RIGA TECHNICAL UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

INSTITUTE OF APPLIED COMPUTER SYSTEMS

Homework #1

“Large Databases”

**Creation of object-relational database(ORDB) data storage structures and data extraction.**

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

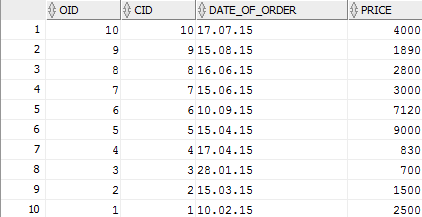
# Goal

Learn more about creation of object-relational database (ORDB) data storage structures and data extraction.

# Task

1. Creation of object table with row type objects (CREATE TYPE, CREATE TABLE), data insert (INSERT), output of metadata (SELECT), output of objects and its components (SELECT), using function VALUE().
2. Creation of object table with column objects (CREATE TYPE, CREATE TABLE), data insert (INSERT), output of metadata (SELECT), output of objects and its components (SELECT), using dot notation.
3. Creation of object table with object collection (nested table) (CREATE TYPE, CREATE TABLE), data insert (INSERT), output of metadata (SELECT), output of objects and its components (SELECT), using function TABLE().
4. Creation of an object view from two tables (CRETE TYPE, CREATE VIEW), data (objects and components) extraction from object view (SELECT).
5. Creation of table with heterogenic objects, using type hierarchy. Data extraction (use of TRAT(), IS OF TYPE(), SYS\_TYPEID()).
6. Data extraction using SUBMULTISET [OF], [NOT] MEMBER [OF], IS [NOT] A SET, CARDINALITY(), [ALL] OR [DISTINCT] MULTISET EXCEPT(), [ALL] OR [DISTINCT] MULTISET INTERSECT, [ALL] OR [DISTINCT] MULTISET UNION(), POWERMULTISET(), POWERMULTISET\_BY\_CARDINALITY(), SET() functions and operators.
7. Two tables connection using object references (REF). Input of object identifiers using function REF(). Data extraction (all objects and object components) using function DEREF().
8. Conclusions (what seems good, what bad, what like, what is problematic).

# Database description



# SQL queries

***1. Query goal (CREATE a table CUSTOMER):*** *Create a table with name CUSTOMER with Customer ID(CID), Customer Name(CNAME), Country, Age attributes and constraint for checking Customer ID(CID)>0.*

***Query SQL code***

*CREATE TABLE CUSTOMER*

*(*

*CID NUMBER(2) NOT NULL,*

*CNAME VARCHAR(25),*

*COUNTRY VARCHAR(25),*

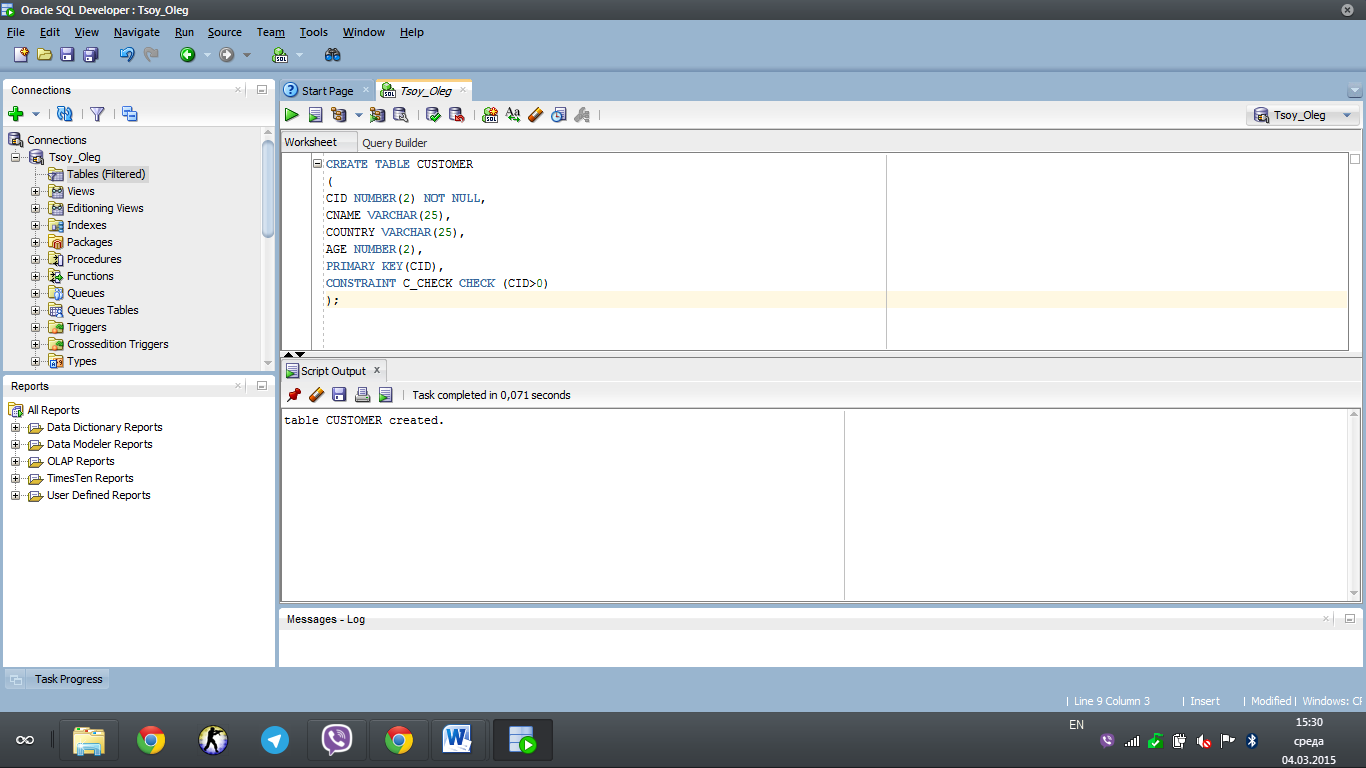
*AGE NUMBER(2),*

*PRIMARY KEY(CID),*

*CONSTRAINT C\_CHECK CHECK (CID>0)*

*);*

***Result of execution***

****

***Analysis of results, what in these data can be seen:*** *According to the requirements of creating the query I have made the table called CUSTOMER with CID with format NUMBER(2) and NOT NULL constraint for it. Customer Name, Country Name are used the format VARCHAR(25). AGE attribute uses 2 possible variables. In this case PRIMARY key is CID. There is only one constraint to check the fill the value CID>0.*

***2. Query goal (CREATE a table CUSTOMER\_ORDER):*** *Create a table with name CUSTOMER\_ORDER with Order ID(OID), Customer ID(CID), Date of order, Price attributes.*

***Query SQL code:***

*CREATE TABLE CUSTOMER\_ORDER*

*(*

*OID NUMBER(2) NOT NULL,*

*CID NUMBER(2) NOT NULL,*

*DATE\_OF\_ORDER DATE,*

*PRICE NUMBER(4),*

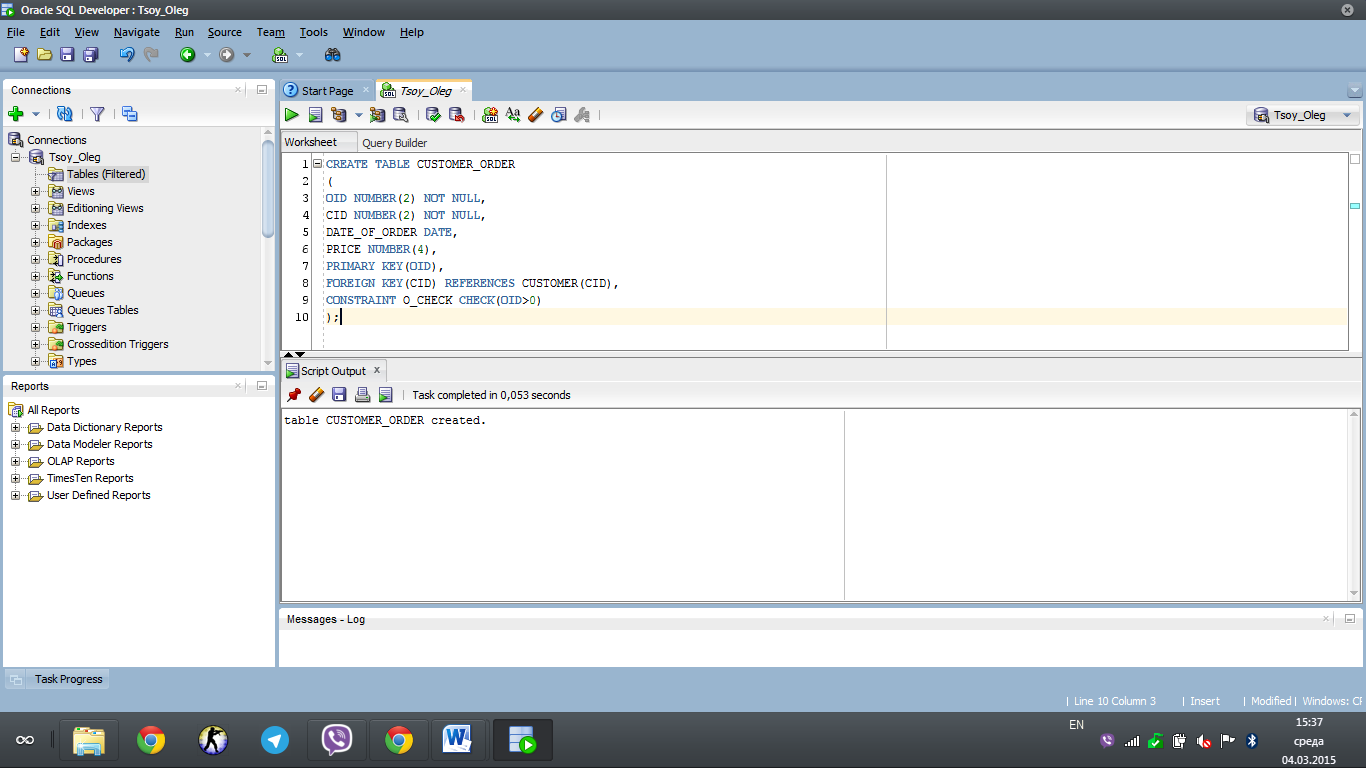
*PRIMARY KEY(OID),*

*FOREIGN KEY(CID) REFERENCES CUSTOMER(CID),*

*CONSTRAINT O\_CHECK CHECK(OID>0)*

*);*

***Result of execution***

**

***Analysis of results, what in these data can be seen:*** *There are several requirements which were used for constructing this table. In this table Order ID us NUMBER(2) and NOT NULL, CID also is NUMBER(2) and NOT NULL. In this case we had different format DATE for DATE of ORDER. Price is NUMBER. PRIMARY key is Order ID and Foreign key is Customer ID. Constraint in this case is the Order Check function that is why Order ID always should be more than zero.*

***3. Query goal (CREATE a table CUSTOMER\_ADDRESS):*** *Create a table CUSTOMER\_ADDRESS with Customer ID(CID), Customer Address ID(CADID), Customer Name(CNAME), Country, City, Zip attributes.*

***Query SQL code:***

*CREATE TABLE CUSTOMER\_ADDRESS*

*(*

*CID NUMBER(2) NOT NULL,*

*CADID NUMBER(2) NOT NULL,*

*CNAME VARCHAR(25),*

*COUNTRY VARCHAR(25),*

*CITY VARCHAR(25),*

*ZIP VARCHAR(7),*

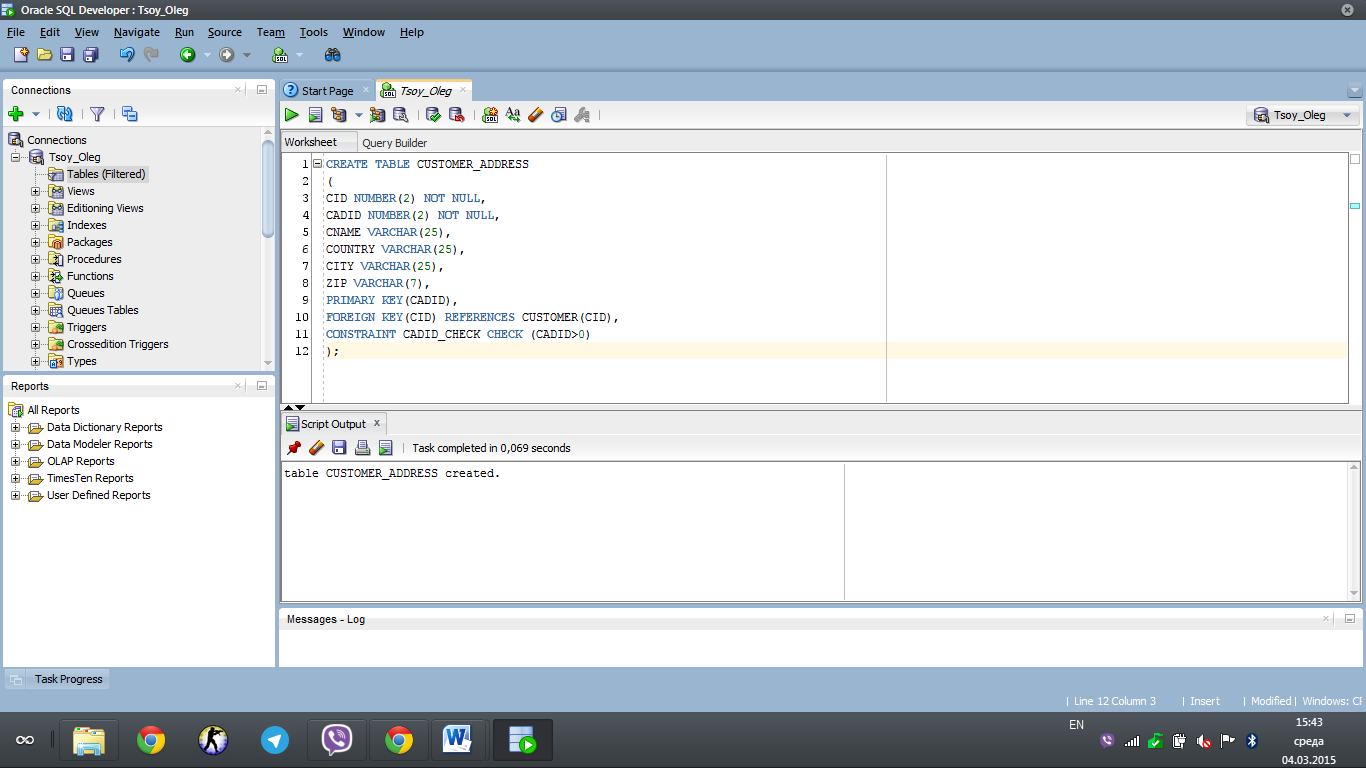
*PRIMARY KEY(CADID),*

*FOREIGN KEY(CID) REFERENCES CUSTOMER(CID),*

*CONSTRAINT CADID\_CHECK CHECK (CADID>0)*

*);*

***Result of execution:***

**

***Analysis of results, what in these data can be seen:*** *There are several reqiurements which are given in this table. In this case OID has the format NUMBER(2) and it is NOT NULL as the Customer Address ID. Customer Name, Country name and City are Varchar(25). Also ZIP attribute is NUMBER. In this case PRIMARY key is Customer Address ID while Customer ID is the Foreign key.*

***4. Query goal (To create a Type Person as Object):***

***Query SQL code:***

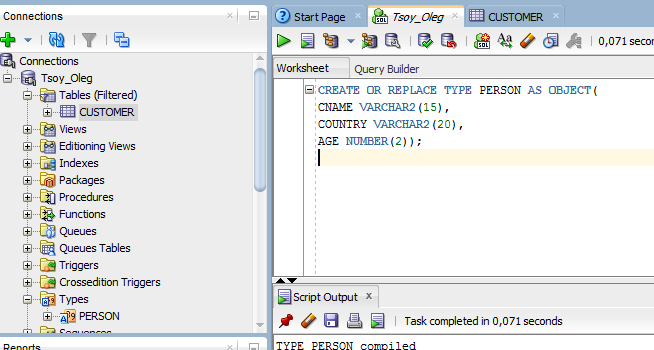
*CREATE OR REPLACE TYPE PERSON AS OBJECT(*

*CNAME VARCHAR2(15),*

*COUNTRY VARCHAR2(20),*

*AGE NUMBER(2));*

***Result of execution:***

****

***Analysis of results:*** *According to the requirements the Object PERSON was constructed. The main features of this object are Customer Name (CNAME), Country of the customer and Age of customer.*

***5. Query goal (To create a Type Person as Object):***

***Query SQL code:***

*create or replace type c\_order as object(*

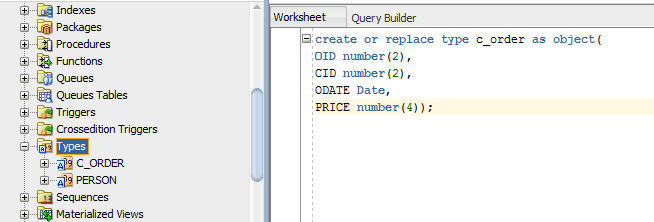
*OID number(2),*

*CID number(2),*

*ODATE Date,*

*PRICE number(4));*

*Result of execution:*



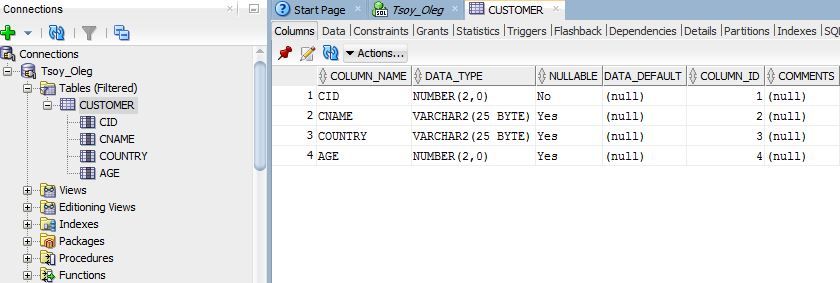
**Analysis of results:** According to requirements one additional type c\_order (customer order) was created for demonstrating the insertation of an objects to the row.

***6. Query goal (To create table with type object):***

***Query SQL code:***

*CREATE table CUSTOMER of PERSON;*

***Result of execution:***



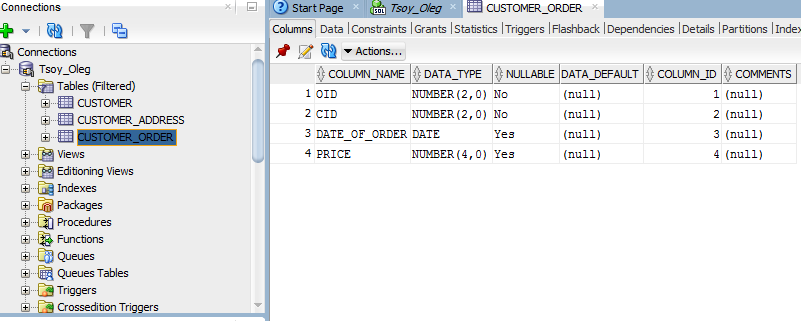
***Analysis of results:*** *According to the requirements we can see that the main features of the table customer are displayed on the screen, also the nullable and data type things are presented.*

***7. Query goal (To create table with type object):***

***Query SQL code:***

*CREATE table CUSTOMER\_ORDER of c\_order;*

***Result of execution:***



**Analysis of results:** In this case the data type, column name and nullable variables were demonstrated in customer\_order table.

***8. Query goal (INSERT INTO a table CUSTOMER VALUES):*** *Insert the values into CUSTOMER table with needed attributes.*

***Query SQL code:***

*begin*

*INSERT INTO CUSTOMER VALUES(person(01,'ERIC PEARCE','USA',27));*

*INSERT INTO CUSTOMER VALUES(person(02,'VANESSA GEAR','CZECH REPUBLIC',21));*

*INSERT INTO CUSTOMER VALUES(person(03,'OLEG TSOY','UZBEKISTAN',20));*

*INSERT INTO CUSTOMER VALUES(person(04,'FARZUNA KHAMITOVA','UZBEKISTAN',20));*

*INSERT INTO CUSTOMER VALUES(person(05,'DURBEK FAYZULLAYEV','UZBEKISTAN',20));*

*INSERT INTO CUSTOMER VALUES(person(06,'ZARINA BEGULOVA','UZBEKISTAN',22));*

*INSERT INTO CUSTOMER VALUES(person(07,'SHOKHRUZ SATTAROV','UZBEKISTAN',20));*

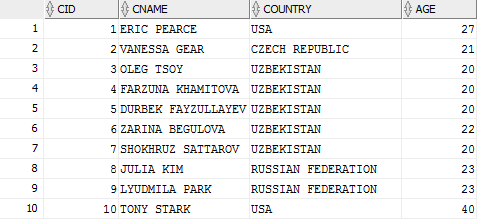
*INSERT INTO CUSTOMER VALUES(person(08,'JULIA KIM','RUSSIAN FEDERATION',23));*

*INSERT INTO CUSTOMER VALUES(person(09,'LYUDMILA PARK','RUSSIAN FEDERATION',23));*

*INSERT INTO CUSTOMER VALUES(person(10,'TONY STARK','USA',40));*

*end;*

***Result of execution:***



***Analysis of results, what in these data can be seen:*** *In this case we input all needed values into the table according to the rules of filling and constraint execution.*

***9. Query goal (INSERT INTO a table CUSTOMER\_ORDER):*** *Insert the values into the CUSTOMER\_ORDER table with needed attributes.*

***Query SQL code:***

*BEGIN*

*INSERT INTO CUSTOMER\_ORDER VALUES(c\_order(01,01,'10-02-2015',2500));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(02,02,'15-03-2015',1500));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(03,03,'28-01-2015',700));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(04,04,'17-04-2015',830));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(05,05,'15-04-2015',9000));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(06,06,'10-09-2015',7120));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(07,07,'15-06-2015',3000));*

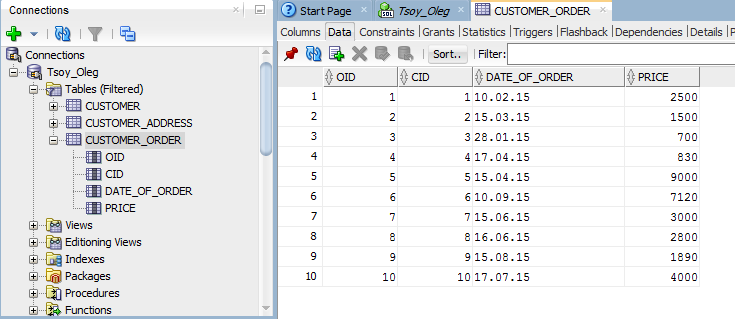
*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(08,08,'16-06-2015',2800));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(09,09,'15-08-2015',1890));*

*INSERT INTO CUSTOMER\_ORDER VALUES (c\_order(10,10,'17-07-2015',4000));*

*END;*

***Result of execution:***



***Analysis of results, what in these data can be seen:*** *In this case we input all needed values into the table according to the rules of filling and constraint execution.*

***10. Query goal (INSERT INTO a table CUSTOMER\_ADDRESS):*** *Insert the values into the CUSTOMER\_ADDRESS table with needed attributes.*

***Query SQL code:***

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(01,01,'ERIC PEARCE','USA','DENVER',897378));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(02,02,'VANESSA GEAR','CZECH REPUBLIC','PRAGUE',897895));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(03,03,'OLEG TSOY','UZBEKISTAN','TASHKENT',100124));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(04,04,'FARZUNA KHAMITOVA','UZBEKISTAN','TASHKENT',100220));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(05,05,'DURBEK FAYZULLAYEV','UZBEKISTAN','TASHKENT',100425));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(06,06,'ZARINA BEGULOVA','UZBEKISTAN','TASHKENT',100122));*

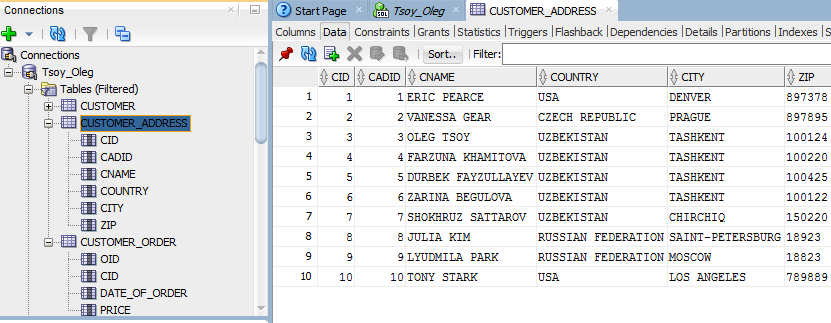
*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(07,07,'SHOKHRUZ SATTAROV','UZBEKISTAN','CHIRCHIQ',150220));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(08,08,'JULIA KIM','RUSSIAN FEDERATION','SAINT-PETERSBURG',18923));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(09,09,'LYUDMILA PARK','RUSSIAN FEDERATION','MOSCOW',18823));*

*INSERT INTO CUSTOMER\_ADDRESS VALUES (PERSON(10,10,'TONY STARK','USA','LOS ANGELES',789889));*

***Result of execution:***



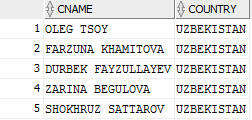
***Analysis of results, what in these data can be seen:*** *In this case we input all needed values into the table according to the rules of filling and constraint execution.*

***11. Query goal (Output of objects and its components (SELECT), using function Value():***

***Query SQL code:***

***select VALUE(A) from CUSTOMER A where VALUE(A).COUNTRY='UZBEKISTAN';***

***Result of execution:***

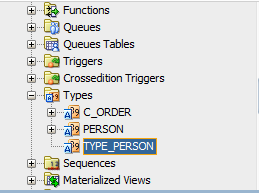


***12. Query goal (create a type Type\_Person for showing the detailed construction of Table of Person):***

***Query SQL code:***

*CREATE OR REPLACE TYPE TYPE\_PERSON AS TABLE OF PERSON;*

***Result of execution:***



***Analysis of results:*** *As can be seen this type was created specially for detailed self-understanding in my problematic issue during this practical assignment.*

***13. Query goal (Output of objects and its components (SELECT), using function Table():***

***Query SQL code:***

CREATE TABLE COMPANY(

COMPANY\_ID NUMBER PRIMARY KEY,

COMPANY\_NAME VARCHAR2(30),

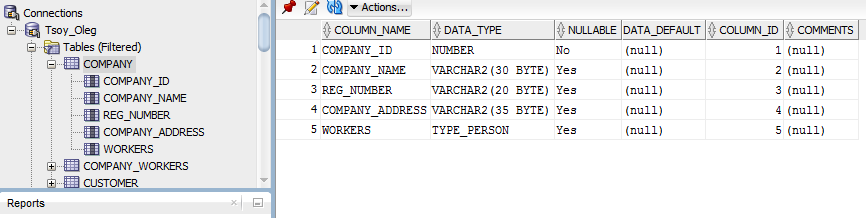
REG\_NUMBER VARCHAR2(20),

COMPANY\_ADDRESS VARCHAR2(35),

WORKERS TYPE\_PERSON)

NESTED TABLE WORKERS STORE AS COMPANY\_WORKERS;

**Result of execution:**



**Analysis of results:** In this given case we can observe that now there 1 table Company and one nested table COMPANY\_WORKERS.

***14. Query goal (Output of objects and its components (SELECT), using function Table(): Create a sequence of COMPANY\_ROW & INSERTION COMPANY VALUES:***

***Query SQL code:***

CREATE SEQUENCE COMPANY\_ROW

START WITH 1

MINVALUE 1

MAXVALUE 999;

INSERT INTO COMPANY VALUES(

1,

'PARTNERSHIP TENDER',

'91204',

'GAUSTOVENOU',

TYPE\_PERSON(

PERSON('JOHN','SMITH','12012-13013','UZBEKISTAN','MALE'),

PERSON('JANE','OSTIN','14014-15015','ENGLAND','FEMALE'),

PERSON('JOSH','KIRK','17017-19019','UZBEKISTAN','MALE')

));

INSERT INTO COMPANY VALUES(

1,

‘General Armor Distributors',

'8591204',

'Glasgow str 88',

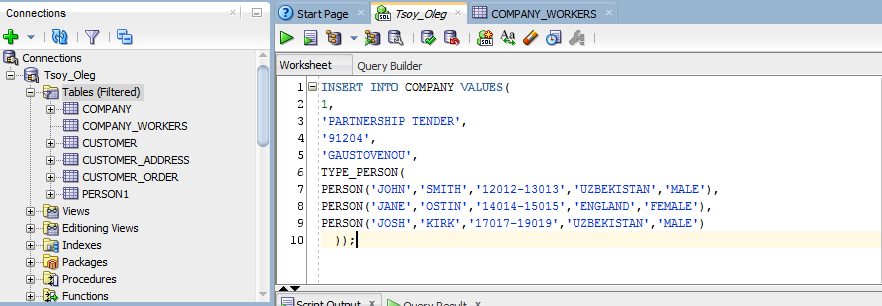
TYPE\_PERSON(

PERSON('WALTER','SUMMER','88012-13013','MEXICO','MALE'),

PERSON('ALAN','FADE','14854-15015','CANADA','MALE'),

PERSON('JAKE','MARIE','18617-19019','SWEDEN','MALE')

));

***Result of execution:***

***Analysis of results:*** *In this screenshot we can see the insertation of values into COMPANY VALUES for the first company and second company.*

***15. Query goal (Make a view of CUSTOMER TABLE by SQL statements):***

***Query SQL code:***

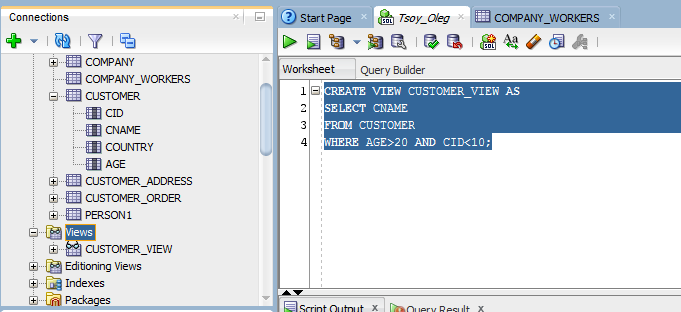
CREATE VIEW CUSTOMER\_VIEW AS

SELECT CNAME

FROM CUSTOMER

WHERE AGE>20 AND CID<10;

***Result of execution:***



***Analysis of execution:*** *In this case I observe the creation of the view of the table by SQL statement the main basics of the view became CUSTOMER table with specialized requirements(conditions).*

***16. Query goal (Metadata):***

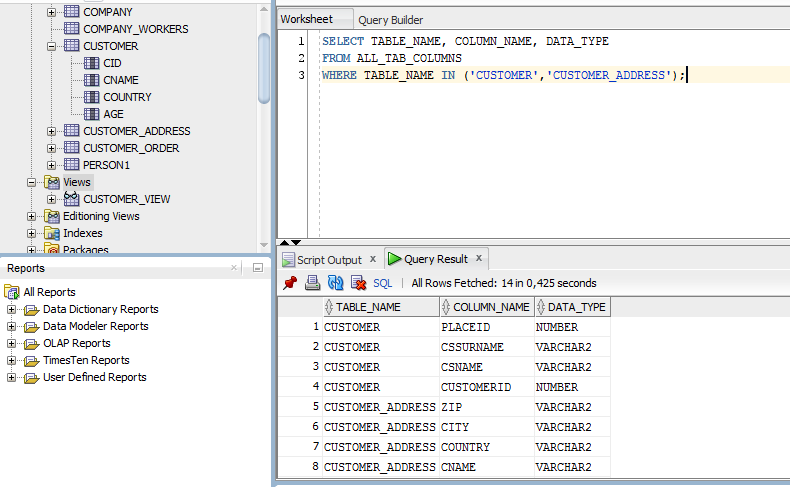
***Query SQL code:***

SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE

FROM ALL\_TAB\_COLUMNS

WHERE TABLE\_NAME IN ('CUSTOMER','CUSTOMER\_ADDRESS');

***Result of execution:***



***Analusis of results:*** *In metadata we can see all data type with column name and table names which were chosen by user in SQL statement.*

***17. Query goal (MAKE\_REF()):***

***Query SQL code:***

CREATE TABLE INVEST

(

INVEST\_ID NUMBER,

INVEST\_NAME VARCHAR2(25),

INPUT NUMBER(8,2),

PRIMARY KEY (INVEST\_ID, INVEST\_NAME)

);

CREATE OR REPLACE type INV

AS

object

(

INVEST\_ID NUMBER,

INVEST\_NAME VARCHAR2(25),

INPUT NUMBER(8,2));

CREATE VIEW VIEW\_INVEST OF INV

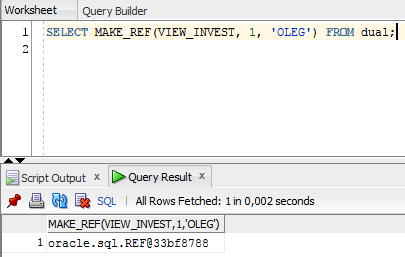
WITH object identifier(INVEST\_ID, INVEST\_NAME) AS

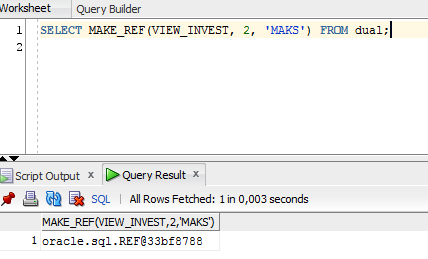
SELECT \* FROM INVEST;

SELECT MAKE\_REF(VIEW\_INVEST, 1, 'OLEG') FROM dual;

SELECT MAKE\_REF(VIEW\_INVEST, 2, 'MAKS') FROM dual;

***Result of execution:***

******



***Analysis of results:*** *In usage of make\_ref function in SQL developer it shows the oracle SQL code refences which contains only machine-read information.*

***18. Query goal SUBMULTISET [OF]:***

***Query SQL code:***

CREATE OR REPLACE type TYPE\_PERSON

AS

TABLE OF PERSON;

CREATE TABLE COMPANY

(

COMPANY\_ID NUMBER PRIMARY KEY,

COMPANY\_NAME VARCHAR2(30),

REG\_NUM VARCHAR2(20),

ADDRESS VARCHAR2(35),

PARTICIPANTS TYPE\_PERSON

)

nested TABLE WORKERS store AS COMPANY\_WORKERS;

INSERT INTO COMPANY VALUES(

1,

'OLEG CORP.',

'108885',

'UGANDA 45',

TYPE\_PERSON(

PERSON('Vlad','Prada','150596-16895','Russia','male'),

PERSON('Andy','Fixer','120795-12592','USA','male'),

PERSON('Jet','Zeror','010986-10258','China','male'),

PERSON('Po','Wert','150596-16895','Japan','male')

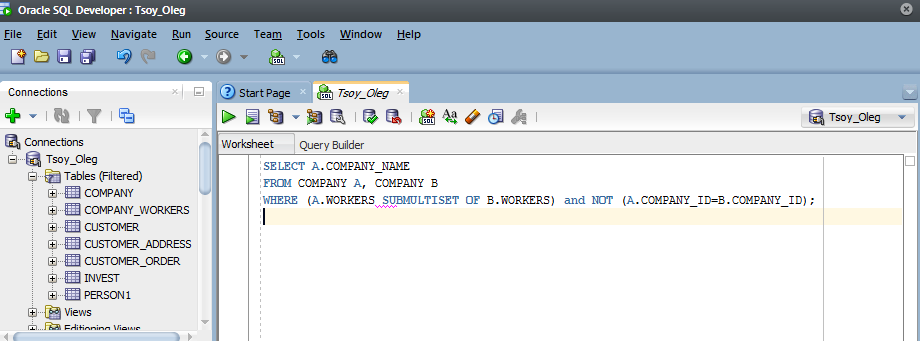
));

SELECT A.COMPANY\_NAME

FROM COMPANY A, COMPANY B

WHERE (A.WORKERS SUBMULTISET OF B.WORKERS) and NOT (A.COMPANY\_ID=B.COMPANY\_ID);

***Result of execution:***



***Analysis of results:*** *This code will show name of OLEG CORP. during the execution. Because of the first statement of execution is based on the table with inforamtion about OLEG CORP.*

***19. Query goal MEMBER [OF]:***

***Query SQL code:***

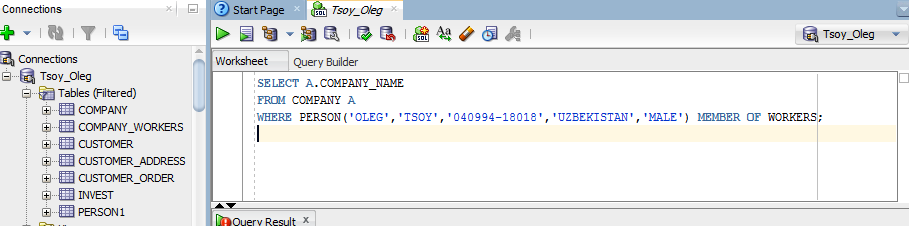
SELECT A.COMPANY\_NAME

FROM COMPANY A

WHERE PERSON('OLEG','TSOY','040994-18018','UZBEKISTAN','MALE') MEMBER OF

WORKERS;

***Result of execution:***



***Analysis of results:*** *This SQL statement will show OLEG TSOY as a participant in a compamy membership due to the usage of MEMBER [OF] function.*

***20. Query goal IS [NOT] A SET:***

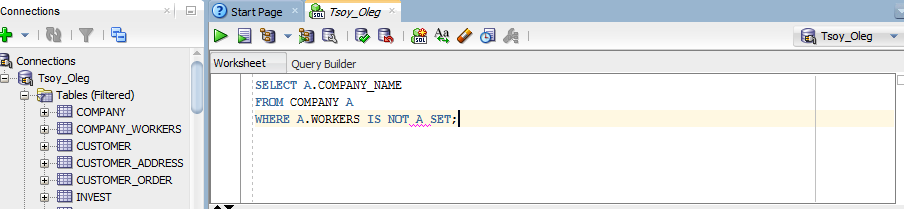
***Query SQL code:***

SELECT A.COMPANY\_NAME

FROM COMPANY A

WHERE A.WORKERS IS NOT A SET;

***Result of execution:***

******

***Analysis of results:*** *This SQL statement will show that “no rows selected” because of usage IS NOT A SET. According to my conditions I do not have needed rows and data in it that is why computer will answer by simple sentence (no rows selected).*

***21. Query goal CARDINALITY():***

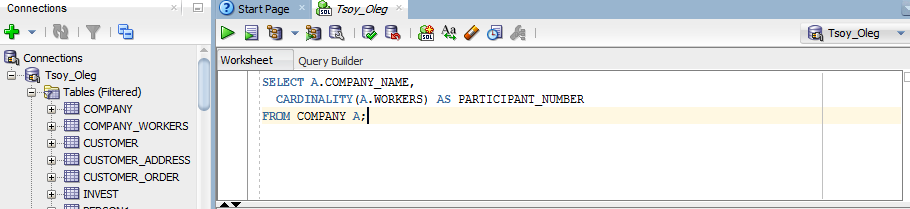
***Query SQL code:***

SELECT A.COMPANY\_NAME,

CARDINALITY(A.WORKERS) AS PARTICIPANT\_NUMBER

FROM COMPANY A;

***Result of execution:***



***Analysis of results:*** *The CARDINALITY finction demonstrates the calculation results of number of employees(participants) in all 3 companies which were registered in the system.*

# Conclusion

According to my first homework I can say that the task seems not so difficult at the moment but duting my preparations I found some disadvantages connected with my SQL knowledge. Actually I am not strong in SQL because of I started to learn previus semester and before that experience I have never use this useful language at all. However, now my experience could not be called ‘Professional or Amateur’ but I think that I am on the right way. In this semester the main problem is a time for this important and difficult cource. I think I have enough strength to finish this cource for this short period of time. While, my first homework does not approve it but I think I will improve it more and more before the end of this semester.

# References

1. C. A. Mackay. SQL Injection Attacks and Some Tips on How to Prevent Them. Technical report, The Code Project, January 2005. <http://www.codeproject.com/cs/database/SqlInjectionAttacks.asp>.
2. O. Maor and A. Shulman. SQL Injection Signatures Evasion. White paper, Imperva, April 2004. [http://www.imperva.com/application defense center/white papers/sql injection signatures evasion.html](http://www.imperva.com/application%20defense%20center/white%20papers/sql%20injection%20signatures%20evasion.html).
3. M. Martin, B. Livshits, and M. S. Lam. Finding Application Errors and Security Flaws Using PQL: A Program Query Language. In Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming systems languages and applications(OOPSLA 2005), pages 365–383, 2005.
4. R. McClure and I. Kruger. SQL DOM: Compile Time Checking of ¨ Dynamic SQL Statements. In Proceedings of the 27th International Conference on Software Engineering (ICSE 05), pages 88–96, 2005.
5. S. McDonald. SQL Injection: Modes of attack, defense, and why it matters. White paper, GovernmentSecurity.org, April 2002.