Page No. 1
Date

Assignment - 1

Q.1)) Explain overall architecture of D.B.M.S in detail.

Ans) The architecture of a database System is greatly influenced by the underlying Computer System on which databases is running:

i.) Centralized
ii.) Client-Server.
iii.) Parallel (multi-Processor).
iv.) Distributed.

Explaination of Fig a. Database System Architecture is given below:

(1) Data base users and User Interfaces:-

(a) There are 4 different types of database-System users. (I) Naive users (II) Application-Programmers (III) Sophisticated users. (IV) Specialized users.

(b) Naive users:- These are unsophisticated users who interact with the system by invoking one of the application programs. They comes View-lvl abstraction.

c) Application programmers:- These are computer professionals who write application programs. They use DDL & DML to create backend and link it with frontend of application programs for ease of Naive users. They come under logical-lvl abstraction.

d) Sophisticated users:- They write database query which is fed to query processor, which function is to breakdown DML statements. Analysts which who submit queries to explore data fall under this category. They come under logical-lvl abstraction.

e) Specialized users:- These users write ↓ specialized database applications (frame works) for the ease of application programmers.

(ii) Database Administrator (DBA):-

a) The Database administrator have central control over both the data and the programs that access those data.
b) The functions of DBA include:-
c) Schema definition
d) Storage Structure and access-method definition.
e) Schema and Physical-organization modification.
f) Granting of authorization for data access.
g) Routine maintenance.

h) They come under Physical & view level abstraction.

(iii) The Query Processor includes:-

a) DDL interpreter, which interpret DDL State-ments and records the definitions in the data dictionary.
b) DML compiler, which translates DML Statements in a query language.
c) Query evaluation engine, which executes low-lvl instructions generated by the DML compiler

(iv) Storage Manager:-

a) A Storage Manager is a program module that provides the interface between the low level data Stored in the database and the application programs and queries submitted to the System.

The Storage manager components include:

b) Authorization and integrity manager, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.
c) Transaction manager, ensures Atomicity.
d) File manager, which manages allocation of space on disk Storage and the data structures used to represent information Stored on disk
e) Buffer manager fetches data from disk storage into main memory.

(v) Disk Storage maintains all files, dictionary and hashes at physical level.

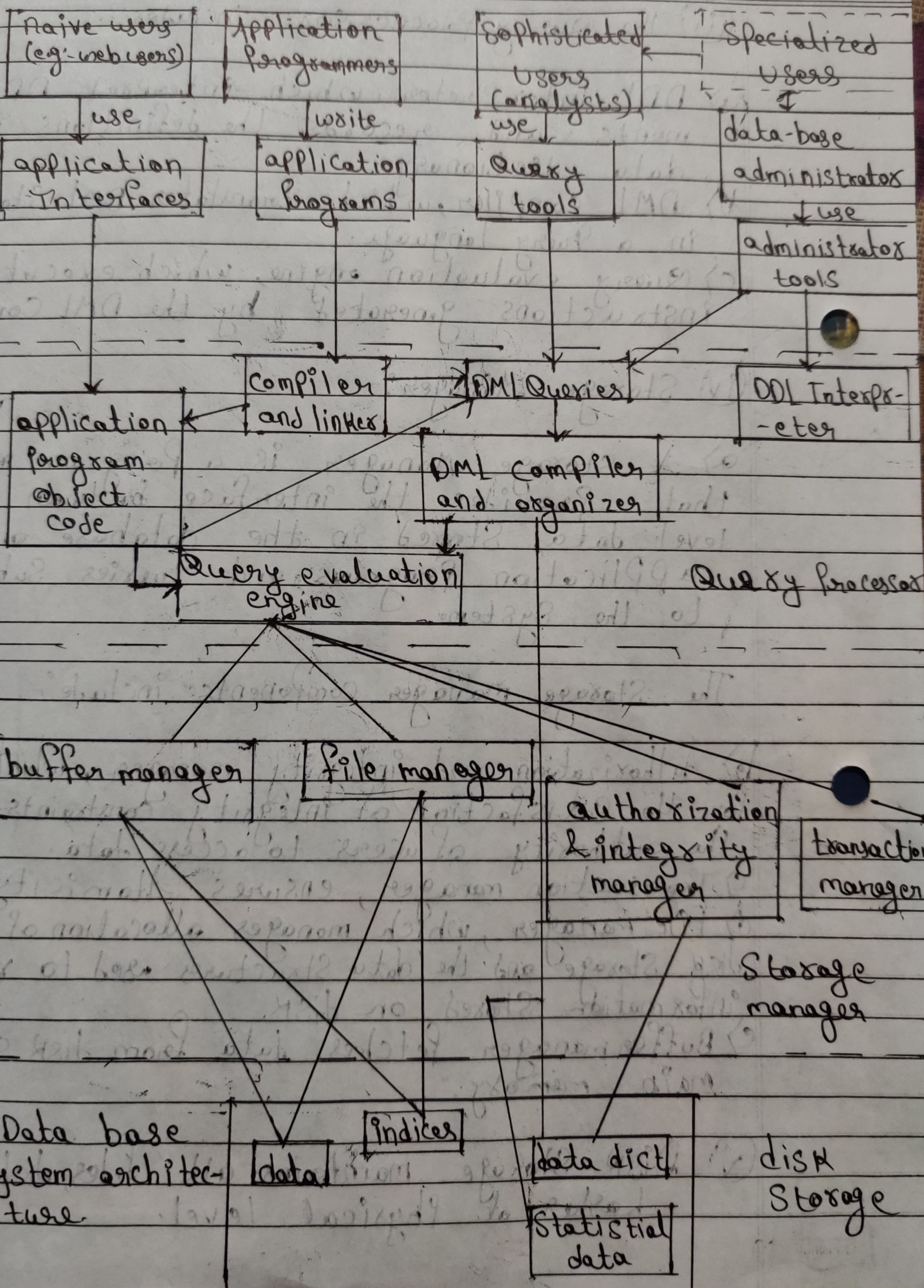
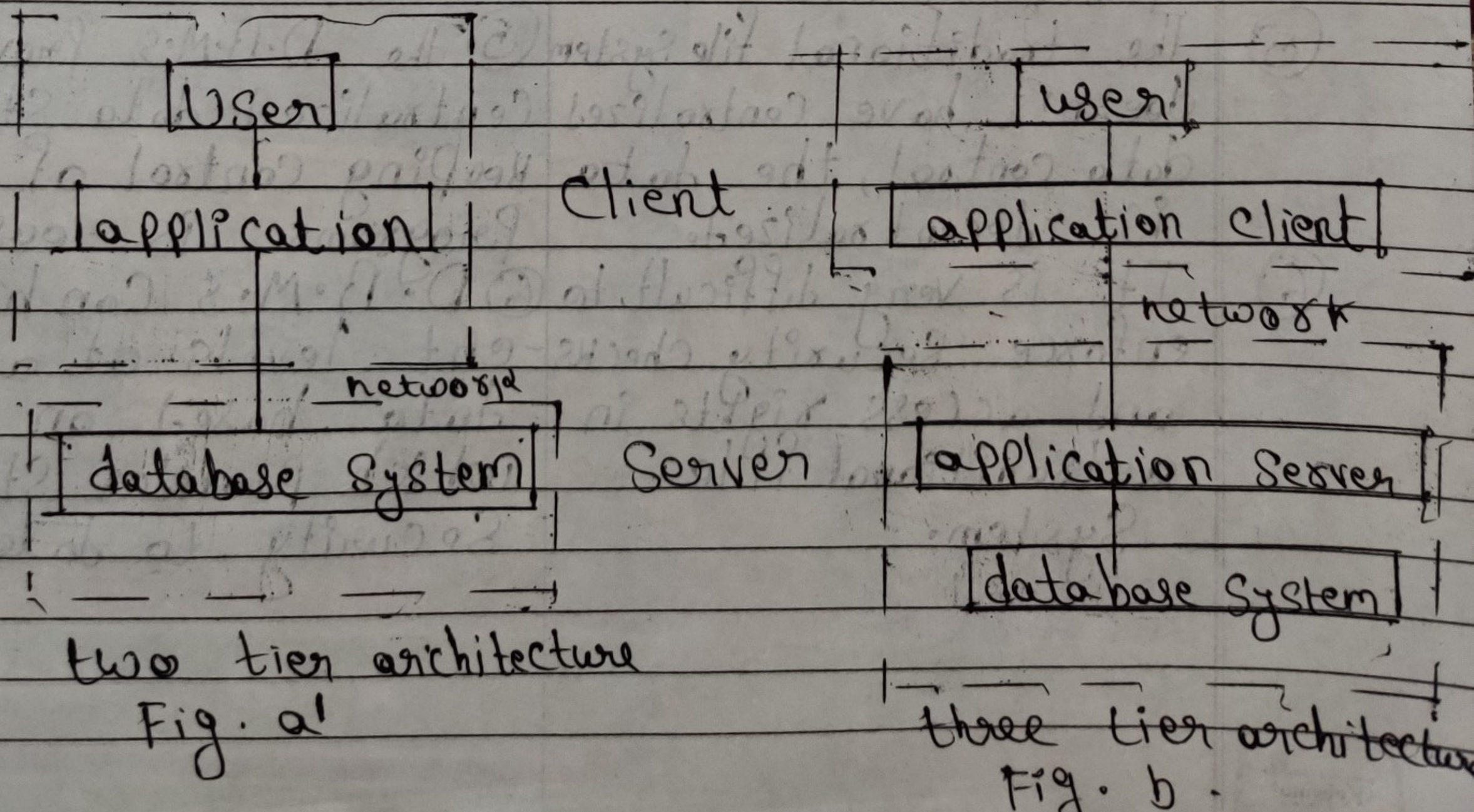


Fig a:- Data base System architecture

(v) Database applications are usually partitioned into two parts, as in Figure a, b

a) In a two tier architecture, the application is partitioned into component that resides at the client machine, which invokes database functionally at the serve machine through query language statements. Application program interface standards like ODBC & JDBC are used for interaction b/w the client and the server.

b) In a three architier architecture the client machine acts as merely a frontend and doesn't contain any direct database calls. Instead, the client end communications with the application server, usually through form interface. The application server in turns communicates with a database system to access data.



Q.2) Difference between File Processing & Database Management System.

Ans)

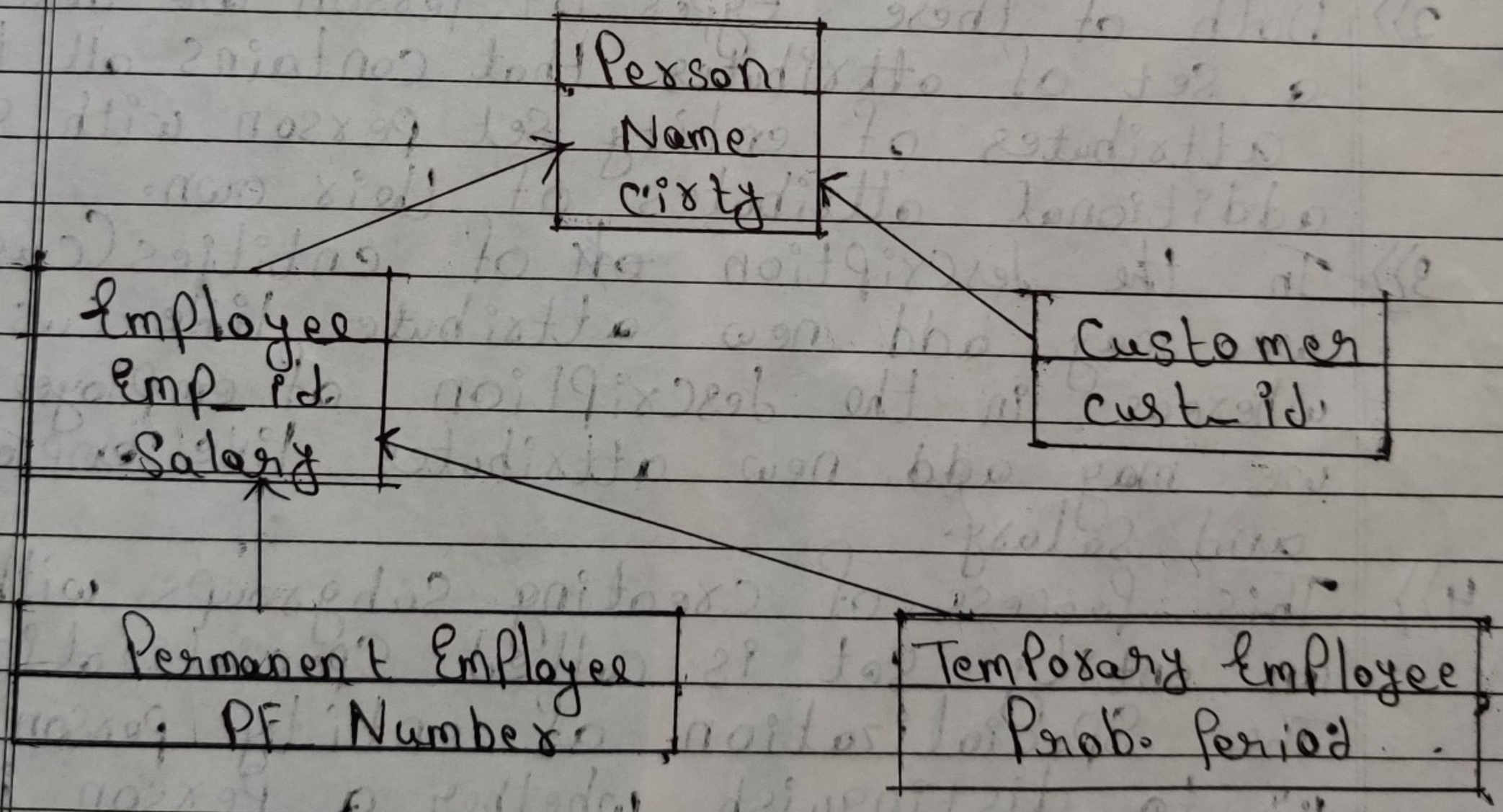
File Processing System	Database Management System
① Duplicate data may exist in multiple files which leads to data redundancy.	① The data is integrated into a single database which avoids data redundancy.
② Data inconsistency occurs when data is not updated in all files.	② The data consistency is maintained since, data is integ-rated in a database.
③ It is difficult to share data in traditional file syster	③ In D.B.M.S data can be easily shared with multiple apps.
④ In files data is stored in specific format. If the format of any of the file is changed, then we have to make changes in the program ~~with~~ which processes file.	④ In D.B.M.S we can comple-tely seperate the data stru-cture of database & pro-grams or applications which are used to access data.
⑤ The traditional file system doesn't have centralized data control, the data is decentralized.	⑤ The D.B.M.S provides the centralized data storage. Hence keeping control of data & programs is easy.
⑥ It is very difficult to enforce security checks and access rights in a traditional file System.	⑥ D.B.M.S can have differ-ent levels of access to data based on their roles which provides storage security to data.

Q.3) Explain Extended ER features.

Ans) (A) Generalization:-

① The refinement from an initial entity set into subgroups depending upon distinct features shows top-down approach.

② Generalization is termed as containment relation-ship that exists between higher level entity set and one or more lower level entity sets.



Generalization

③ Here, the employee is higher level entity set while the entities 'Permanent employee' and 'Temporary employee' are the lower level entity set.

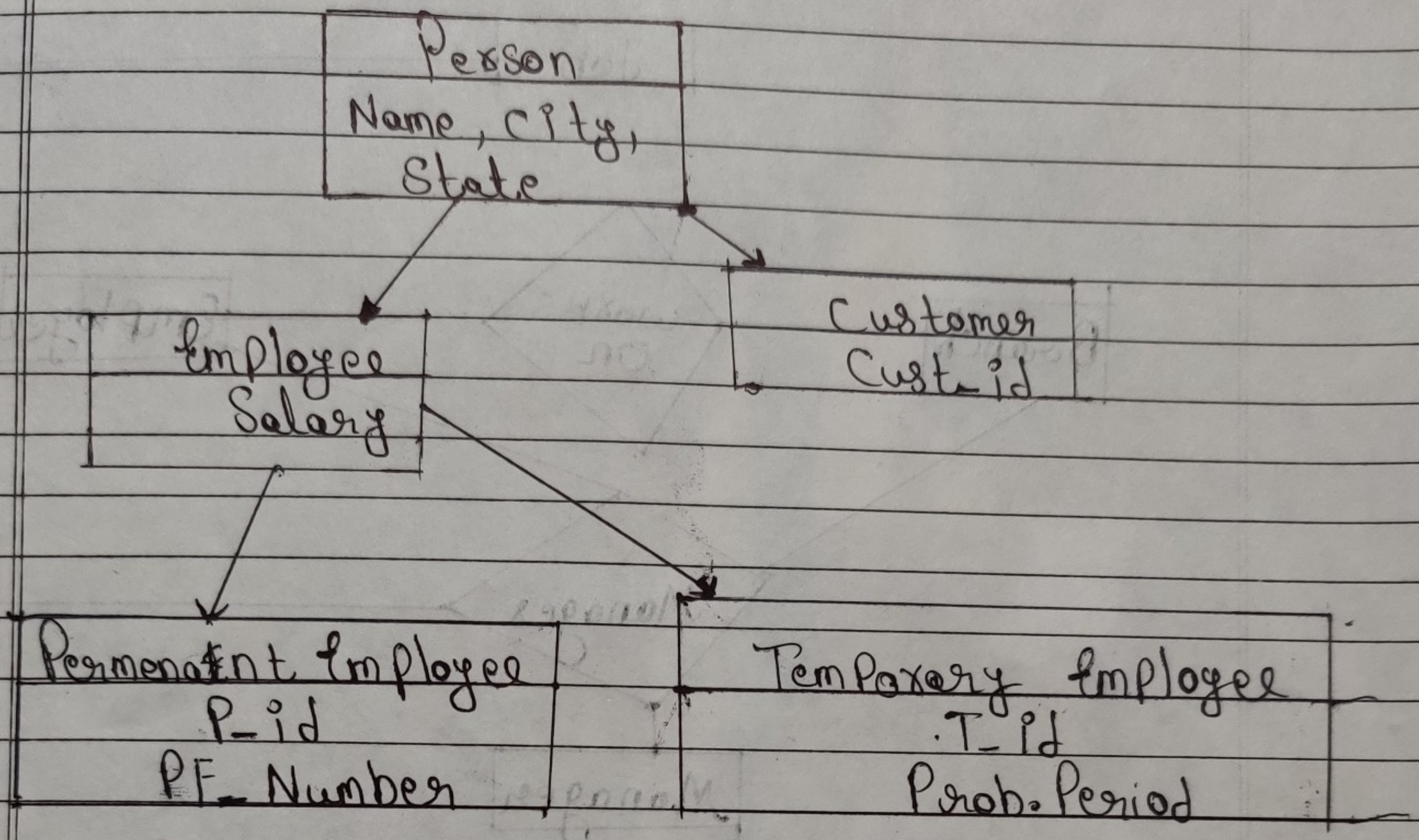
④ The higher level entity set is also known as super class while the lower level entity set also

4)) Known as Subclass.
5)) The attributes of the higher level entity sets are generally considered to be inherited from the lower level entity sets.

(B) Specialization:-

1)) Consider, an example of entity set person having attribute name, city and state. The entity person may be further classified as follows:-
- Customer
- employee

2)) Both of these types of person are described by a set of attributes that contains all the attributes of entity set person with some additional attributes of their own.
3)) In the description ~~of~~ of entities (customer), we may add new attribute like Cust_id whereas in the description of employee entities we may add new attributes like employee-id, and salary.
4)) This process of creating subgroups within an entity set is called Specialization. The Specialization of entity person helps us to distinguish whether a person is an employee or customer depending upon the attribute
5)) An entity set maybe specialized by more than one distinguishing character.



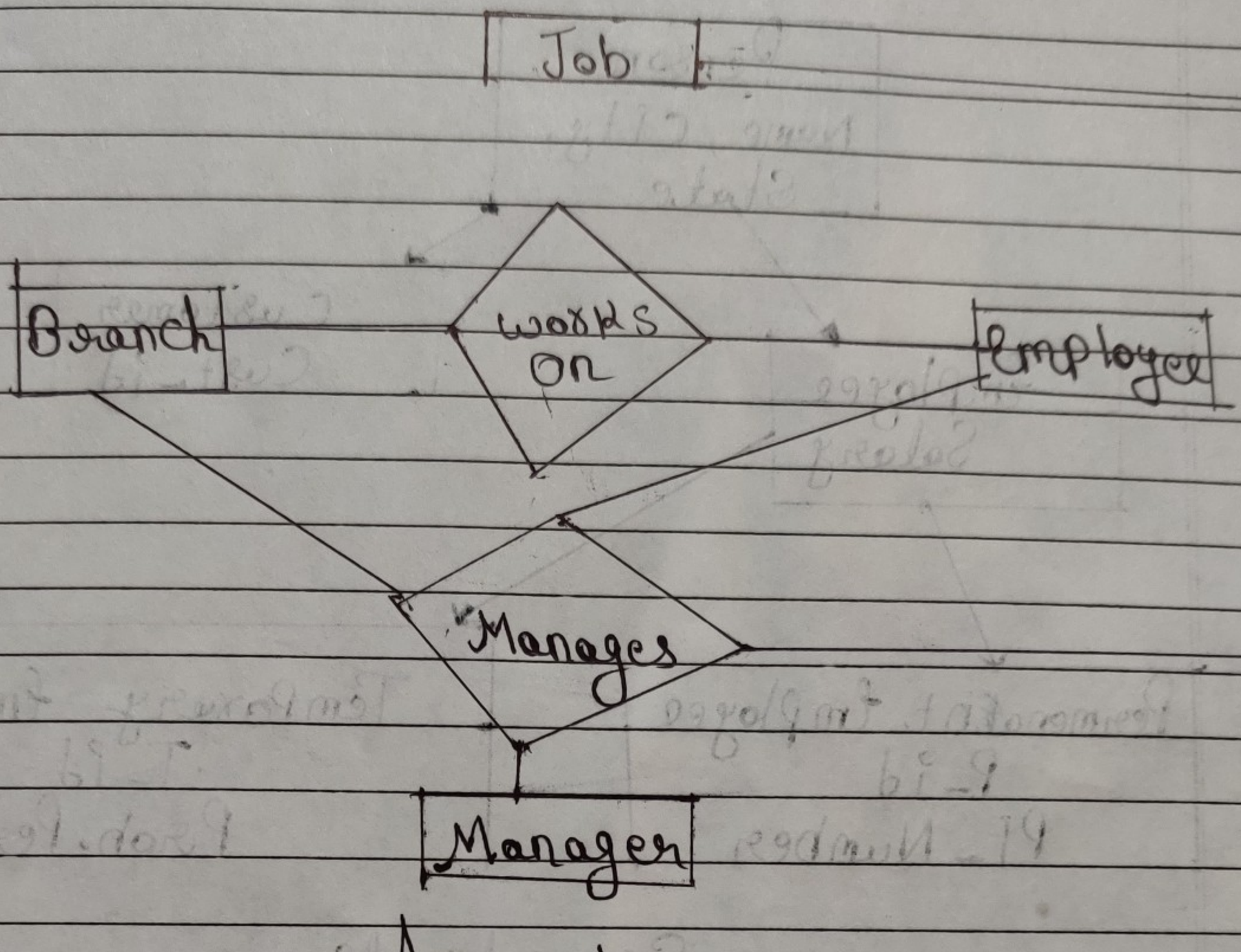
Specialization

(C) Aggregation:-

① In E-R model there is limitation i.e: it cannot express relationships among relationship. Consider, the ternary relationships 'Works-on' between an employee, branch & job.

② Consider we want to record managers for the above combination. An entity set manager is available

③ To represent this relationships, we can set a quaternary relationships manages between employee, branch, Job and manager. Using the basic E-R modelling constructs, we obtain the ~~above~~ below E-R diagram.

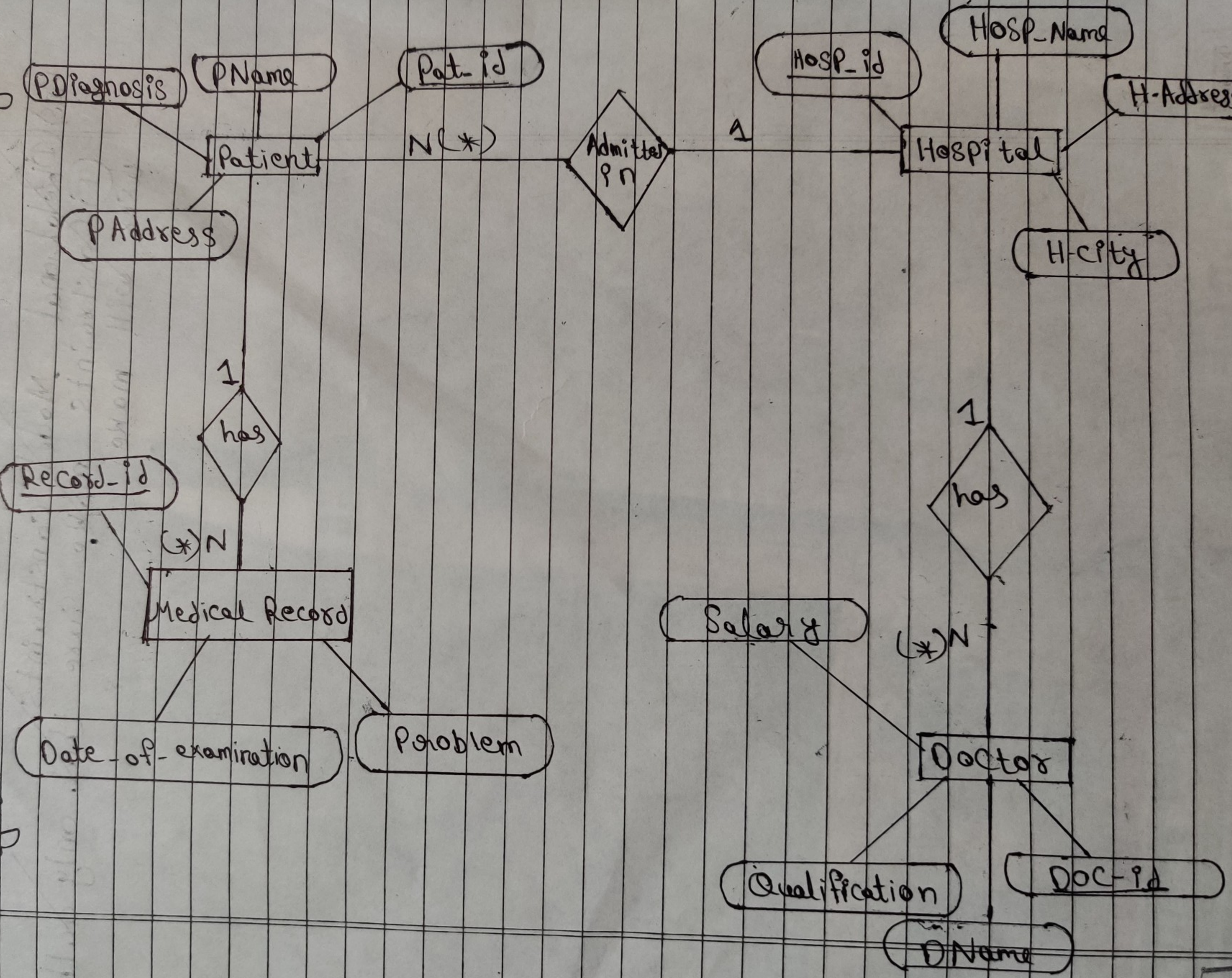


Aggregation

4)) Here, the relationship set 'works-on' and 'manages' can be combined into one single relation ship set. But Combining them is difficult, as some employee, branch, job Combination may not have a manager.

Page No. 11
Date

Q.4) Draw an E-R diagram of Hospital management System.



Q.5] Explain different types of constraints.

Ans) (I) Constraints enforce limits to the data or type data that can be inserted/deleted/updated from a table.

(II) The whole purpose of constraints is to maintain the data Integrity during an update/delete/insert queries into a table.

(III) Types of Constraints:-

(1) NOT NULL:- NOT NULL constraint makes sure that a column doesn't hold NULL value. If we don't specify this Constraint and don't provide value to particular column during insertion then by default it takes NULL.

(2) UNIQUE: UNIQUE constraint ensures that ~~to~~ two rows don't have same values for same column. Duplication of values in a column cannot be done if it is provided UNIQUE Constraint.

(3) DEFAULT: The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

(4) CHECK:- When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

(5) Key constraints:-

(i) PRIMARY KEY:- PRIMARY KEY is a Candidate Key selected by data base designer, used to differentiate between two rows of a table since, Primary key is uniquety for each entity in a entity set and it cannot be null.

(ii) FOREIGN KEY:- Foreign keys are the columns of a table that points to the unique or primary key of another table. They act as a cross-reference between tables.

(6) DOMAIN constraints:- (i) Domain constraints restrict attributes to have values only on their domains.
E.g:- Integer, float, number can only have numeric values
Varchar & varchar2 only have Strings of ≤ predefined length.

(ii) And it also restrict values on basis of constraint we have seen until now.

(7) Relationship Constraints:-

(i) Mapping cardinalities or cardinality ratio:-

(I) One-to-one:- We will consider this relation and all other cardinalities have binary relation (Includes two entity sets)

(I) one-to-one:- Let, us consider consider two entity sets A & B connected through Relation R. In one-to-one cardinality constr-aint one entity of A is related to at most one entity of B and vice-versa.

(II) one-to-many:- In this constraint one entity of A is related to one or more entities of B. one entity of B is related to atmost one entity of A.
If we reverse the meaning of above statements ~~saya~~ (inversion) we would get many-to-one constraint.

(III) many-to-many:- In this constraint one entity of A is related to one or more entities of B & vice-versa.

7.) (i) Participation constraint:- Consider, two entity sets A and B & Relation R
(I) ~~Total-Relation~~:
(I) Total-Participation:- If each entity of A is associated in a relationship R then A is said to have total-participation with R, Same goes for B.

(II) Partial-Participation:- If only some entities of A are associated in a relationship R the A is said to have partial-participation with R, Same goes for B.

Q.6) Consider the following database tables:-

Employee (empname, street, city, date_of_joining)
Work (empname, company_name, Salary)
Company (company_name, city)
Manages (empname, manager_name)

Write SQL Queries for the following Statements:

i) Modify the database so that employee "Amruta" now leave in "KonaKan"

Ans) update Employee set city = "KonaKan"
where empname = "Amruta".;

ii) Find the number of employees in each city with date_of_joining as "01-Jul-2018"

Ans) Select count(empname) from Employee
where date_of_joining = "01-Jul-2018";

iii) List name of companies starting with letter "A

Ans) ~~Select *~~
Select (company_name) from Company
where company_name like 'A%';

iv) Display empname, manager_name, street, city only for employees having manager

Ans) Select Manages.empname, Manages.manager_name, Employee.street, Employee.city
from Manages INNER JOIN Employee
On Manages.empname = Employee.empname;

v) Give total number of employee.

Ans) Select Count(empname) from Employee;