CAT - 1

Programme	:	**B. Tech**			
Course Title	:	**Database Management Systems**	Code	:	**CSE2007**
Time	:	**01:30 Hours**	Max. Marks	:	**50**

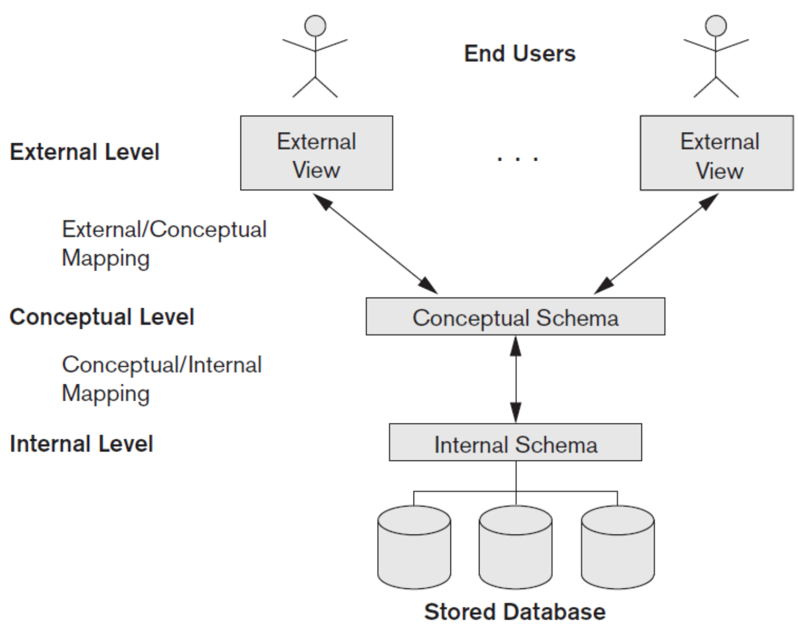
Part A- Answer all the Questions (5 x 10= 50 marks)

1. Describe the three-schema architecture. Why do we need mappings among schema levels? How do different schema definition languages support this architecture? **10**

Answer:

The Three-Schema Architecture

The goal of the three-schema architecture, illustrated in Figure, is to separate the user applications from the physical database. In this architecture, schemas can be defined at the following three levels:



1. The **internal level** has an **internal schema**, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
2. The **conceptual level** has a **conceptual schema**, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This

implementation conceptual schema is often based on a *conceptual schema design* in a high-level data model.

3. The **external** or **view level** includes a number of **external schemas** or **user views**. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level conceptual data model.

The three-schema architecture is a convenient tool with which the user can visualize the schema levels in a database system. Most DBMSs do not separate the three levels completely and explicitly, but they support the three-schema architecture to some extent. Some older DBMSs may include physical-level details in the conceptual

schema. The three-level ANSI architecture has an important place in database technology development because it clearly separates the users' external level, the database's conceptual level, and the internal storage level for designing a database. It is very much applicable in the design of DBMSs, even today. In most DBMSs that support user views, external schemas are specified in the same data model that describes the conceptual-level information (for example, a relational DBMS like Oracle or SQLServer uses SQL for this).

Notice that the three schemas are only *descriptions* of data; the actual data is stored at the physical level only. In the three-schema architecture, each user group refers to its own external schema. Hence, the DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database. If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view. The processes of transforming requests and results between levels are called **mappings**. These mappings may be time-consuming, so some DBMSs—especially those that are meant to support small databases—do not support external views. Even in such systems, however, it is necessary to transform requests between the conceptual and internal levels.

Data Independence

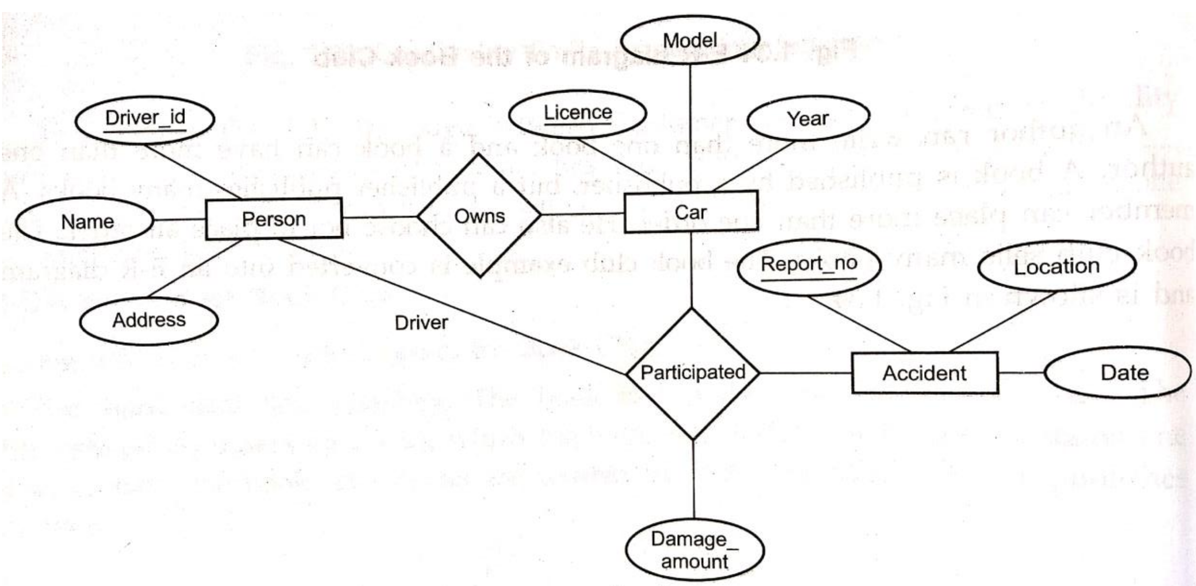
The three-schema architecture can be used to further explain the concept of **data independence**, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. We can define two types of data independence:

1. Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). In the last case, external schemas that refer only to the remaining data should not be affected. For example, the external schema of Figure 1.5(a) should not be affected by changing the GRADE_REPORT file (or record type) shown in Figure 1.2 into the one shown in Figure 1.6(a). Only the view definition and the mappings need to be changed in a DBMS that supports logical data independence. After the conceptual schema undergoes a logical reorganization, application programs that reference the external schema constructs must work as before. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

2. Physical data independence is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schemas need not be changed as well. Changes to the internal schema may be needed because some physical files were reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema. For example, providing an access path to improve retrieval speed of SECTION records (Figure) by semester and year should not require a query such as *list all sections offered in fall 2008* to be changed, although the query would be executed more efficiently by the DBMS by utilizing the new access path.

2. Construct the ER diagram for the given scenario: **10**

Consider a car-insurance company whose customers own one or more cars each. Each car has unique license no, model name and year in which the car is introduced. Each customer has name, unique driver ID and address. If the customer participated in the car accident, the car-insurance company collect certain details such unique report number, location, date and damage amount for every car accident.



3. Consider the following relation **10**

Relation: Employee

EMP_ID	EMP_SSN	EMP_NAME	EMP_DEPT	EMP_DESIGNATION
EMP7001	74634	Codd	HR	Manager
EMP7002	54354	Charles	Finance	Manager
EMP7003	34254	Azar	HR	Executive
EMP7004	64342	Samson	Development	Developer
EMP7005	53246	Thilak	Finance	Executive
EMP7006	74612	Shruthi	HR	Executive

Identify the following and justify your answer

1) candidate key
2) primary key
3) super key
4) alternate key
5) composite key

Identify the following and justify your answer

1) candidate key
   Candidate Key: are individual columns in a table that qualifies for uniqueness of all the rows. Here in Employee table EMP_ID & EMP_SSN are Candidate keys.

2) primary key
   Primary Key: is the columns you choose to maintain uniqueness in a table. Here in Employee table you can choose either EMP_ID or EMP_SSN columns, EMP_ID is preferable choice, as EMP_SSN is a secure value.

3) super key
   Super Key: If you add any other column/attribute to a Primary Key then it become a super key, like EMP_ID + EMP_NAME is a Super Key.

4) alternate key
   Alternate Key: Candidate column other the Primary column, like if EMP_ID is PK then EMP_SSN would be the Alternate key.

5) composite key
   Composite Key: If a table do have a single columns that qualifies for a Candidate key, then you have to select 2 or more columns to make a row unique. Like if there is no EMP_ID or EMP_SSN columns, then you can make EMP_NAME + EMP_DOB as Composite primary Key. But still there can be a narrow chance of duplicate row.

4. Consider the two relations given below **10**

Employee:

Emp_name	Street	City
Ram	Civil line	Mumbai
Shyam	Park street	Kolkata
Ravi	M.g. Street	Delhi
Hari	Nehru nagar	Hyderabad

Fact_workers:

Emp_name	Branch	Salary
Ram	Infosys	10000
Shyam	Wipro	20000
Kuber	HCL	30000
Hari	TCS	50000

Find the following:

a) $\pi_{(\mathrm{Emp_{name}},\mathrm{Salary})}(\mathrm{EMPLOYEE} \bowtie \mathrm{FACT\_WORKERS})$

b) $\sigma_{(\mathrm{Salary}<30000)}(\mathrm{EMPLOYEE} ⟕ \mathrm{FACT\_WORKERS})$

Answer:

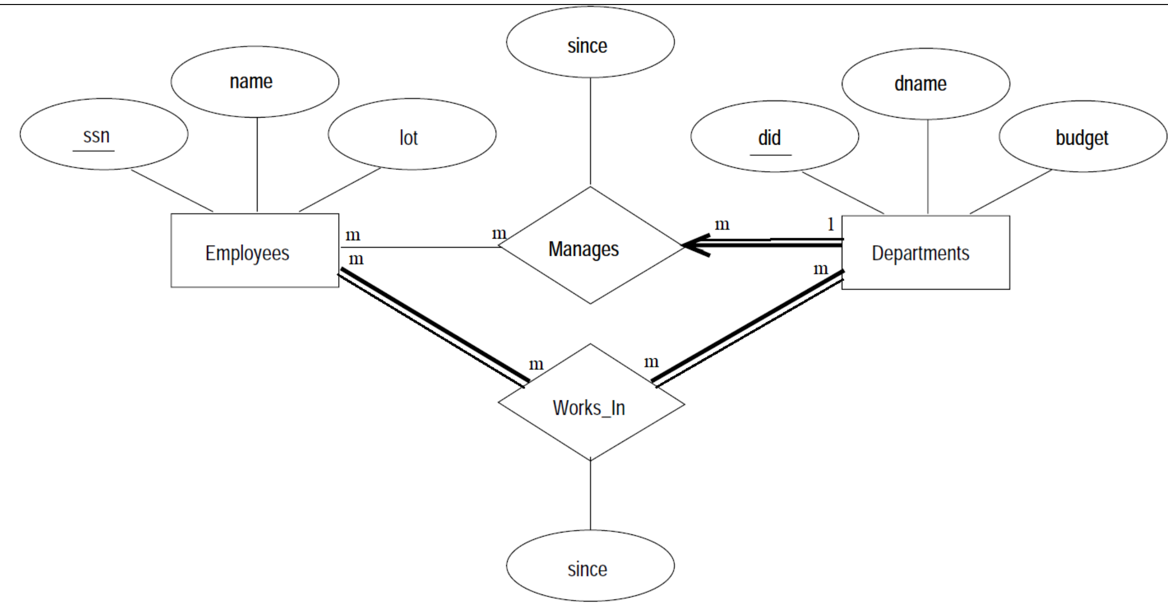
a) $\pi_{(\mathrm{Emp_{name}},\mathrm{Salary})}(\mathrm{EMPLOYEE} \bowtie \mathrm{FACT\_WORKERS})$

Emp_name	Salary
Ram	10000
Shyam	20000
Hari	50000

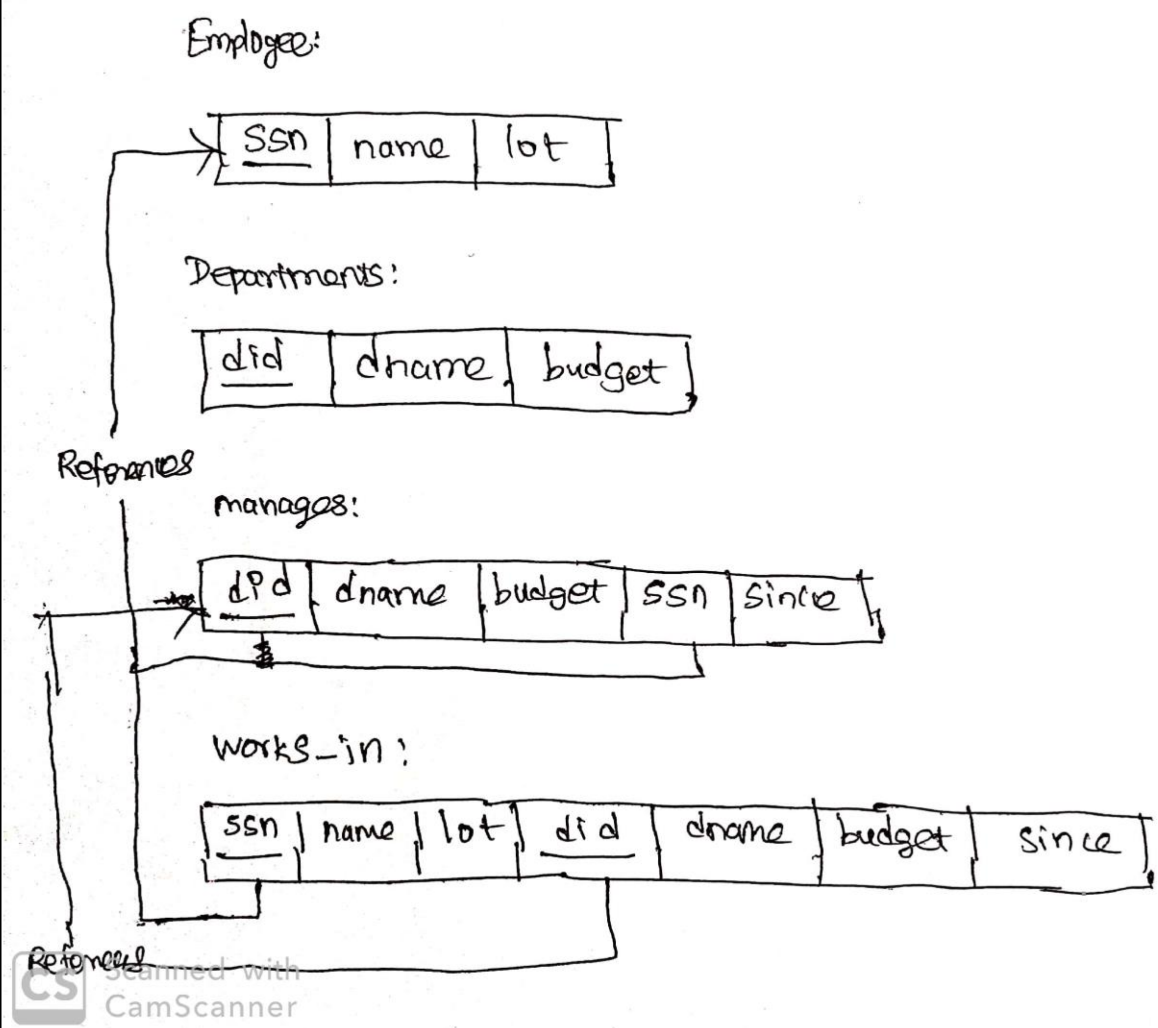
b) $\sigma_{(\mathrm{Salary}<30000)}(\mathrm{EMPLOYEE} ⟕ \mathrm{FACT\_WORKERS})$

Emp_name	Street	City	Emp_name	Branch	Salary
Ram	Civil line	Mumbai	Ram	Infosys	10000
Shyam	Park street	Kolkata	Shyam	Wipro	20000
Ravi	M.g. Street	Delhi	-	-	-

5.	Construct a Relational model for the given ER model	**10**



Answer:



Note:

- Mentioning domain (data type) and create query is optional.

	• if the students written only the create query for the above schema full marks can be awarded.	