(f) Find the names of all branches that have assets greater than atleast one branch located in Brrnn...

Qs:1 Give the advantage of a DBMS over File processing system.

Ans:1 The advantages of a DBMS over File processing system are:

(i) DATA REDUNDANCY AND INCONSISTENCY

Redundancy means repeatition of data. The data in DBMS is checked for redundancy and consistency i.e. after transaction data is in correct state.

(ii) DIFFICULTY IN ACCESSING DATA

Accessing data in DBMS is much more easier than in File processing system.

(iii) DATA ISOLATION

Isolation refers to separation of data.

(iv) ATOMICITY

Atomicity refers to either all or none. The transaction should be completed either at entirenty or none at all.

(v) INTEGRITY PROBLEM

In DBMS rules are applied for inserting data while in File processing system there is no such rules.

(vi) SECURITY

Data in DBMS file is much more secure than in file processing system.

Qs:2 Define integrity rules.

A:2
- ENTITY INTEGRITY: This deals with primary key of the relational database. It says that the primary key of a relation is not NULL and unique.
- REFERENTIAL INTEGRITY: This deals with the foreign key of the relation. It deals with the value that the foreign key will take.
- DOMAIN CONSTRAINT: This deals with the domain of the relation. It says that relevant domain names should be used and the range of values taken by a specific attribute reflects the type and nature of the attribute.

Q1:3 What is Data Independence? Discuss types.

A13 It refers to the characteristic of being able to modify the schema at one level of the DBMS without altering the schema at the next higher level. It is of 2 types:

1) Physical Data Independence → The data stored in a database must be independent of the application programs accessing this database.

2) Logical Data Independence → The logical data in a database should be independent of the users view application i.e. any change in the logical data must not affect the applications using it.

Q:4 What do you understand by Data models? Define types.

A4 A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. Types of data:

1) HIERARCHICAL MODEL → This database model organises data into a tree-like structure, with a single root, to which all other data is linked.

2) NETWORK MODEL → This is an extension of Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

3) ENTITY - RELATIONSHIP MODEL → In this model, relationship are created by dividing of interest into entity and its characteristics into attributes.

4) RELATION MODEL → In this model, data is organised in two-dimensional tables and the relationship is maintained by storing a common field.

(f) Find the names of all branches that have assets greater than atleast one branch located in Brooklyn.

Qs:5 Define Entity, Relation, Entity set, Relationship set.

A:5 1) ENTITY: An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

2) RELATION: The association among entities is called a relation. It is represented by ◇.
Eg: an employee works-at a department.

3) RELATIONSHIP SET: A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes.

4) ENTITY SET: An entity set is a group of similar kind of entities. It may contain entities with attributes sharing similar values.

Qs:6 Write difference between two-tier and three-tier architecture.

Two-Tier Architecture	Three-Tier Architecture
• It is similar to a basic client-server model. The application at client end directly communicates with the database at the server side.	In this type, there is another layer between the client and the server. The client does not directly communicate with the server.
• The maintenance and understanding is easier, compatible with existing system.	It has enhanced scalability, data integrity and is secured.

Scanned by CamScanner

Q.9 write SQL statements for the following:
& 10.

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

(a) Find the names of all branches in the loan relation.

SQL: select branch-name from loan;

Relational Algebra: $\pi_{branch\text{-}name}$ (loan);

(b) Find the loan number of those loans with loan amounts between \$ 90,000 and \$ 1,00,000

SQL: select loan-number from loan where amount between \$ 90,000 and \$100000.

Relational Algebra: $\pi_{loan\text{-}number}(\sigma_{amount > \$90000 \wedge amount < \$100000}(loan))$;

(b) Find the loan number for loans made at Perryridge branch with loan amount greater than \$ 1200.

SQL: select loan-number from loan where branch-name = 'Perryridge' and amount > \$ 1200;

Relational Algebra: $\pi_{loan\text{-}number}(\sigma_{branch\text{-}name = 'Perryridge' \wedge amount > \$1200})$ (Loan)

(d) Find the customer-names, loan-numbers and loan-amounts for all customers who have a loan from the bank.

SQL: select borrower.customer-name, loan.loan-number, loan.amount from borrower FULL JOIN Loan ON Loan.loan-number = borrower.loan-number;

Relational Algebra: T = loan ⋈ borrower;

$\pi_{customer\text{-}name,\ loan\text{-}number,\ amount}(T)$;

(e) Find the customer-names, loan numbers and loan amounts for all loans at the Perryridge branch.

SQL: select borrower.customer-name, loan.loan-number, loan.amount from borrower FULL JOIN loan ON loan.loan-number = borrower.loan-number where branch-name = 'Perryridge';

Relational Algebra: T = loan ⋈ borrower;

R = $\sigma_{branch\text{-}name = 'Perryridge'}$;

$\pi_{customer\text{-}name,\ loan\text{-}number,\ amount}(R)$;

(f) Find the names of all branches that have assets greater than atleast one branch located in Brooklyn.

SQL: select branch-name from branch where assets > (select asset from branch where branch-name = 'Brooklyn');

Relational Algebra: $R = \pi_{asset}(\sigma_{branch\text{-}name = 'Brooklyn'}(branch))$;

$\pi_{branch\text{-}name}(\sigma_{asset > T}(branch))$;

(g) Find the names of all customers whose street address includes the substring 'Main'.

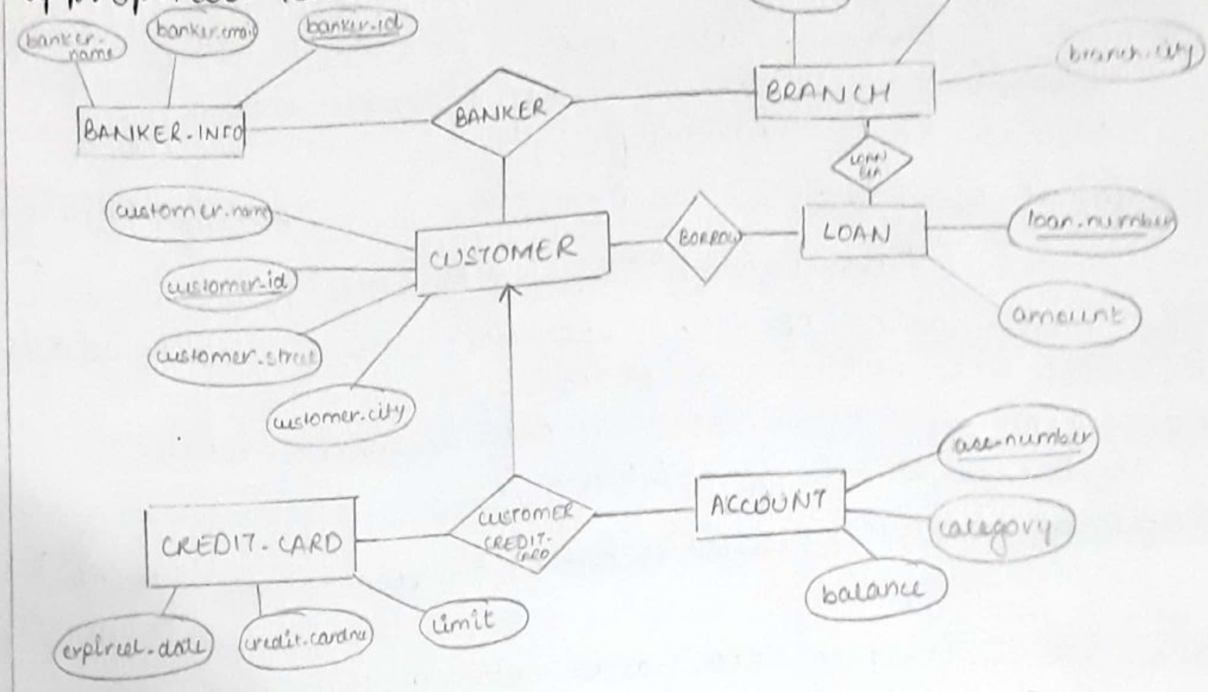
SQL: Select customer-name from customer where customer-street like %Main%;

Relational Algebra: $\pi_{customer\text{-}name}((\sigma_{customerstreet = '\%Main\%'})(customer))$;

(h) Find all customers who have both a loan and an account at the bank.

SQL: select customer-name from loan natural join borrower; ~~where branch-no on~~ and customer-name ~~[illegible]~~ in (select customer-name from account natural join depositor);

6 construct an E-R diagram for Banking system. Construct appropriate table also.



(1) BANKER_INFO

banker-id	banker-name	banker-email

(2) BRANCH

branch-name	branch-city	assets

(3) CUSTOMER

customer-id	customer-name	customerstreet	customer-city

(4) LOAN

loan-number	account

(5) CREDIT-CARD

credit-cardnumber	expired-date	limit

(6) ACCOUNT

acc-number	balance	category

(7) BANKER

banker-id	branch-name	customer-id

(8) BORROW

customer-id	loan-number

(9) CUSTOMER-CREDIT CARD

customer-id	creditcardnumber	acc.numb

7 construct an E-R diagram for Hospital system. Construct app. table

(1) DOCTOR

Did	Name	Address	Department	Phno	Gender

(2) PATIENT

Pid	Name	Age	Address	PhNo	Gender

(3) BILL

BillNo	Doc-charges	Roomcha

(4) ROOM

Room-id	Type

(5) TESTS

Testid	Test-patientid	Testname	Testtype	Testcost	Test-desc	Test-report

(6) ISSUED

Pid	BillNo

(7) ASSIGN

Pid	Room-id

(8) CONDUCTED

Pid	Test-id