RIGA TECHNICAL UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

INSTITUTE OF APPLIED COMPUTER SYSTEMS

Homework #3

“Large Databases”

**Use of object methods for data extraction from ORDB data storage structures. Java language use in database server.**

Author: Oleg Tsoy

Course, group: ADBD0

Student card no.: 131ADB042

2014 / 2015 study year

**Content**

1 Goal 3

2 Task 3

3 Database description 4

4 SQL queries 5

5 Conclusions 25

6 References 26

# Goal

Learn more about usage of object methods for data extraction from ORDB data storage structures. How we can use Java programming language in the database server.

# Task

1. Creation and test of **MAP** type method.

2. Creation and test of **ORDER** type method.

3. Creation and test of **MEMBER** type method.

4. Creation and test of user-defined constructor method.

5. Java class creation and loading in database:

1) using LOADJAVA program;

2) using CREATE JAVA command.

Java class meta-date inspection (SELECT).

Creation of PL/SQL function which call Java method. Execution of Java method.

6. Java class creation, loading and testing for SELECT query realization.

7. Conclusions (what seems good, what bad, what like, what is problematic).

# SQL queries

***1. Query goal (MAP type method):***

***Query SQL code***

*CREATE TYPE STUDENT\_TYPE2*

*AS*

*OBJECT*

*(*

*STUDENT\_ID NUMBER,*

*MATHEMATICS\_MARK NUMBER,*

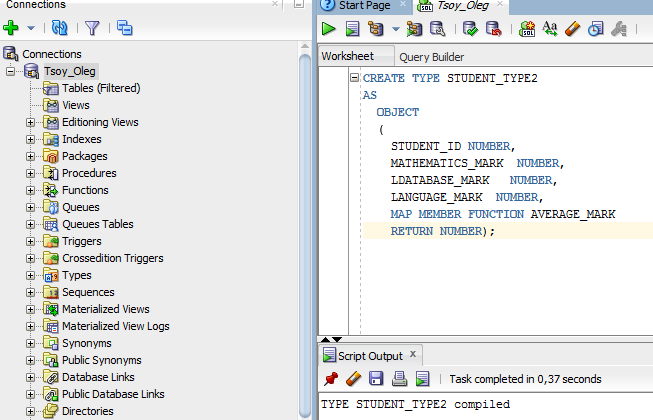
*LDATABASE\_MARK NUMBER,*

*LANGUAGE\_MARK NUMBER,*

*MAP MEMBER FUNCTION AVERAGE\_MARK*

*RETURN NUMBER);*

***Result of execution***



***Analysis of results, what in these data can be seen:*** *As can be seen from the screenshot I have created the STUDENT\_TYPE2 as an object for my project.*

***2. Query goal (MAP type method):***

***Query SQL code***

*CREATE OR REPLACE TYPE BODY STUDENT\_TYPE2*

*AS*

*MAP MEMBER FUNCTION AVERAGE\_MARK*

*RETURN NUMBER*

*IS*

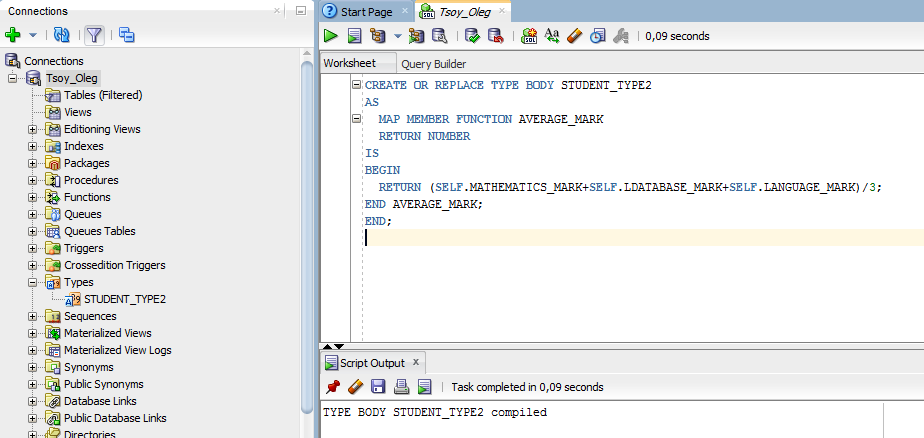
*BEGIN*

*RETURN (SELF.MATHEMATICS\_MARK+SELF.LDATABASE\_MARK+SELF.LANGUAGE\_MARK)/3;*

*END AVERAGE\_MARK;*

*END;*

***Result of execution***



***Analysis of results, what in these data can be seen:*** *In this case I have compiled STUDENT\_TYPE2 with MAP MEMBER FUNCTION.*

***3. Query goal (MAP type method):***

***Query SQL code***

*CREATE TABLE STUDENTS\_WITH\_MARKS OF STUDENT\_TYPE2;*

*INSERT INTO STUDENTS\_WITH\_MARKS VALUES*

*( STUDENT\_TYPE2(10,8,9,9)*

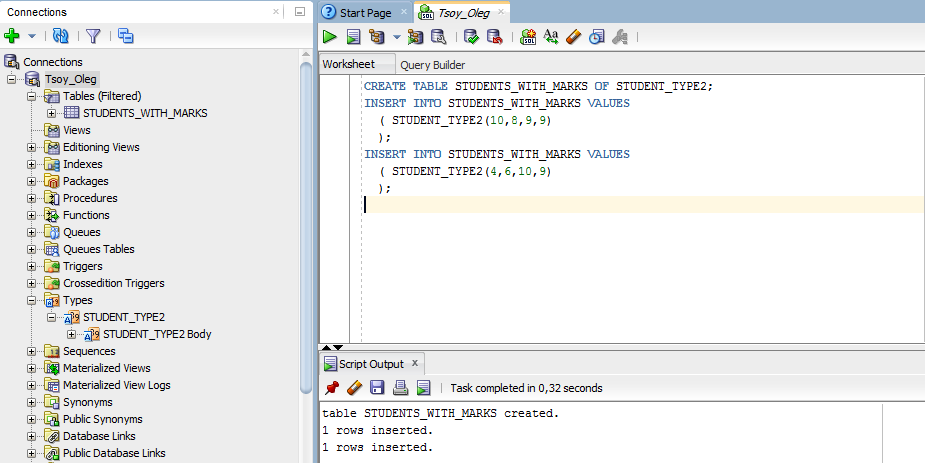
*);*

*INSERT INTO STUDENTS\_WITH\_MARKS VALUES*

*( STUDENT\_TYPE2(4,6,10,9)*

*);*

***Result of execution***



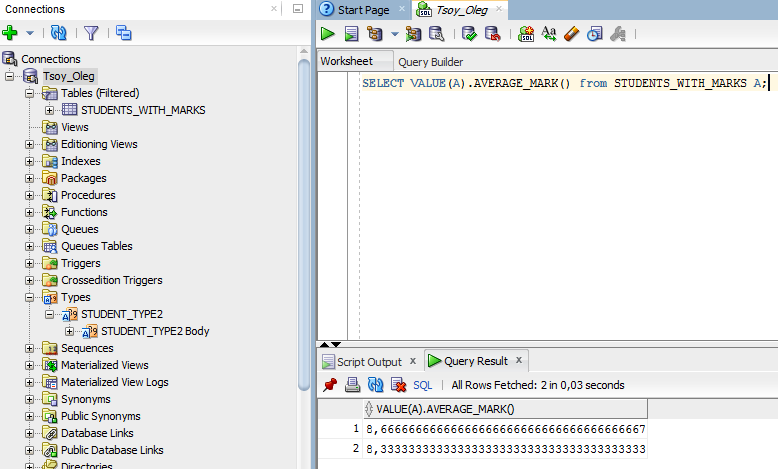
***Analysis of results, what in these data can be seen:*** *I have created STUDENT\_WITH\_MARKS table and 2 rows of data were inserted into this table.*

***4. Query goal (MAP type method):***

***Query SQL code***

*SELECT VALUE(A).AVERAGE\_MARK() from STUDENTS\_WITH\_MARKS A;*

***Result of execution***



***Analysis of results, what in these data can be seen:*** *By this select statement function I presented the calculation function which was given in previous type body state. As can be seen from screenshot it gave us the AVERAGE\_MARK of inputed marks of 2 students and automatically calculate them.*

***5. Query goal (ORDER type method):***

***Query SQL code***

CREATE OR REPLACE TYPE TYPE\_ORDER

AS

OBJECT

(

LEVEL\_NUMBER1 NUMBER,

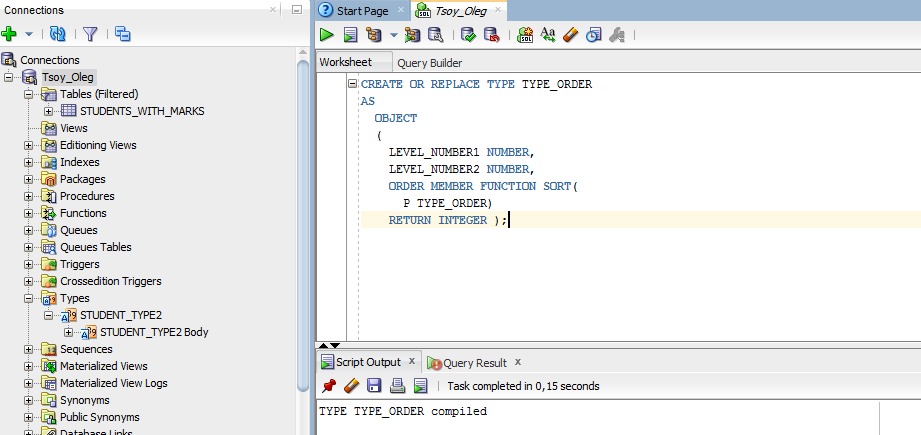
LEVEL\_NUMBER2 NUMBER,

ORDER MEMBER FUNCTION SORT(

P TYPE\_ORDER)

RETURN INTEGER );

***Result of execution***



***Analysis of results, what in these data can be seen:*** *According to the query goal I have created TYPE\_ORDER as an object to realize the ORDER function in my project.*

***6. Query goal (ORDER type method):***

***Query SQL code***

CREATE OR REPLACE TYPE BODY TYPE\_ORDER

AS

ORDER MEMBER FUNCTION SORT(

P TYPE\_ORDER)

RETURN INTEGER

IS

BEGIN

IF LEVEL\_NUMBER1 < P.LEVEL\_NUMBER1 THEN

RETURN -1;

ELSIF LEVEL\_NUMBER1 > P.LEVEL\_NUMBER1 THEN

RETURN 1;

ELSE

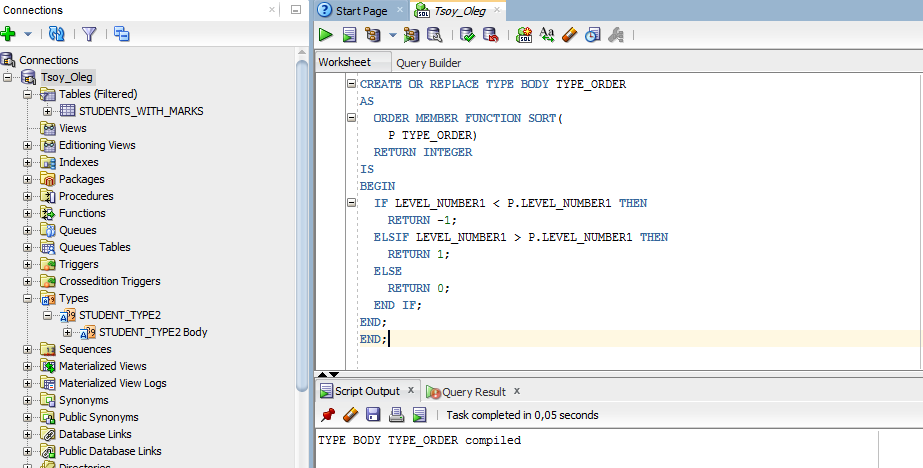
RETURN 0;

END IF;

END;

END;

***Result of execution***



***Analysis of results, what in these data can be seen:*** *In this case I have created the TYPE\_ORDER body as order member function sort where some requirements were included.*

***7. Query goal (ORDER type method):***

***Query SQL code***

CREATE TABLE TESTER1 OF TYPE\_ORDER;

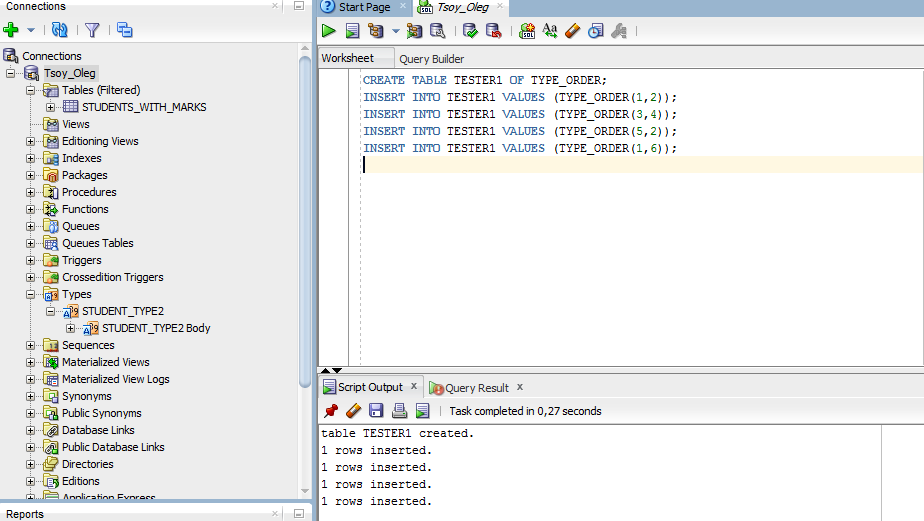
INSERT INTO TESTER1 VALUES (TYPE\_ORDER(1,2));

INSERT INTO TESTER1 VALUES (TYPE\_ORDER(3,4));

INSERT INTO TESTER1 VALUES (TYPE\_ORDER(5,2));

INSERT INTO TESTER1 VALUES (TYPE\_ORDER(1,6));

***Result of execution***



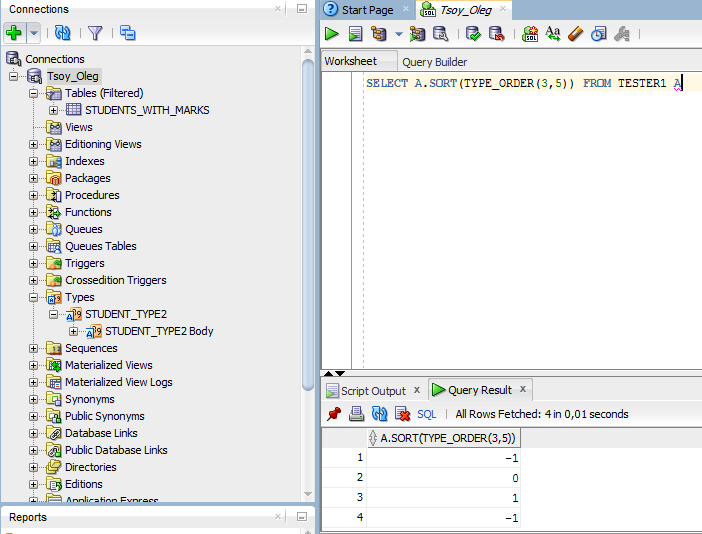
***Analysis of results, what in these data can be seen:*** *In this query I have created table TESTER1 of TYPE\_ORDER, where I inserted 4 rows with needed values to show the main performance of this project in ORDER MEMBER function.*

***8. Query goal (ORDER type method):***

***Query SQL code***

SELECT A.SORT(TYPE\_ORDER(3,5)) FROM TESTER1 A

***Result of execution***



***Analysis of results, what in these data can be seen:*** *The SELECT statement in this case shows the sort of TYPE\_ORDER values which were inserted below FROM table TESTER1.*

***9. Query goal (MEMBER type method):***

***Query SQL code***

CREATE TYPE STUDENT\_TYPE3

AS

OBJECT

(

STUDENT\_ID NUMBER,

NAME VARCHAR2(50),

MATHEMATICS\_MARK NUMBER,

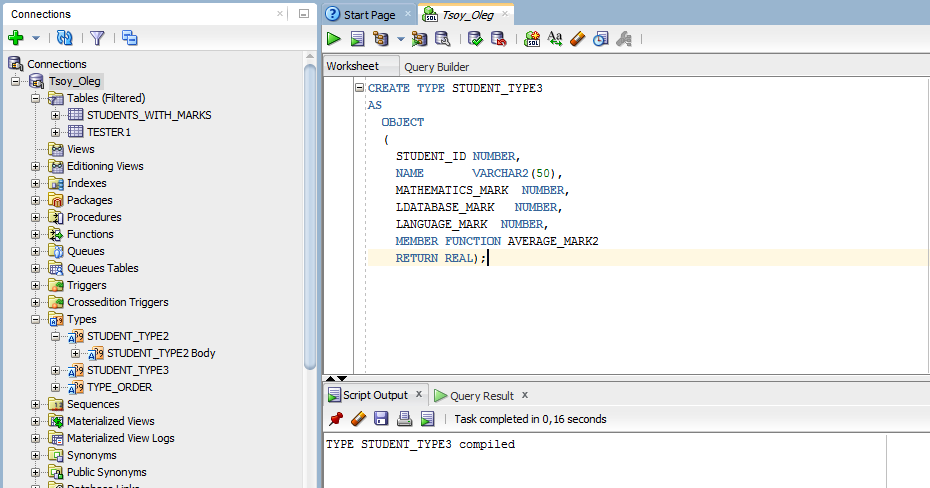
LDATABASE\_MARK NUMBER,

LANGUAGE\_MARK NUMBER,

MEMBER FUNCTION AVERAGE\_MARK2

RETURN REAL);

***Result of execution***



***Analysis of results, what in these data can be seen:*** *I have created the 3rd type student as an object with ID, Name, Marks for subjects and use member function AVERAGE\_MARK2 which returned real numbers.*

***10. Query goal (MEMBER type method):***

***Query SQL code***

CREATE OR REPLACE TYPE BODY STUDENT\_TYPE3

AS

MEMBER FUNCTION AVERAGE\_MARK2

RETURN REAL

IS

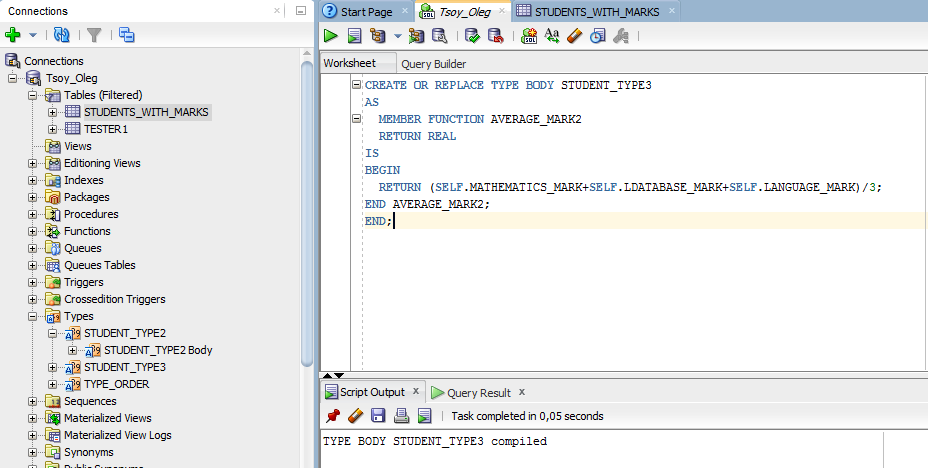
BEGIN

RETURN (SELF.MATHEMATICS\_MARK+SELF.LDATABASE\_MARK+SELF.LANGUAGE\_MARK)/3;

END AVERAGE\_MARK2;

END;

***Result of execution***



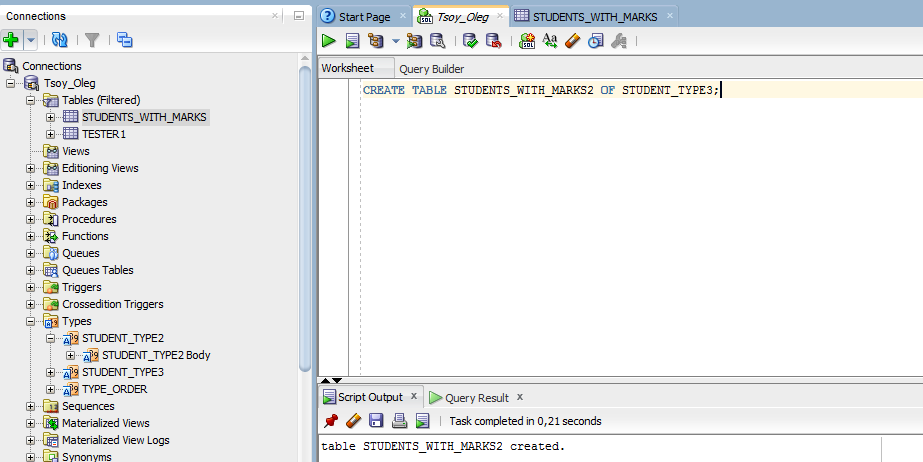
***Analysis of results, what in these data can be seen:*** *In this case the Body of student\_type 3 was created with registered member function AVERAGE\_MARK2, also the algorithm of calculating the AVERAGE\_MARK2 was given in the code.*

***11. Query goal (MEMBER type method):***

***Query SQL code***

CREATE TABLE STUDENTS\_WITH\_MARKS2 OF STUDENT\_TYPE3;

***Result of execution***



***Analysis of results, what in these data can be seen:*** *The table STUDENTS\_WITH\_MARKS2 was created for STUDENT\_TYPE3 observations.*

***12. Query goal (MEMBER type method):***

***Query SQL code***

INSERT

INTO STUDENTS\_WITH\_MARKS2 VALUES

(

STUDENT\_TYPE3(1,'OLEG TSOY',8,9,10)

);

INSERT

INTO STUDENTS\_WITH\_MARKS2 VALUES

(

STUDENT\_TYPE3(2,'VLADISLAV TSOY',9,9,8)

);

INSERT

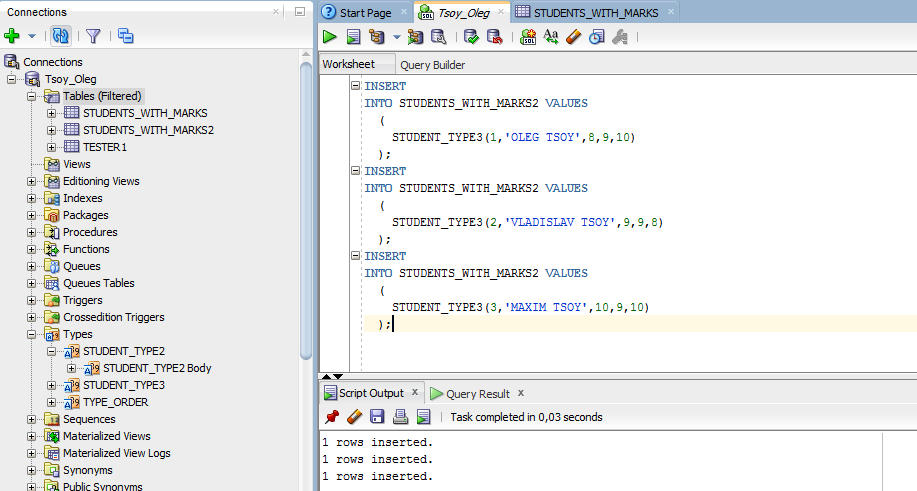
INTO STUDENTS\_WITH\_MARKS2 VALUES

(

STUDENT\_TYPE3(3,'MAXIM TSOY',10,9,10)

);

***Result of execution***



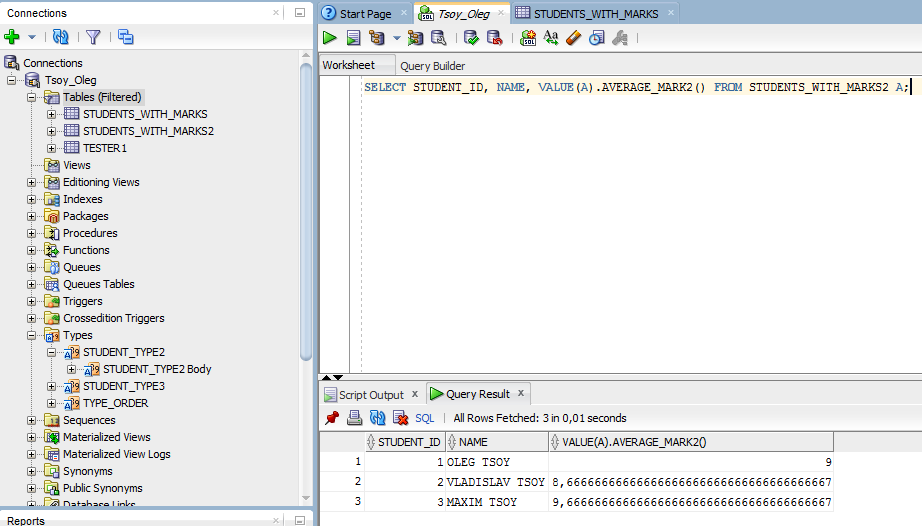
***Analysis of results, what in these data can be seen:*** *In this screenshot the inserted rows values were indicated for future manipulations.*

***13. Query goal (MEMBER type method):***

***Query SQL code***

*SELECT STUDENT\_ID, NAME, VALUE(A).AVERAGE\_MARK2() FROM STUDENTS\_WITH\_MARKS2 A;*

***Result of execution***



***Analysis of results, what in these data can be seen:*** *The demonstration of inserted data and the performance of the member function working calculations with inserted values.*

***14. Query goal (Using of LOADJAVA program):***

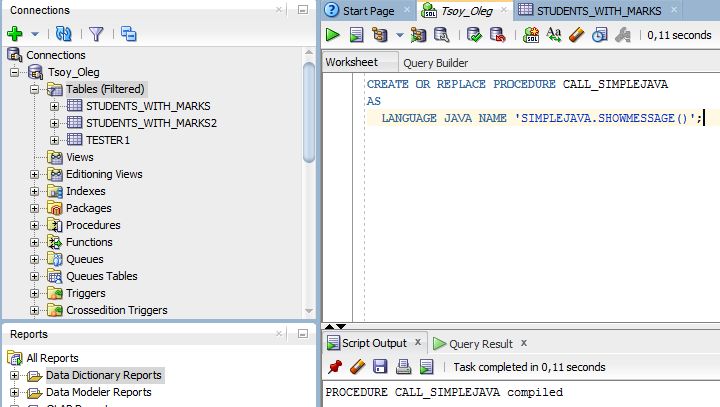
***Query SQL code***

CREATE OR REPLACE PROCEDURE CALL\_SIMPLEJAVA

AS

LANGUAGE JAVA NAME 'SIMPLEJAVA.SHOWMESSAGE()';

***Result of execution***



***Analysis of results, what in these data can be seen:*** *I have created call\_simplejava as language java in my Database system.*

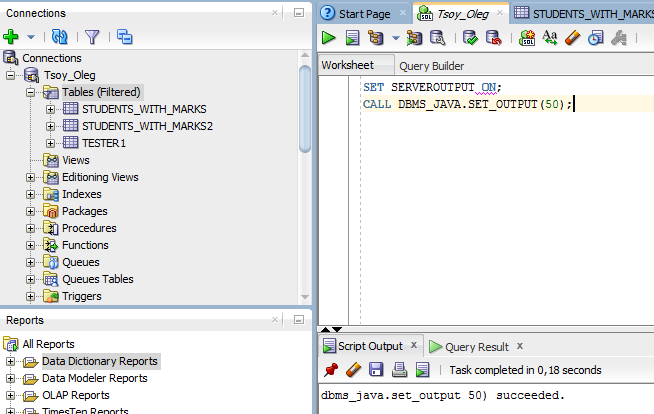
***15. Query goal (Using of LOADJAVA program):***

***Query SQL code***

SET SERVEROUTPUT ON;

CALL DBMS\_JAVA.SET\_OUTPUT(50);

***Result of execution***



***Analysis of results, what in these data can be seen:*** *The server output by java programming language was succeeded.*

***16. Query goal (JAVA SELECT):***

***Query SQL code***

CREATE OR REPLACE JAVA SOURCE NAMED "MY\_WORK" AS

IMPORT JAVA.SQL\*;

IMPORT ORACLE.JDBC\*;

IMPORT ORACLE.JDBC.POOL.ORACLEDATASOURCE;

CLASS MY\_WORK {

PUBLIC STATIC VOID MAIN (STRING ARGS[])

THROWS SQLEXCEPTION {

STRING TEXT = NULL;

ORACLEDATASOURCE ODS = NEW ORACLEDATASOURCE();

ODS.SETURL(“JDBC:DEFAULT:CONNECTION:”);

CONNECTION CONN = ODS.GETCONNECTION();

STATEMENT COMMAND = NULL;

COMMAND = CONN.CREATESTATEMENT();

RESULTSET RESULTS = COMMAND.EXECUTEQUERY (“SELECT NAME FROM STUDENTS\_WITH\_MARKS2 WHERE STUDENT\_ID = 1”);

WHILE (RESULTS NEXT()) { TEXT = RESULTS.GETSTRING(“NAME”);}

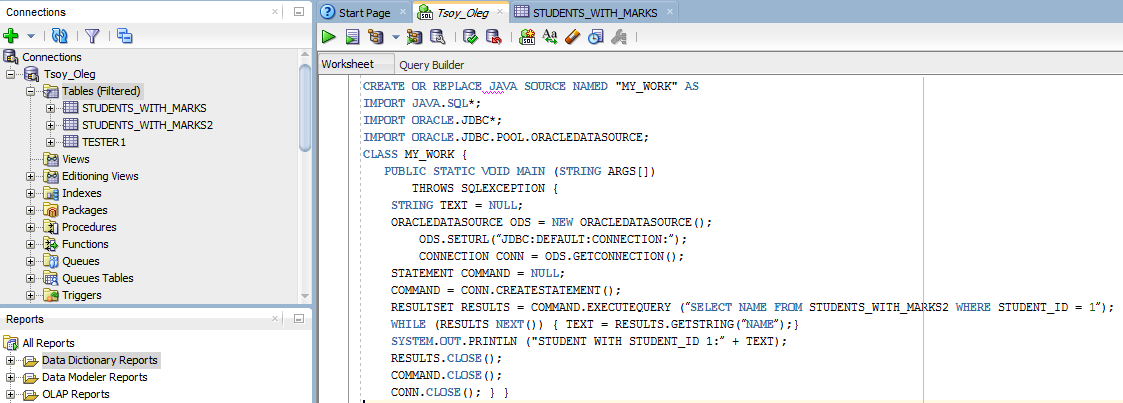
SYSTEM.OUT.PRINTLN ("STUDENT WITH STUDENT\_ID 1:” + TEXT);

RESULTS.CLOSE();

COMMAND.CLOSE();

CONN.CLOSE(); } }

***Result of execution***





***Analysis of results, what in these data can be seen:*** *Execution of JAVA language meta-data inspection.*

***17. Query goal (CREATE JAVA procedure):***

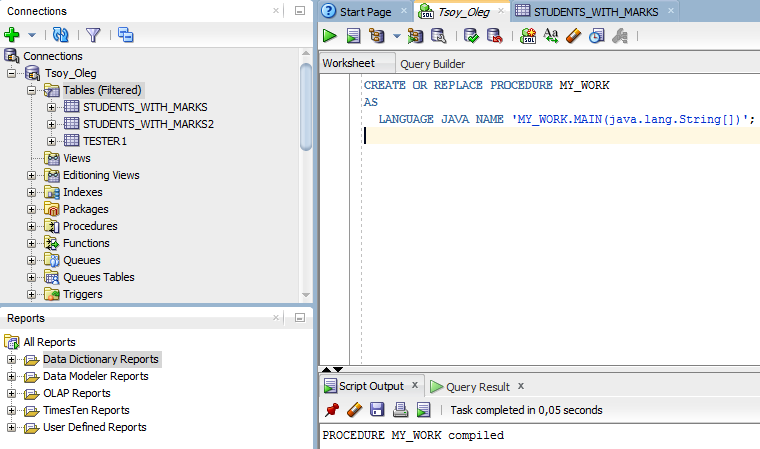
***Query SQL code***

CREATE OR REPLACE PROCEDURE MY\_WORK

AS

LANGUAGE JAVA NAME 'MY\_WORK.MAIN(java.lang.String[])';

***Result of execution***



***Analysis of results, what in these data can be seen:*** *Procedure MY\_WORK was compiled successfully. That shows that creation of the procedure MY\_WORK which is MAIN format has language JAVA as a priority created language.*

***18. Query goal (serveroutput of JAVA procedure):***

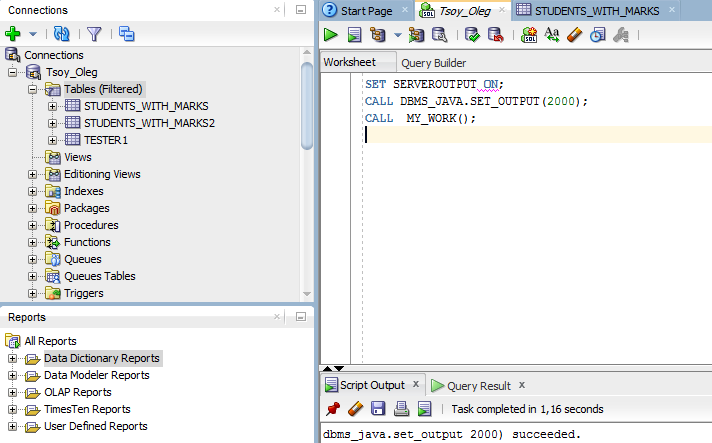
***Query SQL code***

SET SERVEROUTPUT ON;

CALL DBMS\_JAVA.SET\_OUTPUT(2000);

CALL MY\_WORK();

***Result of execution***



***Analysis of results, what in these data can be seen:*** *In this case we can observe that this query call the execution of MY\_WORK procedure.*

# Conclusion

*As can be concluded from this project description I can say that it was the most difficult database project for me because of the existence of JAVA programming language for comparing all data. While, I found some advantages in using of the programming language or IDE which can provide the highest performance in my procedures and simplify the initial code structure. In my case I used Eclipse IDE for compiling JAVA with Oracle Database Management System as can be seen from the most cases in my preview. The problematic part of this host was to find the path how to realize the code description due to the SQL developer and SQL\* Plus developer have the different structure. However, I was late after reading the materials about them that is why it was my main problem in this case.*