Ques:- Give the advantages of DBMS over file processing system.

Ans:- A database management System (DBMS) is a system software that allows users to effeciently define, create, maintain and share databases.

Advantages of DBMs over File System-

- Data redundancy and inconsistency ⇒ Redundancy is the concept of repitition of data, i.e each data may have more than a single copy. The file system cannot control redundancy of data as each user defines and maintains the needed files for a specific application to run. There may be a possibility that two users are maintaining same files data for different applications. Hence changes made by one user does not reflect in files used by second users, which leads in inconsistency of data. Whereas DBMS control redundancy by maintaining a single repository of data that is defined once and is accessed by many users. As there is no or less redundancy, data remains consistent.
- Data sharing ⇒ File system does not allow sharing of data or sharing is too complex. Whereas in DBMS data can be shares easily due to ~~ce~~centralized system.

Security Rules

- Data concurring - Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occurs when changes made by one user gets lost because of changes made by other user. File system does not provide any procedure to stop anomalies where as DBMS provides a locking system to stop anomalies to occur.
- Data Searching => For every search operation performs on file system, a different application programs has to be written. While DBMS provides inbuilt searching operations. User only have to write a small query to retrieve data from database.
- Data Integrity -> There may be cases when some constraints needs to be applied on the data before inserting it in database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS maintains data integrity by enforcing user defined constraints on data by itself.

Q2:- Define Integrity Rules

- Integrity rules are needed to inform the DBMS about certain constraints in the real world.
- Specific integrity rules apply to one specific database. Eg - parts weights must be greatet than zero.
- General integrity rules apply to all databases

Two general rules will be discussed to deal with: primary keys and foreign keys

PRIMARY KEYS

- There could be several candidate keys as long as they satisfy two properties:

1) Uniqueness
2) Minimality

- From the set of candidate keys, one is chosen to be the primary key.
- The others beome alternate keys

The ENTITY INTEGRITY RULE

- No component of the primary key of a base relation is allowed to accept nulls.

FOREIGN KEYS

- A foreign key is an attribute of one relation R2, where values are required to match those of the primary key of some other relation R1.

The REFRENTIAL INTEGRITY RULE

- The database must not contain any unmatchy foreign key values.

Q3:- What is Data Independence ? Discuss types

Ans:- The acquired skill to change a conceptual pattern by not altering the conceptual pattern of the next superior level is defined as the data independence. The conventional data processing does not provide data independance in application programs so, any kind of changes in the information, layouts or arrangements need the change in application program also but in the database system the data independence become easy because of its multilayer feature and DBMS furnish interface in application program and data to have data independencing.

Types of Data Independence

The data independence of two types

- Logical Data Independance - Logical Data independence points out that the conceptual pattern can be altered by undamaging the current external patterns or schemas. It also protects and isolates application programs from actions like combination of dual records into a single records or separating a single record into two or more records

- Physical Data Independence - Physical data independance prints out the physical strong patterns changes by undamaging conceptual structures or arrangements the presence of internal level in the architecture of database and the operation of changes from the conceptual level to internal level archieves the physical data independence.

Q4:- What do you understand by Data Models? Define types.

Underlying the structure of database is the data model: a collection of conceptual model tools for describing data, data relationships, data semantics, and consistency constraints. A data model provides a way to describe the design of a database at the physical, logical and view levels

Some of its types are:-

- Relational Model - The relational model uses a collection of tables to represent both data and relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations. The relational model is an example of a record-based model.

Entity - Relationship Model - The entity - relationship (E-R) data model uses a collection of basic objects called entities and relationships among these objects. An entity is a "thing" or "object" in the real world.

Object - Based Data Model - Object - oriented programming has become the dominant software development methodology. This led to the development of object - oriented data model that can be seen as extending the E-R model with notions of encapsulation, methods and objects identity.

Define Entity, Relations, Entity set, Relationship set.

Entity - An entity is any singular, identifiable and seperate object. It refers to individuals, organizations, systems, bits of data or even distinct system component that are considered significant in and of themselves.

Relations - It is originally defines as a set of tuples $(d_1, d_2, \ldots d_n)$ where each element d_j is a membet of D_j a data domain. Codd's original defination notwithstanding and contrary to the usual defination, there is no ordering to the elements of the tuples of a relation.

ite SQL

anch (branch name, branch city, assets)
customer street, customer city)

Entity set - Entity set is a set of entity all of which have the same set of attributes. Entity sets need not to be dispute.

Relationship set - It is a set of relation-ship of the same type. Formally it is a mathematical relation on $n \geq 2$ sets. If $E_1, E_2, \ldots E_n$ are entity sets, then a relationship set R is a subset

Q6:- Differentiate between two-tier and three-tier architecture.

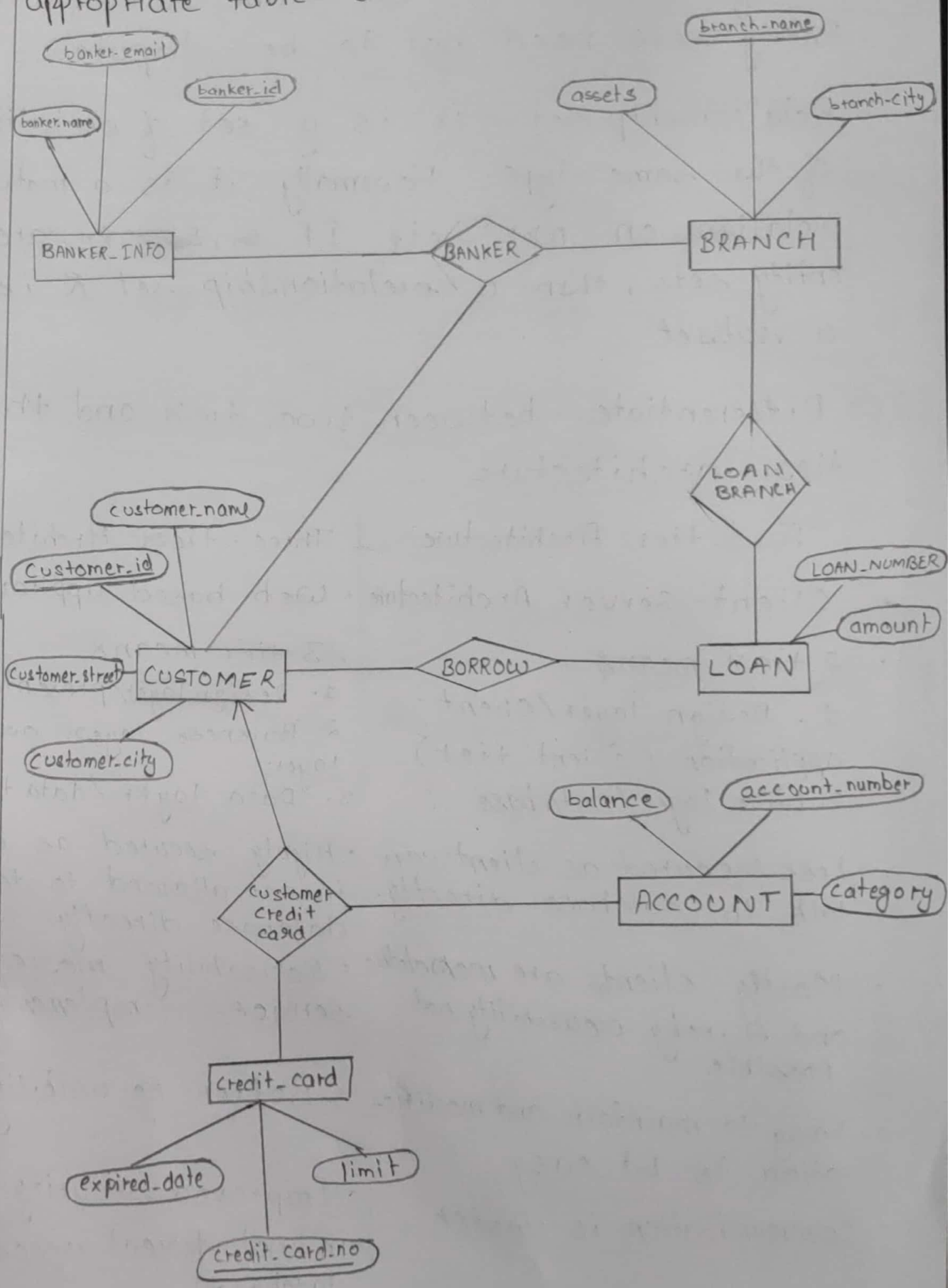
Two-tier Architecture	Three-tier Architecture
• Client-Server Architecture	• Web based application
• 2 tier means 1. Design layer/Client application (client tier) 2. Data layer/Database	• 3 tier means 1. Design layer/presentation. 2. Buisness layer or logic layer 3. Data layer/data tier
• Less secured as client can talk to database directly.	• Highly secured as client is not allowed to talk to database directly.
• Mostly clients are monolothic and thereby reusability not possible.	• Reusability more with services implementation.
• Easy to maintain and modification is bit easy.	• Better Re-usability
• communication is faster.	• Improved security client is not direct access to database.

Q9:- Write SQL statements for the following:

BANKER_INFO

BRANCH

Q7:- Construct an E-R Diagram for banking system. Construct appropriate table also



Q9:-

BANKER_INFO

banker_id	banker_name	banker_email

BRANCH

branch_name	assets	branch_city

LOAN

Laan_number	amount

CUSTOMER

Customer_id	customer_name	customer_street	customer_city

ACCOUNT

Account_number	balance	Category

CREDIT_CARD

Credit_card_number	expired_date	limit

BANKER

Banker_id	Branch_name	customer_id

Loan_Branch

Branch_name	loan_number

Borrow

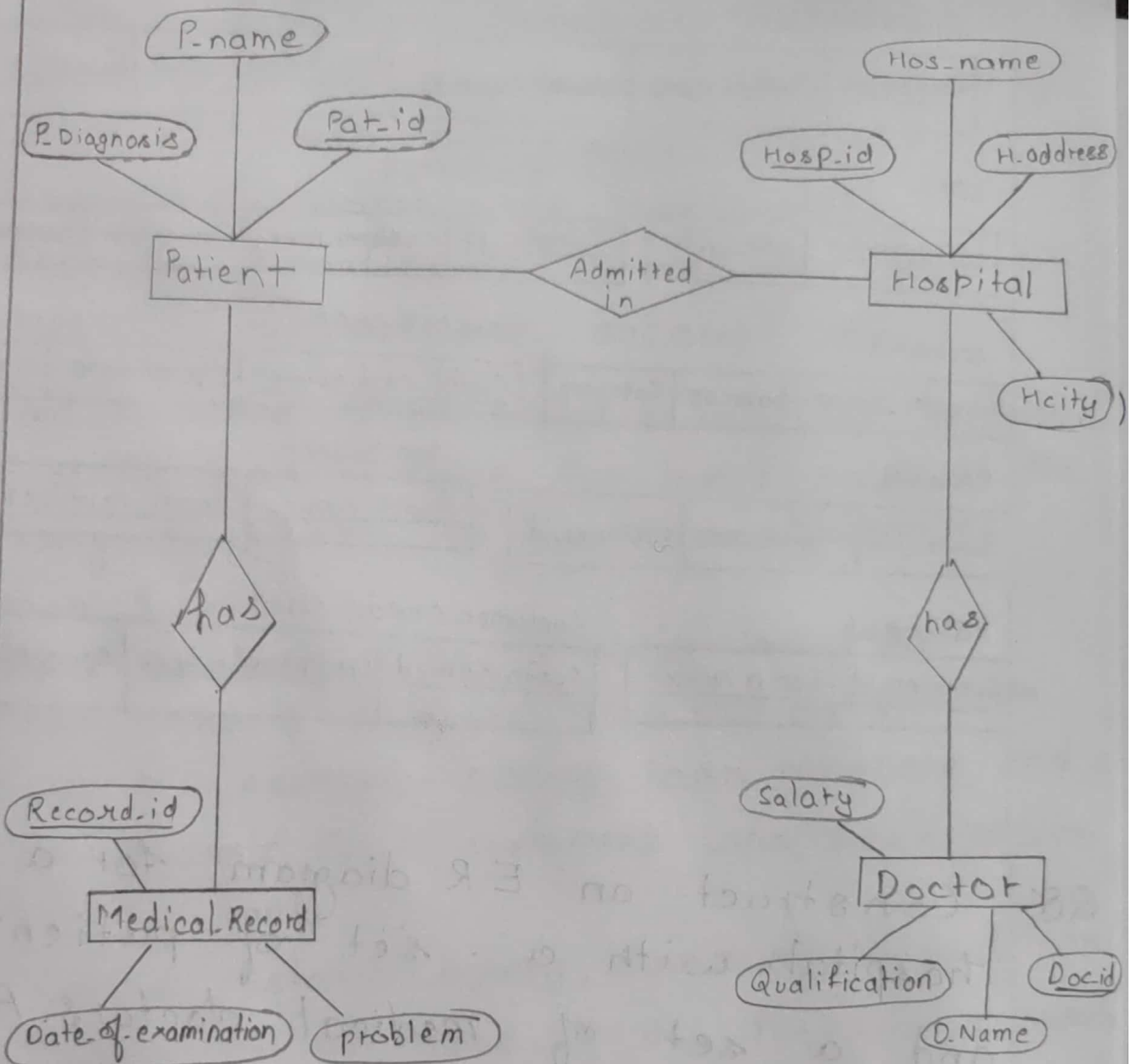
Customer_id	Loan_number

Customer_credit_card

Customer_id	Credit_card_number	Account_number

Q8:- Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Construct appropriate tables for the above ER Diagram.

... statements for the following:



Patient

Pat_id	P_name	P_Diagonostic

Hospital

Hosp_id	Hos_Name	HAddress	Hcity

Doctor

Doc_id	DName	Qualification	Salary

Medical_Record

Record_id	Date_of_examination	Problem

Admitted_in

Pat_id	Hosp_id

has

Hosp_id	Doc_id

has

Pat_id	Record_id

Q9:- Write SQL statements for the following:

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 loan relation
(b) Find all loan numbers for loans made at the Perryridge branch with loan amounts greater than $1200
(c) Find the loan number of those loans with loan amounts between $90,000 and $100,000.
(d) Find the customer names, loan numbers and loan amounts for all customers who have a loan from the bank.
(e) Find the customer names, loan numbers and loan amounts for all loans at the Perryridge branch
(f) Find the names of all branches that have assets greater than atleast one branch located in Brooklyn.
(g) Find the names of all customers whose streets address include the substring 'Main'.
(i) Find all customers who have an account but no loan at the bank.
(j) Find the average account balance of each branch.

Ans 4

(a) Select branch name from loan;

(b) select loan_number from loan where branch name = 'Perryridge' and amount > 1200;

(c) Select loan_number from loan where amount between 90000 and 100000;

(d) Select customer_name, loan_number, amount from borrow, loan;

(e) select customer_name, loan_number, amount from borrower, loan where branch_name = 'Perryridge';

(f) Select branch_name from branch where assets > (select max(assets) from branch where branch_city = "Brooklyn");

(g) Select customer_name where customer_street like '%.Main%.';

(h) Select * from depositor where depositor.customer_name = borrower.customer_name;

(i) Select * from depositors where not depositor.customer_name = borrower.customer_name;

(j) Select avg(balance) as avg.balance from account group by branch_name;

Q10: Give Relational Algebra queries for the following Q9 (a-j):

Ans:

(a) $\Pi_{branch\_name}(loan)$

(b) $\Pi_{loan\_numbers}(\sigma_{branch\_name = 'Perryridge' \wedge amount > 1200}(loan))$

(c) $T = \sigma_{amount >= 90000 \wedge amount <= 100000}(loan)$

$\Pi_{loan\_number}(T)$

(d) $\Pi_{customer\_name, loan\_number, amount}(borrower \bowtie loan)$

(e) $T = \sigma_{branch\_name = "Perryridge"}(borrow \bowtie loan)$

$\Pi_{customer\_name, loan\_number, amount}(T)$

(h) $\Pi_{customer\_name, account.number}(depositor \bowtie borrower)$ ~~, amount (T)~~

(i) $T = \sigma_{depositor.customer\_name != borrower.customername}(depositor \bowtie borrower)$

$\Pi_{customer\_name, account\_number}(T)$

(j) $_{branchname}\mathcal{G}_{avg(balance)}(account)$