

RIGA TECHNICAL UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

INSTITUTE OF APPLIED COMPUTER SYSTEMS

“Technology of Large database”

Practical task 1#

**Object-relational DB storage structures**

Author: Batuhan Arslanoglu

Studentcardno.: 151ADB096

2017 / 2018 study year

Contents

[1. Task description 4](#_Toc505462506)

[2. Table with row type objects. 4](#_Toc505462507)

[2.1 Map Member function 4](#_Toc505462508)

[Creation 4](#_Toc505462509)

[Input of data 5](#_Toc505462510)

[Output of data 5](#_Toc505462511)

[2.2 Member type method 5](#_Toc505462512)

[Creation 5](#_Toc505462513)

[Input of data 6](#_Toc505462514)

[Output of data 6](#_Toc505462515)

[2.4 Member Procedure 7](#_Toc505462516)

[Creation 7](#_Toc505462517)

[Input of data 7](#_Toc505462518)

[Output of data 8](#_Toc505462519)

[3. Creation of object column and object table 8](#_Toc505462520)

[Creation 8](#_Toc505462521)

[Input of data 9](#_Toc505462522)

[Output of data 9](#_Toc505462523)

[4. Object View 10](#_Toc505462524)

[Creation 10](#_Toc505462525)

[Input of data 12](#_Toc505462526)

[Output of data 12](#_Toc505462527)

[5. Table with object collection. 13](#_Toc505462528)

[Creation 13](#_Toc505462529)

[Input of data 14](#_Toc505462530)

[Output of data 14](#_Toc505462531)

[6. Creation of object type 15](#_Toc505462532)

[Creation 15](#_Toc505462533)

[Input of data 16](#_Toc505462534)

[Output of data 16](#_Toc505462535)

[7. Conclusion 18](#_Toc505462536)

# Task description

* Creation of object table with row type objects. Input of data (INSERT). Output of metadata (SELECT), output of objects and its components (SELECT) using function Value(). Map or ORDER and MEMBER methods creation for row type objects. Queries with methods (min. 3).
* Creation of one to many (1 : N) object relationship between relational table with object column and object table. Input of data. Input of object identifiers using function REF(). Data extraction (all objects and object components) using function DEREF().
* Creation of object view with 2 methods from 2 tables (relational table and relational table with column objects). Realization of queries from object view (min. 2).
* Creation of table with collection. For collection objects is created Member type method. Data input. Queries with method (min.2).
* Creation of object type hierarchy (min. 4 object types) with methods. Creation of object table with different type of objects (according defined type hierarchy). Data input, queries with TREAT(), IS OF TYPE(), SYS\_TTYPEID(). Realization of methods inheritance. Queries.
* Conclusions.

# [Table with row type objects.](https://datubaze.files.wordpress.com/2015/09/l_ang_db2_3_tabula_ar_rindas_tipa_objektiem.docx)

I have created tables with row type objects for three methods of Map, Order, Member:

## 2.1 Map Member function

### Creation

|  |
| --- |
| CREATE OR REPLACE TYPE LV\_Learner AS OBJECT(  ID NUMBER,  NAME VARCHAR2(30),  SURNAME VARCHAR2(30),  WRITING NUMBER,  READING NUMBER,  LISTENING NUMBER,  SPEAKING NUMBER,  MAP member function OVERALL return number);  /  create or replace type body LV\_Learner AS  MAP member function OVERALL return number is  begin  return (SELF.WRITING+SELF.READING+SELF.LISTENING+SELF.SPEAKING)/4;  end OVERALL;  end;  /  create table LV\_learners of LV\_Learner;  / |

### Input of data

|  |
| --- |
| insert into LV\_learners values(LV\_Learner(1, 'John', 'Smith', 8, 7, 9, 6));  insert into LV\_learners values(LV\_Learner(2, 'Batuhan', 'Arslanoglu', 9, 5, 10, 7));  insert into LV\_learners values(LV\_Learner(3, 'Batuhan', 'Arslanoglu', 4, 6, 5, 3));  / |

### Output of data

|  |
| --- |
| select Value(A).NAME, Value(A).SURNAME, Value(A).OVERALL()  from LV\_learners A;  /  A screenshot of a cell phone  Description generated with very high confidence |

## 2.2 Member type method

### Creation

|  |
| --- |
| CREATE OR REPLACE TYPE Bank AS OBJECT(  ID NUMBER,  SALARY NUMBER,  EXPENSES NUMBER,  BALANCE NUMBER,  MEMBER function GET\_SALARY return number,  MEMBER function CURRENT\_BALANCE return number);  /  create or replace type body Bank AS  MEMBER function CURRENT\_BALANCE return number is  begin  return SELF.BALANCE+SELF.SALARY-SELF.EXPENSES;  end CURRENT\_BALANCE;  MEMBER function GET\_SALARY return number is  begin  return SELF.SALARY;  end GET\_SALARY;  end;  /  create table bank\_account of Bank;  /  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| insert into bank\_account values(Bank(1, 500, 250, 1500));  insert into bank\_account values(Bank(2, 800, 400, 600));  insert into bank\_account values(Bank(3, 500, 250, 800));  insert into bank\_account values(Bank(4, 450, 150, 1000));  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1#:**  select Value(A).ID, Value(A).GET\_SALARY(), Value(A).CURRENT\_BALANCE()  from bank\_account A;  A screenshot of a cell phone  Description generated with very high confidence |

## 2.4 Member Procedure

### Creation

|  |
| --- |
| create type Person\_typ as object(  id number,  name varchar2(20),  surname varchar2(20),  gender varchar2(20),  MEMBER procedure FULL\_NAME (SELF IN OUT NOCOPY Person\_typ));  create type body Person\_typ as  MEMBER PROCEDURE FULL\_NAME (SELF IN OUT NOCOPY Person\_typ) IS  BEGIN  DBMS\_OUTPUT.PUT\_LINE('Full Name: ' || name || ' - ' || surname);  END;  end;  /  create table person\_tab of Person\_typ;  /  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| insert into person\_tab values(Person\_typ(1, 'Batuhan', 'Anrsaloglu', 'male'));  insert into person\_tab values(Person\_typ(2, 'Jane', 'Kurser', 'female'));  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1#:**  select Value(A).id, Value(A).name from person\_tab A;  A screenshot of a cell phone  Description generated with very high confidence |

# Creation of object column and object table

I have created table with object column type to store data of INSURANCE company`s client:

### Creation

|  |
| --- |
| create type item\_typ as object(  name varchar2(20),  description varchar2(20),  price number);  /  create type address\_typ as object(  street VARCHAR2(15),  city VARCHAR2(15),  state CHAR(2)  );  /  create type person\_typ as object(  name varchar2(20),  surname varchar2(20),  gender varchar2(20)  );  /  create type insurance as object(  id number,  person person\_typ,  address address\_typ,  item item\_typ  );  /  create table insurance\_tab of insurance;  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| insert into insurance\_tab values (insurance( 1,  person\_typ('Janis', 'Grabis', 'male'),  address\_typ('kalku 1', 'Riga', 'LV'),  item\_typ('Car', 'BWM car', 2000)));  insert into insurance\_tab values (insurance( 2,  person\_typ('Ruslan', 'Yurinkov', 'male'),  address\_typ('Elizabetes 103', 'Berlin', 'DE'),  item\_typ('Tank', 'Toyato car', 2000)));    insert into insurance\_tab values (insurance( 3,  person\_typ('thompson', 'Kear', 'male'),  address\_typ('Medved 15', 'Moscow', 'RU'),  item\_typ('Car', 'BWM car', 2000)));  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1#:**  select Value(A).person.name, Value(A).address.city, Value(A).item.name  from insurance\_tab A;  A screenshot of a cell phone  Description generated with very high confidence |

# Object View

I have created database for a company where they can store the product which they produce and there will be data about product name, testing result and others.

### Creation

|  |
| --- |
| CREATE TABLE product (  pk\_id INTEGER NOT NULL,  name VARCHAR2(512),  price INTEGER,  CONSTRAINT product\_pk PRIMARY KEY (pk\_id));  /  CREATE TABLE test\_product (  test\_id INTEGER NOT NULL,  result\_test VARCHAR2(45) NOT NULL,  product INTEGER NOT NULL,  CONSTRAINT test\_product\_pk PRIMARY KEY (test\_id, result\_test, product),  CONSTRAINT product\_fk FOREIGN KEY (product) REFERENCES product (pk\_id));  /  CREATE TYPE test\_product\_typ AS TABLE OF VARCHAR2(45);  CREATE TYPE product\_typ AS OBJECT (  pt\_id INTEGER,  pt\_name VARCHAR2(512),  price INTEGER,  test\_result test\_product\_typ,  MEMBER FUNCTION set\_product (new\_product\_name IN VARCHAR2,  new\_product\_type IN VARCHAR2, new\_price IN INTEGER)  RETURN product\_typ,  MEMBER FUNCTION set\_testing (new\_testing IN test\_product\_typ)  RETURN product\_typ,  PRAGMA RESTRICT\_REFERENCES (DEFAULT, RNDS, WNDS, RNPS, WNPS)  );  /  CREATE TYPE BODY product\_typ  AS  MEMBER FUNCTION set\_product (new\_product\_name IN VARCHAR2,  new\_product\_type IN VARCHAR2, new\_price IN INTEGER)  RETURN product\_typ  IS  pt\_container product\_typ := SELF;  BEGIN  pt\_container.pt\_name := new\_product\_name;  pt\_container.price := new\_price;  RETURN pt\_container;  END;  MEMBER FUNCTION set\_testing (new\_testing IN test\_product\_typ)  RETURN product\_typ  IS  test\_container product\_typ := SELF;  BEGIN  test\_container.test\_result := new\_testing;  RETURN test\_container;  END;  END;  /  CREATE VIEW product\_v  OF product\_typ  WITH OBJECT OID (pt\_id)  AS  SELECT p.pk\_id, p.name, p.price,  CAST (MULTISET (SELECT result\_test  FROM test\_product t  WHERE t.test\_id = p.pk\_id)  AS test\_product\_typ)  FROM product P;  /  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| insert into product values(1, 'headphone', 5000);  insert into product values(2, 'Cover for iphone 5s', 4100);  insert into product values(3, 'Glasses with black cover', 1000);  insert into product values(4, 'Keyboard for PC', 14000);  insert into test\_product values(1, 'Successfully', 1);  insert into test\_product values(2, 'Failed', 2);  insert into test\_product values(3, 'Successfully', 3);  insert into test\_product values(4, 'Successfully', 4);  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1#:**  SELECT pt\_id, pt\_name, price, test\_result  FROM product\_v;  /  A screenshot of a cell phone  Description generated with very high confidence  **QUERY\_2#:**  SELECT pt\_id, pt\_name, price, test\_result  FROM product\_v  WHERE price > 5000;  /  A screenshot of a cell phone  Description generated with very high confidence |

# [Table with object collection.](https://datubaze.files.wordpress.com/2015/09/l_ang_db2_5_tabula_ar-kolekciju.docx)

### Creation

|  |
| --- |
| CREATE TYPE BUDGET AS OBJECT(  MONTH VARCHAR2(20),  SALARY NUMBER,  EXPENSES NUMBER,  BALANCE NUMBER,  MEMBER function GET\_SALARY return number,  MEMBER function CURRENT\_BALANCE return number);  /  CREATE TYPE BODY BUDGET AS  MEMBER function CURRENT\_BALANCE return number is  BEGIN  RETURN SELF.BALANCE+SELF.SALARY-SELF.EXPENSES;  END CURRENT\_BALANCE;  MEMBER function GET\_SALARY RETURN NUMBER IS  BEGIN  RETURN SELF.SALARY;  END GET\_SALARY;  END;  /  CREATE TYPE PERSON\_BUDGET AS TABLE OF BUDGET;  /  CREATE TABLE PERSON\_BUDGET\_T(  IDNO NUMBER,  BUDGET\_C PERSON\_BUDGET)  NESTED TABLE BUDGET\_C STORE AS BUDGET\_NESTED;  /  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('JANUARY', 250, 50, 400)));  INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('MARCH', 150, 150, 700)));  INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('MARCH', 500, 150, 800)));  INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('MAY', 400, 150, 700)));  INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('SEPTEMBER', 100, 150, 200)));  INSERT INTO PERSON\_BUDGET\_T VALUES(1, PERSON\_BUDGET(BUDGET('DECEMBER', 50, 10, 400)));  /  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1#:**  SELECT N.MONTH, N.BALANCE, N.SALARY, N.EXPENSES, N.GET\_SALARY(), N.CURRENT\_BALANCE()  FROM PERSON\_BUDGET\_T P,  TABLE(P.BUDGET\_C) N;  A screenshot of a cell phone  Description generated with very high confidence  **QUERY\_2#:**  SELECT N.MONTH, N.GET\_SALARY(), N.CURRENT\_BALANCE()  FROM PERSON\_BUDGET\_T P,  TABLE(P.BUDGET\_C) N  WHERE N.MONTH = 'MARCH';  A screenshot of a cell phone  Description generated with very high confidence |

# Creation of object type

### Creation

|  |
| --- |
| CREATE OR REPLACE TYPE PRODUCT\_TYP AS OBJECT(  product\_id NUMBER,  product\_name VARCHAR2(20),  description VARCHAR2(20)) NOT FINAL;    CREATE OR REPLACE TYPE LAMP\_TYP UNDER PRODUCT\_TYP(  type VARCHAR2(20),  made\_year DATE,  price NUMBER) NOT FINAL;  CREATE OR REPLACE TYPE CLOTHES\_TYP UNDER PRODUCT\_TYP(  COLOR varchar2(20),  TYPE varchar2(20)  );  CREATE OR REPLACE TYPE VEGETABLES\_TYP UNDER PRODUCT\_TYP(  DELIVERED\_DATE DATE,  price NUMBER  );  CREATE TABLE LAMP OF LAMP\_TYP;  CREATE TABLE VEGETABLE OF VEGETABLES\_TYP;  CREATE TABLE CLOTHE OF CLOTHES\_TYP;  A screenshot of a cell phone  Description generated with very high confidence |

### Input of data

|  |
| --- |
| INSERT INTO LAMP VALUES(LAMP\_TYP(100, 'Eco lamp', 'new lamp', 'economical', '03-MAY-17', 50));  INSERT INTO LAMP VALUES(LAMP\_TYP(101, 'Light lamp', 'Plastic', 'economical', '03-SEP-15', 70));  INSERT INTO CLOTHE VALUES(CLOTHES\_TYP(201, 'T-SHIRT', 'Cotton t-shirt', 'red', 'x size'));  INSERT INTO CLOTHE VALUES(CLOTHES\_TYP(202, 'T-SHIRT', 'Tight t-shirt', 'yellow', 'medium size'));  A screenshot of a cell phone  Description generated with very high confidence |

### Output of data

|  |
| --- |
| **QUERY\_1:**  select VALUE(A).product\_id, VALUE(A).product\_name, VALUE(A).COLOR from CLOTHE A;  A screenshot of a cell phone  Description generated with very high confidence  **QUERY\_2:**  SELECT A.\* FROM LAMP A;  A screenshot of a cell phone  Description generated with very high confidence  **QUERY\_3: FUNCTION TREAT()**  select TREAT(VALUE(A) as PRODUCT\_TYP)  from LAMP A;  **A screenshot of a cell phone  Description generated with very high confidence** |

|  |
| --- |
| **QUERY\_4: FUNCTION TREAT()**  select A.\*  from CLOTHE A  where TREAT(VALUE(A) as CLOTHES\_TYP).COLOR = 'red';  A screenshot of a cell phone  Description generated with very high confidence  **QUERY\_4:** IS OF TYPE()  select A.\*  from LAMP A  where VALUE(A) IS OF TYPE (LAMP\_TYP);  A screenshot of a cell phone  Description generated with very high confidence |

# Conclusion

In this practical work, I have learned object relational database principles. I have had difficult time during the implementation of this practical work, but I have completey fixed all error and finished it.

It is clear to me that I have learned many thing by doing this practical work and I understood how to work with objects, because before I used Relational Database and this is different to me. There are many similarity and I did easily it.