CMR INSTITUTE OF TECHNOLOGY		USN							

Internal Assesment Test – I, June 2021

Sub:	DATABASE MANAGEMENT SYSTEM							Code:	20MCA21
Date:	22-06-2021	Duration:	90 mins	Max Marks:	50	Sem:	II	Branch:	MCA

Note: Answer one full question from each part. All questions carry equal marks. Total Marks: 50

	Marks	OBE CO	OBE RBT
PART-I			
1. Explain about actors on the scene and workers behind the scene.	10	CO1	L1
(OR)			
2. List and explain the advantages of using the DBMS approach.	10	CO1	L1
PART-II			
3. Explain about data models, schemas and instances.	10	CO1	L1
(OR)			
4. Explain with necessary diagram the Database system environment architecture.	10	CO1	L1
PART-III			
5. Explain about the Centralized and client/server architecture with diagram.	10	CO1	L1
(OR)			
6. Define the following terms: (i) Join (ii) Intersect (iii) Cartesian Product (iv) Union (v) Set Difference.	10	CO2	L2
PART-IV			
7 Design an ER diagram for the following scenario: The ABC COMPANY PVT LTD., database keeps track of a company’s employees, departments, and projects. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations. A department controls a number of projects, each of which has a unique name, a unique number, and a single location. We store each employee’s name, Social Security number, address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee). We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.	10	CO2	L4
(OR)			
8 Explain about the unary relational operators with appropriate syntax and example.	10	CO2	L1
PART-V			
9 Explain the following terms with appropriate example: (i) Entity (ii) Attribute (iii) Attribute value (iv) Domain and (v) Degree	10	CO2	L1
(OR)			
10 Explain about (i) Functional dependencies and (ii) normal forms based on primary key.	10	CO2	L1

CMR Institute of Technology, Bengalore – 560 037
Department of Computer Applications
Scheme and Solutions for Internal Assessment Test – I

18MCA31 – DATABASE MANAGEMENT SYSTEM **Semester / Section: II-A/B** **Date of Test: 22-06-2021**

---

1. Explain about actors on the scene and workers behind the scene. (10)

(i) Actors on the Scene:

- the people whose jobs involve the day-to-day use of a large database

i. Database Administrators: responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed

ii. Database Designers: responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data.

- develop views of the database that meet the data and processing requirements of these groups.

iii. End Users: the people whose jobs require access to the database for querying, updating, and generating reports

- several categories of end users:
  - Casual end users: occasionally access the database, but they may need different information each time (middle or higher-level managers)
  - Naive or parametric end users: constantly querying and updating the database (bank tellers, reservation agents)
  - Sophisticated end users: include engineers, scientists, business analysts, and others who thoroughly familiarize themselves with the facilities of the DBMS in order to implement their own applications to meet their complex requirements.
  - Standalone users: maintain personal databases by using ready-made program packages that provide easy-to-use menu-based or graphics-based interfaces.

iv. System Analysts and Application Programmers

- System analysts determine the requirements of end users, especially naive and parametric end users, and develop specifications for standard canned transactions that meet these requirements.
- Application programmers implement these specifications as programs; then they test, debug, document, and maintain these canned transactions.

(ii) Workers behind the scene:

- others are associated with the design, development, and operation of the DBMS software and system environment.
- they include the following categories:

i. DBMS system designers and implementers: design and implement the DBMS modules and interfaces as a software package

ii. Tool developers: the software packages that facilitate database modeling and design, database system design, and improved performance.

iii. Operators and maintenance personnel: responsible for the actual running and maintenance of the hardware and software environment for the database system

---

2. List and explain the Advantages of DBMS approach. (10)

i. Controlling Redundancy

- redundancy in storing the same data multiple times leads to several problems
  - there is the need to perform a single logical update
  - storage space is wasted when the same data is stored repeatedly
  - Files that represent the same data may become inconsistent

ii. Restricting Unauthorized Access

- A DBMS should provide a security and authorization subsystem
- most users will not be authorized to access all information in the database

iii. Providing Persistent Storage for Program Objects

- Databases can be used to provide persistent storage for program objects and data structures

iv. Providing Storage Structures and Search Techniques for Efficient Query Processing

- the DBMS must provide specialized data structures and search techniques to speed up disk search for the desired records – index
- The DBMS often has a buffering or caching module that maintains parts of the database in main memory buffers
- The query processing and optimization module of the DBMS is responsible for choosing an efficient query execution plan for each query based on the existing storage structures.

v. Providing Backup and Recovery

- responsible for recovery
- the database is restored to the state it was in before the transaction started executing

vi. Providing Multiple User Interfaces

- a DBMS should provide a variety of user interfaces
  - query languages for casual users,
  - programming language interfaces for application programmers,
  - forms and command codes for parametric users, and
  - menu-driven interfaces and natural language interfaces for standalone users

vii. Representing Complex Relationships among Data

- A DBMS must have the capability to represent a variety of complex relationships among the data, to define new relationships as they arise, and to retrieve and update related data easily and efficiently

viii. Enforcing Integrity Constraints

- The simplest type of integrity constraint involves specifying a data type for each data item
- referential integrity constraint: specifying that a record in one file must be related to records in other files
- Primary key constraint: uniqueness on data item values

ix. Permitting inferencing and Actions Using Rules: A trigger is a form of a rule activated by updates to the table, which results in performing some additional operations to some other tables, sending messages, and so on.

x. Additional Implications of Using the Database Approach

- Potential for Enforcing Standards: The DBA can enforce standards in a centralized database environment
- Reduced Application Development Time: Development time using a DBMS is estimated to be one-sixth to one-fourth of that for a traditional file system
- Flexibility: Modern DBMSs allow certain types of evolutionary changes to the structure of the database without affecting the stored data and the existing application programs
- Availability of Up-to-Date Information: As soon as one user's update is applied to the database, all other users can immediately see this update
- Economies of Scale: reduce the wasteful overlap activities thereby reducing the overall costs of operation and management

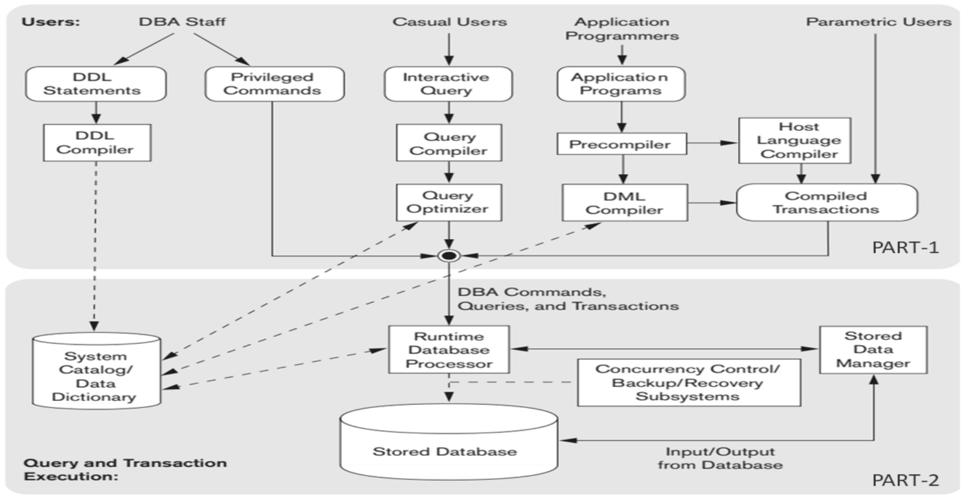
3. Explain about data models, schemas and instances. (10)

- Data model: provides the necessary means to achieve data abstraction
- Categories of Data Models:
  - High-level or Conceptual data models: provide concepts that are close to the way many users perceive data
  - Low-level or Physical data models: provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks
  - Representational (or Implementation) data models: provide concepts that may be easily understood by end users but that are not too far removed from the way data is organized in computer storage
- Conceptual data models use concepts such as entities, attributes, and relationships.
  - An entity represents a real-world object or concept (Eg. Employee)
  - An attribute represents some property of interest that further describes an entity (Eg. employee's name or salary)
  - A relationship among two or more entities represents an association among the entities
- Physical data models describe how data is stored as files in the computer by representing information such as record formats, record orderings, and access paths
- database schema: The description of a database
- database state or snapshot (also called occurrence or instance): The data in the database at a particular moment in time
  - Empty state: the database state with no data
  - Initial state: the database is first populated or loaded with the initial data
  - Current state: every time an update operation is applied to the database, we get another database state
  - valid state: a state that satisfies the structure and constraints specified in the schema

- The schema is sometimes called the intension, and a database state is called an extension of the schema
- schema evolution: changes occasionally need to be applied to the schema as the application requirements change

4. Explain with necessary diagram the Database system environment architecture. (10)

- A DBMS is a complex software system
- discuss the types of software components that constitute a DBMS and the types of computer system software with which the DBMS interacts
- two parts:
  - The top part: various users and their interfaces
  - The lower part: storage of data and processing of transactions
- Access to the disk is controlled primarily by the operating system (OS)
- buffer management module to schedule disk read/write, because this has a considerable effect on performance
- A higher-level stored data manager module: controls access to DBMS information that is stored on disk, whether it is part of the database or the catalog
- DDL Statement: used by the DBA for defining the database and tuning it
- DDL Compiler: processes schema definitions and stores descriptions of the schemas in the DBMS catalog
- Privileged commands: The commands which are exclusively used by the DBA
- Interactive query: The interface by which the casual users and persons with occasional need for information from the database interact
- Query compiler: Queries are parsed and validated for correctness of the query syntax
- Query optimizer: Responsible for optimising the query and its execution



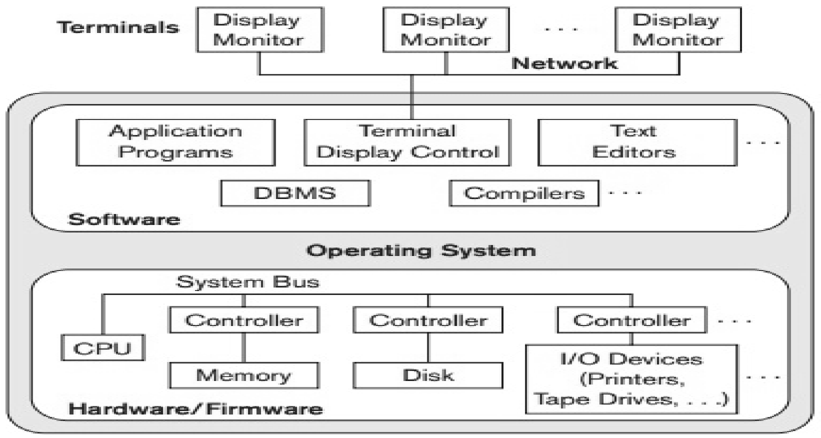
Part-1: users and their interfaces Part-2: Internals of the DBMS

- Pre-compiler: Extracts DML commands from an application program written in a host programming language
- DML compiler: Compilation of pre-compiled DMLC commands into object code for database access

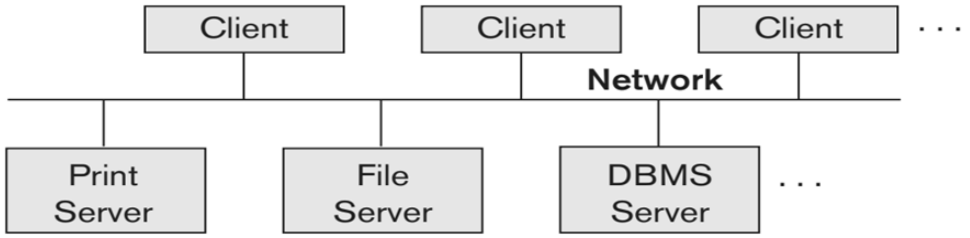
- Host language compiler: The rest of the program is sent to the host language compiler.
- Compiled transactions: Canned transactions are executed repeatedly by parametric users, who simply supply the parameters to the transactions

5. Explain about the Centralized and client/server architecture (10) with diagram.

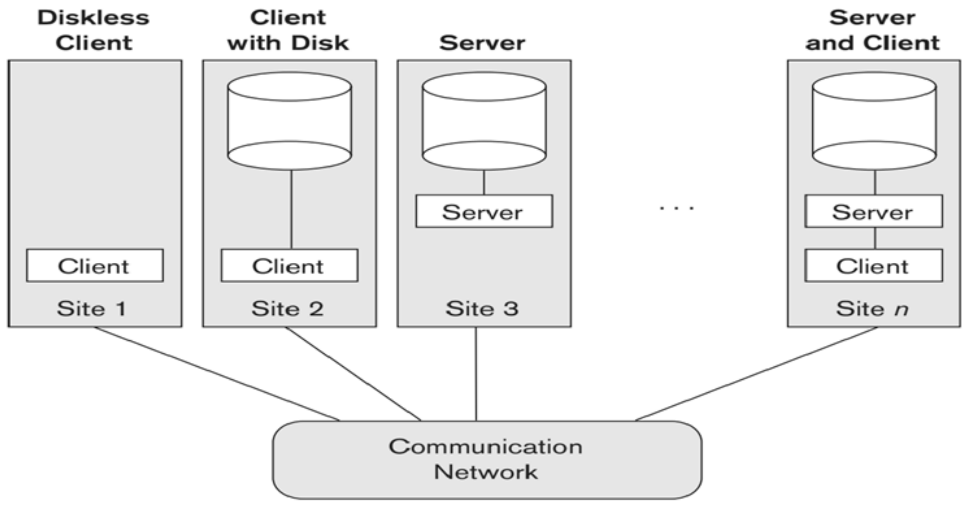
- Earlier architectures used mainframe computers to provide the main processing for all system functions, including user application programs and user interface programs, as well as all the DBMS functionality
- centralized DBMS in which all the DBMS functionality, application program execution, and user interface processing were carried out on one machine
- all processing was performed remotely on the computer system
- only display information and controls were sent from the computer to the display terminals



- Basic Client/Server Architectures
- define specialized servers with specific functionalities
- A client is typically a user machine that provides user interface capabilities and local processing
- A server is a system containing both hardware and software that can provide services to the client machines, such as file access, printing, archiving, or database access
- The client/server architecture was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, data base servers, Web servers, e-mail servers, and other software and equipment are connected via a network



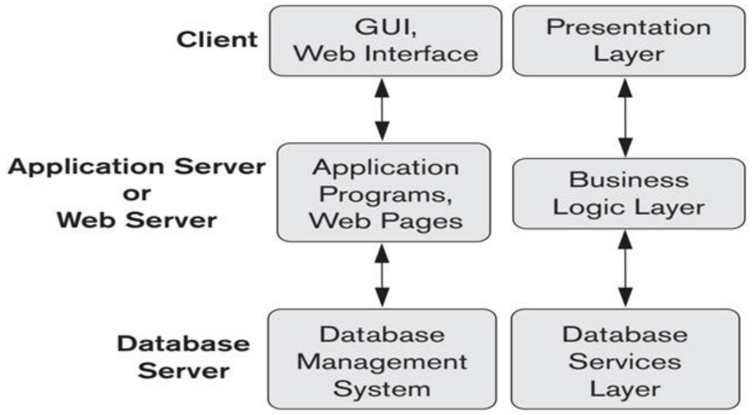
Logical two-tier client/server architecture



Physical two-tier client/server architecture

- Two main types of basic DBMS architectures were created on this underlying client/server framework: two-tier and three-tier
- Two-Tier Client/Server Architectures for DBMSs:
- the software components are distributed over two systems: client and server
  - client side: has the user interface and application programs
  - server side: the query and transaction functionality related to SQL processing
- The different approach to two-tier client/server architecture was taken by some object-oriented DBMSs
- the client level: handle the user interface; data dictionary functions; DBMS interactions with programming language compilers; global query optimization, concurrency control, and recovery across multiple servers; structuring of complex objects from the data in the buffers; and other such functions
- the server level: the DBMS software responsible for handling data storage on disk pages, local concurrency control and recovery, buffering and caching of disk pages, and other such functions
- Advantages: simplicity and seamless compatibility with existing systems
- Three-Tier and n-Tier Architectures for Web Applications
- The emergence of the Web changed the roles of clients and servers, leading to the three-tier architecture
- adds an intermediate layer (middle tier – application layer or web server) between the client and the database server
- runs the application programs and storing business rules (procedures or constraints) that are used to access data from the database server

- The presentation layer: displays information to the user and allows data entry.
- The business logic layer: handles intermediate rules and constraints before data is passed up to the user or down to the DBMS.
- The middle layer can also act as a Webserver, which retrieves query results from the database server and formats them into dynamic Web pages that are viewed by the Web browser at the client side
- The bottom layer includes all data management services.



6. Define the following terms: (i) Join (ii) Division (iii) Cartesian Product (iv) Union (v) Set Difference. **(10)**

(i) Join: Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition

(ii) Intersect: returns all tuples that are in both R and S.

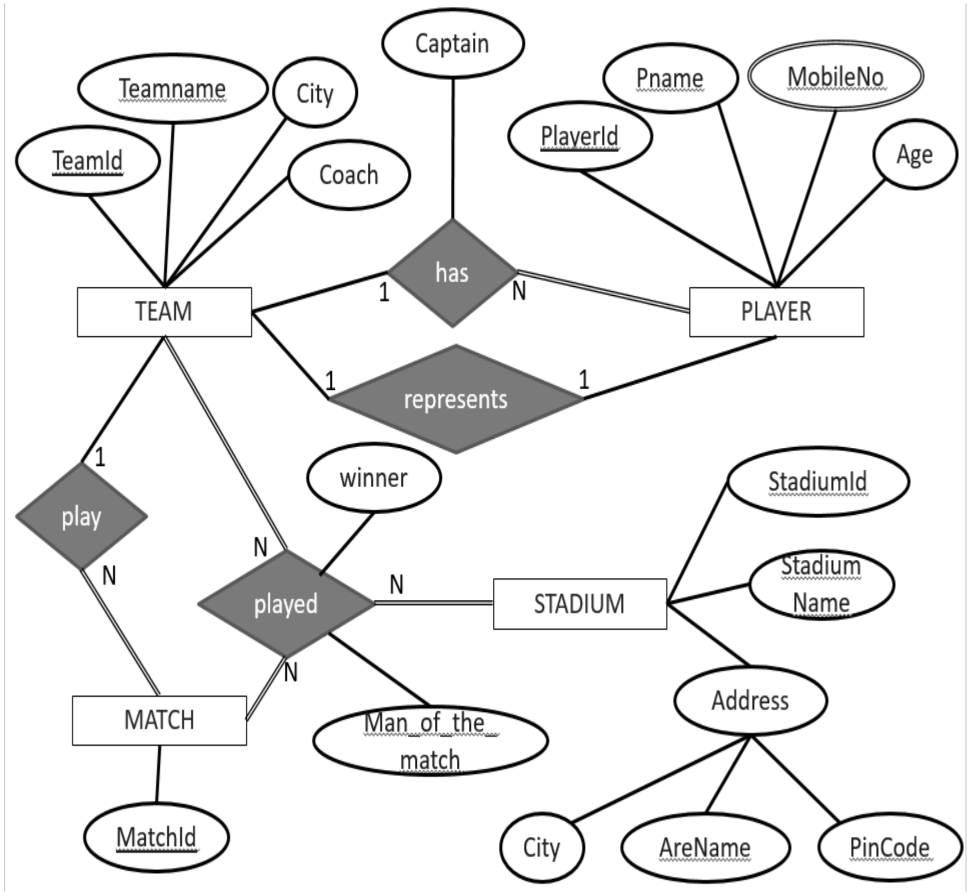
(iii) Cartesian Product: Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2

(iv) Union: Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible

(v) Set Difference: Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible

7. **Design an ER diagram for the following scenario.** **(10)**
   The ABC COMPANY PVT LTD., database keeps track of a company's employees, departments, and projects. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations. A department controls a number of projects, each of which has a unique name, a unique number, and a single location. We store each employee's name, Social Security number, address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who

is another employee). We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.



8. Explain about the unary relational operators with appropriate syntax and example. (10)

(i) The SELECT Operation (σ)

- used to choose a subset of the tuples from a relation that satisfies a selection condition
- the SELECT operation is denoted by: $\sigma_{<\text{selection condition}>}(R)$
  - Eg.: $\sigma_{\text{salary}>30000}$ (EMPLOYEE)
- The Boolean expression specified in <selection condition> is made up of a number of clauses of the form:
  - <attribute name> <comparison op> <constant value> or
  - <attribute name> <comparison op> <attribute name>
- (Eg.) $\sigma_{(\text{Dno}=4 \text{ AND Salary}>25000) \text{ OR } (\text{Dno}=5 \text{ AND Salary}>30000)}$(EMPLOYEE)
- The Boolean conditions AND, OR, and NOT have their normal interpretation, as follows:
  - (cond1 AND cond2) is TRUE if both (cond1) and (cond2) are TRUE; otherwise, it is FALSE.
  - (cond1 OR cond2) is TRUE if either (cond1) or (cond2) or both are TRUE; otherwise, it is FALSE.
  - (NOT cond) is TRUE if cond is FALSE; otherwise, it is FALSE.
- a sequence of SELECTs can be applied in any order.
- In addition, we can always combine a cascade (or sequence) of SELECT operations into a single SELECT operation with a conjunctive (AND) condition;
- that is,

$\sigma_{<cond1>}(\sigma_{<cond2>}(...(\sigma_{<condn>}(R)) ...))=\sigma_{<cond1>\ AND<cond2>\ AND...AND\ <condn>}(R)$

(ii) The PROJECT operation (π)

- selects certain columns from the table and discards the other columns
- the result of the PROJECT operation can be visualized as a vertical partition of the relation into two relations:
  - one has the needed columns (attributes) and contains the result of the operation, and
  - the other contains the discarded columns
- The general form of the PROJECT operation is: $\pi_{<attribute\ list>}(R)$
  - Eg.: $\pi_{Lname, Fname, Salary}$ (EMPLOYEE)
- The PROJECT operation removes any duplicate tuples, so the result of the PROJECT operation is a set of distinct tuples (duplicate elimination)
- $\pi_{Sex, Salary}$(EMPLOYEE) is equivalent to

select DISTINCT sex, salary from EMPLOYEE;

9. Explain the following terms with appropriate example: (i) Entity (ii) Attribute (iii) Attribute value (iv) Domain and (v) Degree. (10)

(i) Entity: The basic object that the ER model represents is an entity, which is a thing in the real world with an independent existence.

- An entity may be an object with a physical existence (for example, a particular person, car, house, or employee) or
- it may be an object with a conceptual existence (for instance, a company, a job, or a university course).

(ii) Attribute: the particular properties that describe an entity.

(For example, an EMPLOYEE entity may be described by the employee's name, age, address, salary, and job.)

(iii) Attribute value: A particular entity will have a value for each of its attributes.

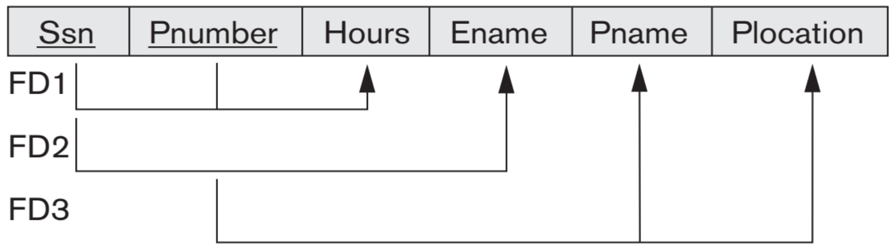
(iv) Domain: a set of permitted values of an attribute.

(v) Degree: the number of attributes in a relation.

10. Explain about (i) Functional dependencies and (ii) normal forms based on primary key. (10)

(i) Functional Dependency:

- a formal tool for analysis of relational schemas that enables to detect and describe some of the problems in precise terms
- A functional dependency (denoted by X → Y) between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation state r of R. The constraint is that, for any two tuples t1 and t2 in r that have t1[X] = t2[X], they must also have t1[Y] = t2[Y]
- The values of Y component of a tuple in r depend on, or determined by, the values of the X component. There is a functional dependency from X to Y, or that Y is functionally dependent on X.



- Ssn → Ename
- Pnumber → Pname, Plocation
- {Ssn, Pnumber} → Hours

(ii) Normal forms based on primary key:

- Normalization of data can be considered a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of:

(1) minimizing redundancy and

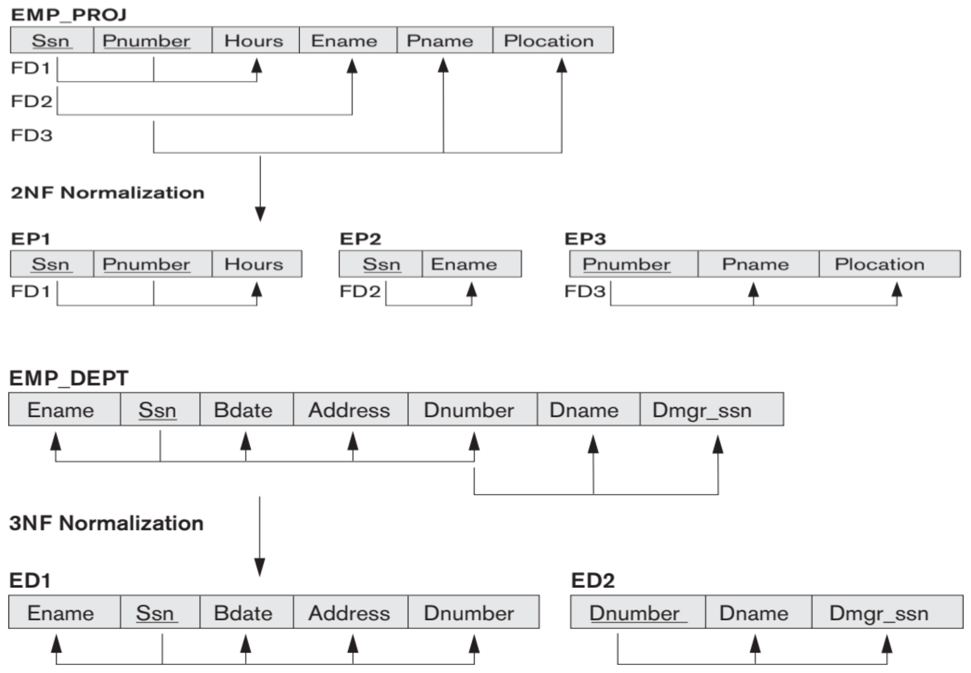
(2) minimizing the insertion, deletion, and update anomalies

Definition: The normal form of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized

- First Normal Form (1NF)

Definition: The relation is said to be in 1NF if all the attributes in a relation must have atomic domains

- Second Normal Form (2NF): based on functional dependency
- Definition: A relation schema R is in 2NF if every non-prime attribute A in R is fully functionally dependent on the primary key of R
- Third Normal Form (3NF): based on the concept of transitive dependency
- Definition: a relation schema R is in 3NF if it satisfies 2NF and no non-prime attribute of R is transitively dependent on the primary key


EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3
2NF Normalization
EP1
EP2
EP3
FD1
FD2
FD3
EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
3NF Normalization
ED1
ED2