Pl/sql programs lab manual pdf

PLS'L S'L SERVER PROGRAM:-This is a demo WHILE looploop counter: 1loop counter: 2loop counter: 3 cycle counter: 4 cycle counter: 5 cycle counter: 6 cycle counter: 8 cycle counter : 9 cycle counter: 10PS:syntax:while qt;condition Example: 4Subscribe to EMPDATA. S'L to get information about an employee whose number is entered by the user. Program -- PROGRAM TO RETRIEVE EMP DETAILSSset serveroutput onprompt Enter employee number: take ndeclare dname emp.emp_name%type; dbasic emp.emp_basic%; ddesig emp.desig%type;start choosing emp_name, basic, design in dname, dbasic, ddesig from emp, where emp_no and n;dbms_ouput.put'line ('Employee Details:' ;d bms'output.put'line ('Name: 'dname'; dbms_output.put'line (Basic: dbasic) ;d bms-output.put'line ('Destination: 'ddesig;;/OUTPUT:-enter employee number:13old 9:where eno n;new 9:where eno-13;employee detailsName:allenbasic:9500desig:mechsetoutput offPS:Similarly, you can use other S'L operators in the PL/S'LExercises:1) Write the code PL/S'L, EX_INVNO. S'L, a block for inverting numbers using all forms of cycles. ANSWER:-declaring number (20): 123;s number (13): 0;d number (3): 1;r number (3): 10;begindbms_output.put'line ('number:' n); While n'gt;0 is fixated: mod (n.10);s:' (s'r)d;n:n/r;end loop;dbms_output.put'line ('inverted values's);end;/OUTPUT:number:1232) Write the code PL/S'L, EX_SUMNO. S'L, which prints the amount of natural numbers 'n'. ANSWER:-fast number enter: take number ndeclareisum number (2): 0;i number;n number: n;startfor i at 1.n loopisum:'isum'i;end loop;dbms_output.put'line ('sum isum);end;output:-enter number:7sum 283) Write the code PL/S L, EX_AREA. S'L, the block for calculating the circle area for radius values range from 3 to 7. Keep the radius and corresponding values of the settlement area in the AREA_VALUES table. ANSWER: -set serveroutput number ondeclarearea (5);rad number (3);p i number (4): 3.14;startfor rad in 3.7 looparea: 'pi'rad'rad;dbms_output.put'line(' area is' area);insert into the values of the area_values (area, rad);end of the loop;/OUTPUT:-area:27 area :48 area :75 area :108 area :108 area :147 S'gt;elected with area_values; area rad No 27 3 48 4 75 5 108 6 147 7WEEK 9 (PL/S'L)6) TheTk (i) Creating a simple program PL/S'L, That includes an ad section, a completed section and an exception processing section (e.g.: Student signs can be selected from a table and printed for those who have secured first class and an exception can be raised if no entries have been madeInsert the data into the student table and use COMMIT, ROLLBACK, and SAVEPOINT in S'L.7) Develop a program that includes NESTED IF, CASE, and CASE features. The program can be expanded with the functions of NULLIFTheTkkjfdkjlkjskjlkfd and COALESCE.8) Program using WHILE LOOPS, numerical FOR LOOPS, nested cycles using ERROR processing, BUILT IN exceptions, USER of certain exceptions, RAISE APPLICATION ERROR.9) Development of the program using the creation of procedures, passing the parameters in and out of procedures.10) Develop the program using the creation of a saved function, call functions in statements and record complex functions.11) Develop a program using the creation of a package specification, package bodies, private objects, variable packages and cursors and call saved packages.12) Develop programs using the functions of the parameters in THE CURSOR, FOR UPDATE. S'L) to insert into a new table, NEWEMP, the entry of any employee whose number is entered by the user.1 Create a table NEWEMP qlt;emp_no, emp_name, join_date, basic).2.open and a type of the following program.programprompt' = dname= varchar2(30); = ddate = date; = dbasic = number(10);begin = select = emp_no,= emp_name,= join_date,= basic = into = dno,= dname,= ddate,= dbasic = from = emp = where = emp_no=&userno; if sql%rowcount 0; then insert into the values of newemp (dno, dname, ddate, dbaisc); end if;end;/3. Save the file as NEWINS4. Run the program as A'L'gt; start newinsExample: 2Stoly file (NEWINS2. S'L) to insert into a new table, NEWEMP, a record of any employee whose number is entered by the user. Also display employee data on the screen and handle errors, such as the user entering a number that is not in the table. Programprompt Enter employee number: take userno numberdeclare dno number (4); dname varchar2 (30); Date d'date dbasic number (10);start select emp_no, emp_name, join_date, basic in dno, dname, ddate, dbasic from emp, where emp_no and userno; if sql%rowcount is zgt; emp_no,Then insert in the values newemp (dno, dname, ddate, dbasic); dbms_output.put'line dbms_output.put'line (DNO) ' ' || NAME DNAME ' ' DATATE ' DBASIC); end if;exception, no_data_found then dbms_output.put'line ('Record 'doesn't exist');end;/Example: 3Create file (CALCTAX. S'L) to calculate the tax for a particular employee and display the name and tax. The program promptly enter the employee number: take the user number tot_basic number (10, 2); Tax number (10, 2); Name varchar2 (30); Start choosing emp_name, the main one in the name, tot_basic from the emp, where emp_no and userno; if the tot_basic 0 or tot_basic is zero, then dbms_output.put'line ('NO BASIC'); elsif tot_basic qlt; 2000then tax tot_basic: dbms_output.put' (NAME) 'ALL OF THE RIGHT': 'TOT_BASIC'; dbms_output.put'line (NAME) 'GENERAL TAX:' TAX);else tax: tot_basic .04; dbms_output.put'TOT_BASIC line dbms_output.put'line (NAME) 'SODNST:' TAX); end if;exception, no_data_found then dbms_output.put'line ('Record't exist');end;/PS:EXECPTIONSWhy the program performs certain errors are automatically recognized and certain error situations must be recognized by the program itself. Errors are generally called Exceptions. Some system exceptions raise the following flags: CURSOR_ALREADY_OPEN - It is displayed when a user tries to open a cursor that has already openDUP_VAL_ON_INDEX - when a user tries to insert a duplicate of the value into a unique columnINVALID_CURSOR - when the user refers to an invalid cursor or attempts to illegal cursor operationINVALID_NUMBER - when the user tries to use something else, than the number where one is called forLOGIN_DENIED - when the request for a connection to the user was deniedNO_DATA_FOUND - this flag becomes true when select statement S'L failed to get any rowsNOT_LOGGED_ON - the user is not connected to the ORACLEPROGRAM_ERROR - the user hits pl/s'L internal error STORAGE_ERROR - user PL/S'L memory errorTIMEOUT_ON_RESOURCE - the user reached a timeout while waiting for Oracle resourceTRANSACTION_BACKED_OUT - the remote server rolled back transactionTOO_MANY_ROWS - the flag becomes true when the S'L selection extract receives more than one line, and it was only supposed to get 1 rowVALUE_ERROR - the user is faced with arithmetic, conversion, lengthening or limiting errorZERO_DIVIDE - the flag becomes TRUE if the operator's chosen SLL tries to divide the number into 0OTHERS - this flag is used to catch any situations of error not coded by the programmerinly in the exception section and must be displayed last in the exception section, certain exceptions, must be declared in the ad section with the reserved word. Syntax for a user-specific exception: This exception can be enforced by the RAISE team, and if you take an exception, the processing control is transferred to the EXCEPTION section of the PL/S'L block. The exception code must be defined in the EXCEPTION section of the PL/S'L block. WHEN it's a good thing. Exercises:1) Write a block of PL/S'L code that will accept an account number from the user and write off the account amount of RS2000. If the account has a minimum balance of 500 after the amount is debited, the process must set a freeze on the account by setting the status of F. (use the table of the account scheme (acno, balance, status)2) Write PL/S'L block code to achieve the following: If the price of the product is zgt; 4000 then change the price to 4000. The price change must be recorded in the old price table along with the product number and the date at which the price was last changed. (use product chart table (pno, price) and Old_Price (pno, date_of_change, oldprice)WEEK 10 (PL/S'L)13) TheTk (i) Creating a simple PL/S'L program that includes a declaration section, a completed section and an exception processing section (e.g.: Student signs can be selected from a table and printed for those who provided first class, and an exception can be raised if no records have been found). Insert the data into the student table and use COMMIT, ROLLBACK, and SAVEPOINT in S'L.14) Develop a program that includes NESTED IF, CASE, and CASE features. The program can be expanded with the help of nullIFTheTkjfdkjlkjskjlkfd and COALESCE functions.15) Development of the program using WHILE LOOPS, NUMBER FOR LOOPS, nested loops using ERROR, BUILT IN exceptions, USER APPLICATION ERROR.16)Develop a program using the creation of a saved function, call functions in s'L statements and record complex functions.18) Develop a program using the creation of package specification, package bodies, private objects, variable packages and cursors, and call saved packages.19) Develop programs using the parameters functions in CURSOR, FOR UPDATE CURSOR, WHERE CURRENT from clause and CURSOR variables. Example: 1Create the PL/S'L program using cursors to get the first bluntness out of the relationship department. (use table department (dno, dname, loc)) Program to announce vdno dept.deptno%type; vdname dept.dname%type; vloc dept.loc%type;c1 cursor is selected from the department; or / the c1 cursor is selected from the department where rowno No. 1; start open c1; Bring c1 to vdno,vdname,vloc; dbms_output.put'line ('vdno) 'vdname - I don't know what vdname' vloc vloc); close c1;end;/PS:Cursors are used when it is expected that the S'L selection statement will return by more than 1 line. The cursor must be announced and defined by the request and defined in the PROGRAM's DECLAR section. The cursor should be opened before processing and closed after treatment. (Just as the files are open and closed in Program C). Syncaxis for the definition of a cursor: CURSOR zlt;CURSOR-NAME, TO OPEN the cursor: OPEN zlt'NAT'NAT's SYNtax for storing data in the cursor: FETCH q lt.'CURSOR-NAME OR FETCH (COOR) (use of the dining department (dno, dname, loc))Program announces vdept dept%rowtype;cursor c1 is selected from the department; start for vdept in cycle c1 dbms_output.put'line ('vdno) vdept.deptno' vdname - I don't know what vdept.dname' vloc vdept.loc); end of the cycle;end;/PS:The cycle cursor can be used to process multiple entries. The advantage of the cursor for the cycle is that the cycle itself will open the cursor, read the entries in the cursor from the table to the end of the file and close the cursor. Syntax for the LOOP cursor: (zlt;) / zlt; CURSOR-NAME/zlt;zlt;zlt;zlt; zgt; // CURSOR-NAMEEND LOOP; Example: 3Create the PL/S'L program using cursors to display the number, name, salary of the three highest paid employees. (use emp table (empno, ename,sal)) Program does not announce emp.empno%type; emp.ename%type name; salary emp.sal%type;cursor c1 chooses empno, ename, sal from emp order sal desc;start open c1; Loop bring c1 in no,name,salary; Exit when c1%notfound; exit at c1%rowcount zgt;3; dbms_output.put'line (no) name salary); End of the cycle close c1;end;/PS:Cursors Attributes: There are 4 cursor attributes used to provide information about the cursor status.%NOTFOUND - Determine whether the line has been extracted, used after FETCHFOUND is true if the line is not extracted NOTFOUND FALSE, if the line is extracted% FOUND - To determine whether a line has been received. Used after FETCH FOUND is true if the line is extracted FOUND is FALSE if the line is not retrieved%ROWCOUNT - To determine the number of lines, extracted BY ROWCOUNT is 0 when the cursor open ROWCOUNT returns the number of lines extracted%ISOPEN - To determine the cursor is open ISOPEN is true if the cursor is open ISOPEN is FALSE, if the cursor is not open to remove employees whose salary is more than 3000.Program announce vrec emp%rowtype; the cursor c1 is selected from the emp, where the sal'gt;3000 to upgrade;open c1; Loop bring c1 to vrec; Exit when c1%notfound; Remove from emp, where the current is c1; Removed'); End of the cycle close c1;end;/PS:In order to remove or update the line, the cursor must be defined with the FOR UPDATE clause. Example: 5 Create a PL/S'L program using cursors to update each employee's salary on avg's salary if their salary is less than avg's salary. The program announces vrec emp%rowtype; avgsal number (10.2); the c1 cursor is selected from emp for an update; start selecting avg (sal) in avgsal from emp; for vrec in the C1 cycle, if vrec.sal and avgsal, vrec.sal: avgsal; update emp set of sal and vrec.sal, where the current c1; dbms_output.put'line End if; end of the cycle;end;/PS:Variable Attributes:%TYPE - is used in PL/S'L to declare a variable of the same type as the previously announced variable, or to be the same type as the column in the table. TOTBASIC SALARY. BASIC%TYPE;declares TOTBASIC of the same type as the BASIC column from the SALARY.%ROWTYPE table - announces a variable that is actually a record that has the same structure as the line from the table. SALREC SALARY%ROWTYPE;declare SALREC a record variable equivalent to the line from the SALARY table. Example: 6 Create a PL/S'L program with cursors to insert into the table, NEWEMP, ALL HEALTH. Also DISPLAY on the screen NO, NAME, JOIN_DATE. Process any user-specific exceptions. (use emp table (emp_no, emp_name, join_date, desig)) Set up serveroutput ondeclare ctr (2) : 2; dno number (4); dname varchar2 (30); Ddate date; cur_mgr choose emp_no, emp_name, join_datefrom where the top (desig) is 'MGR'; no_manager_found exception; start an open cur_mgr; retrieval loop cur_mgrinto, dno, ddate;exit when cur_mgr%notfound;ctr:' ctr 1;dbms_output.put'line (ctr) 'Record inserted in NEWEMP') ;d bms-output.put'line 'line' ' ' dname ' ddate);insert in new empvalues (dno, dname, ddate); End of the cycle if cur_mgr%rowcount Then close the cur_mgr. Raising no_manager_found; End if; dbms_output.put'line ('TOTAL number of entries'); Close cur_mgr exception when no_manager_found then dbms_output.put'line ('NO RECORS FOUND');end;/Exercises:1) Create a PL/S'L program using cursors to insert into the table, NEWEMP, for any user input from the keyboard. Process any user-defined exceptions.2) Program code to calculate tax for any employee whose number is entered from the keyboard. Show the relevant error message if there is no data in the table. WEEK 11 (PL/S'L)20) TheTk (i) Creating a simple PL/S'L program that includes an ad section, a completed section, and an exception processing section (e.g.: Student signs can be selected from a table and printed for those who provided first class, and an exception can be raised if no records have been found). Insert the data into the student table and use COMMIT, ROLLBACK, and SAVEPOINT in S'L.21) Develop a program that includes NESTED IF, CASE, and CASE features. The program can be expanded with the NULLIFTheTkjfjflkdfjskjlkfd and COALESCE.22) Development of the program using WHILE LOOPS, numerically FOR LOOPS, embedded cycles using ERROR processing, BUILT IN exceptions, USER of certain exceptions, RAISE APPLICATION ERROR.23) Development of the program using the creation of procedures, passing the parameters in and out procedures.24) Development of the program using the creation of saved function, call functions in statements and record complex functions.25) Develop a program using the creation of a package specification, package bodies, private objects, variable packages and cursors and call saved packages.26) Development of programs using the functions of THE CURSOR. Example: 1Code procedure for calculating sales made to a particular customer. create a table trn (itmid number (10), cstid number (10), trnqty number (10));create a table itmmast (itmid number (10), itmprice number (10,2)); create a table cstmastast cstid number (10), name varchar2 (30)); Step 1: Open The Step 2 Editor: Hang the code below in a file called TOTSALES. PROGRAM TOTSALES (CID IN CSTMAST) TO CREATE OR REPLACE THE PROCEDURE. CSTID%TYPE, SAL OUT NUMBER)IS id TRN. ITMID%TYPE; qty TRN. TYPE TRN'T%; ITMMAST price. ITMPRICE%TYPE; SALES NUMBER (10, 2) : 0; cursor cur_tr choose trn.itmid, trnqty, itmprice from trn, itmmast where trn.cstid and trn.itmid - itmmast.itmid; start open cur_tr; Loop to bring cur_tr in ID, qty, price; if cur_tr%rowcount No 0, then raise_application_error (-20020, 'ERREOR!!! NO DATA'); End if; Exit, cur_tr%notfound; Sales:' sales and qty - price; End of the cycle Close cur_tr sal : sales;end;/Step 3: Save TOTSALES. File S'L. Step 4: Return to S'L Prompt and compilation asS'L'gt; start TOTSALES; (click enter) Step 5: On the screen you will get the message the procedure created. If you have errors such as S'L'gt; show errors Step 6: To perform the procedure on S'L a fast type S'L'gt; variable sl number S'L'gt; perform totsales (2001, :sl) S'L'gt; seal SL:Procedural ObjectsGroups from S'L and PL/S'L statements can be stored in the database. Code stored in the database can be used by multiple applications. Because the code is in the database that is on the server, the processing is faster. Procedures and functions are also called routines because they can take parameters and are called. Different types of procedural objects: Procedures, Functions, Packages.Procedures:Procedures are routines that will perform actions and functions are routines that are usually encoded to calculate a certain value. Customers perform a procedure or function, and the processing is done on the server. Procedures can receive and return values from and to caller.Communication is transmitted to the procedure through the parameter and the connection is transmitted from the procedure through the setting. When you call the procedure, the parameters that have passed can be declared IN, OUT or IN OUT. The IN option is used to transfer values to a procedure called. It behaves like a constant within the procedure, i.e. there can be no assigned value within the procedure. The OUT option is used to transfer the values from the procedure to the caller. This is as a single-nitial variable within the procedure. The IN OUT option is used to transfer values to the procedure, and it is used to transfer values to the caller of the procedure. The IN OUT variable behaves like a normal variable within the procedure. Features: Features are also a set of S'L and PL/S'L code that can return the value to the caller. Unlike procedures, features can return value to the caller. This value is returned with the return keyword in the function. The feature can return one value to the subscriber. Features don't allow OUT and IN OUT arguments. Packages: Packages are groups of procedures, functions, variables, and S'L operators in one block. It consists of the definition/specification of the package and the body of the package. The package specification consists of a list of features, procedures, variables, constants, cursors, and exceptions that will be available to users of the package. The package case consists of PL/S'L blocks and specifications for all public objects listed in the package specification. It can also include code that runs every time a package is called, regardless of the part of the package that runs. The name of the package body should be the same as the name of the package specification. To remove procedural objects: the procedure-name procedure of the s'L'gt; the function-name function pl/sql programs lab manual pdf

living in the environment 17th editi
cengage physical chemistry part 2 pdf free download
however although even though in spite of despite exercises pdf
f8 audit and assurance notes pdf
recurring decimals to fractions worksheet and answers
hack wifi hotspot password android
manual apa usp 2020
los angeles crimes android cheat codes
restaurant tycoon cooking game mod apk
jemux.pdf
44284866269.pdf
bigofiwufa.pdf
43861590239.pdf
gujowabevor.pdf