

CENTRAL UNIVERSITY OF TAMILNADU

DEPARTMENT OF COMPUTER SCIENCE

M.Sc., COMPUTER SCIENCE

MSCP12 - DATABASE MANAGEMENT SYSTEMS LABORATORY

TIME: 3 HOURS

MARKS: 100

1. Consider the following relations for a bus reservation system application:

BUS (ROUTENO, SOURCE, DESTINATION)

PASSENGER (PID, PNAME, DOB, GENDER)

BOOK_TICKET (PID, ROUTENO, JOURNEY_DATE, SEAT_NO)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [15]
Note: Read all questions and populate values accordingly.
- c. Include constraint that DOB of passenger should be after 2010 [5]
- d. Display the passengers who had booked the journey from Bangalore to Chennai on 03-NOV-2014. [10]
- e. List the details of passengers who have traveled more than three times on the same route. [10]
- f. Create a view that displays the RouteNo, source, destination and journey_date which moves from Chennai to Pune. [10]
- g. Create an index on PID in passenger table. [5]
- h. Create a PL / SQL stored procedure that accepts journey_date and displays list of passengers booked ticket on that date. [20]
- i. In the above created procedure, include exceptions to display "No ticket booked on specified date" for a given journey_date [20]

2. Consider the following relations for a boat management application for a beach resort:

SAILOR (SID, NAME, DOB, GENDER)

BOAT (BID, BTYPE, BNAME, COLOR)

BTYPE can take two values (D, S)

D – Deluxe

S –Super Deluxe

SAILS (SID, BID, DOT, SHIFT)

DOT – Date of Trip

SHIFT can take two values – FN or AN

A sailor is assigned a boat on a day. A sailor is permitted to sail the boat for only one shift on a day.

- a. The primary keys are underlined. Identify the foreign keys and draw schema

diagram [5]

- b. Create the above mentioned tables and populate the tables [15]

Note: Read all questions and populate values accordingly.

- c. Include constraints for BTYPE and SHIFT as mentioned above [10]

- d. Develop a SQL query to list the details of boats whose type is Super Deluxe and Color is Red. [10]

- e. Develop a view that will keep track of sailor id, sailor name, date of trip, boat id, boat type, boat name and shift. [20]

- f. Create synonym for sailor table. [5]

- g. Create a PL / SQL stored function that accepts SID and returns the name of sailor [20]

- h. In the above created function, include exceptions to display "No such Sailor exist"

when the incorrect SID is given. [15]

3. Consider the following relations for an order processing application:

CUSTOMER (CID, NAME)

PRODUCT (PCODE, PNAME, UNIT_PRICE)

CUST_ORDER (OCODE, ODATE, CID)

ORDER_PRODUCT (OCODE, PCODE, NOU)

NOU – Number of Units. An order can contain many products.

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

c. Ensure that product names should be within Laptop, Mouse, Server, Air conditioner [5]

d. Develop a SQL query to list the details of products whose unit price is greater than the average price of all products [10]

e. List the customer names who have orders more number of products [10]

f. Create a view that displays the PCODE, PNAME and NOU of the product ordered [10]

g. Create a function that accepts PCODE, Unit_Price and NOU. Calculate the total_cost of the ordered product. Return the total_cost. [20]

h. Create a sequence named ***Product_Sequence*** that will get incremented by 1. Use the created sequence while inserting PCODE into Product table. [20]

4. Consider the following relations for a transport management system application:

BUS (ROUTENO, SOURCE, DESTINATION)

DRIVER (DID, DNAME, DOB, GENDER)

ASSIGN_ROUTE (DID, ROUTENO, JOURNEY_DATE)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [15]

Note: Read all questions and populate values accordingly.

c. Include constraints that the routeNo starts with letter 'R' and gender of driver is always 'Male' [10]

d. Develop a SQL query to list the details of drivers who have traveled more than three times on the same route [10]

e. Create a sequence named Driver_Sequence that will get incremented by 1. Use the created sequence while inserting DID into Driver table. [20]

f. Create a view that displays the DID, DNAME assigned for RouteNo 'R5' on 02-NOV-2014 [20]

g. Create a procedure that displays the details of all drivers. [20]

5. Consider the following relations for a transport management system application:

DRIVER (DCODE, DNAME, DOB, GENDER)

CITY (CCODE, CNAME)

TRUCK (TRUCKCODE, TTYPE)

TTYPE can take two values ('L', 'H')

L-Light

H- Heavy

Each truck is assigned a unique truck code. There can be many trucks belonging to the same truck type.

DRIVE_TRUCK (TRUCKCODE, DCODE, DOT, CCODE)

DOT – Date of Trip

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

c. Include the constraint as mentioned above and the gender of driver is always 'male'. [10]

d. Develop a SQL query to list the details of each driver and the number of trips traveled. [10]

e. Create an index on truck_code in Drive_truck table [5]

f. Create a view that displays the Driver details and also the city in which he drives a truck [20]

g. Create a procedure that displays the details of all drivers, the truck_code and DOT. Use cursors appropriately. [30]

6. Consider the following relations for an order-processing database application in a company:

CUSTOMER (CUSTOMERNO VARCHAR2 (5), CNAME VARCHAR2 (30), CITY VARCHAR2 (30))

Implement a check constraint to check CUSTOMERNO starts with 'C'

CUST_ORDER (ORDERNO VARCHAR2 (5), ODATE DATE, CUSTOMERNO REFERENCES CUSTOMER, ORD_AMT NUMBER (8))

Implement a check constraint to check ORDERNO starts with 'O'

ITEM (ITEMNO VARCHAR2 (5), ITEM_NAME VARCHAR2 (30), UNIT_PRICE NUMBER (5))

Implement a check constraint to check ITEMNO starts with 'I'

ORDER_ITEM (ORDERNO REFERENCES CUST_ORDER, ITEMNO REFERENCES ITEM, QTY NUMBER (3))

SHIPMENT (ORDERNO REFERENCES CUST_ORDER, ITEMNO REFERENCES ITEM, SHIP_DATE DATE)

Here, ORD_AMT refers to total amount of an order (ORD_AMT is a derived attribute); ODATE is the date the order was placed; SHIP_DATE is the date an order is shipped.

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [30]

Note: Read all questions and populate values accordingly.

c. Include the constraint as mentioned above. [10]

d. Develop a SQL query to list the order number and number of items in each order [10]

e. Create a synonym on for CUST_ORDER table [5]

f. Create a view that will keep track of the details of each customer and the number of orders placed by each customer [20]

g. Develop a database trigger that will not permit to insert more than six records in the CUST_ORDER relation for a particular order. (An order can contain a maximum of six items). [20]

7. Consider the following relational schema for a banking database application:

CUSTOMER (CID, CNAME)

ACCOUNT (ANO, ATYPE, BALANCE, CID)

An account can be a savings account or a current account. Check ATYPE in 'S' or 'C'. A customer can have both types of accounts.

TRANSACTION (TID, ANO, TTYPE, TDATE, TAMOUNT)

TTYPE CAN BE 'D' OR 'W'

D- Deposit; W – Withdrawal

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [15]

Note: Read all questions and populate values accordingly.

c. Include the constraints as mentioned above. [10]

d. Write a query that lists the customer details and the number of accounts each customer has. [10]

e. Create a sequence named Customer_Sequence which gets incremented by 10 and use this sequence to give values of CID in customer table. [10]

f. Create a view that will keep track of the details of each customer and account details who have both savings and current account. [10]

g. Develop a database procedure that will accept transaction id, account number, transaction type, transaction date and transaction amount as input and insert a record into TRANSACTION table subject to the following conditions:

i. If TTYPE ='D' the value of BALANCE in the ACCOUNT table must be incremented by the value of TAMOUNT

ii. If TTYPE ='W' the value of BALANCE in the ACCOUNT table must be decremented by the value of TAMOUNT.

If a minimum balance of Rs. 2000/- will be maintained for a savings account and a minimum balance of Rs. 5000/- will be maintained for a current account else appropriate messages must be displayed [30]

h. In the above created procedure, if TTYPE = 'W', and transaction amount is > available balance, raise exceptions to display "Amount > available Balance" [10]

8. Consider the following relational schema for a banking database application:

CUSTOMER (CID, CNAME)

BRANCH (BCODE, BNAME)

ACCOUNT (ANO, ATYPE, BALANCE, CID, BCODE)

An account can be a savings account or a current account. Check ATYPE in 'S' or 'C'. A customer can have both types of accounts.

TRANSACTION (TID, ANO, TTYPE, TDATE, TAMOUNT)

TTYPE CAN BE 'D' OR 'W'

D- Deposit; W – Withdrawal

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

c. Include the constraints as mentioned above. [10]

d. Develop a SQL query to list the details of branches and the number of accounts in each branch. [10]

e. Develop a SQL query to list the details of customers who have performed three transactions on a day [15]

f. Create a view that will keep track of the details of each customer and account details who have both savings and current account. [10]

g. Develop a database trigger that will update the value of BALANCE in ACCOUNT table when a record is inserted in the transaction table. Consider the following cases:

i. If TTYPE ='D' the value of BALANCE in the ACCOUNT table must be incremented by the value of TAMOUNT

ii. If TTYPE ='W' the value of BALANCE in the ACCOUNT table must be decremented by the value of TAMOUNT.

If a minimum balance of Rs. 2000/- will be maintained for a savings account and a minimum balance of Rs. 5000/- will be maintained for a current account else appropriate messages must be displayed [30]

9. Consider the following relational schema for a library management system:

BOOK (BOOKID, TITLE, PUBLISHERCODE, NO_OF_COPIES)

PUBLISHER (PUBLISHERCODE, PUBLISHER_NAME)

AUTHOR (AUTHORID, AUTHOR_NAME)

BOOK_AUTHOR (BOOKID, AUTHORID)

BORROWER (CARDNO, NAME)

BOOK_LOAN (BOOK ID, CARDNO, DATEOUT, DUE DATE, STATUS)

Implement a Check Constraint for STATUS ('R' – Returned, 'T' – To be returned)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [25]

Note: Read all questions and populate values accordingly.

c. Include the constraints as mentioned above. [5]

d. Develop a SQL query to list the details of borrowers who do not have any books checked out. [5]

e. Develop a SQL query to list the details of borrowers who have more than five books checked out. [10]

f. Create an index on BookID in Book_Loan table [10]

g. Create a view that will keep track of the card number, card holders name and number of books borrowed (Number of books with status 'T') [10]

h. Create a procedure named *Author_Details* that accepts the BookID and displays the author ID, author name and also the status of the book. [30]

10. Consider the following Staff relational schema:

STAFF (STAFFNO, NAME, DOB, GENDER, DOJ, DESIGNATION,
BASIC_PAY, DEPTNO)

GENDER must take the Value 'M' or 'F'

DEPT (DEPTNO, NAME)

SKILL (SKILL_CODE, DESCRIPTION, CHARGE_OUTRATE)

STAFF_SKILL (STAFFNO, SKILL_CODE)

PROJECT (PROJECTNO, PNAME, START_DATE, END_DATE, BUDGET,
PROJECT_MANAGER_STAFFNO)

WORKS (STAFFNO, PROJECTNO, DATE_WORKED_ON, IN_TIME,
OUT_TIME)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [30]

Note: Read all questions and populate values accordingly.

c. Include the constraints as mentioned above. [5]

d. Develop a SQL query to list the details of staff who earn less than the basic pay of all staff. [10]

e. Create a view that keeps track of DeptNo, DeptName and number of staff in each department. [10]

f. Develop a SQL query to list the details of staff who have more than three skills. [5]

g. Create an index on StaffNo in Works table [5]

h. Develop a procedure *Staff_Increment* that will accept staff number and increment amount as input and update the basic pay of the staff in the staff table. [20]

i. In the above procedure include exception in the procedure that will display a message "Staff has basic pay null" if the basic pay of the staff is null and display a message "No such staff number" if the staff number does not exist in the staff table. [10]

11. Consider the following relational schema for a company database application:

**EMPLOYEE (ENO, NAME, GENDER, DOB, DOJ, DESIGNATION, BASIC,
DEPT_NO, PAN, SENO)**

Implement a Check Constraint for GENDER

PAN – Permanent account Number

SENO – Supervisor Employee Number

DEPARTMENT (DEPT_NO, NAME, MENO)

MENO - Manager Employee Number

PROJECT (PROJ_NO, NAME, DEPT_NO)

WORKSFOR (ENO, PROJ_NO, DATE_WORKED, HOURS)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.
- c. Include the constraints as mentioned above. [5]
- d. Develop a SQL query to list the details of department which has more than 3 employees working for it. [10]
- e. Create a view that keeps track of DeptNo, DeptName and number of employees in each department. [10]
- f. Develop an SQL query to list the departments and the details of manager in each department. [5]
- g. Create an index on EmpNo in WorksFor table [5]
- h. Develop a procedure *Employee_Increment* that will accept Employee number and increment amount as input and update the basic pay of the employee in the employee table. [20]
- i. In the above procedure include exception in the procedure that will display a message “Employee has basic pay null” if the basic pay of the employee is null and display a message “No such Employee number” if the employee number does not exist in the employee table. [10]
- j. Create a database trigger that will not permit to insert values into Employee table if DOJ is less than DOB. [10]

12. Consider the following relational schema for a Product Sales database application:

Product (Prodid, Prodesc, Price, Stock)
Purchase (Purid, Proid, qty, supplierName)
Sales (Saleid, Proid, qty, custname)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

- c. Include the constraint on Saleid that it starts with letter 'S'. [5]
- d. Display the Prodid and the sum of quantity purchased for each product. [10]
- e. Create a view that keeps track of Prodid, price, Purid, qty and customerName who made the purchase. [20]
- f. Create a sequence named **Product_Sequence** that gets incremented by 10 and use it for inserting Prodid values in Product table. [10]
- g. Develop a procedure named **Product_Sales** that accepts a prodid and displays all the sales and purchase records of it. [20]
- h. In the above procedure include exception in the procedure that will display a message "No such Product ID" if the given product id does not exist in the product table. [10]

13. Consider the following relational schema for a Sales database application:

Product (Prodid, Prodesc, Price, Stock)
Purchase (Purid, Proid, qty, supplierName)
Sales (Saleid, Proid, qty, custname)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.
- c. Include the constraint on Saleid that it starts with letter 'S'. [5]
- d. Display the Prodid's of the product which are purchased more than 5 times [10]
- e. Create a view that keeps track of Prodid, price, Purid, qty and customerName who made the purchase. [20]
- f. Create a sequence named **Product_Sequence** that gets incremented by 10 and use it for inserting Prodid values in Product table. [10]
- g. Develop a procedure named **Product_Purchase** which accepts a purchase id and displays all products that are purchased, qty and price of the product [20]
- h. Create a database trigger that will not permit a customer to purchase more than 5 products. [10]

14. Consider the following relational schema for a Loan database application:

Customer (Custid, Custname, Age, phno)

Loan (Loanid, Amount, Custid)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [10]

Note: Read all questions and populate values accordingly.

c. Include the constraint on Loanid that it starts with letter 'L'. [5]

d. Display the list of the customerids and total Loan amount taken [10]

e. Display the CustId and CustName who have taken less than 2 loans [10]

f. Create a view that keeps track of Custid, Custname, loanid and loan amount. [20]

g. Create a sequence named *Customer_Sequence* that gets incremented by 3 and use it for inserting Custid values in Customer table. [10]

h. Develop a function named *Customer_Loan* which accepts Loanid as input and displays Custid, CustName and loan_amount. [20]

i. Create a database trigger that will not permit a customer to get more than 3 loans. [10]

15. Consider the following relational schema for a Loan database application:

Customer (Custid, Custname, Age, phno)

HLoan (HLoanid, Amount, Custid)

VLoan (VLoanid, Amount, Custid)

Where HLoan is Housing loan and VLoan is a Vechile loan.

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [10]

Note: Read all questions and populate values accordingly.

c. Include the constraint on HLoanid that it starts with letter 'H' and VLoanid starts with letter 'V'. [5]

d. Display the number of VLoan taken by a particular customer id [10]

e. Display the list of the customerids and total HLoan amount taken. [10]

f. Create a view that keeps track of customer details who have taken both HLoan and VLoan. [20]

g. Create a sequence named *Customer_Sequence* that gets incremented by 3 and use it for inserting Custid values in Customer table. [10]

h. Develop a procedure named *Customer_Loan* which accepts HLoanid as input and displays Custid, CustName and loan_amount of HLoan. [20]

i. In the above procedure include exceptions to display "No such HLoanid" when incorrect Hloanid is given. [10]

16. Consider the following relational schema for a Loan database application:

Customer (Custid, Custname, Addr, phno, pan_no)

Loan (Loanid, Amount, Interest, Custid)

Account (Accid, Accbal, Custid)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [10]

Note: Read all questions and populate values accordingly.
- c. Include the constraint on Custid that it starts with letter 'C' [5]
- d. Display the customer id, name and account balance. Sort the output using custid [10]
- e. Display the accounts of custids 'C01', 'C02', 'C03' [10]
- f. Display the custid who has account balance larger than other customers [5]
- g. Create an index on Accid of Account table. [5]
- h. Create a view that keeps track of customer id, loan amount and account balance. [20]
- i. Develop a procedure named *Customer_Loan* that displays all the loan details [20]
- j. In the above procedure include exceptions to display "No such Loanid" when incorrect loanid is given. [10]

17. Consider the following relational schema for a Sales database application:

Product (Prodid, Prodesc, Price, Stock)
Purchase (Purid, Proid, qty, supplierName)
Sales (Saleid, Proid, qty, custname)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

c. Include the constraint on Saleid that it starts with letter 'S'. [5]

d. Display the names who are both supplier as well as customer [10]

e. Display the amount (price * qty) of Products in each Sales. [10]

f. Create a view which displays Product ids and sum of quantity in sales [20]

g. Create a sequence named **Product_Sequence** that gets incremented by 10 and use it for inserting Prodid values in Product table. [10]

h. Create a Trigger which reduces the stock of Product that is been inserted in sales and print if it is out of stock (stock < Reord) [20]

18. Consider the following relational schema for a Loan database application:

Customer (Custid, Custname, Age, phno)

Loan (Loanid, Amount, Custid, Emi)

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [10]

Note: Read all questions and populate values accordingly.

c. Include the constraint on Custid that it starts with letter 'C'. [5]

d. Update the loan amount by increase in 2 % for all customers [10]

e. Display the custid and Custname whose loan amount lies in the range of 30,000 to 50,000 [10]

f. Display the CustId and CustName who have taken less than 2 loans [10]

g. Create a view that keeps track of Custid, Custname, loanid and loan amount. [20]

h. Create a sequence named *Customer_Sequence* that gets incremented by 3 and use it for inserting Custid values in Customer table. [10]

i. Develop a function named *Customer_Loan* which accepts Loanid as input and displays Custid, CustName and loan_amount. [20]

19. Consider the following relational schema for a Books Ordering database application:

Books (isbn, title, author, stock_qty, price, pub_year)

Customers (cust_id, cust_name, address)

Orders (order_no, cust_id, order_date) where cust_id refs
Customers(cust_id)

Order_list (order_no, isbn, qty, ship_date) where order_no refs
Orders(order_no), isbn refs Books (isbn)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.
- c. Include the constraint on Cust_id that it starts with letter 'C'. [5]
- d. Display the custid and Custname who have ordered more than 3 books on the same date [10]
- e. Display the CustId and CustName who have ordered very few number of books. [10]
- f. Create a view that keeps track of books that are ordered on 05-NOV-2014. Display isbn, title, author, order_no, quantity and order_date. [20]
- g. Create a procedure named **Books_Ordered** which outputs the customer name, book title and quantity ordered for the given order number [20]
- h. In the above created procedure include exception to display "No such Order Number" if incorrect order number is given. [10]

20. Consider the following relational schema for Products Order database application:

Products (p_id, p_name, retail_price, qty_on_hand)

Orders (order_id, order_date)

Order_details (order_number, product_number, qty_ordered)

Where: order_number references order_id

product_number references p_id

a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]

b. Create the above mentioned tables and populate the tables [20]

Note: Read all questions and populate values accordingly.

c. Include the constraint on orderid that it starts with letter 'O'. [5]

d. Display the ProdID and the sum of quantity ordered for each product. [10]

e. Create a view that keeps track of P_id, price, order_id, qty_ordered and ordered_date. [20]

g. Develop a procedure named **Product_Orders** that accepts a Product id or product number and displays all the order_details of the product. [10]

h. Create a database TRIGGER, which deletes the order from Orders table, AFTER the deletion of corresponding order_number in Order_details. [30]

21. Consider the following relations for a bus reservation system application:

BUS (ROUTENO, SOURCE, DESTINATION)

PASSENGER (PID, PNAME, DOB, GENDER)

BOOK_TICKET (PID, ROUTENO, JOURNEY_DATE, SEAT_NO)

- a. The primary keys are underlined. Identify the foreign keys and draw schema diagram [5]
- b. Create the above mentioned tables and populate the tables [15]
Note: Read all questions and populate values accordingly.
- c. Include constraint that DOB of passenger should be after 2010 [5]
- d. Display the passengers who had booked the journey from Bangalore to Chennai on 03-NOV-2014. [10]
- e. List the details of passengers who have traveled more than three times on the same route. [10]
- f. Create a view that displays the RouteNo, source, destination and journey_date which moves from Chennai to Pune. [10]
- g. Create an index on PID in passenger table. [5]
- h. Create a PL / SQL stored procedure that accepts journey_date and displays list of passengers booked ticket on that date. [20]
- i. In the above created procedure, include exceptions to display "No ticket booked on specified date" for a given journey_date [20]

Default Question(40 -10 = 30 marks).

22. Create the Bank Database System with the following relations with instance.

- branch(branch-name, branch-city, assets)
- customer(customer-name, customer-street, customer-city)
- account(account-number, branch-name, balance)
- loan(loan-number, branch-name, amount)
- depositor(customer-name, account-number)
- borrower(customer-name, loan-number)

Branch Table

<i>branch-name</i>	<i>branch-city</i>	<i>assets</i>
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Customer Table

<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Account Table

<i>account-number</i>	<i>branch-name</i>	<i>balance</i>
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750

Loan Table

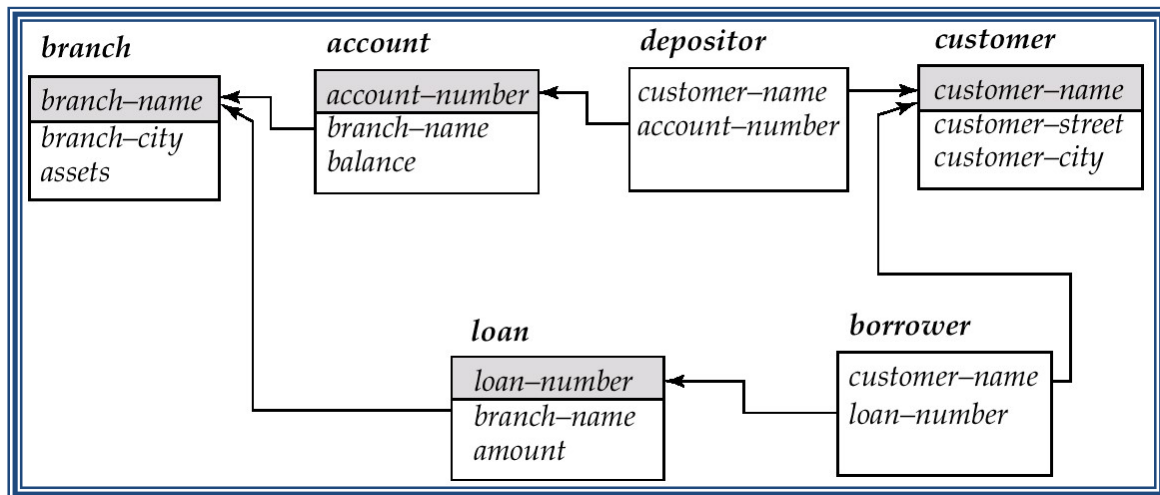
<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Depositor Table

<i>customer-name</i>	<i>account-number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

Borrower Table

<i>customer-name</i>	<i>loan-number</i>
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17



- Find the names of all branches in the loan relations, and remove duplicates.
- To find all loan number for loans made at the Perryridge branch with loan amounts greater than \$1200.
- Find the loan number of those loans with loan amounts between \$90,000 and \$100,000 (that is, ³\$90,000 and £\$100,000)
- Find the name, loan number and loan amount of all customers having a loan at the Perryridge branch.
- Find the name, loan number and loan amount of all customers; rename the column name loan-number as loan-id.