1	Consider the following Table in a banking database: Account(**Accno**, Accname, Type, Balance) Write a PL/SQL program for withdraw and deposit of amount on a given account with following features: The program should automatically detect conditions for insufficient fund and raise error. For successful transaction, the program must insert records of the transactions in a new table called Transaction with following definition: **Transaction**(Tr_id, Accno, Tr_type, Amt, date of Tr)	
2	Consider following two relations Book_stock(**Book_id**, Title, No of Copies) Book_Issue(**card_no**, cholder_name, **book_id**, issue_date, due_date) Book_return(**card_no**, cholder_name, **book_id**, return_date, issue_date) Write a PL/SQL program for issuing/return of books with following conditions: A member is not allowed to issue more than 5 books. A member is not allowed to issue more than one copy of same bookid during a period. After issuing/return of book the no of copies must be automatically updated. If the return date of the book is later than the due date; insert a record in a new table called Fine (card_no, amt). Assume a fixed late fine amt.	
3	Consider a bank database with only one relation transaction (transno, acctno, date, amount) The amount attribute value is positive for deposits and negative for withdrawals (a) Define an SQL view TP containing the information (accno, T1.date, T2.amount) for every pair of transactions T1, T2 such that T1 and T2 are transactions on the same account and the date of T2 is $\leq$ the date of T1. (b) Using only the above view TP, write a query to find for each account the minimum balance it ever reached (not including the 0 balance when the account is created). Assume there is at most one transaction per day on each account and each account has had at least one transaction since it was created. To simply your query, break it up into 2 steps by defining an intermediate view V.	
4	(a) Given a relation graph (P,Q, cost) where P and Q attributes represent vertices associated to edges in the graph and cost represent weight. Write SQL/PL-SQL program for the followings (i) Find the vertices with max and min degree. (ii) List all the path of length 2 with total cost less than 10. (b) List all the loops of length 3. Display all the nodes in each loop (if exists) (c) Write a PL/SQL program to find minimum spanning Tree of the graph	
5	(a) Given a relation graph (P,Q, cost) where P and Q attributes represent vertices associated to edges in the graph and cost represent weight. Write SQL/PL-SQL program for the followings (i) Find the vertices with max and min degree. (ii) List all the path of length 2 with total cost less than 10. (b) Test whether the graph is a tree	

	(c) Create a table for representing a complete graph with n vertices labelled {1,2,3,…,n}, where cost of each edhe eij = |i-j|. Write a PL/SQL program to find the minimum spanning tree of the graph.	
6	Write a PL/SQL code to print the frequency of numbers between 1 and 1000 such that the recursive sum of squares of digits of a number converges cyclically to a one-digit number. (15/20) Example - 1 -> 1 -> 1 (so on) 2 -> 4 -> 16 -> 37 -> 58 -> 89 -> 145 -> 42 -> 20 -> 4 (cyclical convergence to 4) 3 -> 9 -> 81 -> 65 -> 61 -> 37 -> 58 -> 89 -> 145 -> 42 -> 20 -> 4 (cyclical convergence to 4) Answer - For 1 to 3, frequency table is	

Number	Frequency
0	0
1	1
2	0
3	0
4	2
5	0
6	0
7	0

7	Considering the HR database in SQL Plus (Oracle Database 11g), write PL\SQL codes for the following: a. Write a function taking a manager ID and returning the names of employees who report to this manager. (Use cursor internally) b. Write a procedure taking department ID and changing the manager ID for the department to the employee in the department with minimum salary. (Use Exceptions). c. Write the following details into job history whenever a job is changed for an employee - (Employee ID, old job ID, old department ID, hire date of the employee for start date, system date for end date). If a row is already present for employee job history then the start date should be the end date of that row + 1.	

8

Set-A

1. Create the following tables with relevant tuples/records.

EMP(eno, ename, sal, J-date, dno) → foreign key (DEPT)

DEPT(dno, dname, dloc, dmgrno) → foreign key (EMP)

Write SQL/~~PL~~-SQL Program for the following:

(i) Find the dname wise total manpower cost and no. of employees

(ii) Write a PL-SQL Program to update salary of employees with following conditions:

dname	% of increase
HR	5
PRODUCT	10
SALES	7
RESEARCH	10

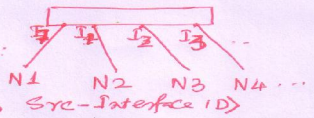
(iii) Suppose 'RESEARCH' dept to be closed. Delete details of the dept from DEPT table. Write a PL/SQL Program to implement automatic deletion of referenced tuples from EMP table.

(iv) Write a PL/SQL Program using cursors to find names of maximum salary earner in each dept.

9

5. (a) Consider a network where each node is labelled with node-ID. All the nodes are connected to a switch through interfaces. Each Interface is labelled with interface-ID. Now, assume data transfer requests are issued by nodes randomly.

Structure of Request is as follows:



<src-node ID, Dst-node ID, Req NO, Src-Interface ID>

The switch maintains a switch table ~~consisting of~~ where each entry contains <Node-ID, Interface-ID>. The switch table is initially empty and gets dynamically populated with connectivity info. based on requests. On receipt of a request, switch inserts an entry in the ST <Src-Node ID, Src-Interface-ID> if the entry is not already in place. Subsequently, switch searches the ST indexed by Dst-Node-ID (of request). If there is a match between Dst-Node ID (of request) and any entry in ST (Node-ID), it displays "# Req NO, # Src-Node ID.. Forwarded to # interface ID" else it inserts an entry in a new table called Broadcast (Req NO, Src-Node ID, Dest Node ID, Sys Time).

(i) Write a PL/SQL Program which will ~~randomly~~ issue a number of data transfer requests (intervals of Nodes) by providing the required as input and accordingly will populate ST and Broadcast Table automatically.

(ii) Display all broadcast ~~requests~~ records after a given Sys-time.

* Validate that after few requests, no. of broadcast records gets reduced. *