PL/SQL EXERCISE

GENERAL PL/SQL BLOCKS:
1) WAP to input two numbers and find out what is the output of all arithmetic operations. (Addition, Subtraction, Multiplication, Division etc.)
2) WAP to input rollno and three subject marks. Find out total, percentage, result and grade for the student from the entered data.
3) WAP to print first 10 odd numbers using for loop.
4) WAP to print prime numbers upto 10 using while loop.
5) WAP to input three nos and find out maximum and minimum from it.
6) WAP to input empno from keybord. Check whether inputted empno exist in emp table or not. If not give error message otherwise display name and salary of that employee.
7) WAP to insert record in Customer table.
Customer(cust_id,cust_name,address,city);

CURSORS:
1) Create a cursor for the emp table. Produce the output in following format:
{empname} employee working in department {deptno} earns Rs. {salary}.
EMP(empno, empname, salary, deptno);
2) Create a cursor for updating the salary of emp working in deptno 10 by 20%.
If any rows are affected than display the no of rows affected. Use implicit cursor.
3) Create a cursor for updating the salary of emp working in deptno 10 by 20%.
Use explicit cursor.
EMP(empno, empname, salary, deptno);
4) WAP that will display the name, department and salary of the first 10 employees getting the highest salary.
5) WAP using *parameterized cursor* to display all the information of employee living in specified city. Ask the city from user.
6) WAP which display the sum of salary department wise.
7) Create a cursor to generate defferent two tables from one master table.
Student(Rno, Name, Std, B_date, Sex);
Girl_Table(Rno, Name, Std, B_date);
Boy_Table(Rno, Name, Std, B_date);
First fetch the row from Student table. If sex is ‘M’ then insert that row in Boy_Table and if ‘F’ then insert that row in Girl_Table.
In both table Rollno entry must be in Sequence(Using create sequence command).

FUNCTIONS :
1) WAF which accepts the name from user and returns the length of that name.
2) WAF which accepts one number and return TRUE if no is prime and return FALSE if no is not prime.
3) Write a function which accept the department no and returns maximum salary of that department. Handle the error if deptno does not exist or select statement return more than one row.
EMP(Empno, deptno, salary).
4) Write a function to display whether the inputed employee no is exists or not.
5) WAF which accepts one no and returns that no+100. Use INOUT mode.
6) WAF which accepts the empno. If salary<10000 than give raise by 30%.
If salary<20000 and salary>=10000 than give raise by 20%. If salary>20000 than give raise by 10%. Handle the error if any.

7) WAF which accepts the empno and returns the experience in years. Handle the error if empno does not exist.
EMP(Empno, Empname, DOJ);

PROCEDURES:

1) Write a procedure which accepts the empno and returns the associated empname. If empno does not exist than give proper error message.
EMP(Empno, Empname).
2) WAP which accepts the student rollno and returns the name,city and marks of all the subjects of that student.
STUDENT(Stud_ID,Stud_name,m1,m2,m3).
3) WAP which accepts the name from the user. Return UPPER if name is in uppercase, LOWER if name is in lowercase, MIXCASE if name is entered using both the case.
4) WAP which accepts the student rollno and returns the highest percent and name of that student to the calling block.
STUDENT(Stud_ID,Stud_name,percent);
5) WAP which accepts the date of joining for specific employee and returns the years of experience along with its name. Accept the Employee no from user.
EMP(empno, empname, DOJ);
6) WAP which accepts the student rollno and returns the result (in the form of class: first class, second class, third class or fail).
STUDENT(Stud_ID,Stud_name,m1,m2,m3).

TRIGGERS:

1) Write a Trigger that stores the old data table of student table in student_backup while updating the student table.
Student_backup (Stud_ID, Stud_name, Address, Contact_no, Branch, Operation_date)
Student (Stud_ID, Stud_name, Address, Contact_no, Branch)
2) Write a trigger, that ensures the empno of emp table is in a format 'E00001' (empno must start with 'E' and must be 6 characters long). If not than complete empno with this format before inserting into the employee table.
3) Write a trigger which checks the age of employee while inserting the record in emp table. If it is negative than generate the error and display proper message.
4) Write a trigger which converts the employee name in upper case if it is inserted in any other case. Change should be done before the insertion only.
5) WAT that stores the data of emp table in emp_backup table for every delete operation and store the old data for every update operation.
EMP(Empno, Empname, salary);
Emp_Backup(Empno,Empname,Date_of_operation,Type_of_operation(i.e.update or delete));
6) WAT which display the message 'Updating','Deleting' or 'Inserting' when Update, Delete or Insert operation is performed on the emp table respectively.
7) WAT which generate an error if any user try to delete from product_master table on weekends (i.e. Saturday and Sunday).
8) WAT which inserts the value of client_no in the client_master table whenever user tries to insert data in the emp table. Generate primary key using sequence and enter the client_no using that sequence.
Client_Master(client_no,client_name,address,city);
9) WAT to calculate the Income Tax amount and insert it in emp table..
EMP(emp_no,emp_name, emp_income, income_tax);

If emp_income <100000 and >=50000 then incometax = 10%
If emp_income <200000 and >=100000 then incometax = 15%
If emp_income <300000 and >=200000 then incometax = 20%

PACKAGE:

1) Write a package which consist of one function f_itemchk and two procedures proc_insert and proc_update.
Function will check the existence of empno. If it exists than update the record for that employee using procedure proc_update. Otherwise insert the record using procedure proc_insert.
2) Write a package which consist of two functions named SUMMATION().
In one function accept two number arguments and return the addition of them.
In the second function accept two character type arguments and return concatenated string. (Use of Function Overloading.)
3) Create a package which consist of three procedures.
First procedure check for the number is > 0 or not.
Second procedure accepts one date argument and check that is < sysdate or not.
Third procedure accepts a name and check whether it is in uppercase or not.