAT(PERF) FROM dds0001.employee;

--Hex Returns the Hexadecimal value
SELECT HEX(HIREDTE) FROM dds0001.employee;

--Hour extracts the Hour portion of a time or timestamp value
SELECT HOUR(SHIFT) FROM dds0001.employee;

--Year extracts the year portion of a date
SELECT YEAR(HIREDTE) FROM dds0001.employee;

--Month extracts the month portion of a date
SELECT MONTH(HIREDTE) FROM dds0001.employee;

SELECT DAY(HIREDTE) FROM dds0001.employee;
```

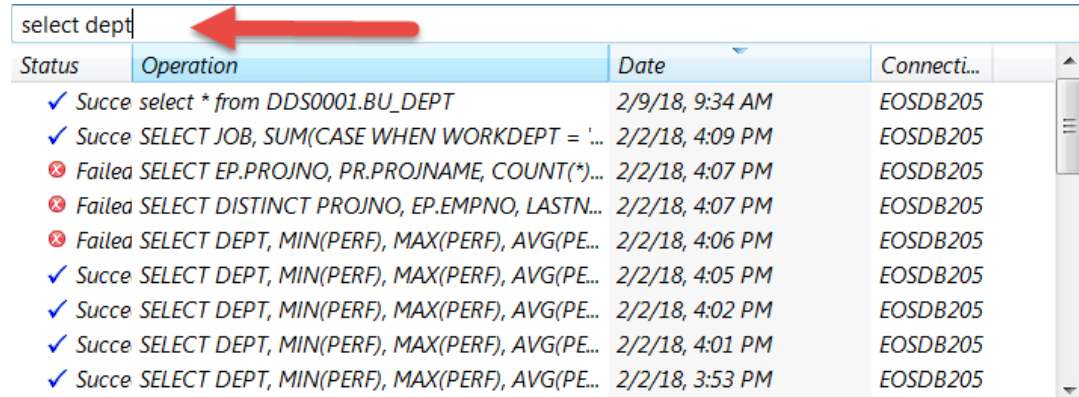
Run SQL

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

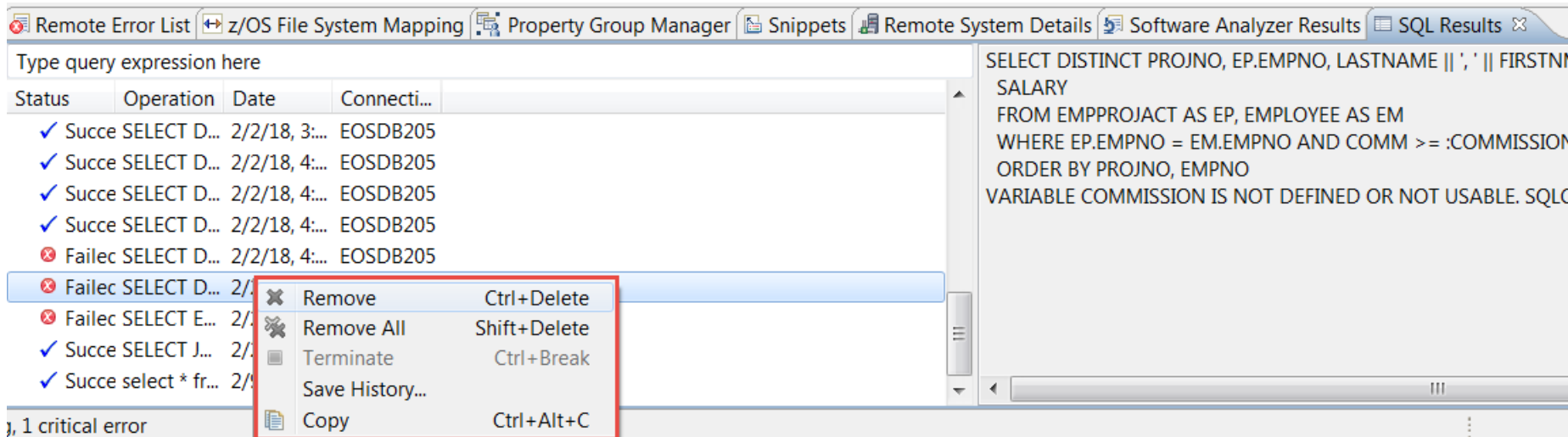
The SQL Results view has a number of useful actions for development/test

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



Status	Operation	Date	Connecti...
✓	Succe select * from DDS0001.BU_DEPT	2/9/18, 9:34 AM	EOSDB205
✓	Succe SELECT JOB, SUM(CASE WHEN WORKDEPT = '...	2/2/18, 4:09 PM	EOSDB205
✗	Failea SELECT EP.PROJNO, PR.PROJNAME, COUNT(*)...	2/2/18, 4:07 PM	EOSDB205
✗	Failea SELECT DISTINCT PROJNO, EP.EMPNO, LASTN...	2/2/18, 4:07 PM	EOSDB205
✗	Failea SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PE...	2/2/18, 4:06 PM	EOSDB205
✓	Succe SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PE...	2/2/18, 4:05 PM	EOSDB205
✓	Succe SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PE...	2/2/18, 4:02 PM	EOSDB205
✓	Succe SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PE...	2/2/18, 4:01 PM	EOSDB205
✓	Succe SELECT DEPT, MIN(PERF), MAX(PERF), AVG(PE...	2/2/18, 3:53 PM	EOSDB205

- 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



Remote Error List | z/OS File System Mapping | Property Group Manager | Snippets | Remote System Details | Software Analyzer Results | SQL Results

Type query expression here

Status	Operation	Date	Connecti...
✓	Succe SELECT D...	2/2/18, 3:...	EOSDB205
✓	Succe SELECT D...	2/2/18, 4:...	EOSDB205
✓	Succe SELECT D...	2/2/18, 4:...	EOSDB205
✓	Succe SELECT D...	2/2/18, 4:...	EOSDB205
✗	Failec SELECT D...	2/2/18, 4:...	EOSDB205
✗	Failec SELECT D...	2/2/18, 4:...	EOSDB205
✗	Failec SELECT E...	2/2/18, 4:...	EOSDB205
✓	Succe SELECT J...	2/2/18, 4:...	EOSDB205
✓	Succe select * fr...	2/2/18, 4:...	EOSDB205

- Remove Ctrl+Delete
- Remove All Shift+Delete
- Terminate Ctrl+Break
- Save History...
- Copy Ctrl+Alt+C

1 critical error

```
SELECT DISTINCT PROJNO, EP.EMPNO, LASTNAME || ' ' || FIRSTN  
SALARY  
FROM EMPPROJECT AS EP, EMPLOYEE AS EM  
WHERE EP.EMPNO = EM.EMPNO AND COMM >= :COMMISSION  
ORDER BY PROJNO, EMPNO  
VARIABLE COMMISSION IS NOT DEFINED OR NOT USABLE. SQLC
```


Additional SQL Results View Options – 2 of 4

Display result in single tab...

The screenshot shows the 'SQL Results' tab in a development environment. On the left, a list of operations is shown with their status. A red callout bubble points to the 'Sample Contents' operation, stating: 'Provides a single-view of both the SQL and its result'. On the right, the 'Sample Contents' view is displayed, showing a table with columns 'EDLEVEL' and 'OCCURRENCE'. A red arrow points to a button in the top right corner of the results pane, with a callout that says: 'Display result in single tab'.

ID	EDLEVEL	OCCURRENCE
1	12	4
2	14	5
3	15	3
4	16	12
5	17	6
6	18	4
7	19	1
8	20	1
9	22	1

Copy Results to Clipboard

Paste into Windows file

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

This block contains two screenshots. The left screenshot shows a 'Notepad' window with a table of employee data. A green callout bubble states: 'Results Pasted into Notepad'. The right screenshot shows the 'SQL Results' window with a table of employee data. A red arrow points to a button in the top right corner of the results pane, with a callout that says: 'Copy Results to clipboard'.

ID	EDLEVEL	OCCURRENCE
1	12	4
2	14	5
3	15	3
4	16	12
5	17	6
6	18	4
7	19	1
8	20	1
9	22	1

Additional SQL Results View Options – 3 of 4

Vertical Orientation

The screenshot shows the SQL Results window with the 'Vertical View Orientation' selected in the 'Preferences...' menu. The results are displayed in a vertical table format. A red arrow points to the 'Vertical View Orientation' option in the menu. A red callout bubble points to the results table with the text 'Good for queries returning many columns'.

Status	Operation	Date	Connecti...
Critical Error	"DSN81110"."PROJ"	2/7/19	EOSDB20...
Succeeded	SELECT * FROM dds0001.empl E, dds0001.pay PA, dds0001.proj PR	2/7/19, 1...	EOSDB20...
Succeeded	SELECT EMPNO FROM DSN81110.EMP E WHERE NOT EXISTS	2/7/19, 1...	EOSDB20...
Failed	SPUFIZOS.sql	2/7/19, 1...	EOSDB20...
Succeeded	SELECT LASTNAME FROM DSN81110.EMP	2/7/19, 1...	EOSDB20...
Succeeded	SELECT * from DDS0001.CUST WHERE ST in ('NJ','CT') ORDER BY 2 desc	2/7/19, 1...	EOSDB20...
Succeeded	Script1.sql	2/7/19, 1...	EOSDB20...
Succeeded	Sample Contents	2/7/19, 1...	EOSDB20...
Succeeded	SELECT MIN(WORKDEPT), MAX(WORKDEPT), AVG(EDLEVEL) FROM dds0...	2/18/19, ...	EOSDB205

ID	LNAME	FNAME	CITY	ST
03	Sparrow	JACK	HARTFORD	CT
03	Sparrow	JACK	HARTFORD	CT
08	Ravi	Sujeet	Raleigh	CT
08	Ravi	Sujeet	Raleigh	CT
89	Nebel	David	Vestal	CT
89	Nebel	David	Vestal	CT

Total 14 records shown

Display result in text mode...

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

The screenshot shows the SQL Results window with the 'Text' view selected. The results are displayed in a plain-text format. A red arrow points to the 'Text' view option in the 'Preferences...' menu. A red callout bubble points to the results table with the text 'Like SPUFI Results, plain-text output from the SQL statement'. A context menu is open over the results table, showing options like 'Copy', 'Paste', 'Delete', 'Select All', 'Find/Replace', 'Save All Results...', 'Export All Results...', and 'Print All Results...'.

Status	Operation
Succeeded	Sample Contents
Succeeded	Sample Contents
Succeeded	Sample Contents
Critical Error	"DSN81110"."PROJ"
Succeeded	SELECT * FROM dds0001.empl E, dds0001.pay PA, dds0001.proj PR
Succeeded	SELECT EMPNO FROM DSN81110.EMP E WHERE NOT EXISTS
Failed	SPUFIZOS.sql
Succeeded	SELECT LASTNAME FROM DSN81110.EMP
Succeeded	SELECT * from DDS0001.CUST WHERE ST in ('NJ','CT') ORDER BY 2 desc
Succeeded	Script1.sql
Succeeded	Sample Contents
Succeeded	SELECT MIN(WORKDEPT), MAX(WORKDEPT), AVG(EDLEVEL) FROM dds0..
Succeeded	SELECT MIN(WORKDEPT), MAX(WORKDEPT), AVG(EDLEVEL) FROM dds0..

EDLEVEL	OCCURREN
12	
14	
15	
16	
17	
18	
19	
20	
22	
23	
27	1
33	1
34	2

Additional SQL Results View Options – 4 of 4

Toggle Tabbed Layout View

Remote Error List Property Group Manager SQL Results Debug Properties SQL Outline Access Plan Diagram

Toggle Tabbed Layout View

Status	Operation	Date	Connection Profile
Failed	SPUFISQL.sql	2/18/19, 10:59 PM	EOSDB205
Succeeded	SELECT JOB, SUM(CASE W...	2/18/19, 11:02 PM	NDCDB201
Failed	SELECT EMPNO, SALARY F...	2/18/19, 11:05 PM	NDCDB201
Failed	SELECT EMPNO, SALARY F...	2/18/19, 11:07 PM	NDCDB201
Succeeded	SELECT EMPNO, SALARY F...	2/18/19, 11:07 PM	NDCDB201
Succeeded	SELECT DEPT FROM dds000...	2/18/19, 11:09 PM	EOSDB205
Succeeded	SELECT DEPT FROM dds000...	2/18/19, 11:10 PM	EOSDB205
Succeeded	SELECT DEPT, MIN(PERF), ...	2/18/19, 11:41 PM	NDCDB201
Failed		2/18/19, 11:51 PM	NDCDB201
Failed		2/18/19, 11:51 PM	NDCDB201
Succeeded		2/19/19, 12:15 AM	EOSDB205
Succeeded		2/19/19, 12:17 AM	EOSDB205

History Result

Tabs on the bottom of View

Toggles between tab layout views

Remote Error List Property Group Manager Debug Properties SQL Results SQL Outline Access Plan Diagram

Toggle Tabbed Layout View

Type query expression here

Status	Operation
Succeeded	Sample Contents
Succeeded	Sample Contents
Succeeded	Sample Contents
Critical Error	"DSN81110"."PROJ"
Succeeded	SELECT * FROM dds0001.empl E, dds0001.pay PA, dds0001.
Succeeded	SELECT EMPNO FROM DSN81110.EMP E W
Failed	SPUFIZOS.sql
Succeeded	SELECT LASTNAME FROM DSN81110.EMP
Succeeded	SELECT * from DDS0001.CUST WHERE ST in ('NJ','CT') ORC
Succeeded	Script1.sql
Succeeded	Sample Contents
Succeeded	SELECT MIN(WORKDEPT), MAX(WORKDEPT), AVG(EDLEVEI
Succeeded	SELECT MIN(WORKDEPT), MAX(WORKDEPT), AVG(EDLEVEI

Sample Contents

Starting run

SELECT DISTIN

Run successful

UNT(*) AS OCCURRENCE FROM DSN81110.EMP

Tabs on top of 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?