

Subject Code : 1CS2010307	Subject Title: PROCEDURE PROGRAMMING IN SQL
Pre-requisite :	Database Management System Concepts, SQL

Course Objective:

The objectives of the course are to:

- To be able to understand the concepts of PL/SQL
- To be able to understand various database objects stored procedure, function and Packages
- To be able to understand about various database triggers

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory		Practical		Total
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	--	3	6	60	40	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	Advance SQL Hierarchical Queries, Advanced Queries (Running sum and Decode, Top N queries) Indexes, Views, SEQUENCES	5	10
2	Introduction to PL/SQL Overview, PL/SQL, Advantages of PL/SQL, Generic PL/SQL Block PL/SQL Fundamentals: PL/SQL Variables and PL/SQL Data types, Variable attribute (%type, %rowtype) PL/SQL Control Structure (looping)	5	15
3	PL/SQL Cursor Management Implicit Cursor, Cursor Attributes, Writing Explicit Cursors (Cursor function, declaring, opening, fetching data from cursor), Cursor FOR loop. Advanced Explicit Cursors (Cursor with Parameter)	5	10
4	Error handling in PL/SQL & Security Exception Types, Predefined Exception, User Defined Exception, Raise_Application_Error. Locks, Types of locks, Application of Locks Application of Save Point, Commit and Rollback	6	15
5	PL/SQL Database Objects Anonymous PL/SQL and named Block. Developing Procedures, formal and actual parameters, IN, OUT and INOUT parameters. Creating Functions, Stored Functions, User Defined Functions. Creating Packages: - Component of Package, Developing Package, Package Specification, Package Body. Overloading procedures and Functions.	5	15

6	Triggers Types of triggers, Creating DML Triggers (Row trigger, Statement triggers, Before and after triggers, Using OLD and NEW qualifier), Drop Trigger and Applications of Trigger.	5	15
7	Collection in PL/SQL Object Types, Nested Tables, Variable Arrays, Associative Arrays/ Index by table Array.	5	20

Course Outcome:

At the end of this course, the student would be able

- Ability to understand the concept of PL/SQL
- Ability to build and manage database objects like stored procedure, Function and Package
- Ability to understand the usage of database triggers in application.

List of References

1. Ivan Byross, SQL, PL/SQL the programming Language of Oracle, BPB
2. P.S.Deshpande, SQL and PL/SQL for Oracle 11g”, Black Book, dreamtech publication.
3. Kevin Loney ,Oracle Database 11g :The Complete Reference, Oracle Press, McGrawHill
4. Ms. Anjali Jivani and Ms. Amisha Shingala , “ Practice book on SQL and PL/SQL with examples”, Nirav and Roopal Publications, Third Edition.

Indicative Practical List:

PART – I (PL/SQL Basic Programs)

1.	Write a PL/SQL block to print multiplication table.
2	Write a PL/SQL block to find factorial of given number.
3	Write a PL/SQL block which concatenates two string and print it in reverse order in UPPER case
4	Write a PL/SQL block to display a string and find number of vowels and words in a given string.
5	Write a PL/SQL block to find whether inputted year is Leap year or not.
6	Student database consists of studno, name, marks of 5 subjects. Find out the class (distinction, first, second and pass class) of particular student using IF statement. Also implement the same using CASE statement
7	Write a PL/SQL block to generate the Fibonacci series.
8	Write a PL/SQL block to find whether given number is prime or not.
9	Write a PL/SQL block to print nos from 1 to 500 in reverse order as 100 will be 001 254 will be 452
10	Write a PL/SQL block to find value of 'A' depending on following formula $A = B + C * D ^ 2 - E / 10$

PART - II (PL SQL based on Cursors)

Implicit cursor

EMP(empno, empnm, empadd, salary, date_birth, joindt, deptno)

DEPT(deptno, deptnm)

Note: Make table with appropriate constraints and insert proper data.

1.	Display all fields of EMP table using record type variable.
2.	Delete the record for a given employee.
3.	Update the salary of all employee by 10%.
4.	Display all fields of EMP table using Cursor for FOR loop.
5.	Display record of employee for particular department using Cursor Variable.

Explicit cursor

1.	Write a PL/SQL block (table above EMP-DEPT table) which takes as input Department name And displays all the employees of this department who has been working since last five years.
2.	<p>From the following schema , Write a PL/SQL for the following</p> <p>APPLICANT(AID, Aname, Addr, bdt) ENTRANCE_TEST(ETID, etname, max_score, cut_score) ETEST_CENTRE(ETCID, location, incharge, capacity) ETEST_DETAILS(AID, ETID, ETCID, etest_dt, score)</p> <ol style="list-style-type: none"> Accept the Entrace test ID and test centre ID from the user and display the details about all those applicants who has score more than average for that test. Also display the details about the dates on which the test centre has full attendance. Display details about the applicants and details of all the test the applicant has appeared for with his scores.

3. Consider the EMPLOYEE schema as :
 EMPMAST(empno, name, pfno, empbasic, deptno, designation) HOLIDAY(year, month, holidays, weekoff)
 EMPTRAN(empno, month, year, presence, loan(amt))
 BANK(accno, empno, bank name, bank branch)

Rules : HRA = 15% of basic
 DA = 50% of basic
 Medical = 100
 PF = 8.33% of basic
 Salary is given for (attendance + holidays + weekoff) days.

Print salary slip in foll. Format.

SALARY SLIP of Month : Year

Accno : Name of EMP Designation Dept

Bank name : _____ Branch : _____ Dt of Join _____

Presence : Absent days : Sal days :

EARNINGS : DEDUCTIONS BASIC :

LOAN :

DA : PF :

HRA : Prof tax : 100

EDICAL : 100

TOTAL earnings: _____ TOTAL Deductions: _____

NET PAY :

4. From the following schema, Write a PL/SQL for the following
 Item (Item_code, item_name, unit_price)
 Bill (bill_number, bill_date)
 Bill_item (bill_number, item_code, number_units_purchased)

(a) For a given bill number, print the bill in the following format.

Bill no.: **Bill Date:**

Sr. No.	Item code	Item name	Units purchased	unit price	total price
###	xxxxx	xxxxxxxxxxxxxxxxxxx	#####	####.##	#####.##

Bill Amount: #####.##

(b) Add one more attribute named bill_amount in the Bill table. Update the Bill table to store bill amount.

5. Write a PL/SQL block, which will give a raise in salary to the employees as per the following:
 USE above EMP table .
 If salary + comm. < 5000 increase by 10% of salary
 If salary + comm. >= 5000 increase by 500 + 12% of salary above 7000
 Note : SELECT ... FOR UPDATE BEING USED IN CURSOR and CURSOR – FOR LOOP USED

6.	<p>PART_MASTER (P#, PNAME, PRICE, TOT_QTY, REORDER_LEVEL) PART_TRANS (O#, P#, QTY_REQ) Where O# is the order no. TEMP (P#, QTY)</p> <p>Write a PL/SQL block which will read records one by one from the PART_TRANS table and make necessary updates in the PART_MASTER table. However if after the update the TOT_QTY is going below the REORDER_LEVEL then the PART_MASTER is not to be updated and the necessary details are stored in the TEMP table.</p>																						
7	<p>ISSUE (ROLLNO, BOOKNO, ISSUE_DATE, RETURN_DATE)</p> <p>For the above table, check which students have to pay fine and the amount of fine to be paid. A student can keep the book for 15 days. If the number of days has exceeded 15 then, the fine is calculated as follows: Up to 7 days : 50 paise per day 8 – 15 days : Re 1 per day (to be counted from the 8th day onwards) More than 15 : Re 1.5 per day (to be counted from the 16th day onwards)</p> <p>After calculating the fine store the required information in the FINE table so that a report can later be printed out. FINE (ROLLNO, BOOKNO, ISSUE_DATE, RETURN_DATE, TOT_DAYS, FINE)</p>																						
8	<p>Conside the patient schema as :</p> <p>Patient (pcode, name, age, gender , address) Appoint(pcode, app_dt, time , doctor_id) Doctor(doctor_id, name, specialization)</p> <p>Prepare a report from above table taking date, shift, , doctor_id as parameters. Shift is Morning if time are from 9.00 Am to 1.00 pm and Shift is Evening if time are from 5.00 to 9.30 pm.</p> <p>APPOINTMENT SCHEDULE</p> <table data-bbox="199 1433 957 1702"> <tr> <td>Date :</td> <td>Doctor name :</td> </tr> <tr> <td>Shift :</td> <td>Specialization :</td> </tr> <tr> <td>Time</td> <td>Patient name</td> <td>Age</td> </tr> <tr> <td>9.00</td> <td></td> <td></td> </tr> <tr> <td>9.10</td> <td></td> <td></td> </tr> <tr> <td>9.20</td> <td></td> <td></td> </tr> <tr> <td>.....</td> <td></td> <td></td> </tr> <tr> <td colspan="3">Total no. of patients : _____</td> </tr> </table>	Date :	Doctor name :	Shift :	Specialization :	Time	Patient name	Age	9.00			9.10			9.20					Total no. of patients : _____		
Date :	Doctor name :																						
Shift :	Specialization :																						
Time	Patient name	Age																					
9.00																							
9.10																							
9.20																							
.....																							
Total no. of patients : _____																							

9.	<p>Consider the Bank schema as</p> <p>ACCOUNT (AC_NO, NAME, AC_TYPE, BALANCE_AMT, BALANCE_DATE) TRANSACTION (AC_NO, DATE, TR_TYPE, AMOUNT, PREV_BALANCE, REMARK)</p> <p>Note : 1. AC_type may be S for saving or C for current, 2. TR_type may be D for deposit or W for withdrawal.</p> <p>a. Write a procedure to print the Bank Transaction details by passing from and to dates.</p> <p style="text-align: center;">BANK TRANSACTION STATEMENT DATE:</p> <p>---- ACCOUNT NO :</p> <p>FROM DATE: TO DATE:</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">DATE</th> <th style="text-align: left;">DEPOSIT</th> <th style="text-align: left;">WITHDRAWAL</th> <th style="text-align: left;">BALANCE</th> </tr> </thead> <tbody> <tr> <td>DD/MM/YY</td> <td>#####.##</td> <td>-----</td> <td>#####.##</td> </tr> <tr> <td>DD/MM/YY</td> <td>-----</td> <td>#####.##</td> <td>#####.##-----</td> </tr> </tbody> </table> <p>----- ACCOUNT NO :</p> <p>FROM DATE: TO DATE:</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">DATE</th> <th style="text-align: left;">DEPOSIT</th> <th style="text-align: left;">WITHDRAWAL</th> <th style="text-align: left;">BALANCE</th> </tr> </thead> <tbody> <tr> <td>DD/MM/YY</td> <td>#####.##</td> <td>-----</td> <td>#####.##</td> </tr> <tr> <td>DD/MM/YY</td> <td>-----</td> <td>#####.##</td> <td>#####.##</td> </tr> </tbody> </table>	DATE	DEPOSIT	WITHDRAWAL	BALANCE	DD/MM/YY	#####.##	-----	#####.##	DD/MM/YY	-----	#####.##	#####.##-----	DATE	DEPOSIT	WITHDRAWAL	BALANCE	DD/MM/YY	#####.##	-----	#####.##	DD/MM/YY	-----	#####.##	#####.##
DATE	DEPOSIT	WITHDRAWAL	BALANCE																						
DD/MM/YY	#####.##	-----	#####.##																						
DD/MM/YY	-----	#####.##	#####.##-----																						
DATE	DEPOSIT	WITHDRAWAL	BALANCE																						
DD/MM/YY	#####.##	-----	#####.##																						
DD/MM/YY	-----	#####.##	#####.##																						

Part-III (Procedure, Functions and Packages)

1.	Write a function whether given number is odd or even.
2.	Write a function that returns total number of incomplete jobs, using table JOB(jobid, type_of_job, status)
3.	Write a function which displays the number of items whose weight fall between a given range for a particular color using table ITEM(itemno, name, color, weight)
4.	Write a function which returns number of workers with same skills as that of the worker who pass as a parameter using table WORKER(worker_id, name, wage_per_hour, specialization, manager_id).

5.	<p>Consider the following tables :</p> <p>Institute(instid,name) Courses(courseid, name) Offers(instid, courseid, max_capacity) Students(sid,name, city, pincode, phoneno) Registers(sid, instid, reg_date)</p> <ol style="list-style-type: none"> 1. Create a sequence for student id 2. Write a procedure for full task : When student comes for registration, check if course is supported by the given institute or not and the no. of already registered students not exceeding the max capacity for that course and that institute. 3. Create a function to return count of all colleges where no. of students registered in technical board. This function is called in procedure which displays the output in Proper format.
6.	<p>WORKER(workerid, name, wage_per_hour, specialized_in, manager_id)</p> <p>Write a procedure to display top five highest paid workers who are specialized in "PAINTING"</p>
7.	<p>Consider the following tables or take above patient –doctor-table.:</p> <p>Patient (pcode, name, age, gender , address) Appoint(pcode, app_dt, time , doctor_id) Doctor(doctor_id, name, specialization)</p> <p>Write a procedure which displays the details of doctor who has max. appointment. A local function returns the doctor_id having max. appointment.</p>
8.	<p>CUST(accno, name, add, city , phone, email) ACCOUNT(accno, amt)</p> <p>Write a package for banking purpose which perform foll. Task :</p> <ol style="list-style-type: none"> 1. Create a new account 2. Deposit funds 3. Check account balance 4. Close the account

PART- IV: (Triggers)

1.	<p>Given EMP(empno, nm, hra ,da, tot)</p> <ol style="list-style-type: none"> a. Write a trigger for insert or update to check that HRA should be more than 3000 and DA should be in between 40 and 50. b. Write a trigger to update temp table TEMP(tnm, tdt) whenever any data is inserted / deleted / modified in EMP table. c. Write a trigger which does not allow records of EMP to be deleted on 30th of any month.
----	---

2.	<p>Given a STUD(rollno, name, birthdate) RESULT(rollno, marks)</p> <ol style="list-style-type: none"> Write a trigger which accept student registration if her/his age is between 18 And 30. Write a trigger which does not allow student to be enrolled on weekend i.e. Saturday or Sunday or outside office hours (9am to 5 pm is office hours) Write a trigger which converts all lowercase student name into upper case Before inserting in Student table. Write a trigger which deletes every related record from RESULT table if student record is deleted.
3.	<p>Given a table</p> <p>ITEM(itemno, description, unit_measure, qty_hand, type, weight) DEPT(deptno, name, mgr) ISSUE(issueno, issue_dt, deptno)\ ISSUE_ITEM(issueno, itemno, qty)</p> <ol style="list-style-type: none"> Write a trigger which allows only those items to be inserted which starts from 'I' Or 'I'. Write a trigger which check whether quantity ordered is less than quantity on hand Before allowing that record to be inserted in ISSUE_ITEM table. Write a trigger on table ISSUE_ITEM which reduces the quantity on hand in ITEM table as per the value in QTY field after valid row is inserted. Write a trigger which checks that if item being inserted in ISSUE_ITEM table is 'I100' then quantity should in units of 100, if item is 'I200' then quantity should be in units of 500.
4.	<p>DEMP(EMP_ID, EMPNM) DEPT(DEPT_ID,DNAME, DLIMIT) EMP_DEPT(EMP_ID, DEPT_ID)</p> <ol style="list-style-type: none"> WHILE TRANSFERING EMPLOYEE FROM ONE DEPT TO ANOTHER, CHECK THAT NUMBER OF EMPLOYEE IN DEPT DOES NOT EXCEED THE LIMIT. AN EMPLOYEE CANNOT WORK IN MORE THAN TWO DEPARTMENT C. NO ALLOCATION OR TRANSFER IS DONE ON SUNDAY OR OUTSIDE OFFICE HOURS. WRITE TRIGGER TO CHECK WHETHER EMP NAMES ARE INSERTED IN UPPERCASE, IF USER GIVES IN LOWER CASE INSERT INTO UPPERCASE. UPDATE THE DEPT TABLE FOR LIMIT AND UPDATE THE LIMIT FOR DEPTNO = 2 AND ALSO CHECK THAT LIMIT UPDATED SHOULD BE GREATER THAN ORIGINAL LIMIT. WRITE A INSTEAD OF TRIGGER THAT DELETE ROWS IN EMP_DEPT TABLE THAT MATHCHES THE ROW ENTERED.

5.	<p>GIVEN THE TABLE, IMPLEMENT 5 TRIGGERS.</p> <p>EMPDET(EMPNO , EMPNM , HRA , DA , TOT) TEMPDET(TNAME , DATE1)</p> <p>A. WRITE A TRIGGER FOR INSERT OR UPDATE TO CHECK THAT HRA SHOULD NOT BE MORE THAN 3000 DA SHOULD BE IN BETWEEN 40-50</p> <p>B. WRITE A TRIGGER TO UPDATE TEMPDET TABLE WHENEVER DATA IN EMP TABLE IS INSERTED, DELETED OR MODIFIED.</p> <p>C. WRITE A TRIGGER TO INSERT THE EMPLOYEE NUMBER AUTOGENERATEDE FORM SEQUENCE BEFORE INSERT ON EMPDET TABLE.</p> <p>D. WRITE A TRIGGER FOR INSERT OR UPDATE TO CHECK THAT EMP NAME STARTS WITH 'M%' FOR MR AND MRS.</p> <p>E. WRITE A TRIGGER WHICH DOESN'T ALLOW RECORDS TO BE DELETED ON 30TH OF ANY MONTH.</p>
----	---