2-14 – base Scalar Data types (char, varchar2, number, date, timestamp)

110 – OPERATORS

130 - %TYPE

133 - INSERT, UPDATE, DELETE, MERGE

154 – if else

157 – CASE

161 – logic table

165 – basic loop

167 – while loop

170 – for loop

192 – Record (%rowtype)

228 – cursor

231 – cursor attributes

1

Benefits of PL/SQL

Integration of procedural constructs with SQL – The most important advantage of PL/SQL is the integration of procedural constructs with SQL.SQL is a nonprocedural language.

Improved performance - Without PL/SQL, you would not be able to logically combine SQL

statements as one unit.

Modularized program development - Modularized program development: A basic unit in all PL/SQL programs is the block. Blocks can be in a sequence or they can be nested in other blocks.

Integration with Oracle tools - The PL/SQL engine is integrated in Oracle tools such as Oracle Forms, Oracle Reports, and so on.

Portability - Oracle Reports, and so on. When you use these

Exception handling - PL/SQL enables you to handle exceptions efficiently.

BLOCK TYPES

Anonymous(executed at runtime, are not stored in databases)

Procedure(they are named PL/SQL blocks and they are stored in the databases)

Function(similar to a procedure, except that funcxtopn must return a value)

2

Must start with a letter, can include numbers, letters special characters

Must contain no more than 30 words

Must not include reserved words.

Variables = declare in declare section, in executable assigned new values, pass parameters to subprograms, hold the output of a subprogram

DELIMITER - := q’!FATHER’s day!’; // []

VARIABLES: Scalar, Composite (RECORD and TABLES), Refernece(pointers), LOB Data types (large object)

Define date: ime DATE =: SYSDATE +7;

Define constant: ime CONSTANT NUMBER(3,2) := 8.25;

%TYPE – declare a database column definition (identifier table.column\_name%TYPE;)

%TYPE => database column, another declared variable

BIND Variables (VARIABLE b\_result NUMBER) , printing with PRINT

ROWTYPE% => declare record can hold entire row of a table or view.

3

Identifiers are the named given to PL/SQL objects.

Only single-row functions are available in procedural statements.(AVG, MIN, MAX, COUNT, SUM, STDDEV AND VARIANCE)

SELECT NVL(ROUND(salary/1000) - na strain kao resenje 314

LENGTH(vl); – za stringu

MONTHS\_BETWEEN (CURRENT\_DATE, v\_hiredate);

my\_seq.NEXTVAL

Converts data type implicit and explicit (FUNCTION – TO\_CHAR, Date, number, tipestamp)

**IMPLICIT CONVERSION:** PL/SQL attempts to convert data types dynamically if they are mixed in a statement.

**EXPLICIT CONVERSION:** to convert from one to another data type use built-in function.

Nested blocks

BEGIN <<sale>>

DECLARE

BEGIN

cc VARCHAR2(20) := ‘nikolic’;

DECLARE

BEGIN

BDMS\_OUTPU.PUT\_LINE(sale.cc);

END;

END;

END SALE;

OPERATORS – Comparison(IS, NULL, LIKE, BETWEEN, IN)

4

DECLARE

hire\_date employees.hire\_date%TYPE;

BEGIN

SELECT hire\_date, first\_name INTO hire\_d, first\_n FROM employees WHERE employee\_id = 107;

INSERT, UPDATE, DELETE, MERGE

CURSOR – is a pointer to the private memory area allocated by the Oracle server. It is used to handle the result of a SELECT statement.

IMLICIT: Created and managed internally by the Oracle server.

Explicit: Declared explicitly by the programmers.

SQL%FOUND, SQL%NOTFOUND, SQL%ROWCOUNT

5

If , loop, case

6

Composite data types:

Records

Collections( tables or associative arrays, nested tables, VARRAY)

%ROWTYPE (sve column iz tablice, %TYPE jednu column iz tablice)

**RECORD**

TYPE t\_ime IS RECORD

(v\_sal number(8),

V\_rec1 employees%rowtype);

V\_myrec v\_ime;

IDENTIFIER (chosen name of the whole record)

REFERENCE (name of table, view, cursor variable on which the record is to be based)

NOT NULL – must be declared , not import!!

**TABLE**

DECLARE

TYPE name\_table IS TABLE OF

Employees.last\_name%TYPE

INDEX BY PLS\_INTEGER;

moja name\_table;

BEGIN

moja(1) := ’cameron’;

IF moja.EXISTS(7) THEN

…….

END IF;

END;

// using INDEX BY – EXISTS, COUNT, FIRST, LAST, PRIOR, NEXT, DELETE

DECLARE

TYPE moja IS TABLE OF

Employees%ROWTYPE INDEX BY PLS\_INTEGER;

moja\_nova moja;

**VARRAY** – varray is contrained in size

TYPE moj IS VARRAY(3) OF

locations.city%TYPE;

moje\_novo moj;

7

DECLARE

CURSOR c\_emp\_cursor IS

SELECT employee\_id, last\_name FROM employees

WHERE department\_id = 30;

v\_empno employees.employee\_id%TYPE;

v\_lname employees.last\_name%TYPE;

BEGIN

OPEN c\_emp\_cursor;

LOOP

FETCH c\_emp\_cursor INTO v\_empno, v\_lname;

EXIT WHEN c\_emp\_cursor%NOTFOND;

BMS\_OUTPUT.PUT\_LINE(v\_empno || ‘ ‘ || v\_lname);

END LOOP;

CLOSE c\_emp\_cursor;

END;

SHORT VERSION (OPEN, FETCH, CLOSE)

BEGIN

FOR emp\_record IB c\_emp\_cursor

LOOP

MDBS\_OUTPUT.PUT\_LINE(emp\_record.employee\_id)

END LOOP;

END;

%ISOPEN, %NOTFOUND , %FOUND, %ROWCOUNT