1	Create the following table using SQL for an Entrance Exam. Keep a constraint check so that the marks should be within the range 1-100 Student (Sroll,Sname,Sdept,SMarks,Edob) Using PL/SQL implement the following: a) Count the number of Students. b) Find the maximum and minimum marks obtained. c) List the details of the student having the maximum age. d) Write a procedure/function to input the Sroll of a Student and return the Student details. e) Write the code for the Exceptions TOO_MANY_ROWS and NO_DATA_FOUND.
2	Create the following tables and insert suitable records in them maintaining integrity constraints: *Suppliers* (sid, sname, address) *Parts* (pid, pname,color) *Catalog* (sid, pid, cost) (a) Insert at least 6 records in each table. Make proper validation such that every pid must begin with 'PI', the price range of all the items is from Rs.500/- to Rs.25000/-. (b) Write a PL/SQL code using Cursor to list the 3rd maximum price from the Catalog table. (c) Write a function to which will take the pid as an input (from the Parts table) and list the details of the product with that given pid. (d) Implement one built-in and one user-defined Exception.
3	Create the following tables through appropriate SQL commands: Sailors (Sid, Sname, Rating. Age) Boats(Bid,Bname,Colour) Reserves (Sid,Bid,Day) Define the integrity constraints, if required. Populate tables with appropriate data at least 5 records in each table. [7] (a) Answer the following queries: (i) Find the names of sailors who have reserved both a black and a white boat. (ii) Find sailors whose rating is better than "Sherpa". [4+4] (b) Write the PL/SQL code implementing Cursor to give the details of the Sailor having the Minimum age. [7] (c) Implement the Exceptions NO_DATA_FOUND and OTHERS.
4	Create the following Tables maintaining proper Integrity Constraints Student(s_roll,sname,s_address,c-id) Course(c-id,cname,cfees,cstartdate) (a) Insert at least 5 records in each table. Keep proper validation so that the value of course fees (cfees) lies between 2000-15000 and the c-id starts with the letters 'CI'. (b) Write a PL/SQL code using cursor to increase the course fees of the course 'Animation' by 10% and other courses by 5%. Ensure that the updation is properly done. (c) Write a procedure/function to input the c-id of a Course and return the Course details. (d) Use at least one Inbuilt and one User defined Exception.
5	Create the following table using SQL: Employee (E_id,Ename,Edept,Esalary,Edob)

	Using PL/SQL implement the following: (a) Count the number of employees. (8) Find the maximum and minimum salary. (c) List the details of the employee having the maximum age. (d) Write a procedure/function to input the E-id of an Employee and return the employee details. (e) Write the code for the Exceptions TOO_MANY_ROWS and NO_DATA_FOUND
6	Create the following tables with proper integrity constraints: Employee (emp_id, ename, esal, d_id) Department (d_id,dname,d_location) Every employee id must begin with 'EMP', the salary range of an employee should be between 10000 and 100000, and the departments are located in one of the following locations: Kolkata, Mumbai, Delhi and Chennai. (i) Write a PL/SQL code using Cursor to increase the salary of all the employees of Chennai by 25% and decrease the salaries of employees residing in Kolkata by 10% (ii) Write a procedure/function to input the id (emp_id) of an employee and return the corresponding employee details. (iii) Use proper Exception Handling in case of invalid data input.
7	Create a table named Employees to store the detailed information of the employees in an organization. The table should contain the following fields: (Eid,Ename,EDept,date_of_join,ESal) Insert at least 10 records. (a) Write a PL/SQL code to update the salary of all Employees by 20%. (b) Write a PL/SQL code to separately store the records of the first 5 employees having the maximum salary in a separate table named Max_sal. (c) Write a PL/SQL code to separately store the records of the last 5 employees having the maximum salary in a separate table named Min_sal.
8	Create the Following tables with proper integrity constraints: Item (Item_id, name, price, cust_id) Customer (cust_id, name, addr) (a) Make proper validation such that every item id must begin with 'ITM', the price range of all the items is from Rs 50/- to Rs 5000/0. Now add a field 'quantity' to the Item table. (b) Write a PL/SQL code to input the id of an item and to check whether the number of this item present is less than 10. If so, add 100% more items, otherwise add 50% more items. Ensure that the updation is properly done. (c) Write a block to throw an exception if the given item id provided is invalid. (d) Write a function in PL/SQL to input the id of a customer and find the Customer details
9	Employee (e_id, emp_name, street, city) Works (e_id, company_name, salary) Company(company_name, city) Manages(e_id, manager_name) Create and populate the database through appropriate SQL commands maintaining integrity constraints where required. (a) Write the SQL queries for the following:

	(i) Find the names and cities of residence of all employees who work for the company ITC (ii) Find all employees who live in the same city as their managers. (b) Write a procedure/function to input the e_id and return the corresponding employee details.
10	Create the following tables with *proper integrity constraints*: Student (s_id, s_name, s_fees, dept_id) Department (dept_id,dept_name,dept_floor) Every student id must begin with 'STD', the fees range of a student should be between 2000 and 4000, and the departments are located in one of the following floors : First, Second, Third and fourth. Write a PL/SQL code using Cursor to decrease the fees of all the student of first floor and third floor by 20%, second and fourth floors by 10 %. Write a cursor/function to input the id of a student and return the corresponding student details. Use proper Exception Handling in case of invalid data s_id input.
11	Create a table named *Student* to store the detailed information of the Students in a School. The table should contain the following fields: (S_id,Sname,class,date_of_exam, Marks) Insert at least 10 records. (a) Write a PL/SQL code to separately store the records of the first 3 students having the maximum marks in a separate table named Stud_Rank. (b) Write a cursor/procedure/function in PL/SQL to input the id of a student and find th details of that student.
12	Consider the following database: Student (sroll, sname, dept,faculty_id) Faculty (faculty_id, fname,No_of_classes) Create and populate the database through appropriate SQL commands. Use integrity constraints, if required. Write a PL/SQL code to do the following: (a) Find the total number of students using a Cursor. (b) Write a function to input the faculty_id and return the corresponding faculty details. (c) Using Cursor find the Faculty having the maximum number of classes.
13	Suppose there are two tables called Customer and Customer_Discount. The first table stores the details of all the customers of a retail shop. The shop owner has decided to offer some discounts to the customers. The discount is dependant on the total amount purchased by a customer. If a customer has purchased items of more than Rs.5000/- then the discount is 20%, if a customer has purchased items worth Rs.1000/- to Rs.5000/- then the discount is 12% otherwise a flat discount of 5% is offered. The customer name, total amount purchased and the discount amount are to be stored in the second table. Write a PL/SQL code to perform the above operations. At the time of the entering a customer id which is not present in the table, the program will generate proper exception and will display a message.

14	Create a product table to store some relevant fields about different products sold by a shop. Write a PL/SQL code to fetch the first n products according to their price and store their details in a separate table called Expensive_Product.
15	Create a table to store the detailed information of the students of a college. Write a PL/SQL code to separately store odd-numbered records and even-numbered records to two different tables.