**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Subject Name : RDBMS-II**

**Subject Code : BTCS-602**

**Very Short Question Answers**

**1.What is Database Management System? Why do we need a DBMS ?**
DBMS is a collection of interrelated data and a set of programs to access those data. It is to provide a way to store and retrieve information that is both convenient and efficient.

**2.List any two advantages of database systems.**
Security, Integrity, Atomicity, Concurrent access anomalies

**3.What are the limitations of file system ?**
Data redundancy and inconsistency, Difficulty in access data, data isolation, integrity, atomicity, concurrent access anomalies.

**4.Define Data Dictionary?**
A data dictionary is a data structure which stores meta data about the structure of the database ie. the schema of the database.

**5.Define Data independence.**
Application programs are said to exhibit physical data independence if they do not depend on the physical schema and thus need not be rewritten if the physical schema changes.

**6.What is logical data Independence ?**
Application programs are said to exhibit physical data independence if they do not depend on the physical schema and thus need not be rewritten if the physical schema changes.

**7. List any eight applications of DBMS.**
a) Banking
b) Airlines
c) Universities
d) Credit card transactions
e) Tele communication
f) Finance
g) Sales
h) Manufacturing
i) Human resources

**8.What are the advantages of DBMS?.**
a) Controlling redundancy
b) Restricting unauthorized access
c) Providing multiple user interfaces
d) Enforcing integrity constraints.
e) Providing back up and recovery

**9.Give the levels of data abstraction?**
a) Physical level
b) logical level
c) view level

**10.Define instance and schema?**
**Instance:** Collection of data stored in the data base at a particular moment is called an Instance of the database.

**Schema:** The overall design of the data base is called the data base schema.

## Short Question Answers

**Q 1 What are the three levels of data abstraction ?**

**Ans1** a) Physical level b) logical level c) view level

Data abstraction is a process of representing the essential features without including implementation details. many database-systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interactions with the system:

**1) Physical level.**

The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.

**2) Logical level.**

The next-higher level of abstraction describes what data are stored in the database, and what relationships exist among those data. The logical level thus describes the entire database in terms of a small number of relatively simple structures.

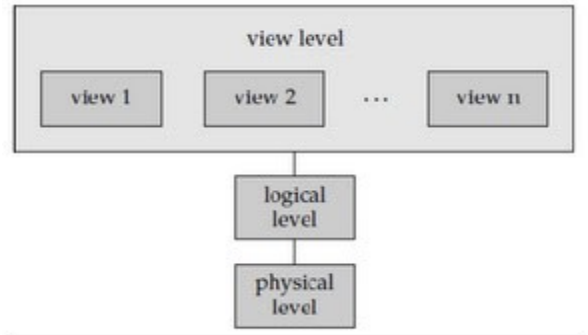



**Figure 3 : Three Levels of Data Abstractions**

**3) View level**.

The highest level of abstraction describes only part of the entire database. The variety of information stored in a large database. Many users of the database system do not need all this information; instead, they need to access only a part of the database. The view level of abstraction exists to simplify their interaction with the system.

**Q 2 What are Data Models ?**

**Ans 2** A data model can be thought of as a diagram or flowchart that illustrates the relationships between data. Although capturing all the possible relationships in a data model can be very time-intensive, it's an important step and shouldn't be rushed. Well-documented models allow stake-holders to identify errors and make changes *before* any programming code has been written. Data modelers often use multiple models to view the same data and ensure that all processes, entities, relationships and data flows have been identified. There are several different approaches to data modeling, including:

**Conceptual Data Modeling** - identifies the highest-level relationships between different entities.

**Enterprise Data Modeling** - similar to conceptual data modeling, but addresses the unique requirements of a specific business.

**Logical Data Modeling** - illustrates the specific entities, attributes and relationships involved in a business function. Serves as the basis for the creation of the physical data model.

**Physical Data Modeling** - represents an application and database-specific implementation of a logical data model.

**Q 3 What are the disadvantages of DBMS ?**

Ans 3 **Although there are many advantages of DBMS, the DBMS may also have some minor disadvantages. These are:**

- **Cost of Hardware and Software**

A processor with high speed of data processing and memory of large size is required to run the DBMS software. It means that you have to up grade the hardware used for file-based system. Similarly, DBMS software is also very costly,.

- **Cost of Data Conversion**

When a computer file-based system is replaced with database system, the data stored into data file must be converted to database file. It is very difficult and costly method to convert data of data file into database. You have to hire database system designers along with application programmers. Alternatively, you have to take the services of some software house. So a lot of money has to be paid for developing software.

- **Cost of Staff Training**

Most database management system are often complex systems so the training for users to use the DBMS is required. Training is required at all levels, including programming, application development, and database administration. The organization has to be paid a lot of amount for the training of staff to run the DBMS.

- **Appointing Technical Staff**

The trained technical persons such as database administrator, application programmers, data entry operations etc. are required to handle the DBMS. You have to pay handsome salaries to these persons. Therefore, the system cost increases.

- **Database Damage**

In most of the organization, all data is integrated into a single database. If database is damaged due to electric failure or database is corrupted on the storage media, the your valuable data may be lost forever.

**Q 4 What is the Entity –Relationship model ?**

Ans 4 An ER model is an abstract way of describing a database. In the case of a relational database, which stores data in tables, some of the data in these tables point to data in other tables - for instance, your entry in the database could point to several entries for each of the phone numbers that are yours. The ER model would say that you are an entity, and each phone number is an entity, and the relationship between you and the phone numbers is 'has a phone number'. Diagrams created to design these entities and relationships are called entity–relationship diagrams or ER diagrams.

**Entity**: an object that is involved in the enterprise and that be distinguished from other objects.

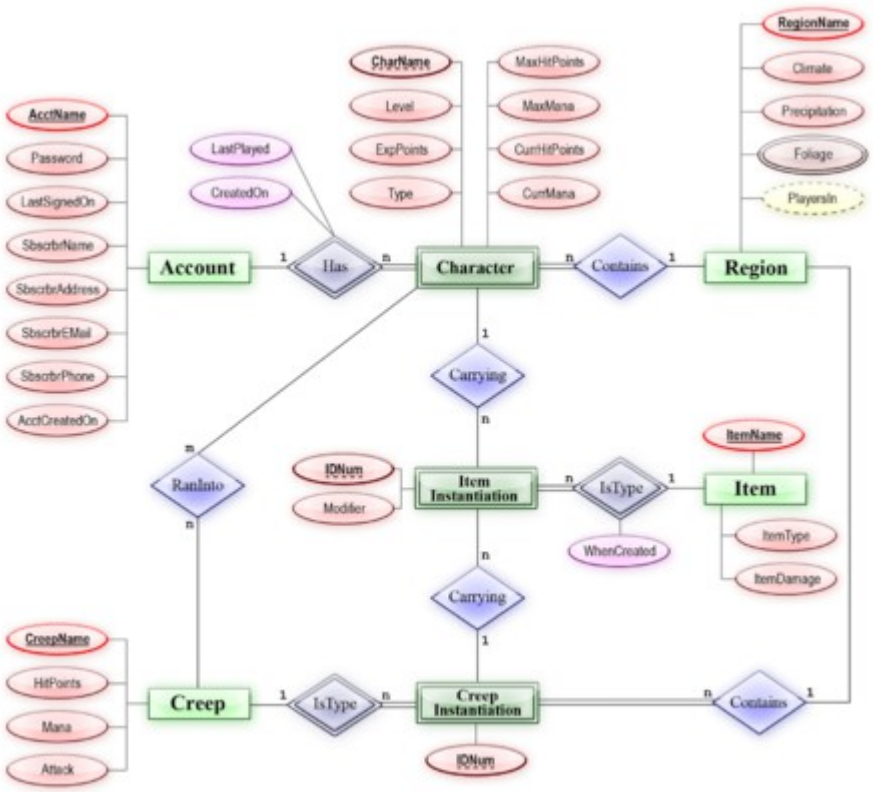
- Can be person, place, event, object, concept in the real world
- Can be physical object or abstraction
- Ex: "John", "CSE305"

**Entity Type**: set of similar objects or a category of entities; they are well defined

- A rectangle represents an entity set
- Ex: ***students***, ***courses***
- We often just say "entity" and mean "entity type"

**Attribute**: describes one aspect of an entity type; usually [and best when] single valued and indivisible (atomic)

- Represented by oval on E-R diagram
- Ex: *name, maximum enrollment*
- May be **multi-valued** – use double oval on E-R diagram
- May be **composite** – attribute has further structure; also use oval for composite attribute, with ovals for components connected to it by lines
- May be **derived** – a virtual attribute, one that is computable from existing data in the database, use dashed oval. This helps reduce redundancy




**ER Diagram**

**Q5 What are integrity constraints ?**

**Ans 5** There are the domain integrity, the entity integrity, the referential integrity and the foreign key integrity constraints.

**Domain Integrity**

Domain integrity means the definition of a valid set of values for an attribute. You define

- data type,
- lenght or size
- is null value allowed

- is the value unique or not
for an attribute.
You may also define the default value, the range (values in between) and/or specific values for the attribute. Some DBMS allow you to define the output format and/or input mask for the attribute. These definitions ensure that a specific attribute will have a right and proper value in the database.

**Entity Integrity Constraint**
The entity integrity constraint states that primary keys can't be null. There must be a proper value in the primary key field. This is because the primary key value is used to identify individual rows in a table. If there were null values for primary keys, it would mean that we could not identify those rows. On the other hand, there can be null values other than primary key fields. Null value means that one doesn't know the value for that field. Null value is different from zero value or space. The entity integrity constraints assure that a spesific row in a table can be identified.

**Referential Integrity Constraint**
The referential integrity constraint is specified between two tables and it is used to maintain the consistency among rows between the two tables.
The rules are:
1. You can't delete a record from a primary table if matching records exist in a related table.
2. You can't change a primary key value in the primary table if that record has related records.
3. You can't enter a value in the foreign key field of the related table that doesn't exist in the primary key of the primary table.
4. However, you can enter a Null value in the foreign key, specifying that the records are unrelated.

**Foreign Key Integrity Constraint**
There are two foreign key integrity constraints: cascade update related fields and cascade delete related rows. These constraints affect the referential integrity constraint.

**Cascade Update Related Fields**
Any time you change the primary key of a row in the primary table, the foreign key values are updated in the matching rows in the related table. This constraint overrules rule 2 in the referential integrity constraints.

## Difficult Terminologies

1. **Instance**: Collection of data stored in the data base at a particular moment is called an Instance of the database.
2. **Schema**: The overall design of the data base is called the data base schema. The schemas at the view level are called subschemas that describe different views of the database.
3. **Cardinality of Relationship**: Cardinality is the number of entity instances to which another entity set can map under the relationship.
4. **Entity type**: An entity type defines a collection of entities that have the same attributes.
5. **Entity set:** The set of all entities of the same type is termed as an entity set.
6. **Key :** A key is a field that you use to sort data. It can also be called a *key field* , *sort key*, *index,* or *key word.*
7. **Super Key** : A ***super key*** is a set of one or more attributes that collectively allows us to identify uniquely an entity in the entity set.
8. **Degree of relationship** : The number of roles in the relationship.
9. **Composite key**: a key requiring more than one attribute
10. **Candidate key**: a superkey such that no proper subset of its attributes is also a superkey (minimal superkey – has no unnecessary attributes).