Pawel DRZYMALA, Henryk WELFLE

Lodz University of Technology

Compatibility, compilation and execution of PL/SQL code on the database system IBM DB2 10

***Abstract**. The article presents the support mechanisms to run applications written in PL/SQL Oracle DB2 environment version 10.1. It presents possibilities compile and execute applications written in PL/SQL using SQL, PL/SQL and Oracle objects. This version implements new elements of support (compatibility) for Oracle in comparison to the previous versions. Paper presents a detailed possibilities of support from DB2 for PL/SQL language and shows how to configure DB2 to work with the language PL/SQL.*

***Streszczenie.** W artykule przedstawiono mechanizmy wsparcia uruchamiania aplikacji napisanych w języku PL/SQL Oracle w środowisku wersji DB2 10.1. Zaprezentowano możliwości kompilacji i wykonywania aplikacji napisanych w języku PL/SQL z użyciem języka SQL, PL/SQL i obiektów Oracle. W wersji tej wprowadzono nowe elementy wsparcia (kompatybilności) dla systemu Oracle w stosunku do poprzednich wersji. Artykuł przedstawia szczegółowe możliwości wsparcia ze strony DB2 dla języka PL/SQL. (**Kompatybilność, kompilacja i uruchamianie kodu PL/SQL po stronie systemu baz danych IBM DB2 10**).*

Keywords: IBM DB2 10.5, Oracle, PL/SQL, DB2_COMPATIBILITY_VECTOR.
Słowa kluczowe: IBM DB2 10.5, Oracle, PL/SQL, DB2_COMPATIBILITY_VECTOR.

doi:10.12915/pe.2014.04.05

Introduction

Database management system IBM DB2 10.1 is designed for managing data in traditional relational structures, as well as a hierarchy XML. DB2 allows to operate Oracle SQL language dialects and PL / SQL (Oracle version 10g, 11g and 12c). Therefore, many applications written for Oracle database management after some changes may be running in the DB2 environment (given in the literature estimated that about 97% of the syntax of PL / SQL, and about 91% of the objects PL / SQL can be directly transferred to DB2 - indicative values are given on the basis of the report generated tool MEET DB2).

It should be noted that, despite the transferability of PL/SQL code for DB2 are not implemented all the features of the language, but each new version of DB2 introduces new features such as object-relational programming (e.g. objects NESTED TABLE). This is an important element affecting feasibility of migrating data and applications between systems. Users who work with the Oracle database can use DB2 of the many new levels of compatibility. The purpose of these changes is the ability to quickly be run on DB2 applications written in PL / SQL.

.

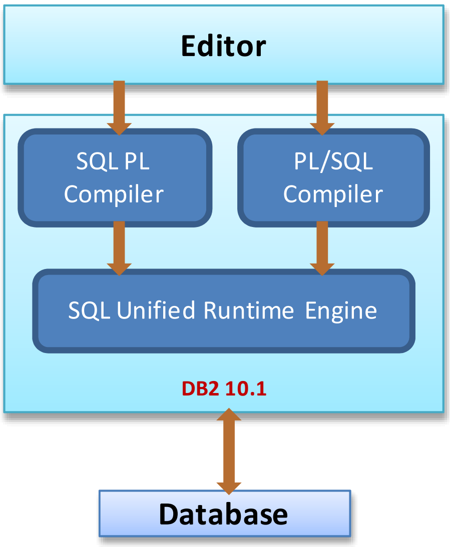
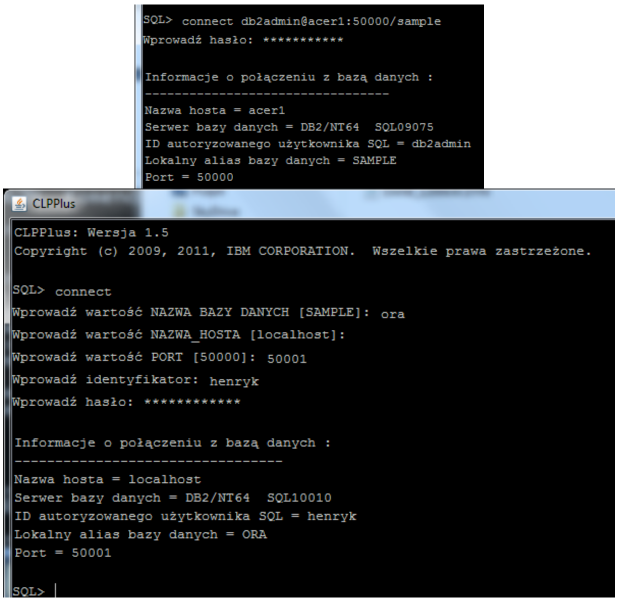


Fig. 1. Support for the language PL/SQL

DB2 can support syntax of PL/SQL and SQL PL. The method of support for PL/SQL and SQL PL is shown in Figure 1.

In order to ensure compatibility with Oracle version of DB2 uses CLPPlus command-line processing utility that allows to run SQL commands and PL / SQL code, and is similar to an existing DB2 command line processor (CLP). CLPPlus environment is presented in figure 2.



```
SQL> connect db2admin@acer1:50000/sample
Wprowadź hasło: ***********

Informacje o połączeniu z bazą danych :
--------------------------------
Nazwa hosta = acer1
Serwer bazy danych = DB2/NT64  SQL09075
ID autoryzowanego użytkownika SQL = db2admin
Lokalny alias bazy danych = SAMPLE
Port = 50000
```

```
CLPPlus: Wersja 1.5
Copyright (c) 2009, 2011, IBM CORPORATION.  Wszelkie prawa zastrzeżone.

SQL> connect
Wprowadź wartość NAZWA BAZY DANYCH [SAMPLE]: ora
Wprowadź wartość NAZWA_HOSTA [localhost]:
Wprowadź wartość PORT [50000]: 50001
Wprowadź identyfikator: henryk
Wprowadź hasło: ************

Informacje o połączeniu z bazą danych :
---------------------------------
Nazwa hosta = localhost
Serwer bazy danych = DB2/NT64  SQL10010
ID autoryzowanego użytkownika SQL = henryk
Lokalny alias bazy danych = ORA
Port = 50001

SQL> |
```

Fig. 2. Environment CLPPlus

Support for most data types of PL / SQL is implemented as a standard on DB2, and for others data types can be enabled through the configuration of registry variable. The registry variable DB2_COMPATIBILITY_VECTOR, briefly described later in this article, can control which of these compatibility is enabled by setting.

DB2 setting to allow the PL / SQL use in order to perform Oracle application

To run Oracle applications on DB2, it must be configured properly. This is done by changing the registry set: DB2_COMPATIBILITY_VECTOR using the db2set command. This registry variable is used to enable one or more DB2 compatibility features. Full compatibility level is achieved by assigning a registry variable DB2_COMPATIBILITY_VECTOR for value ORA. Setting the registry variable to enable handling all compatibility features of Oracle is shown in the following example:

```
    db2set DB2_COMPATIBILITY_VECTOR=ORA
    db2stop
    db2start
```

Setting the registry variable to disable handling all compatibility features of Oracle is shown in the following example:

```
    db2set DB2_COMPATIBILITY_VECTOR=
    db2stop
    db2start
```

The registry variable: DB2_COMPATIBILITY_VECTOR

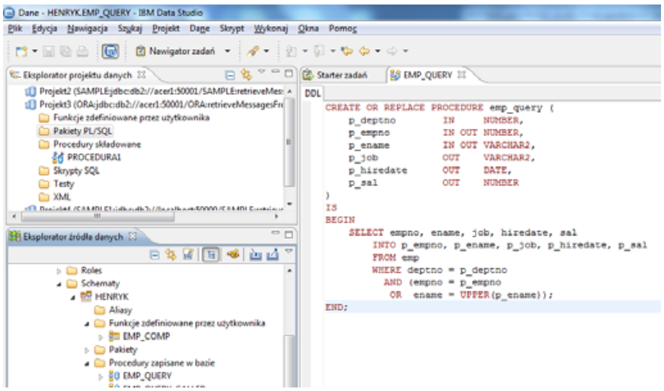
DB2_COMPATIBILITY_VECTOR registry variable can set various functions that exist in the Oracle database. (not all apply to Oracle, some of which applies to Sybase and SQL Server), depending on customer needs. It should be noted that some functionality (marked (*) in Table 1) can be included but are not to be used in existing databases. This would only apply to newly created databases. The value of the registry variable is given as a hexadecimal value, or uses the value of the ORA to include all functionality. The new registry variable setting will not take effect until you stop the instance and restart it.

Table 1. DB2_COMPATIBILITY_VECTOR registry variable values [5]

Bit position	Compatibility feature	Description
1	ROWNUM pseudocolumn	Enables the use of ROWNUM as a synonym for ROW_NUMBER() OVER() and permits ROWNUM to appear in the WHERE clause of SQL statements.
2	DUAL table	Resolves unqualified references to the DUAL table as SYSIBM.DUAL.
3	Outer join operator	Enables support for the outer join operator, which is the plus sign (+).
4	Hierarchical queries	Enables support for hierarchical queries using the CONNECT BY clause.
5	NUMBER data type (*)	Enables support for the NUMBER data type and associated numeric processing.
6	VARCHAR2 data type (*)	Enables support for the VARCHAR2 and NVARCHAR2 data types and associated character string processing.
7	DATE data type (*)	Enables the interpretation of the DATE data type as the TIMESTAMP(0) data type, a combined date and time value.
8	TRUNCATE TABLE	Enables alternative semantics for the TRUNCATE statement, under which IMMEDIATE is an optional keyword that is assumed to be the default if not specified.
9	Character literals	Enables the assignment of the CHAR or GRAPHIC data type, instead of the VARCHAR or VARGRAPHIC data type, to character and graphic string constants whose byte lengths are less than or equal to 254.
10	Collection methods	Enables the use of methods to perform operations on arrays, such as first, last, next, and previous.
11	Oracle data dicticonary-compatible views (*)	Enables the creation of Oracle data dictionary-compatible views.
12	PL/SQL compilation	Enables the compilation and execution of PL/SQL statements and language elements.
13	Insensitive cursors	Enables cursors that are defined with WITH RETURN to be insensitive if the select-statement does not explicitly specify FOR UPDATE.
14	INOUT parameters	Enables the specification of DEFAULT for INOUT parameter declarations.
15	LIMIT and OFFSET clauses	Enables the use of the MySQL- and PostgreSQL-compatible LIMIT and OFFSET clauses on fullselect, UPDATE, and DELETE statements.
17	SQL data-access-level enforcement	Enables routines to enforce SQL data-access levels at run time.
18	Oracle database link syntax	Enables Oracle database link syntax for accessing objects in other databases.

Entered elements of the PL/SQL language, compilation and execution

PL/SQL code can be compiled and executed using DB2 interfaces. This allows the use of existing solutions written in PL/SQL, which ultimately will work with DB2 data server (support of PL / SQL must be included on the DB2 server by listed earlier registry DB2_COMPATIBILITY_VECTOR). Supported interfaces include command line processor (CLP), CLPPlus and IBM Data Studio (Figure 3). Enabling this parameter practically provides the most functionality for PL/SQL code.


Dane - HENRYK.EMP_QUERY - IBM Data Studio
Plik Edycja Nawigacja Szukaj Projekt Dane Skrypt Wykonaj Okna Pomoc
Nawigator zadań
Eksplorator projektu danych
Projekt2 (SAMPLE:jdbc:db2://acer1:50001/SAMPLE:retrieveMes
Projekt3 (ORA:jdbc:db2://acer1:50001/ORA:retrieveMessagesFr
Funkcje zdefiniowane przez użytkownika
Pakiety PL/SQL
Procedury składowane
PROCEDURA1
Skrypty SQL
Testy
XML
Eksplorator źródła danych
Roles
Schematy
HENRYK
Aliasy
Funkcje zdefiniowane przez użytkownika
EMP_COMP
Pakiety
Procedury zapisane w bazie
EMP_QUERY
Starter zadań
EMP_QUERY
DDL

```
CREATE OR REPLACE PROCEDURE emp_query (
    p_deptno        IN      NUMBER,
    p_empno         IN OUT  NUMBER,
    p_ename         IN OUT  VARCHAR2,
    p_job           OUT     VARCHAR2,
    p_hiredate      OUT     DATE,
    p_sal           OUT     NUMBER
)
IS
BEGIN
    SELECT empno, ename, job, hiredate, sal
        INTO p_empno, p_ename, p_job, p_hiredate, p_sal
        FROM emp
        WHERE deptno = p_deptno
          AND (empno = p_empno
           OR  ename = UPPER(p_ename));
END;
```

Fig. 3. Tools DataStudio

There are also restrictions to the language PL/SQL:

- PL/SQL procedures, functions, triggers and packages can only be created from the main partition in a partitioned database environment (DPF - Database Partitioning Feature),
- The NCLOB data type is not supported for use in PL/SQL statements or in PL/SQL contexts when the database is not defined as a Unicode database. In Unicode databases, the NCLOB data type is mapped to a DBCLOB data type,
- The XMLTYPE data type is not supported.

We can execute the following PL/SQL statements:

- Anonymous blocks
  for example:
  SET SERVEROUTPUT ON;
  BEGIN
  dbms_output.put_line('Hello');
  END;

The anonymous block statement, which does not persist in the database, can consist of up to three sections: an optional declaration section, a mandatory executable section, and an optional exception section.

- CREATE OR REPLACE FUNCTION statement (Fig. 4)
- CREATE OR REPLACE PACKAGE statement (Fig. 7)
- CREATE OR REPLACE PACKAGE BODY statement (Fig. 8)
- CREATE OR REPLACE PROCEDURE statement (Fig. 5)
- CREATE OR REPLACE TRIGGER statement
- CREATE OR REPLACE TYPE statement
- DROP PACKAGE statement
- DROP PACKAGE BODY statement

PL/SQL procedures and functions can be invoked from other PL/SQL statements or from DB2 SQL PL statements. We can call a PL/SQL procedure from SQL PL by using the CALL statement (Fig. 6).

The following statements and language elements are supported in PL/SQL contexts:

- Type declarations:
  - Associative arrays
  - Record types
  - VARRAY types
- Subtype declarations
- Variable declarations:
  - %ROWTYPE
  - %TYPE
- Basic statements, clauses, and statement attributes:
  - Assignment statement
  - NULL statement
  - RETURNING INTO clause
  - Statement attributes, including SQL%FOUND, SQL%NOTFOUND, and SQL%ROWCOUNT
- Control statements and structures:
  - Exception handling
  - CASE, EXIT, FOR, GOTO, IF, LOOP, RETURN, WHILE statement
- Static cursors:
- REF CURSOR support:
- Error support:
  - RAISE_APPLICATION_ERROR procedure
  - RAISE statement
  - SQLCODE function
  - SQLERRM function

These elements of the language are very important. Many these elements PL / SQL used in practice in Oracle.

We can create PL/SQL procedures and functions from a DB2 command line processor (CLP) script.

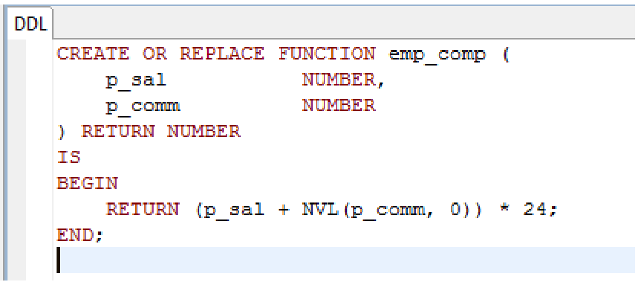
For this purpose formulate PL/SQL procedure or function definitions within a CLP script file. Save the file. Execute the script from the CLP. Next issue the following command:

db2 -td/ -vf script.db2

For procedures, use the CALL statement. For functions, execute queries or other SQL statements that contain references to those functions. For example (Fig. 6):

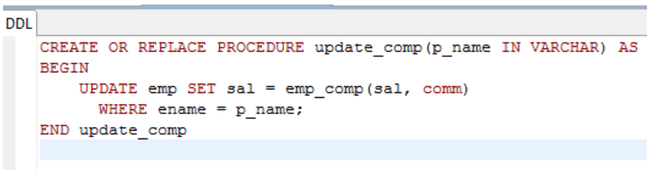
CALL update_comp('test')

Most of the PL/SQL examples are based on a PL/SQL sample schema that represents employees in an organization as in Oracle (Fig 4. and 5.). Code PL/SQL is based on a sample schema given in the documentation [4], [5].



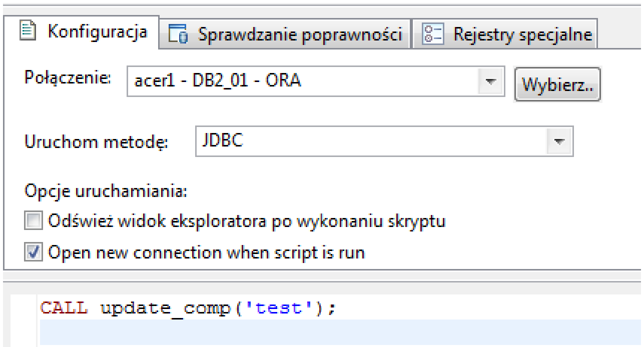
```
CREATE OR REPLACE FUNCTION emp_comp (
    p_sal          NUMBER,
    p_comm         NUMBER
) RETURN NUMBER
IS
BEGIN
    RETURN (p_sal + NVL(p_comm, 0)) * 24;
END;
```

Fig. 4. The sample code object FUNCTION in PL/SQL



```
CREATE OR REPLACE PROCEDURE update_comp(p_name IN VARCHAR) AS
BEGIN
    UPDATE emp SET sal = emp_comp(sal, comm)
      WHERE ename = p_name;
END update_comp
```

Fig. 5. The sample code object PROCEDURE in PL/SQL



Połączenie: acer1 - DB2_01 - ORA

Uruchom metodę: JDBC

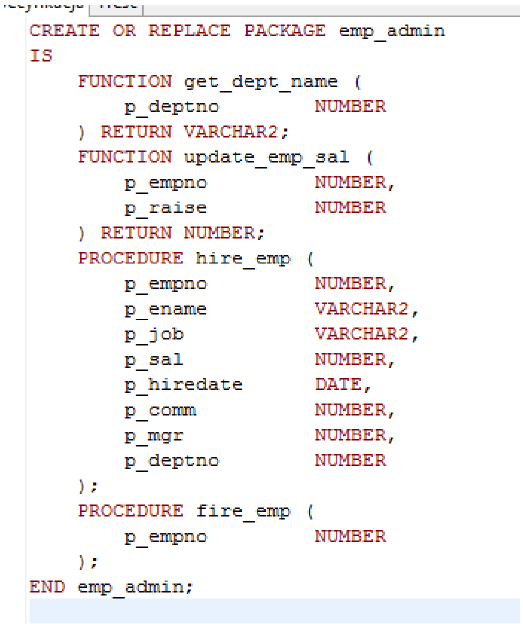
Opcje uruchamiania:

☐ Odśwież widok eksploratora po wykonaniu skryptu

☑ Open new connection when script is run

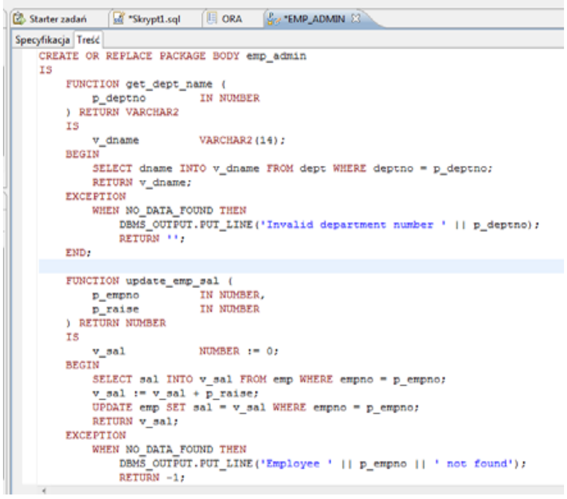
```
CALL update_comp('test');
```

Fig. 6. CALL statement



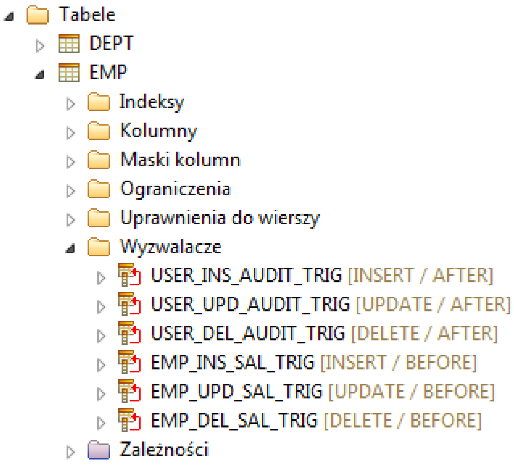
```
CREATE OR REPLACE PACKAGE emp_admin
IS
    FUNCTION get_dept_name (
        p_deptno        NUMBER
    ) RETURN VARCHAR2;
    FUNCTION update_emp_sal (
        p_empno         NUMBER,
        p_raise         NUMBER
    ) RETURN NUMBER;
    PROCEDURE hire_emp (
        p_empno         NUMBER,
        p_ename         VARCHAR2,
        p_job           VARCHAR2,
        p_sal           NUMBER,
        p_hiredate      DATE,
        p_comm          NUMBER,
        p_mgr           NUMBER,
        p_deptno        NUMBER
    );
    PROCEDURE fire_emp (
        p_empno         NUMBER
    );
END emp_admin;
```

Fig. 7. The sample code object PACKAGE in PL/SQL



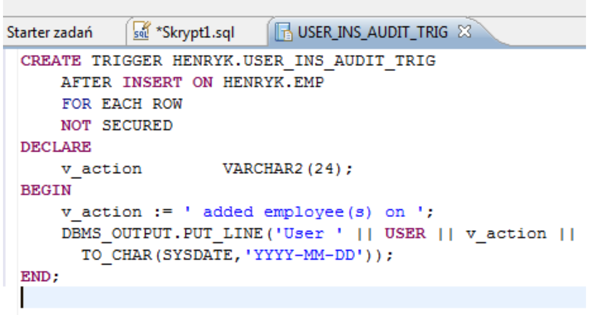
```
CREATE OR REPLACE PACKAGE BODY emp_admin
IS
    FUNCTION get_dept_name (
        p_deptno         IN NUMBER
    ) RETURN VARCHAR2
    IS
        v_dname          VARCHAR2(14);
    BEGIN
        SELECT dname INTO v_dname FROM dept WHERE deptno = p_deptno;
        RETURN v_dname;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Invalid department number ' || p_deptno);
            RETURN '';
    END;

    FUNCTION update_emp_sal (
        p_empno          IN NUMBER,
        p_raise          IN NUMBER
    ) RETURN NUMBER
    IS
        v_sal            NUMBER := 0;
    BEGIN
        SELECT sal INTO v_sal FROM emp WHERE empno = p_empno;
        v_sal := v_sal + p_raise;
        UPDATE emp SET sal = v_sal WHERE empno = p_empno;
        RETURN v_sal;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Employee ' || p_empno || ' not found');
            RETURN -1;
```

Fig. 8. The sample code object PACKAGE BODY in PL/SQL (partial)



- Tabele
  - DEPT
  - EMP
    - Indeksy
    - Kolumny
    - Maski kolumn
    - Ograniczenia
    - Uprawnienia do wierszy
    - Wyzwalacze
      - USER_INS_AUDIT_TRIG [INSERT / AFTER]
      - USER_UPD_AUDIT_TRIG [UPDATE / AFTER]
      - USER_DEL_AUDIT_TRIG [DELETE / AFTER]
      - EMP_INS_SAL_TRIG [INSERT / BEFORE]
      - EMP_UPD_SAL_TRIG [UPDATE / BEFORE]
      - EMP_DEL_SAL_TRIG [DELETE / BEFORE]
    - Zależności

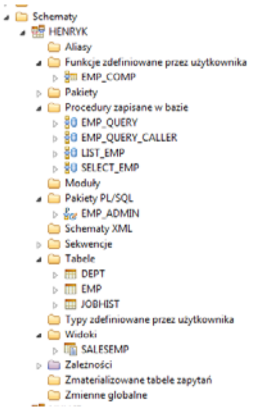
Fig. 9. The sample PL/SQL Trigger in DataStudio



```
CREATE TRIGGER HENRYK.USER_INS_AUDIT_TRIG
    AFTER INSERT ON HENRYK.EMP
    FOR EACH ROW
    NOT SECURED
DECLARE
    v_action          VARCHAR2(24);
BEGIN
    v_action := ' added employee(s) on ';
    DBMS_OUTPUT.PUT_LINE('User ' || USER || v_action ||
      TO_CHAR(SYSDATE,'YYYY-MM-DD'));
END;
```

Fig. 10. The sample code object TRIGGER in PL/SQL

PL/SQL procedures and functions are stored together with other objects DB2 . They are available in the same way as a standard DB2 objects. Figure 11 shows the procedures and functions written in PL / SQL. PL/SQL packages are available in an additional collection 'PL/SQL packages'.



- Schematy
  - HENRYK
    - Aliasy
    - Funkcje zdefiniowane przez użytkownika
      - EMP_COMP
    - Pakiety
    - Procedury zapisane w bazie
      - EMP_QUERY
      - EMP_QUERY_CALLER
      - LIST_EMP
      - SELECT_EMP
    - Moduły
    - Pakiety PL/SQL
      - EMP_ADMIN
    - Schematy XML
    - Sekwencje
    - Tabele
      - DEPT
      - EMP
      - JOBHIST
    - Typy zdefiniowane przez użytkownika
    - Widoki
      - SALESEMP
    - Zależności
    - Zmaterializowane tabele zapytań
    - Zmienne globalne

Fig. 11. PL/SQL Object in DataStudio

Conclusion

Article presents the opportunity to build and execute the Oracle application on DB2 database servers. It show how to configure DB2 to work with the PL / SQL language. For this purpose, a number of Oracle compatibility features are implemented in the BD2 database server and the ability to selectively choose a registry variable. In the DB2 system occurs SQL PL and PL / SQL with a complete range of packages and functions used in Oracle (including SQL standard, which also had to be adapted). DB2 10.5 allows to easily run Oracle applications after earlier migration, with can be perform by using the IBM Data Movement Tool. Ultimately, it is possible to deploy applications on the new DB2 database server instead of the Oracle server. In the system can occur parallel objects written in PL / SQL and SQL PL, which can be executed simultaneously. Article presents the most important PL / SQL objects in DB2 occurring to provide high compatibility with Oracle.

REFERENCES

[1] Drzymała P., Welfle H.: Przetwarzanie zapytań rozproszonych w środowisku InfoSphere Federation Server stowarzyszonym z bazami danych Oracle. Studia Informatica 2011, rocznik 32, nr 2B, s. 359-372

[2] Smarter Planet IBM: http://www.ibm.com/smarterplanet/us/en/overview/ideas/

[3] Raul F. Chong, Clara Liu, Sylvia F. Qi, Dwaine R. Snow: Understanding DB2: Learning Visually with Examples

[4] IBM DB2 10.1 dla systemów Linux, UNIX i Windows - Centrum informacyjne: http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/index.jsp

[5] IBM DB2 10.5 dla systemów Linux, UNIX i Windows - Centrum informacyjne: http://publib.boulder.ibm.com/infocenter/db2luw/v10r5/index.jsp

[6] Whei-Jen Chen, K. Chen, P. Dantressangle, M. Greenstein, S. Lightstone, M. Petrenko, R. Reutlinger, S. Rielau, S. Schormann, M. N Schwenger: Oracle to DB2 Conversion Guide: Compatibility Made Easy, RedBooks -November 2009

[7] Drzymała P., Welfle H.: Mechanizmy wsparcia uruchamiania aplikacji Oracle w środowisku wersji IBM DB2 9.7.5 Express-C przeznaczonej dla społeczności. Studia Informatica 2012, rocznik 33, nr 2B(106), s. 187-199

***Authors**: Ph.D. Pawel Drzymala, Lodz University of Technology, Institute of Mechatronics and Information Systems, Stefanowskiego 18/22, 90-924 Lodz, E-mail: pawel.drzymala@p.lodz.pl; Ph.D. Henryk Welfle, Lodz University of Technology, Institute of Mechatronics and Information Systems, Stefanowskiego 18/22, 90-924 Lodz, E-mail: henryk.welfle@p.lodz.pl.*