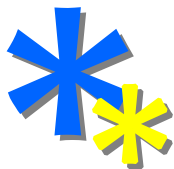




The Best Reliable Partner for High Availability



2008 System I Academy - Coex, Korea

System i에서 Oracle 연동하기

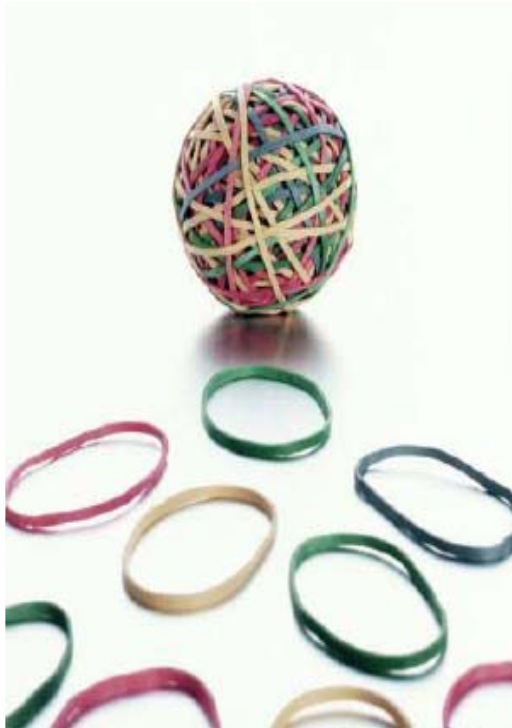
이강민 과장(lkm@kr.ibm.com)
IT Specialist

MTS, IBM Global Technology Services Korea

June 11, 2008



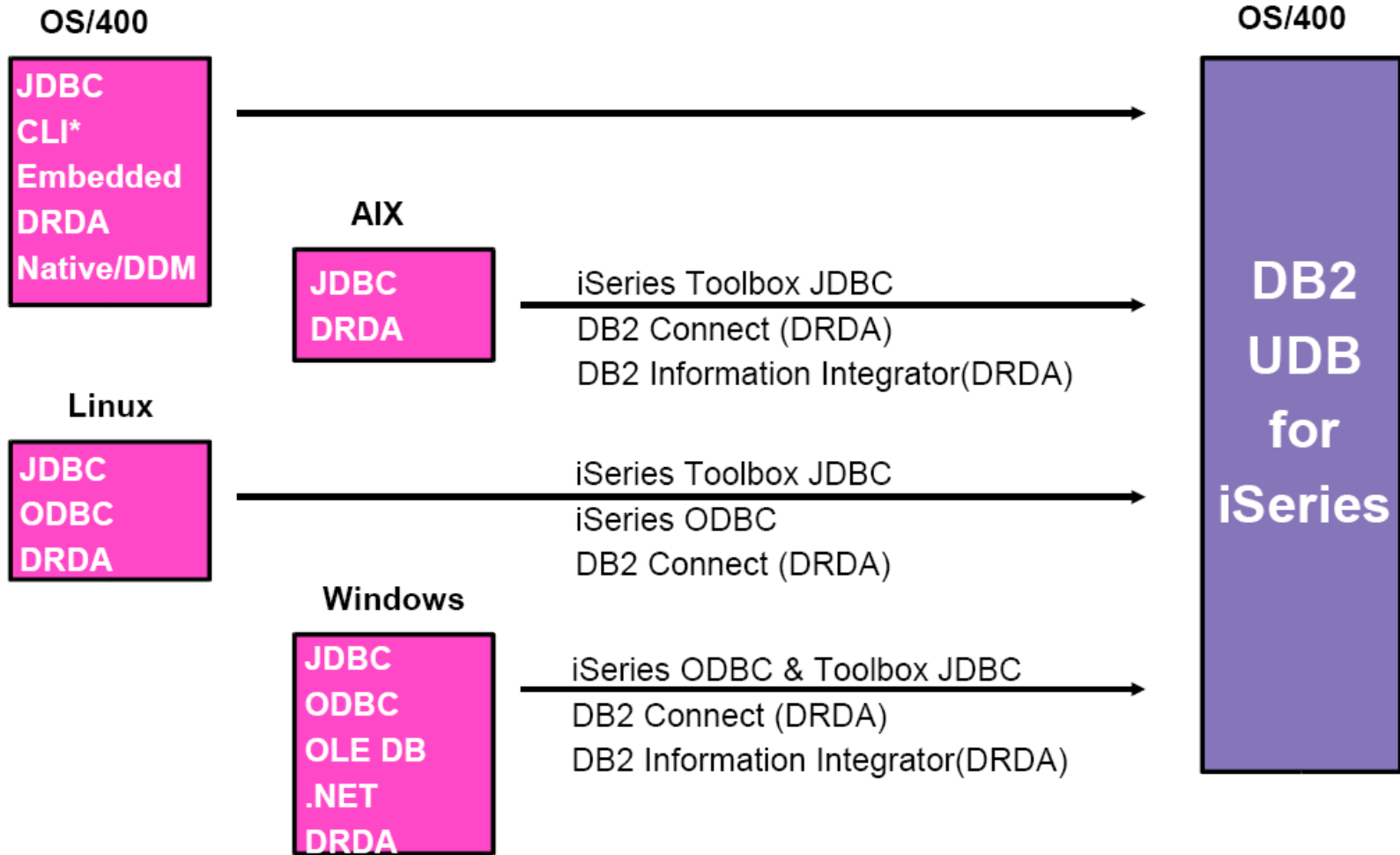
Agenda



1. DB2 UDB for i5/OS as a Server
2. Interoperability with DRDA
3. Interoperability without DRDA
4. Implementation
5. Summary

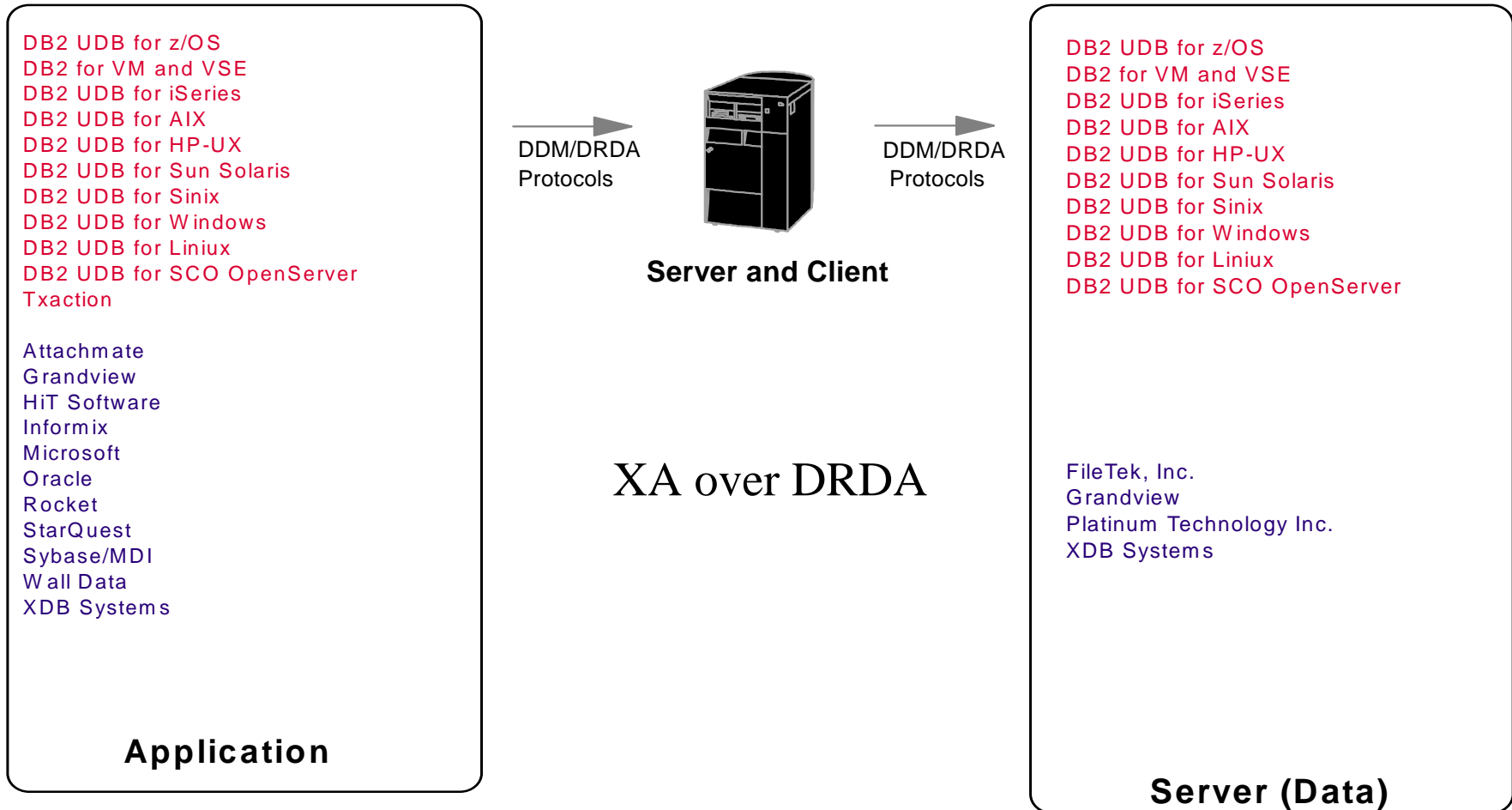
1. DB2 UDB as a Server

- ◆ DB2는 Requester의 요청에 의해 데이터를 저장, 조회, 삭제, 수정



2. Interoperability with DRDA

◆ IBM DB2 server는 DRDA protocol을 통한 연동 및 쿼리문 수행이 가능



Oracle and Microsoft do not support DRDA as a Server

2. Interoperability with DRDA

◆ DB2 for i5/iSeries client to DB2 for i5/iSeries server

Machine 1: "torisc6"
 Hostname: torisc6.ca.ibm.com
 IP Address: 9.89.168.6
 Operating System: OS/400 - iSeries 5.2
 DB2 UDB iSeries is part of the operating system

Machine 2: "mytorisc"
 Hostname: mytorisc.mkm.ibm.com
 IP Address: 9.29.168.62
 DB2 Port: 446
 Operating System: OS/400 - iSeries 5.2
 DB2 UDB iSeries is part of the Operating System



DB2
 Commands
 and Options
 to execute
 from
 OS/400
 Main menu

Part I - Configuring iSeries

From OS/400 main menu type:

wkrdbdire

In panel 'Work with relational database directory entries':

Choose option 1 to add a database, and type 'TORISC6B'

In panel 'Add RDB directory entry':

- Name or address field: 9.29.168.62

- Type: *IP

- Port Number or Service program: *DRDA

- Preferred Method: *ENCRYPTED

Part II - Testing the connection

From OS/400 main menu type: strsql

From 'Enter SQL Statements' Panel:

release all

commit

connect to TORISC6B user johntest using 'psw400'

select * from QIWS.QCUSTCDT

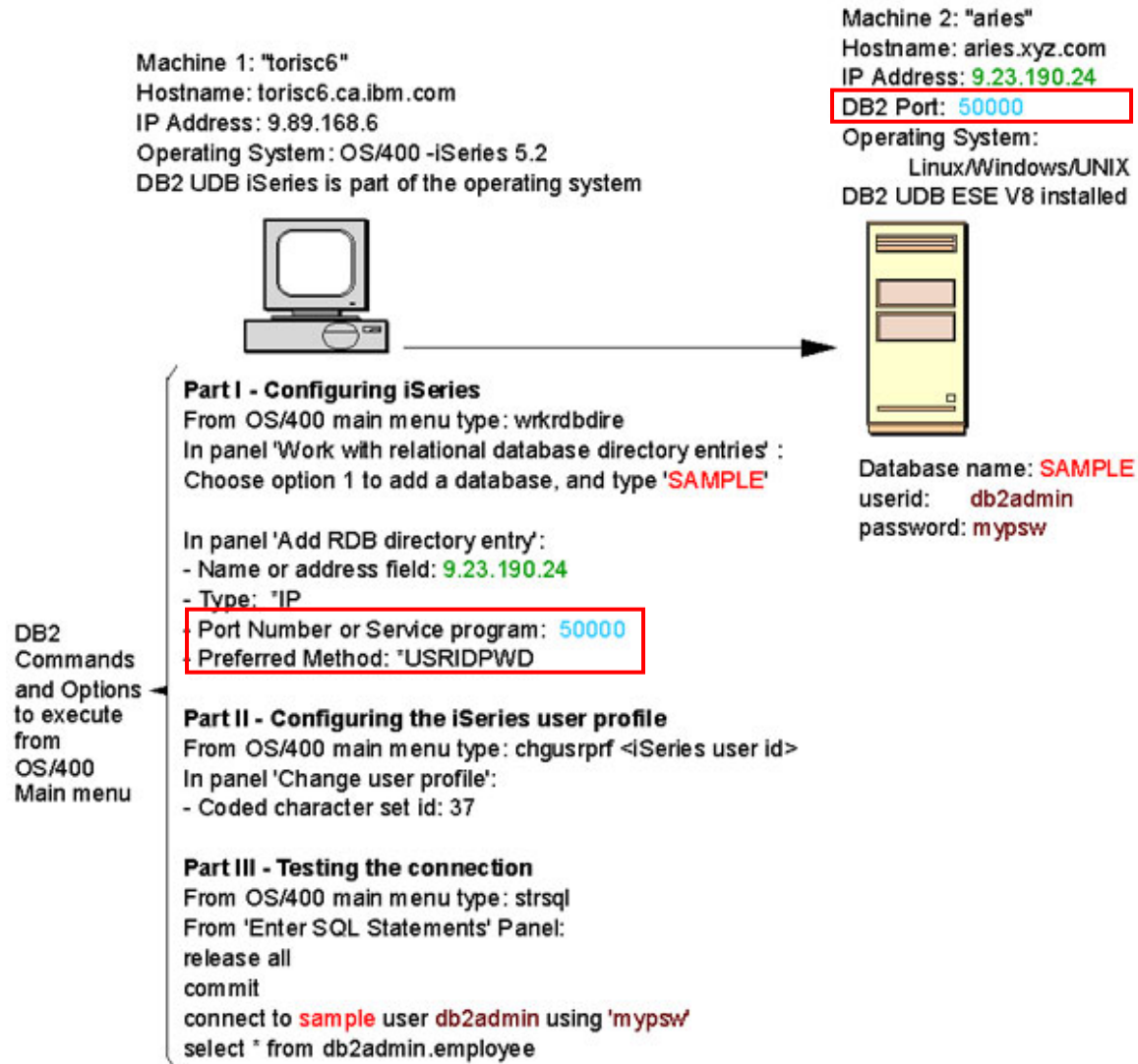
Local RDB Name (Database name): TORISC6B

userid: johntest

password: psw400

2. Interoperability with DRDA

◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server



2. Interoperability with DRDA

◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server

Machine 1 ('torisc6') DB2 UDB for iSeries	Machine 2 ('aries') DB2 UDB for Linux, UNIX and Windows
Part I: Configuring iSeries	
<p>Step 1</p> <p>From OS/400 main menu type wrkrdbdire After pressing enter, this will bring up the Work with relational database directory entries panel.</p>	
<p>From the Work with relational database directory entries panel:</p> <ul style="list-style-type: none"> •Choose Option 1 to Add a database. •type the database name: SAMPLE. 	<p>SAMPLE is the database in this machine 2 that you want to connect from the iSeries machine. If you don't remember the database name, you can issue from the CLP the command: list db directory and look for any entries with a Directory entry type of 'indirect'. These entries would correspond to local databases in your machine.</p>

2. Interoperability with DRDA

◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server

<p>Step 2 In panel 'Add RDB directory entry', specify:</p> <p>Name or address field: 9.23.190.24 Type: *IP</p> <p>Note: The hostname could have been used instead of the IP address.</p>	<p>For this example:</p> <p>9.23.190.24 = IP address of machine 2</p>
<p>Port number or Service program: 50000</p> <p>Notes:</p> <ul style="list-style-type: none"> - The default port number is *DRDA which means port 446. - The service name as defined on this machine1 (client) could have been used instead of the port number. - The default authentication method is *ENCRYPTED which only works between iSeries machines; thus *USRIDPWD is used. 	<p>50000 = The port used for DB2</p> <ul style="list-style-type: none"> - To find out the port used, issue this command from the CLP: get dbm cfg - Then look for the parameter SVCENAME. - If the value of SVCENAME is not the port number but a string, then look in your system for the file 'services' and grep for this string, which is normally based on your DB2 instance name. For example, if you instance name is 'db2inst1', you will normally find a corresponding entry like this: db2cdb2inst1 50000/tcp - The 'services' file can be located at: <ul style="list-style-type: none"> => /etc/services (in UNIX) => X:\WINNT\System32\drivers\etc\services (in Windows)

2. Interoperability with DRDA

◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server

Part II: Configuring the iSeries user profile

Step 3

For this example we logged on to the iSeries machine with user id of 'jmascare'. We need to change the profile for this user so that the CCSID value is not the default of 65335 but something else like 37 (US English). There is no codepage conversion supported for CCSID 65335.

From the OS/400 main menu type: `chqusrprf jmascare`

This command will invoke panel 'Change user profile'. Then specify:

- Coded character set id: 933

Part III: Testing the connection

Step 4

Start the Interactive SQL tool from the OS/400 main menu by typing: `strsql`

From the 'Enter SQL Statements Panel type:

`commit <enter>`

`connect to sample user db2admin using 'mypsw' <enter>`

`select * from db2admin.employee<enter>`

Note:

The password need to be passed in single quotes in order to maintain its case.

Using double quotes for the user id as shown below also works:

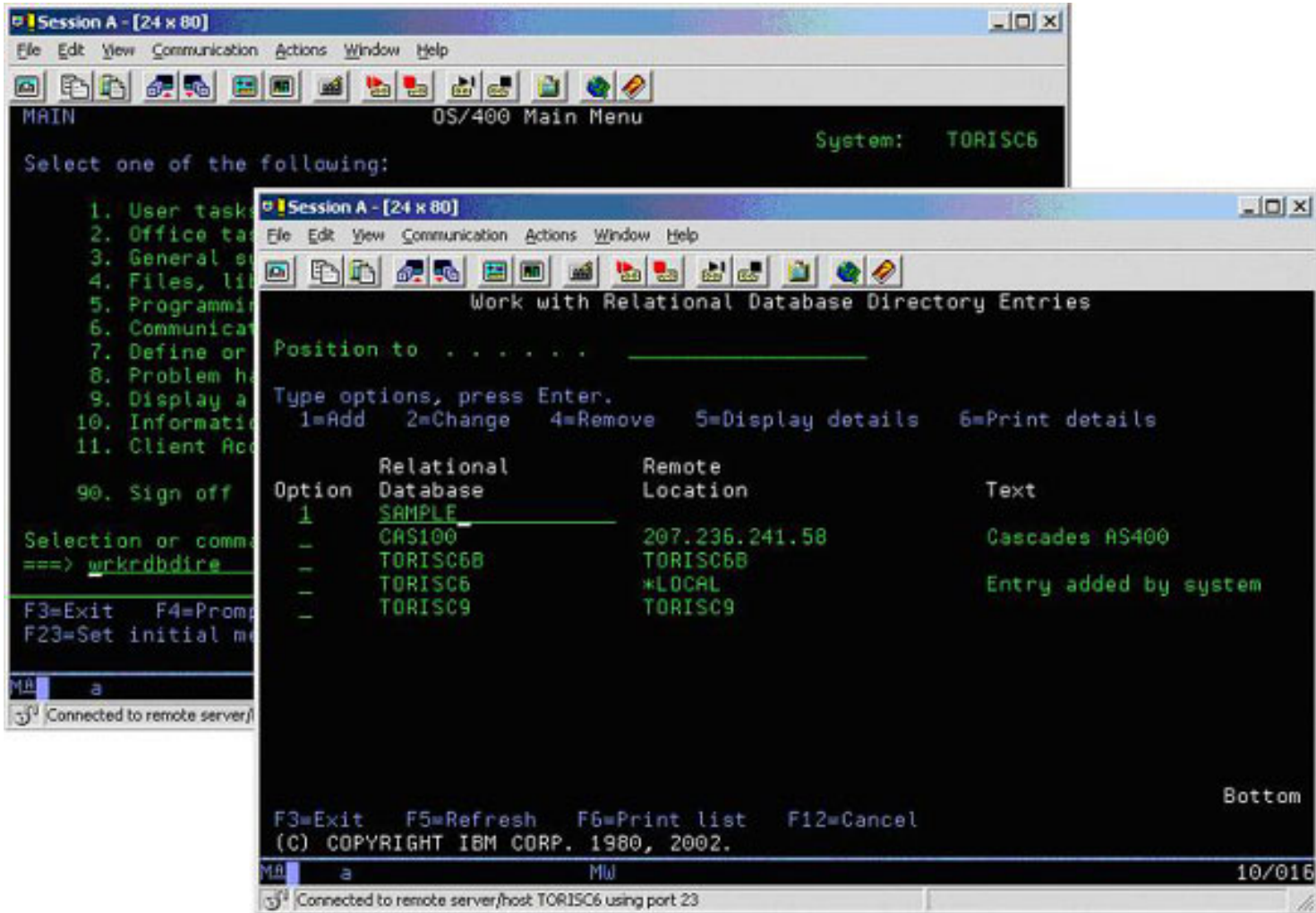
user "**db2admin**"using '**mypsw**'<enter>

db2admin = user id as defined on machine 2

mypsw = password as defined on machine 2

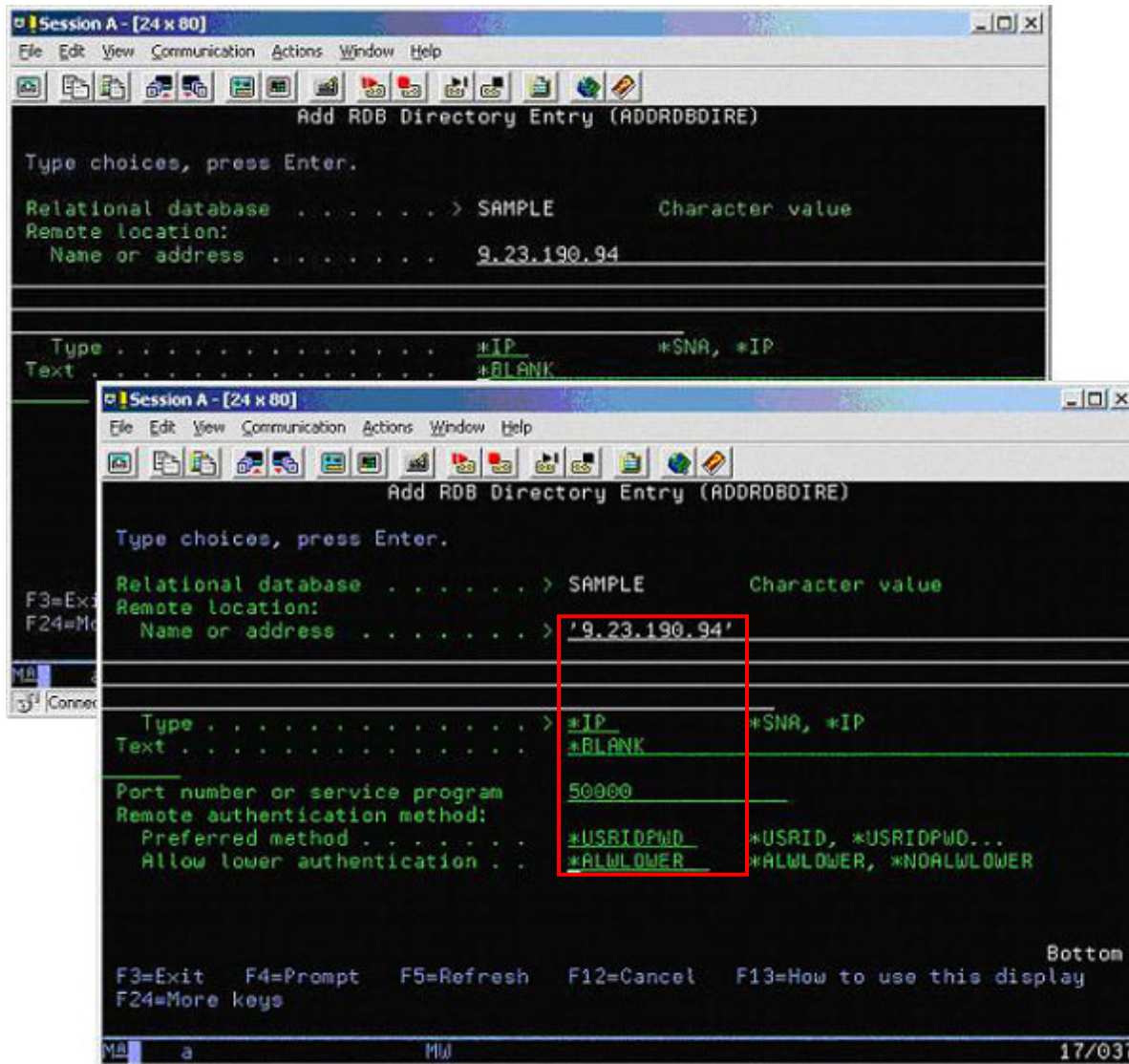
2. Interoperability with DRDA

- ◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server



2. Interoperability with DRDA

- ◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server



2. Interoperability with DRDA

- ◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
MAIN
OS/400 Main Menu
System: TORISC6
Select one of the following:
1. User tasks
2. Office tasks
3. General system
4. Files, libraries
5. Programming
6. Communication
7. Define or change
8. Problem handling
9. Display a screen
10. Information
11. Client Access
90. Sign off
Selection or command
==> strsql_
F3=Exit F4=Prompt
F23=Set initial menu
User profile JMASO
MR a
Connected to remote server/

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Enter SQL Statements
Type SQL statement, press Enter.
> release all
RELEASE of all relational databases completed.
> commit
Commit completed.
> CONNECT TO SAMPLE USER "db2admin" USING ''
Current connection is to relational database SAMPLE.
==> select * from db2admin.employee
Bottom
F3=Exit F4=Prompt F6=Insert line F9=Retrieve F10=Copy line
F12=Cancel F13=Services F24=More keys
MR a MId 10/038
Connected to remote server/host TORISC6 using port 23

```

2. Interoperability with DRDA

- ◆ DB2 for i5/iSeries client to DB2 Linux/Unix/Windows server

Session A - [27 x 132]

File Edit View Communication Actions Window Help

Display Report

Width . . . : 245
Column . . . : 1
Control Line

Line	1	2	3	4	5	6	7	8	9	10	11	12
	EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONE	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	
000001	000010	CHRISTINE	I	HAAS	AD0	3978	01/01/1965	PRES	18	F	08/24/1903	52,
000002	000020	MICHAEL	L	THOMPSON	BD1	3476	10/10/1973	MANAGER	18	M	02/02/1948	41,
000003	000030	SALLY	A	KUAN	CD1	4738	04/05/1975	MANAGER	20	F	05/11/1941	38,
000004	000050	JOHN	B	GEYER	ED1	6789	08/17/1949	MANAGER	16	M	09/15/1925	40,
000005	000060	IRVING	F	STERN	D11	6423	09/14/1973	MANAGER	16	M	07/07/1945	32,
000006	000070	EVA	D	PULASKI	D21	7831	09/30/1980	MANAGER	16	F	05/26/1953	35,
000007	000090	EILEEN	W	HENDERSON	E11	5498	08/15/1970	MANAGER	16	F	05/15/1941	29,
000008	000100	THEODORE	Q	SPENSER	E21	8972	06/19/1980	MANAGER	14	M	12/18/1956	26,
000009	000110	VINCENZO	G	LUCCHESSI	AD0	3490	05/16/1958	SALESREP	19	M	11/05/1929	46,
000010	000120	SEAN	O	O'CONNELL	AD0	2167	12/05/1963	CLERK	14	M	10/18/1942	29,
000011	000130	DOLORES	N	QUINTANA	CD1	4578	07/28/1971	ANALYST	16	F	09/15/1925	23,
000012	000140	HEATHER	A	NICHOLLS	CD1	1793	12/15/1976	ANALYST	18	F	01/19/1946	28,
000013	000150	BRUCE	A	ADAMSON	D11	4510	02/12/1972	DESIGNER	16	M	05/17/1947	25,
000014	000160	ELIZABETH	R	PIANKA	D11	3782	10/11/1977	DESIGNER	17	F	04/12/1955	22,
000015	000170	YOSHITOSHI	J	YOSHIMURA	D11	2890	09/15/1978	DESIGNER	16	M	01/05/1951	24,
000016	000180	MARILYN	S	SCOUTTEN	D11	1682	07/07/1973	DESIGNER	17	F	02/21/1949	21,
000017	000190	JAMES	H	WALKER	D11	2986	07/26/1974	DESIGNER	16	M	06/25/1952	20,

F3=Exit F12=Cancel F19=Left F20=Right F21=Split

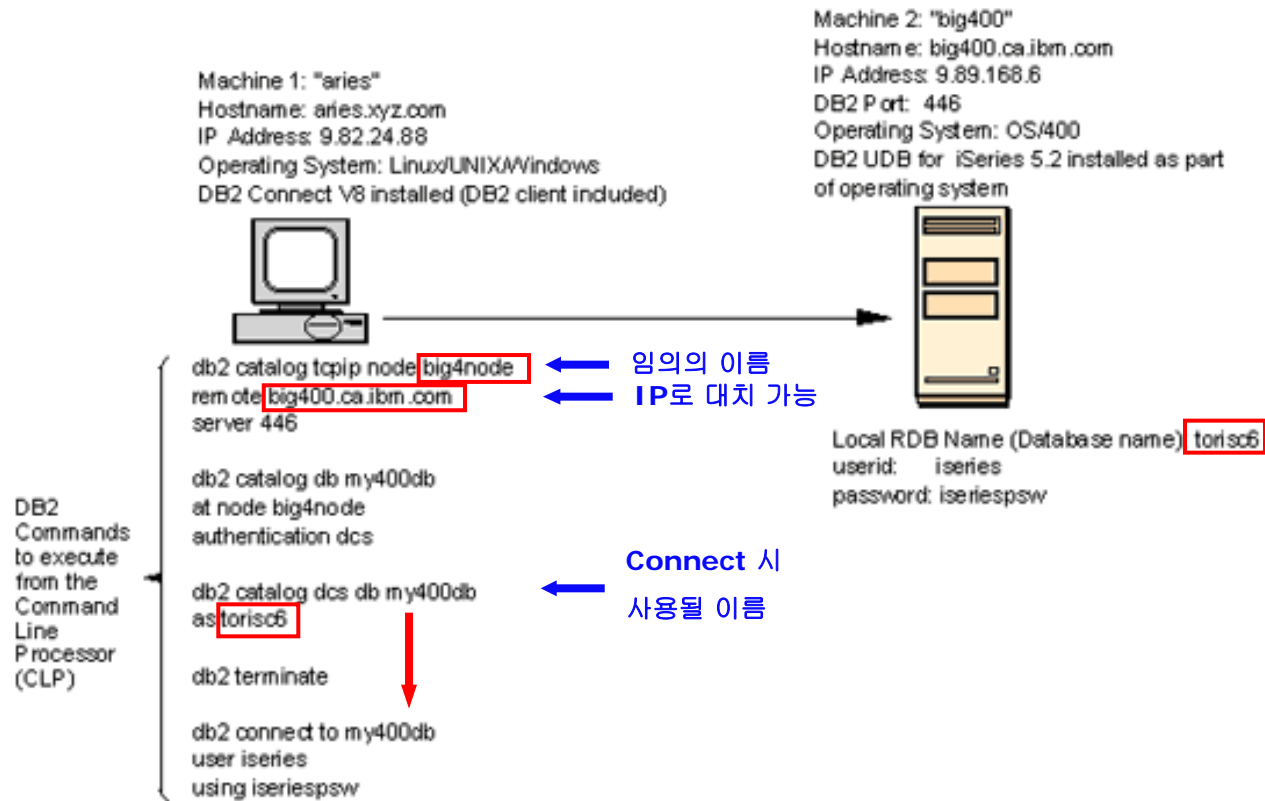
More...

HE a MW 04/021

Connected to remote server/host TORISC6 using port 23

2. Interoperability with DRDA

◆ DB2 Linux/Unix/Windows client to DB2 for i5/iSeries server



✓ **Notes**

⇒ DB2 for LUW가 client인 경우, z/OS나 i5/OS에 drda로 연결하기 위해서는 DB2 Connect 제품이 필요

3. Interoperability without DRDA

◆ **Example** – 데이터가 **System i**와 **Oracle**에 분리되어 있는 경우

✓ **DB2 UDB for i5/OS**

⇒ 사원번호 데이터는 **System i**에 아래와 같이 존재

Column Name	Short Name	Data Type	Length	Nul...	Default Value
EMPNO400	EMPNO400	CHARACTER	5	Yes	Null

	EMPNO400
1	A1111
2	B2222
3	C3333
4	D4444

✓ **Oracle**

⇒ 각 사원번호에 대한 이름과 급여정보는 **Oracle Database**에 존재

Columns	Data	Constraints	Grants	Statistics	Column
ORA_EMPNO	ORA_EMPNAME	SALARY			
1A1111	홍길동	2500			
2B2222	일지매	3500			
3C3333	코난	5000			

이 정보들을 **join** 해서
하나의 화면으로?!



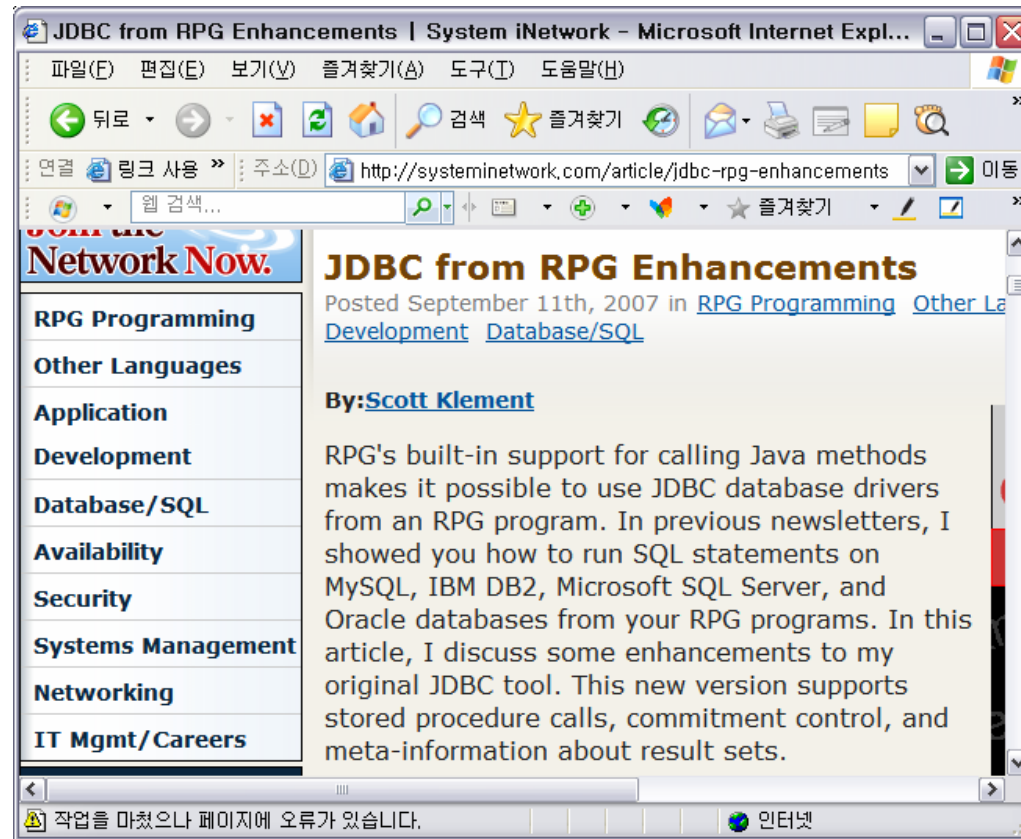
3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ JDBC from RPG Enhancement

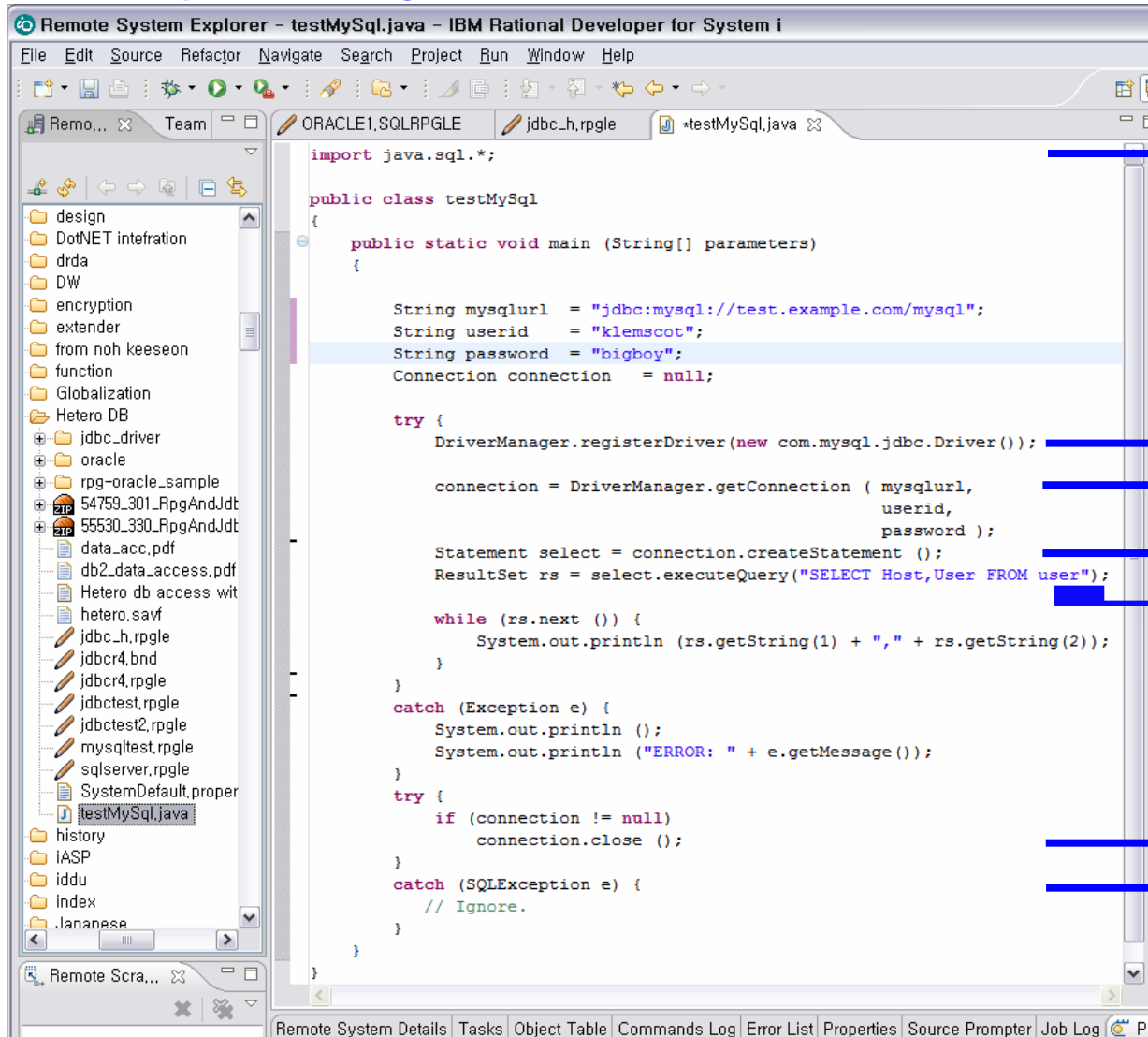
⇒ **Scott Klement** 과 **John Carroll** 은 별도의 Tool(Oracle/400, Information Integrator등) 없이 JDBC를 이용하여 Hetero database와 연동할 수 technique을 소개

⇒ RPG는 java routine을 호출할 수 있는 기능을 포함하고 있고, 대부분의 Database vendor는 jdbc driver를 제공하고 있기 때문에, rpg를 이용하여 Oracle, SQL server, Infomix 그리고 MySql DB 등과 연동을 가능하게 package 화 함.



3. Interoperability without DRDA

◆ Sample JDBC Program



Importing of the java.sql package

Loading the JDBC driver

Connecting to the database

Preparing the SQL statements

Running the SQL statement and
Retrieving the information

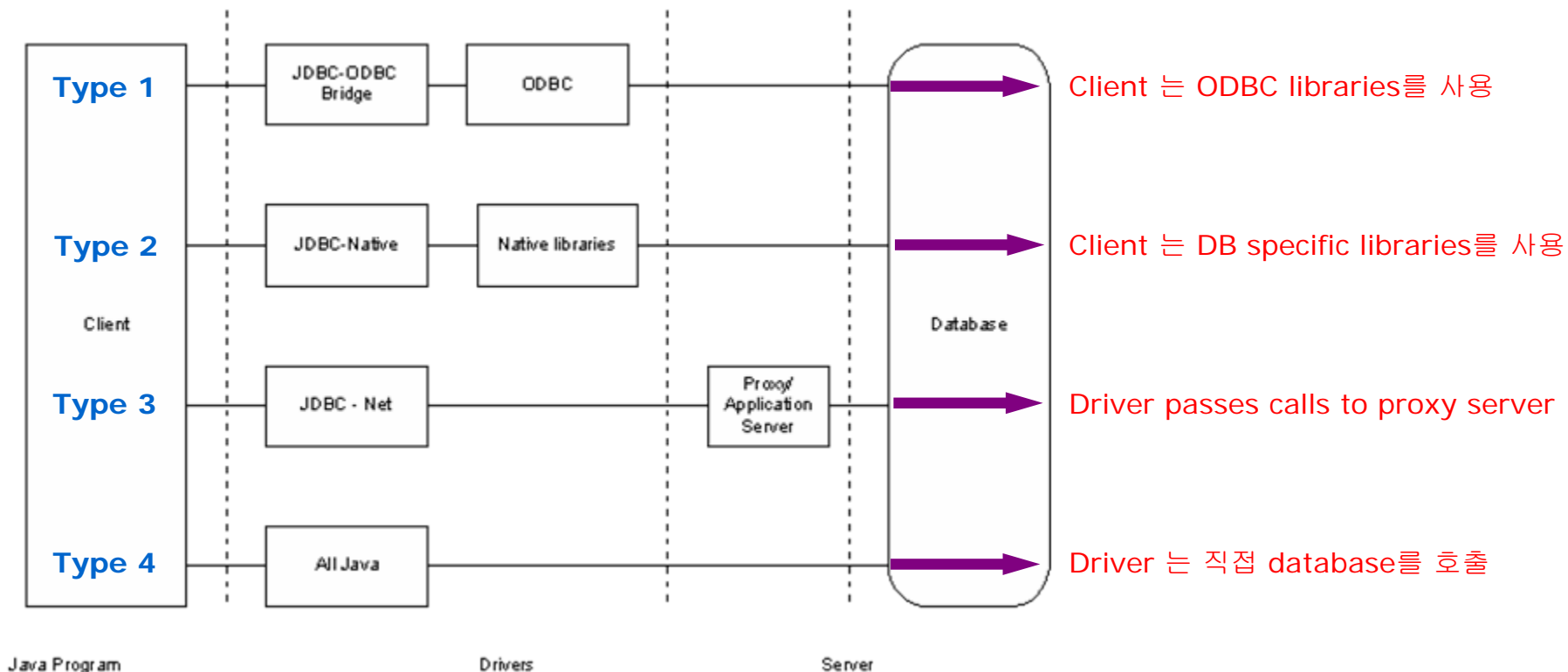
Closing the connection

Handling Errors from the SQL

3. Interoperability without DRDA

◆ JDBC Driver types

- ✓ JDBC 2.0 API : java.sql and javax.sql 를 제공
- ✓ JDBC architecture : JDBC API interfaces 와 classes, driver, database로 구성
- ✓ Driver : Java application 과 database 사이의 매개체로써, Java language를 DB specific language로 mapping 하는 역할



3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ 사전 준비 작업

- ◆ 각 jdbc driver(예: ojdbc14.jar for oracle)를 다운받아 i5 application이 이용할 수 있도록 **/QIBM/UserData/Java400/ext** 폴더에 upload 하거나, 다른 폴더인 경우에는 CLASSPATH로 지정
- ◆ System i에는 1.4 version 이상의 JDK를 설치. 여러 버전의 JDK가 설치되어 있는 경우 QIBM_RPG_JAVA_PROPERTIES 환경변수에 지정
(V5R1 인 경우 SI10069, V5R2 인 경우 SI10101 필요)

Example)

PGM

```
ADDENVVAR ENVVAR(CLASSPATH) +
VALUE('/java/jfreechart/jfreechart-1.0.2.jar:+
      /java/jfreechart/jcommon-1.0.5.jar:+
      /java/poi/poi-3.0.1-FINAL-20070705.jar:+
      /java/poi/poi-contrib-3.0.1-FINAL-20070705.jar:+
      /java/poi/poi-scratchpad-3.0.1-FINAL-20070705.jar:+
      /java/poi/xlparse.jar:+
      /java/jdbc/mysql-connector-java-3.1.12-bin.jar:+
      /java/jdbc/ojdbc14.jar:+
      /java/jdbc/sqljdbc.jar:+ .')
```

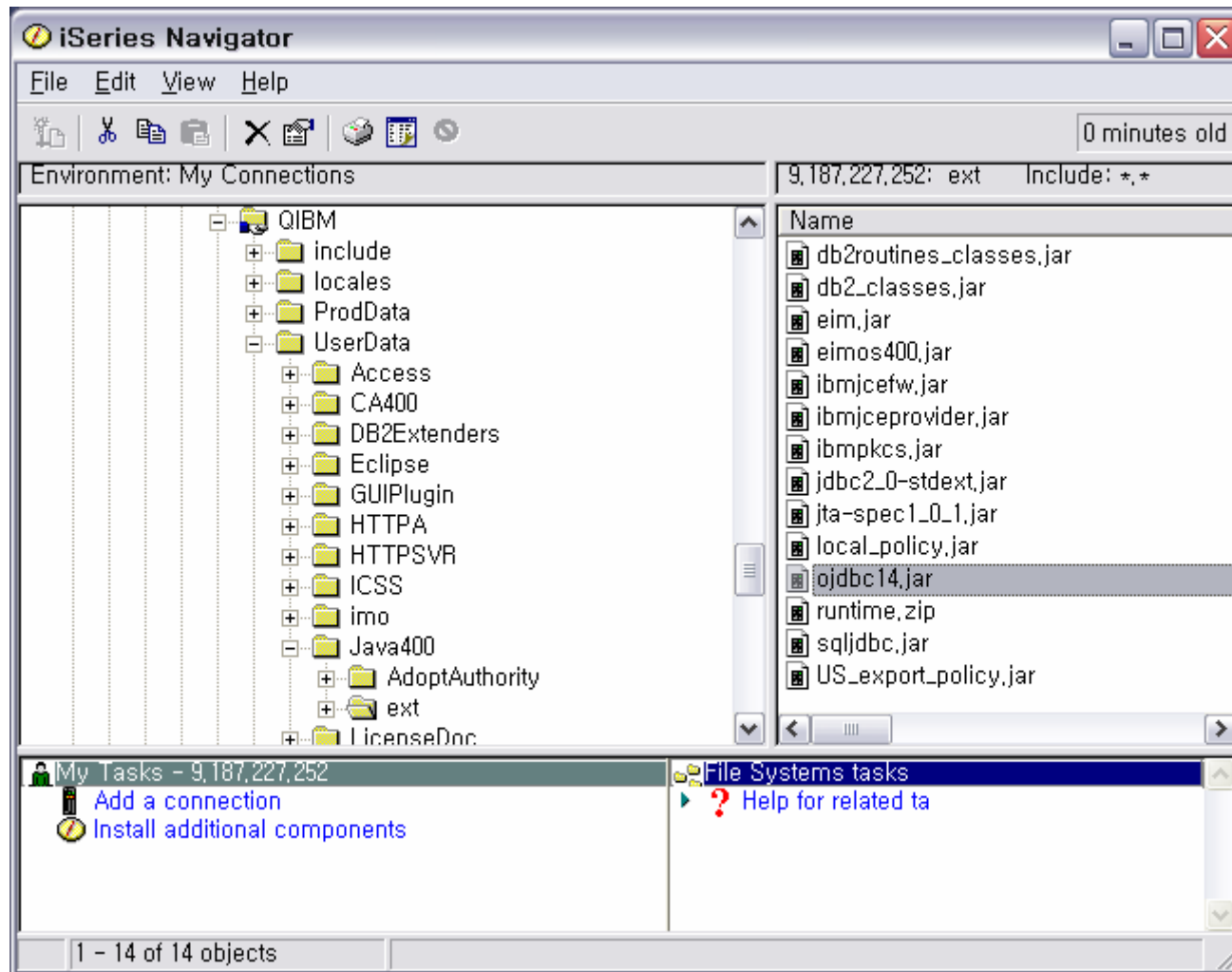
```
ADDENVVAR ENVVAR(QIBM_RPG_JAVA_PROPERTIES) +
      VALUE('-Djava.version=1.4:+
            -Djava.awt.headless=true;+
            -Dos400.awt.native=true;')
```

ENDPGM

3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

- ✓ 사전 준비 작업 - QIBM/UserData/Java400/ext



3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ First Step – Set JDBC Connection Properties

- ◆ 아래 Code는 JDBC R4 service program과 연동되어 jdbc properties를 등록 가능하게 함
- ◆ JDBC R4를 이용함으로써, RPG로 구성되지만, 실제 java와 같은 형태로 코딩이 가능.

Example)

```
/copy JDBC_H
```

```
D user                s 32a
```

```
D passwd             s 32a
```

```
D props                                like(Properties)
```

```
.
```

```
user = 'klemscot';
```

```
passwd = 'bigboy';
```

```
.
```

```
prop = JDBC_Properties();
```

```
JDBC_setProp(prop: 'user' : %trim(userid));
```

```
JDBC_setProp(prop: 'password': %trim(passwd));
```

3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

- ✓ JDBC_H header file

Remote System Explorer - JDBC_H.RPGLE - IBM Rational Developer for System i

File Edit Source Compile(G) Navigate Search Project Run Window Help

ORACLE1.SQLRPGLE ORACLE.SQLRPGLE SQLSERVER.RPGLE JDBC_H.RPGLE

Line 33 Column 43 Replace

```

.....DName+++++ETDsFrom+++To/L+++IDcKeywords+++++Cor
002700
002800      /if defined(MYSQL_H_DEFINED)
002900      /eof
003000      /endif
003100      /define MYSQL_H_DEFINED
003200
003300      D Connection          s          O      CLASS(*JAVA:
003400      D                      'java.sql.Connection')
003500      D ResultSet          s          O      CLASS(*JAVA:
003600      D                      'java.sql.ResultSet')
003700      D PreparedStatement...
003800      D                      s          O      CLASS(*JAVA:
003900      D                      'java.sql.PreparedStatement')
004000      D Properties          s          O      CLASS(*JAVA:
004100      D                      'java.util.Properties')
004200
004300      *+++++
004400      * MySQL_Connect(): Create a connection to a MYSQL server
004500      *

```

3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ JDBC_H header file

```

Remote System Explorer - JDBC_H.RPGLE - IBM Rational Developer for System i
File Edit Source Compile(G) Navigate Search Project Run Window Help
.....1.....2.....3.....4.....5.....6.....7.....8..
006000 *+++++
006100 * JDBC_Connect(): Create a connection using JDBC driver
006200 *
006300 *     driver = (input) Java class name of JDBC driver to use
006400 *     url = (input) JDBC URL of database to connect to
006500 *     userid = (input) userid to log in with
006600 *     password = (input) password to log in with
006700 *
006800 * Returns a connection handle or *NULL upon failure
006900 *+++++
007000 D JDBC_Connect      PR              like(Connection)
007100 D   driver          256A  varying const options(*varsize)
007200 D   url             256A  varying const options(*varsize)
007300 D   userid          50A   varying const options(*varsize)
007400 D   password        50A   varying const options(*varsize)
007500
007600 *+++++
007700 * JDBC_ConnProp(): Connect to database w/properties object
007800 *

```

3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ Second Step – Connecting to Database

- ◆ Database로의 connection은 JDBC4 service program의 JDBC_ConnProp() procedure를 호출하여 수행

- ◆ SQL server database에 연결하는 경우,

Example)

```
conn = JDBC_ConnProp('com.microsoft.sqlserver.jdbc.SQLServerDriver' :  
                    'jdbc:sqlserver://myserver.example.com:1433' :  
                    prop );
```

```
    if (conn = *NULL);
```

```
        return;
```

```
    endif;
```

- ◆ Oracle database에 연결하는 경우도 connection string 부분만 변경해 주면 연결이 가능

```
conn = JDBC_ConnProp('oracle.jdbc.OracleDriver' :  
                    'jdbc:oracle:thin:@123.123.123.123:1521:myDataBase' :  
                    prop );
```

3. Interoperability without DRDA

◆ <http://systeminetwork.com/article/jdbc-rpg-enhancements>

✓ Running SQL Statement

- ◆ 일단 연결이 되고 나면, SQL을 수행할 수 있으며, 이 또한 JDBC R4 service program의 각 procedure를 호출
- ◆ 각 쿼리문의 호출 procedure :

JDBC_ExecUpd(): Runs an SQL statement that does not return any rows or columns as a result. For example, the [Update](#), [Insert](#), [Create Table](#), and [Drop Table](#) SQL statements do not return a result set, because they do not return any rows or columns.

JDBC_ExecQry(): Runs an SQL statement that does return rows and columns
for example, the SQL [Select statement to retrieve records](#) from a database.

JDBC_PrepStmt(): Prepares an SQL statement to be executed later.

Prepared statements can use parameter markers (question marks) to indicate a spot where data will be inserted from variables. After a prepared statement has been created, you can use [JDBC_SetString\(\)](#), [JDBC_setInt\(\)](#), [JDBC_SetDecimal\(\)](#), [JDBC_SetDate\(\)](#), and so forth, to set the values of the parameter markers from variables in your program. You can use the [JDBC_ExecPrepUpd\(\)](#) and [JDBC_ExecPrepQry\(\)](#) procedures to run prepared statements.

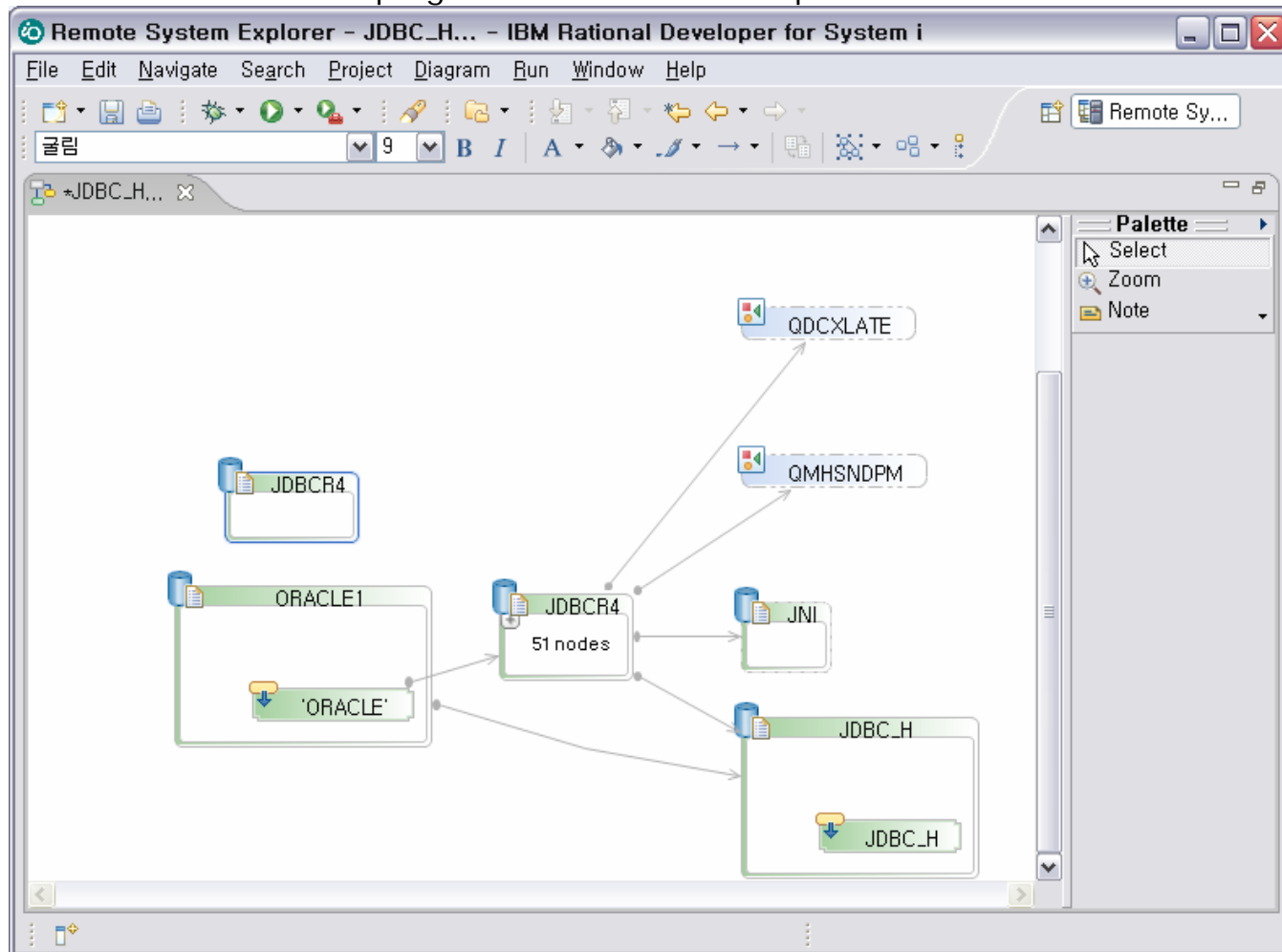
JDBC_PrepCall(): call a [stored procedure](#) instead of running a standard SQL statement

4. Implementation

◆ Program Relations – Real Example

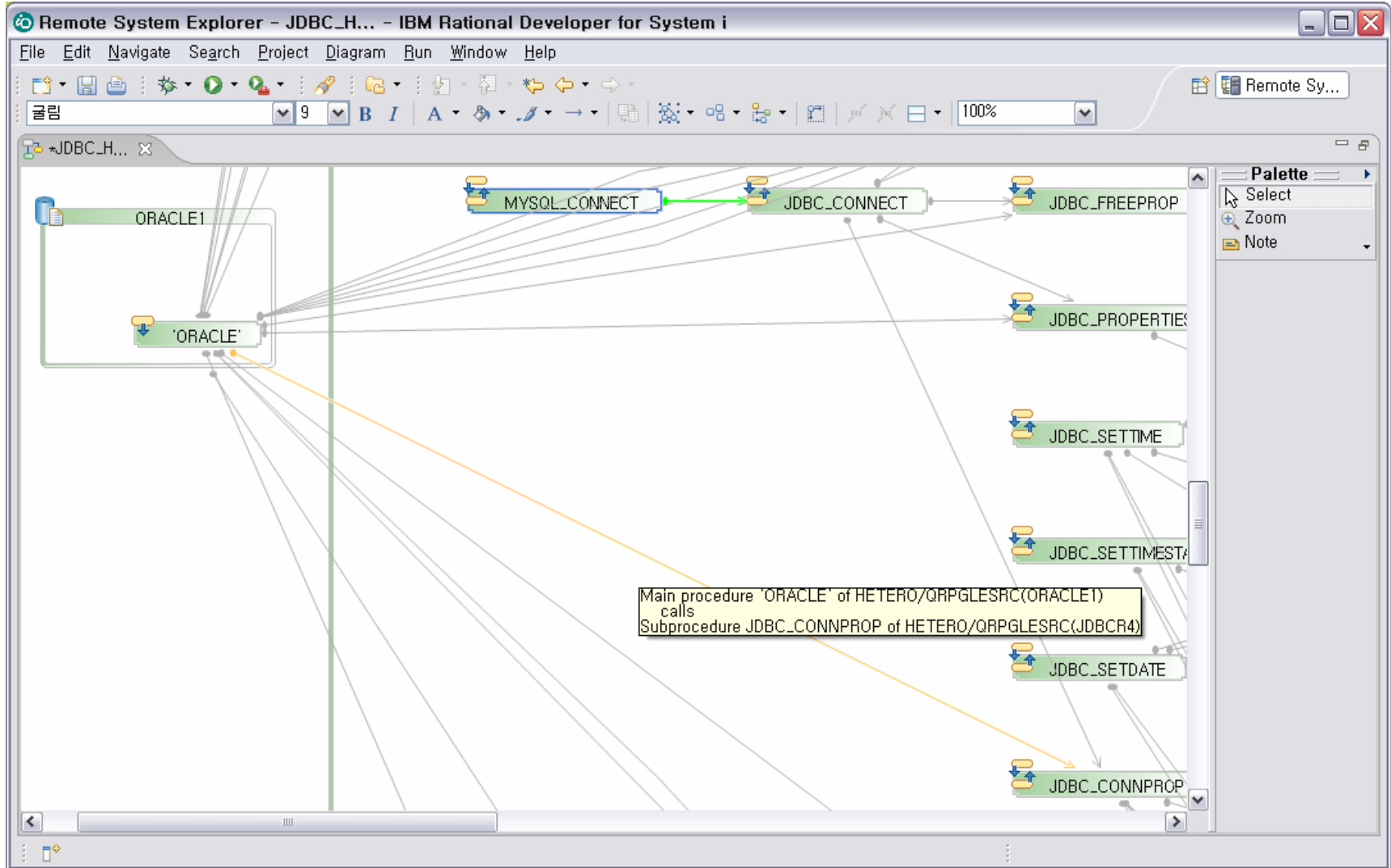
✓ 새로 생성되는 **ORACLE1** 프로그램

◆ Oracle1은 JDBC4 service program에 등록되어 있는 각 procedure를 이용하여 Oracle DB와 연동



4. Implementation

◆ Program Relations – Real Example



4. Implementation

◆ Program Relations – Real Example

✓ Scott Klement이 제공하는 JDBCRC service program 등록절차

- ◆ Library 위치를 실제 사용할 name으로 변경하여 아래 순서에 의해 service program으로 compile

Remote System Explorer - JDBCRC4.RPGLE - IBM Rational Developer for System i

File Edit Source Compile(G) Navigate Search Project Run Window Help

ORACLE1 *ORACLE1 ORACLE1.SQLRPGLE JDBCRC4.RPGLE

Line 45 Column 32 Replace

```

..1...+...2...+...3...+...4...+...5...+...6...+...7...+...
004500
004600 To Compile:
004700 CRTRPGMOD mylib/JDBCRC4 SRCFILE(xxx/QRPGLESRC) DBGVIEW(*LIST)
004800 ( Note: this is the JDBCRC4.rpgle source member. )
004900
005000 CRTSRVPGM SRVPGM(mylib/JDBCRC4) +
005100 EXPORT(*SRCFILE) SRCFILE(xxx/QRVSRVC)
005200 ( Note: this is the JDBCRC4.bnd source member. )
005300
005400 CRTBNDDIR BNDDIR(mylib/JDBC)
005500 ADDBNDDIRE BNDDIR(mylib/JDBC) OBJ((JDBCRC4 *SRVPGM))
005600
005700 -
  
```

Remote System Details Tasks Object Table Commands Log Error List Properties

4. Implementation

◆ Program Relations – Real Example

✓ Scott Klement이 제공하는 JDBCRC service program 등록절차

- ◆ JDBCRC4 service program을 이용하게 될 프로그램을 만드는 경우 아래와 같이 compile

```

Remote System Explorer - ORACLE1.SQLRPGL - IBM Rational Developer for System i
File Edit Source Compile(G) Navigate Search Project Run Window Help
ORACLE1 JDBCRC4 *ORACLE1 ORACLE1.SQLRPGL x
Line 21 Column 1 Replace
.....1.....2.....3.....4.....5.....6.....7.....
000100 * Sample of accessing an SQL Server database. This was posted to
000200 * the SystemiNetwork.com forums by Jon Juracich in the following
000300 * thread:
000400 * http://www.systeminetwork.com/isnetforums/showthread.php?t=48063
000500 *
000600 * To compile:
000700 * - Make sure you change the connection string to the
000800 * correct TCP/IP domain name and port for your system.
000900 * - Make sure you set the user-id and password appropriately
001000 *
001100 * CRTBNDRPG SQLSERVER SRCFILE (xxx/QRPGLESRC) DBGVIEW (*LIST)
001200 *
001300 *
001400 H*DFTACTGRP (*NO)
001500 H OPTION (*NODEBUGIO:*SRCSTMT)
001600 H BNDDIR ('JDBC')
001700
001800 FQSYSPT O F 132 PRINTER
001900
002000 /copy jdbc_h
002100
Remote System Details Tasks Object Table Commands Log Error List Properties

```

4. Implementation

◆ Program Relations – Real Example

✓ Data structure를 이용한 result set 선언

- ◆ DB2 for i5/OS 에서 사번 컬럼과 Oracle로 부터 이름 급여 컬럼을 함께 result set으로 구성

The screenshot shows the Remote System Explorer interface for the program ORACLE1.SQLRPGLE. The main editor displays the following COBOL code snippet:

```

Line 33      Column 1      Replace
...DName+++++ETDsFrom+++To/L+++IDc.Keywords+++++
003000      *****
003100      *Array for result set
003200      *****
003300      Di          s          3  0
003400      Dempinf     ds          occurs (5)
003500      Dempno
003600      Dempname   20a
003700      D salarya  5I  0
003800      *
003900      D ErrMsg   s          50A
004000      D wait     s          1A
004100      D count   s          10I  0
004200      *
004300      D empno1   s          5A
004400      D empname1 s          20A
004500      D salary1  s          5I  0
004600      D count1  s          1p  0
004700      *
004800      D conn     s          like (Connection)
004900      D rs       s          like (ResultSet)
005000      D prop     s          like (Properties)
  
```

The data structure declaration for the result set is highlighted with a red box:

```

003300      Di          s          3  0
003400      Dempinf     ds          occurs (5)
003500      Dempno
003600      Dempname   20a
003700      D salarya  5I  0
  
```

4. Implementation

◆ Program Relations – Real Example

✓ Properties 등록과 Oracle Database 로의 연결 과정

- ◆ Scott과 John 이 제시한 방법대로 간단하게 properties 등록과 연결 부분을 선언

```

Remote System Explorer - ORACLE1.SQLRPGL - IBM Rational Developer for System i
File Edit Source Compile(G) Navigate Search Project Run Window Help
ORACLE1 JDBCR4 *ORACLE1 ORACLE1.SQLRPGL x
Line 65 Column 1 Replace
.....1.....2.....3.....4.....5.....6.....7.....
006400 // *****
006500 // 1. Set up JDBC Properties
006600 // *****
006700
006800 prop = JDBC_Properties();
006900 JDBC_setProp(prop: 'user' : userid);
007000 JDBC_setProp(prop: 'password' : passwd);
007100 JDBC_setProp(prop: 'prompt' : 'false');
007200 JDBC_setProp(prop: 'errors' : 'full');
007300 JDBC_setProp(prop: 'naming' : 'system');
007400
007500 // *****
007600 // 2. Connection String to Oracle
007700 // *****
007800 conn = JDBC_ConnProp('oracle.jdbc.OracleDriver'
007900 : 'jdbc:oracle:thin:@130.1.1.85:1521:IBM'
008000 : prop );
008100
008200 JDBC_freeProp(prop);
008300
008400 if (conn = *NULL);
  
```

4. Implementation

◆ Program Relations – Real Example

✓ Oracle에서 수행될 SQL 준비

- ◆ Prepare statement로 실행될 쿼리문을 준비하고, i5/OS에서 사번을 조회하기 위한 routine 추가

The screenshot shows the 'Remote System Explorer' window for 'ORACLE1.SQLRPGL' in IBM Rational Developer for System i. The main editor displays COBOL code with two sections highlighted in red boxes:

```

Line 89      Column 1      Replace
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
008700      //*****
008800      // 3. Prepare SQL statement string to select Employee name
008900      //   from Oracle database.
009000      //*****
009100      prepstm = JDBC_prepStmt (conn:
009200      'SELECT * from scott.ora_remote where ora_empno = ?');
009300
009400      if (prepstm = *NULL);
009500          jdbc_close(conn);
009600          return;
009700      endif;
009800          eval i = 0;
009900
010000      //*****
010100      // 4. Retrieving Employee number from System i
010200      //*****
010300      exec sql Declare empcur1 cursor for
010400          select empno400 from leelib/ora_test1;
010500      exec sql open empcur1;
010600          exec sql fetch from empcur1 into :empno1;
010700      dow (sqlcode <> 100);
          eval i = i + 1;
  
```

4. Implementation

◆ Program Relations – Real Example

✓ Oracle에서 SQL 실행

- ◆ I5/OS에서 조회된 각 사번을 parameter로 받아 Oracle로 쿼리 수행 및 결과를 받아 data structure에 저장

The screenshot shows the Remote System Explorer interface for an Oracle database. The main window displays a code editor with the following SQL code:

```

Line 118      Column 1      Replace
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
011200      //*****//
011300      // 5. Query the oracle database with employee number from System i
011400      //*****//
011500      JDBC_SetString(prepestm:1:empno1);
011600      rs = jdbc_ExecPrepQry(prepestm);
011700      dow (jdbc_nextRow(rs));
011800          empname1 = jdbc_getCol(rs: 2);
011900          salary1 = %Int(jdbc_getCol(rs: 3));
012000          eval empname = empname1;
012100          eval salarya = salary1;
012200      // except;
012300      enddo;
012400      exec sql fetch from empcur1 into :empno1;
012500      enddo;
012600
012700      JDBC_FreePrepStmt(prepestm);
012800      JDBC_FreeResult(rs);
012900      JDBC_Close(conn);
013000
013100      exec sql close empcur1;
013200      exec sql set result sets array :empinf for :i rows;
013300      return;
  
```

The code is enclosed in a red box, highlighting the main query execution logic. The interface also shows a project browser on the left and a status bar at the bottom with tabs for Remote System Details, Tasks, Object Table, Commands Log, Error List, and Properties.

4. Implementation

◆ Program Relations – Real Example

✓ Module 생성

- ◆ PGM object 전 단계로 module을 생성하고, 이 때 COMMIT mode를 설정

SQL ILE RPG 오브젝트 작성 (CRTSQLRPGI)

오브젝트 :	> ORACLE1	이름
라이브러리 :	> HETERO	이름
소스 파일 :	> QRPGLSRC	이름, QRPGLSRC
라이브러리 :	> HETERO	이름
소스 멤버 :	> ORACLE1	이름
확약 제어 :	*NONE	
관계형 데이터베이스 :	*LOCAL	이름
컴파일 유형 :	*MODULE	
출력 리스트 :	*PRINT	
텍스트 '설명' :	*SRCMBRTXT	문자 값

사전컴파일러 옵션 :

Advanced Parameters

>		Add
*EVENTF		Remove
		Move up
		Move down

대체 :

> *YES

보기 디버깅 :

> *SOURCE

Advanced All Parameters Keywords

CRTSQLRPGI OBJ(HETERO/ORACLE1) SRCFILE(HETERO/QRPGLSRC) SRCMBR(ORACLE1) COMMIT(*NONE)
 OBJTYPE(*MODULE) OUTPUT(*PRINT) OPTION(*EVENTF) REPLACE(*YES) DBGVIEW(*SOURCE)

OK Restore defaults Cancel

4. Implementation

◆ Program Relations – Real Example

✓ PGM object 생성

- ◆ Stored Procedure를 호출하는 client application과 activation group을 같도록 *CALLER로 설정

Property	Value	Description
프로그램 :	PGM	ORACLE1 (이름)
라이브러리 :		+CURLIB (이름)
모듈 :	MODULE	+PGM (이름, 총칭*)
라이브러리 :		+LIBL (이름)
텍스트 '설명' :	TEXT	+ENTMODTXT (문자 값)
Advanced Parameters		
프로그램 입력 프로시저 모듈 :	ENTMOD	+FIRST (이름)
라이브러리 :		+LIBL (이름)
결합 서비스 프로그램 :	BNDSRVPGM	+NONE (이름, 총칭*)
라이브러리 :		+LIBL (이름)
바인딩 디렉토리 :	BNDDIR	+NONE (이름)
라이브러리 :		+LIBL (이름)
활성 그룹 :	ACTGRP	*CALLER (이름)
작성 옵션 :	OPTION	Add(H)
세부사항 나열 :	DETAIL	+NONE
경신 허용 :	ALWUPD	+YES
*SRVPGM 라이브러리 경신허용 :	ALWLIBUPD	+NO
사용자 프로파일 :	USRPRF	+USER
프로그램 대체 :	REPLACE	+YES
권한 :	AUT	+LIBCRTAUT (이름)
목표 릴리스 :	TGTRLS	+CURRENT
재초기화 허용 :	ALWRINZ	+NO
기억영역 모듈 :	STGMDL	+SINGLVL
프로시저 간의 분석 :	IPA	+NO
IPA 제어 파일 :	IPACTLFILE	+NONE (경로명)

Advanced(L)
 All Parameters
 Keywords

CRTPGM PGM(ORACLE1) ACTGRP(*CALLER)

4. Implementation

◆ Program Relations – Real Example

✓ Client Application을 위한 Stored Procedure 작성

- ◆ 앞의 Oracle1을 client application에서 호출하기 위해 stored procedure로 작성
- ◆ Input parameter로는 oracle connection을 위한 user id와 password를 선언
- ◆ Procedure 호출 결과 반환된 result set을 받을 수 있도록 선언

```
CREATE PROCEDURE HETERO.ORACLE1 (  
    IN UID CHAR(5) ,  
    IN PWD CHAR(5) )  
    DYNAMIC RESULT SETS 1  
    LANGUAGE RPGLE  
    SPECIFIC HETERO.ORACLE1  
    NOT DETERMINISTIC  
    MODIFIES SQL DATA  
    CALLED ON NULL INPUT  
    EXTERNAL NAME 'HETERO/ORACLE1'  
    PARAMETER STYLE SQL ;
```

4. Implementation

◆ Program Relations – Real Example

✓ External Stored Procedure 호출 test

- ◆ Run-Sql-Script session을 이용 stored procedure를 호출
- ◆ 반환된 result set에서 사번 컬럼은 i5/OS로부터, 각 사번에 해당하는 이름과 급여는 Oracle로 부터 구성

Examples

```
call hetero.oracle1('scott', 'tiger');
```

EMP...	EMPNAME	SALARYA
A1111	홍길동	2500
B2222	일지매	3500
C3333	코난	5000
D4444		0

4. Implementation

◆ Data type Mapping – DB2 vs. Oracle

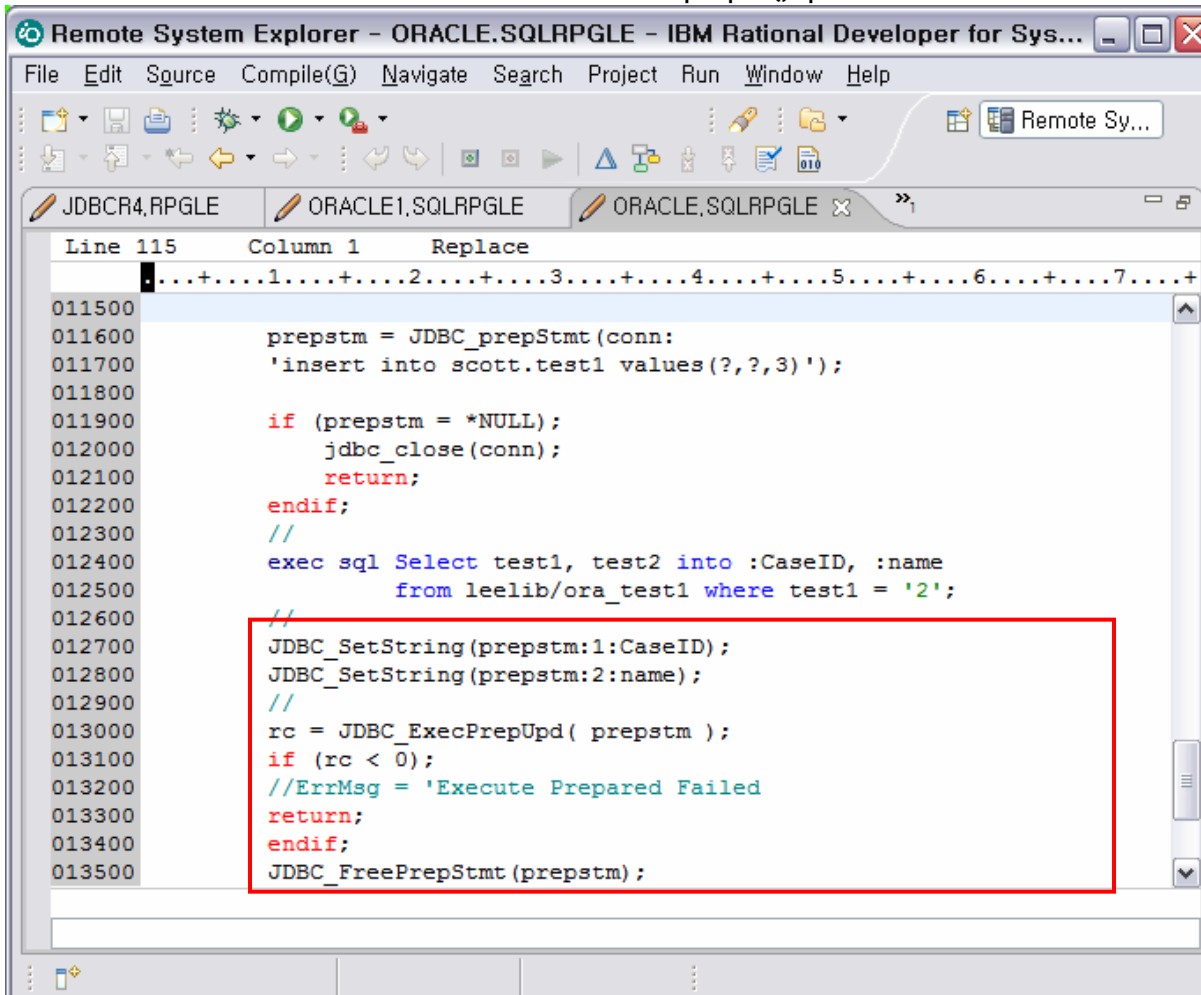
Oracle Data Type	Notes	DB2 UDB Data Type	Notes
DATE		DATE TIME TIMESTAMP	<ul style="list-style-type: none"> — if only MM/DD/YYYY required, use DATE — if only HH:MM:SS required, use TIME — if both date and time required (MM/DD/YYYY-HH:MM:SS.000000), use TIMESTAMP — Use Oracle TO_CHAR() function to format a DATE for subsequent DB2 UDB load. Note that the Oracle default DATE format is DD-MON-YY
VARCHAR2 (n)	n <= 4000	CHAR(n) VARCHAR(n)	n <= 32766, CHAR n <= 32740, VARCHAR
LONG	n <= 2GB	VARCHAR (n) CLOB(n)	<ul style="list-style-type: none"> — if n <= 32K bytes, use CHAR or VARCHAR — if 32K <= n <= 2 GB, use CLOB
RAW & LONG RAW	n <= 255	CHAR(n) FOR BIT DATA VARCHAR(n) FOR BIT DATA BLOB(n)	<ul style="list-style-type: none"> — if n <= 32K, use CHAR(n) FOR BIT DATA or VARCHAR(n) FOR BIT DATA — if n <= 2 GB, use BLOB(n)
BLOB	n <= 4 GB	BLOB(n)	— if n <= 2GB use BLOB(n)
CLOB	n <= 4 GB	CLOB(n)	— if n <= 2GB use CLOB(n)
NCLOB	n <= 4GB	DBCLOB(n)	— if n <= 2GB use DBCLOB(n/2)
NUMBER		SMALLINT / INTEGER / BIGINT DECIMAL(p,s)/NUMERIC(p,s) FLOAT(n) / REAL / DOUBLE	<ul style="list-style-type: none"> — if Oracle decl is NUMBER(p) or NUMBER(p,0), use SMALLINT / INTEGER/ BIGINT — if Oracle decl is NUMBER(p,s), use DECIMAL(p,s) — if Oracle decl is NUMBER, use FLOAT(n) / REAL / DOUBLE

4. Implementation

◆ Program Relations – Real Example

✓ Row insert into Oracle database

- ◆ Update, insert, delete 인 경우, JDBC_ExexPrepUpd() procedure를 호출



```
Remote System Explorer - ORACLE.SQLRPGLE - IBM Rational Developer for Sys...
File Edit Source Compile(G) Navigate Search Project Run Window Help
JDBCRC4.RPGLE ORACLE1.SQLRPGLE ORACLE.SQLRPGLE x
Line 115 Column 1 Replace
...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+
011500
011600     prepstm = JDBC_prepStm(conn:
011700     'insert into scott.test1 values(?,?,3)');
011800
011900     if (prepstm = *NULL);
012000         jdbc_close(conn);
012100         return;
012200     endif;
012300     //
012400     exec sql Select test1, test2 into :CaseID, :name
012500         from leelib/ora_test1 where test1 = '2';
012600     //
012700     JDBC_SetString(prepsm:1:CaseID);
012800     JDBC_SetString(prepsm:2:name);
012900     //
013000     rc = JDBC_ExecPrepUpd( prepstm );
013100     if (rc < 0);
013200         //ErrMsg = 'Execute Prepared Failed
013300         return;
013400     endif;
013500     JDBC_FreePrepStm( prepstm );
```

◆ Summary

✓ **Java method**를 등록, 실행할 수 있는 RPG의 향상된 기능 적용

⇒ System i에서 사용되는 host application은 Java application을 호출 할 수 있을 뿐 아니라, RPG의 경우 각각의 JDBC driver를 이용할 수 있는 java method를 등록할 수 있는 기능이 존재

⇒ Scott과 John에 의해 구현된 service program JDBCR4 을 이용하면, 별도의 tool 이 필요없이 Hetero database로의 연동이 가능하며, 하나의 result set 안에 Oracle, SQL server등으로 부터 관련된 데이터를 취합하여 구성하는 것이 가능

✓ **DRDA**를 이용한 **DB2** 간의 연동

⇒ IBM DB2 for Linux, Unix, Windows, i5/OS 그리고 z/OS 간에는 DRDA라는 protocol을 이용하여 상대 서버와의 연동 및 쿼리 수행이 가능

Reference

<http://systeminetwork.com/article/jdbc-rpg-enhancements>

❖ **Special Notes** from Scott Klement

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Q&A



감사합니다.