

Database Laboratory Assignment # 3

1. Use the company database created in Assignment # 2.

(a) Write a PL/SQL program to raise the salary of employees as follows:

All managers: 20% increase

Dno % of increase

1 10

2 15

3 18

(b) Write a PL/SQL program to find maximum salary earners name (use cursor)

(c) Write a PL/SQL program to find top N/2 salary earners name

2. 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 (acctno, 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 simplify your query, break it up into 2 steps by defining an intermediate view V.

3. 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:

The program will take option I(issue) / R(return) along with card_no from user. Accordingly, insert records in Issue and Return relations. Date of issue/return must be populated with SYSDATE. Before issuing book, check the following constraints: A member is not allowed to issue more than 3 books. A member is not allowed to issue more than one copy of same bookid.

After issuing/return of book the no of copies must be 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.

4. 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.

5. 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 check conditions for insufficient fund (before withdraw) and raise error. For successful transaction, the program must update Account relation and subsequently insert records of the transactions in a new table called Transaction with following definition:

Transaction(Tr_id, Accno, Tr_type, Amt, date of Tr)

Deadline of Submission: 28.01.2020

Database Laboratory Assignment # 4

1. Write database triggers for Q 3 and Q 5 of Assignment # 3. Test the correctness of the triggers with relevant DML operations (e.g. issue/return of books, withdraw/deposit transactions on account)

2. Consider the following relations for an OPD in an hospital:

Patient(patient-id, patient-name, DOB, Sex)

Doctor(Doctor-id, Name, specialization, Unit)

OPD_Schedule(Doctor-id, date, time, fees)

Appointment (appointment-no, patient-id, doctor-id, date)

OPD_payments(appointment-no, patient-id, amt, date_payment)

Write SQL/PL-SQL program for the following tasks:

(i) Display the list of patients who has visited the same doctor in the hospital more than 2 times during a given time-period

(ii) The records in OPD_payments to be automatically updated based on Schedule and Appointment table. If a patient visits twice the same doctor within a week, amt in OPD_payments will be automatically set to 0. For senior citizen female patients, the fees to be reduced by 50% and entry should be automatically inserted in OPD_payments table.

(iii) Display the number of patients visited to the doctors unit-wise and date-wise.

(iv) Change the date and time (not other field) in OPD_Schedule of any given doctor through views.

(v) Create a OPD log including patient name, age, date_of_visits and doctor names for a given period.

(vi) Increase the fees of a given doctor through views.

3. Consider the following database:

Student(Sroll, Name, Branch, Batch, Programme)

Course(CID, Cname, Instructor Name)

Attendance(Sroll, Course ID, Period, Total#class, Attendance)

Period includes start-date and end-date, Total#class states total number of classes conducted during a period.

Write SQL/PL-SQL program for the following tasks:

(i) Display the name of students whose attendance is below 80% in a given course.

(ii) Create a view to list attendance records of the students in all subjects for a given batch.

(iii) Use the view in (ii) to list the student names whose attendance is below 70% in all the subjects.

(iv) Create a view to display attendance of students for a given course ID.

(v) Use this view in (iv) to update the attendance of a student (assume the update is done by a authorized user)

(vi) Internal marks for attendance for a student is to be calculated against a course as per the following rules:

Attendance % Marks

>=95	5
85-95	4
75-85	3
60-75	2
< 60	0

Write a PL/SQL function which takes sroll and course id as input and computes the attendance percentage for that course and returns the marks for that course. Then, Insert the Sroll and Marks in a newly created table called ATT_Marks(Sroll, course-id, Marks). This complete task is to be done in a single PL/SQL program.

Deadline of Submission: 04.02.2020