Notes of DBMS for BSC (HONS) part-III

Syllabus :(Group – D) Data Base Management System : Basic concept, File Management Systems, Advantages of DBMS, ANSI/SPARC Architecture, Physical, Conceptual and External Models, ER Diagram; Data Models: Relational, Hierarchical. Network; File Organization: Sequential, Indexed .Sequential, Random, Inverted; Query Languages, Relational Algebra, Relational Calculus, Functional Dependencies, Normal Forms: INF, 2NF, 3NF and BCNF; Structured Query Languages (SQL), elementary concepts of Security, Integrity.

Q :Define Database.

Ans: A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

In computing, databases are sometimes classified according to their organizational approach. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

Computer databases typically contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles. Typically, a database manager provides users the capabilities of controlling read/write access, specifying report generation, and analyzing usage.

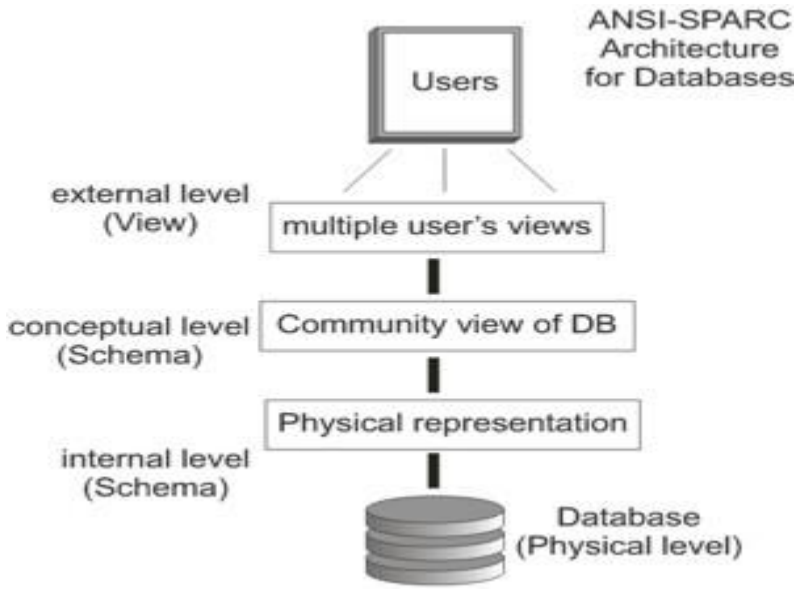
Q: Describe briefly Three-level ANSI SPARC Architecture

The Architecture of most of commercial dbms are available today is mostly based on this ANSI-SPARC database architecture .
ANSI SPARC THREE-TIER architecture has main three levels:

1. **Internal Level**
2. **Conceptual Level**
3. **External Level**

These three levels provide data abstraction ;means hide the low level complexities from end users .
A database system should be efficient in performance and convenient in use.

Using these three levels,it is possible to use complex structures at internal level for efficient operations and to provide simpler convenient interface at external level.



1. Internal level:

- This is the lowest level of data abstraction.
- It describes How the data are actually stored on storage devices.
- It is also known as physical level.
- It provides internal view of physical storage of data.
- It deals with complex low level data structures,file structures and access methods in detail.
- It also deals with Data Compression and Encryption techniques,if used.

2. Conceptual level:

- This is the next higher level than internal level of data abstraction.
- It describes What data are stored in the database and What relationships exist among those data.
- It is also known as Logical level.
- It hides low level complexities of physical storage.
- Database administrator and designers work at this level to determine What data to keep in database.
- Application developers also work on this level.

3. External Level:

- This is the highest level of data abstraction.
- It describes only part of the entire database that a end user concern.
- It is also known as an view level.
- End users need to access only part of the database rather than entire database.
- Different user need different views of database.And so,there can be many view level abstractions of the same database.

Advantages of Three-tier Architecture:

- The main objective of it is to provide data abstraction.
- Same data can be accessed by different users with different customized views.
- The user is not concerned about the physical data storage details.
- Physical storage structure can be changed without requiring changes in internal structure of the database as well as users view.
- Conceptual structure of the database can be changed without affecting end users.

Q: What is an Attribute?

Ans: An attribute in a table is a named column or they are the set of important properties which describes the particular entity. An attribute may consist of name, roll number, age etc. Relations are used to hold information about the object. The attributes in a relation can appear in any order and the relation is the same relation and hence conveys the same meaning. Attributes are set of qualities of qualities which are used to identify a particular entity.

Q: What are Derived Attributes?

Ans: Derived attributes are those attributes which are based on and are derived from the attributes of another table or a relation. The derived attributes may contain new values or the values from the base table from which it was derived. Derived attributes are effectively read-only since there is no place to write them back to. Also, because derived attributes don't directly point to anything in the database, they cannot be used as primary keys. For example: a derived attribute person's full name may be derived from attribute person's first name and the last name.

Q: What do you mean by Meta data?

Ans: Metadata is data about data. An item of metadata may describe an individual data item or a collection of data items. Metadata is used to facilitate the understanding, use and management of data. Metadata defines the nature of the data stored in the database. Metadata consists of pre-determined values that describe various attributes of a given table or a relation. Thus a part of the database which contains information about data stored in the database is called as metadata.

Q: What is Data Dictionary?

Ans: Data dictionary is defined as structured repository of data about data.it means that data dictionary does not contain any actual data from the database but it contains set of all the precise definitions of the terms and symbols used in the database.it is a DBMS catalog which contains information like data types, relationships and data constraints of the database.it also convey information who are the authorized users and what are their access rights.

Q :What are the Advantages and disadvantage of Database Systems (DBMS's) ?

The Database Systems provide the following advantages over the traditional file system.

1) **Controlled redundancy**: In a traditional file system, each application program has its own data, which causes duplication of common data items in more than one file. This duplication/redundancy requires multiple updations for a single transaction and wastes a lot of storage space. We cannot eliminate all redundancy due to technical reasons. But in a database, this duplication can be carefully controlled, that means the database system is aware of the redundancy and it assumes the responsibility for propagating updates.

2) **Data consistency**: The problem of updating multiple files in traditional file system leads to inaccurate data as different files may contain different information of the same data item at a given point of time. This causes incorrect or contradictory information to its users. In database systems, this problem of inconsistent data is automatically solved by controlling the redundancy.

3) **Program data independence**: The traditional file systems are generally data dependent, which implies that the data organization and access strategies are dictated by the needs of the specific application and the application programs are developed accordingly. However, the database systems provide an independence between the file system and application program, that allows for changes at one level of the data without affecting others. This property of database systems allow to change data without changing the application programs that process the data.

4) **Sharing of data**: In database systems, the data is centrally controlled and can be shared by all authorized users. The sharing of data means not only the existing applications programs can also share the data in the database but new application programs can be developed to operate on the existing data. Furthermore, the requirements of the new application programs may be satisfied without creating any new file.

5) **Enforcement of standards**: In database systems, data being stored at one central place, standards can easily be enforced by the DBA. This ensures standardised data formats to facilitate data transfers between systems. Applicable standards might include any or all of the following—departmental, installation, organizational, industry, corporate, national or international.

6) **Improved data integrity**: Data integrity means that the data contained in the database is both accurate and consistent. The centralized control property allow adequate checks can be incorporated to provide data integrity. One integrity check that should be incorporated in the database is to ensure that if there is a reference to certain object, that object must exist.

7) **Improved security**: Database security means protecting the data contained in the database from unauthorised users. The DBA ensures that proper access procedures are followed, including proper authentical schemes for access to the DBMS and additional checks before permitting access to sensitive data. The level of security could be different for various types of data and operations.

8) **Data access is efficient**: The database system utilizes different sophisticated techniques to access the stored data very efficiently.

9) **Conflicting requirements can be balanced**: The DBA resolves the conflicting requirements of various users and applications by knowing the overall requirements of the organization. The DBA can structure the system to provide an overall service that is best for the organization.

10) **Improved backup and recovery facility**: Through its backup and recovery subsystem, the database system provides the facilities for recovering from hardware or software failures. The recovery subsystem of the database system ensures that the database is restored to the state it was in before the program started executing, in case of system crash.

11) **Minimal program maintenance**: In a traditional file system, the application programs with the description of data and the logic for accessing the data are built individually. Thus, changes to the data formats or access methods results in the need to modify the application programs. Therefore, high maintenance effort are required. These are reduced to minimal in database systems due to independence of data and application programs.

12) **Data quality is high**: The quality of data in database systems are very high as compared to traditional file systems. This is possible due to the presence of tools and processes in the database system.

13) **Good data accessibility and responsiveness**: The database systems provide query languages or report writers that allow the users to ask ad hoc queries to obtain the needed information immediately, without the requirement to write application programs (as in case of file system), that access the information from the database. This is possible due to integration in database systems.

14) **Concurrency control**: The database systems are designed to manage simultaneous (concurrent) access of the database by many users. They also prevents any loss of information or loss of integrity due to these concurrent accesses.

15) **Economical to scale**: In database systems, the operational data of an organization is stored in a central database. The application programs that work on this data can be built with very less cost as compared to traditional file system. This reduces overall costs of operation and management of the database that leads to an economical scaling.

16) **Increased programmer productivity**: The database system provides many standard functions that the programmer would generally have to write in file system. The availability of these functions allow the programmers to concentrate on the specific functionality required by the users without worrying about the implementation details. This increases the overall productivity of the programmer and also reduces the development time and cost.

Disadvantages of Database Systems :

In contrast to many advantages of the database systems, there are some disadvantages as well. The disadvantages of a database system are as follows:

1) **Complexity increases**: The data structure may become more complex because of the centralised database supporting many applications in an organization. This may lead to difficulties in its management and may require professionals for management.

2) **Requirement of more disk space**: The wide functionality and more complexity increase the size of DBMS. Thus, it requires much more space to store and run than the traditional file system.

3) **Additional cost of hardware**: The cost of database system's installation is much more. It depends on environment and functionality, size of the hardware and maintenance costs of hardware.

4) **Cost of conversion**: The cost of conversion from old file-system to new database system is very high. In some cases the cost of conversion is so high that the cost of DBMS and extra hardware becomes insignificant. It also includes the cost of training manpower and hiring the specialized manpower to convert and run the system.

5) **Need of additional and specialized manpower**: Any organization having database systems, need to be hire and train its manpower on regular basis to design and implement databases and to provide database administration services.

6) **Need for backup and recovery**: For a database system to be accurate and available all times, a procedure is required to be developed and used for providing backup copies to all its users when damage occurs.

7) **Organizational conflict**: A centralised and shared database system requires a consensus on data definitions and ownership as well as responsibilities for accurate data maintenance.

8) **More installational and management cost**: The big and complete database systems are more costly. They require trained manpower to operate the system and has additional annual maintenance and support costs.

Q :What are basic differences between DBMS and file management system ?
Ans- These are basic differences between DBMS and file management
1) Flexibility
2) Fast response to information requests
3) Multiple access
4) Lower user training costs
5) Less storage

Q:Explain the various keys in DBMS.

Ans: **Primary Key:** The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS . Primary keys may consist of a single attribute or multiple attributes in combination.
Examples: Imagine we have a STUDENTS table that contains a record for each student at a university. The student's unique student ID number would be a good choice for a primary key in the STUDENTS table. The student's first and last name would not be a good choice, as there is always the chance that more than one student might have the same name.

Super Key: A superkey is a combination of attributes that can be uniquely used to identify a database record. A table might have many superkeys. Candidate keys are a special subset of superkeys that do not have any extraneous information in them.

Examples: Imagine a table with the fields <Name>, <Age>, <SSN> and <Phone Extension>. This table has many possible superkeys. Three of these are <SSN>, <Phone Extension, Name> and <SSN, Name>. Of those listed, only <SSN> is a candidate key, as the others contain information not necessary to uniquely identify records.

Candidate Key: A candidate key is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Each table may have one or more candidate keys. One of these candidate keys is selected as the table primary key.

In the relational model of databases, a **candidate key** of a relation is a minimal superkey for that relation; that is, a set of attributes such that

1. the relation does not have two distinct tuples with the same values for these attributes (which means that the set of attributes is a superkey)
2. there is no proper subset of these attributes for which (1) holds (which means that the set is minimal).

Since a relation contains no duplicate tuples, the set of all its attributes is a superkey if NULL values are not used. It follows that every relation will have at least one candidate key.

The candidate keys of a relation tell us all the possible ways we can identify its tuples. As such they are an important concept for the design database schema.

Foreign Key: A foreign key is a field (or fields) that points to the primary key of another table. The purpose of the foreign key is to ensure referential integrity of the data. In other words, only values that are supposed to appear in the database are permitted.

For example, say we have two tables, a CUSTOMER table that includes all customer data, and an ORDERS table that includes all customer orders. The constraint here is that all orders must be associated with a customer that is already in the CUSTOMER table. In this case, we will place a foreign key on the ORDERS table and have it relate to the primary key of the CUSTOMER table. This way, we can ensure that all orders in the ORDERS table are related to a customer in the CUSTOMER table. In other words, the ORDERS table cannot contain information on a customer that is not in the CUSTOMER table.

The structure of these two tables will be as follows:

Table ***CUSTOMER***

column name	characteristic
SID	Primary Key
Last_Name	
First_Name	

Table ***ORDERS***

column name	characteristic
Order_ID	Primary Key
Order_Date	
Customer_SID	Foreign Key
Amount	

In the above example, the Customer_SID column in the ORDERS table is a foreign key pointing to the SID column in the CUSTOMER table.

Composite Key : A composite key is a key that uses more than one column to identify the data as opposed to a single column. This can sometimes be more useful than assigning each row an arbitrary value to use as a key such as an auto number field.

Q: What is the domain of an Attribute?

Ans: Attribute domains are rules that describe the legal values of a field type, providing a method for enforcing data integrity. Attribute domains are used to constrain the values allowed in any particular attribute for a table or feature class. If the features in a feature class or nonspatial objects in a table have been grouped into subtypes, different attribute domains can be assigned to each of the subtypes. A domain is a declaration of acceptable attribute values. Whenever a domain is associated with an attribute field, only the values within that domain are valid for the field. In other words, the field will not accept a value that is not in that domain. Using domains helps ensure data integrity by limiting the choice of values for a particular field.

For example: Rooms in hotel (1-300)
Age (1-99)
Married (yes or no)
Nationality (Sri Lankan, Indian, American, or British)

Q: What do you mean by Entity and Entity set?

Ans: An entity is a thing in the real world with an independent existance. and entity set is collection or set all entities of a particular entity type at any point of time.

* An entity is an object that exists and is distinguishable from other objects. For instance, John Harris with S.I.N. 890-12-3456 is an entity, as he can be uniquely identified as one particular person in the universe.

* An entity may be concrete (a person or a book, for example) or abstract (like a holiday or a concept).

* An entity set is a set of entities of the same type (e.g., all persons having an account at a bank).

* Entity sets need not be disjoint. For example, the entity set employee (all employees of a bank) and the entity set customer (all customers of the bank) may have members in common.

* An entity is represented by a set of attributes.
o E.g. name, S.I.N., street, city for “customer” entity.

Q: What is a Log File?

Ans: A **log file** is a recording of everything that goes in and out of a particular server. It is a concept much like the black box of an airplane that records everything going on with the plane in the event of a problem. The information is frequently recorded chronologically, and is located in the root directory, or occasionally in a secondary folder, depending on how it is set up with the server. The only person who has regular access to the log files of a server is the server administrator, and a log file is generally password protected, so that the server administrator has a record of everyone and everything that wants to look at the log files for a specific server.

The point of a **log file** is to keep track of what is happening with the server. If something should malfunction within a complex system, there may be no other way of identifying the problem. Log files are also used to keep track of complex systems, so that when a problem does occur, it is easy to pinpoint and fix. Log files are also important to keeping track of applications that have little to no human interaction, such as server applications. There are times when log files are too difficult to read or make sense of, and it is then that log file analysis is necessary. Log file analysis is generally performed by some kind of computer program that makes the log file information more concise and readable format. Log files can also be used to correlate data between servers, and find common problems between different systems that might need one major solution to repair them all.

Q:What is Normalization? Why it is requried?

Ans: Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

Why it is requried?

Normalization reduces redundancy. Redundancy is the unnecessary repetition of data. It can cause problems with storage, reterival and updation of data. Redundancy can lead to:

- **Inconsistencies:-**errors are more likely to occur when facts are repeated.
- **Update anomalies:-**inserting, modifying and deleting data may cause inconsistencies. Inconsistency occurs when we perform updation or deletion of data in one relation, while forgetting to make corresponding changes in other relations.

During the process of normalization, you can identify dependencies, which can cause problems when deleting or updating. Normalization also helps to simplify the structure of the tables. A fully normalized record consist of:

- A primary key that identifies that entity.
- A set of attributes that describe that entity.

Q: What is the Difference between primary key and unique key?
Ans: The column holding the primary key constraint cannot accept null values.whereas colum holding the unique constraint can accept null values assume that t3 is a table with two columns t1 having primary key constraint and t2 having unique constraint if u try to insert null into t2 it will accept that values whereas column t1 will not accept null. Each table having only one PRIMARY KEY.And my contain many UNIQUE KEYS.

Q:What is Data Redundancy?

Ans: **Data redundancy** occurs in database systems which have a field that is repeated in two or more tables. For instance, in case when customer data is duplicated and attached with each product bought then redundancy of data is a known source of inconsistency, since customer might appear with different values for given attribute.Data redundancy leads to data anomalies and corruption and generally should be avoided by design.Database normalization prevents redundancy and makes the best possible usage of storage.Proper use of foreign keys can minimize data redundancy and chance of destructive anomalies. However sometimes concerns of efficiency and convenience can result redundant data design despite the risk of corrupting the data.

Q: What is the Difference between Single valued and multi valued attributes ?

Ans: A single valued attribute can have only a single value. For example a person can have only one 'date of birth', 'age' etc. That is a single valued attributes can have only single value. But it can be simple or composite attribute.That is 'date of birth' is a composite attribute , 'age' is a simple attribute. But both are single valued attributes.

Multivalued attributes can have multiple values. For instance a person may have multiple phone numbers,multiple degrees etc.Multivalued attributes are shown by a double line connecting to the entity in the ER diagram.

Single Valued Attribute: Attribute that hold a single value
Example1: Age
Exampe2: City
Example3:Customer id

Multi Valued Attribute: Attribute that hold multiple values.
Example1: A customer can have multiple phone numbers, email id's etc
Example2: A person may have several college degrees.

Q:What is the difference between Strong and weak entity?

Ans: An entity set that does not have sufficient attributes to form a primary key is termed as a weak entity set. An entity set that has a primary key is termed as strong entity set.

A weak entity is existence dependent. That is the existence of a weak entity depends on the existence of a identifying entity set. The discriminator (or partial key) is used to identify other attributes of a weak entity set.The primary key of a weak entity set is formed by primary key of identifying entity set and the discriminator of weak entity set. The existence of a weak entity is indicated by a double rectangle in the ER diagram. We underline the discriminator of a weak entity set with a dashed line in the ER diagram.

Q : What is storage manager?

A storage manager is a program module that provides the interface between the low level data stored in a database and the application programs and queries submitted to the system.

components of storage manager:

The storage manager components include

a) Authorization and integrity manager
b) Transaction manager
c) File manager
d) Buffer manager

Q: What do you mean by Integrity Constraints?

Ans: Integrity constraints are used to ensure accuracy and consistency of data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity. There are many types of integrity constraints that play a role in referential integrity

Entity integrity:-The entity integrity constraint states that no primary key value can be null. This is because the primary key value is used to identify individual tuples in a relation . Having null value for the primary key implies that we cannot identify some tuples.This also specifies that there may not be any duplicate entries in primary key column.

Referential Integrity:-The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples in the two relations. Informally, the referential integrity constraint states that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation.

Domain Integrity:-The domain integrity states that every element from a relation should respect the type and restrictions of its corresponding attribute. A type can have a variable length which needs to be respected. Restrictions could be the range of values that the element can have, the default value if none is provided, and if the element can be NULL.

Q:What is a Data Warehouse?

Ans: A data warehouse (DW) is a database used for reporting . The data is offloaded from the operational systems for reporting. The data may pass through an operational data store for additional operations before it is used in the DW for reporting.

A data warehouse maintains its functions in three layers: staging, integration, and access. *Staging* is used to store raw data for use by developers (analysis and support). The *integration* layer is used to integrate data and to have a level of abstraction from users. The *access* layer is for getting data out for users.

Q: Explain Referential Integrity?

Ans: Referential integrity is a database concept that ensures that relationships between tables remain consistent. When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key unless there is a corresponding record in the linked table. It also includes the techniques known as cascading update and cascading delete, which ensure that changes made to the linked table are reflected in the primary table.

Consider the situation where we have two tables: Employees and Managers. The Employees table has a foreign key attribute entitled ManagedBy which points to the record for that employee's manager in the Managers table. Referential integrity enforces the following three rules:

1. We may not add a record to the Employees table unless the Managed By attribute points to a valid record in the Managers table.
2. If the primary key for a record in the Managers table changes, all corresponding records in the Employees table must be modified using a cascading update.
3. If a record in the Managers table is deleted, all corresponding records in the Employees table must be deleted using a cascading delete.

Q:What are the main tasks performed by DBA?

Ans: One of the main reasons for using DBMS is to have a central control of both data and the programs accessing those data. A person who has such control over the system is called a Database Administrator(DBA). The following are the functions of a Database administrator.

- Schema Definition
- Storage structure and access method definition
- Schema and physical organization modification.
- Granting authorization for data access.
- Routine Maintenance

Schema Definition: The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database table(Relation) like data types of attributes,length of attributes,integrity constraints etc.

Storage structure and access method definition

Database tables or indexes are stored in the following ways: Flat files,Heaps,B+ Tree etc..

Schema and physical organization modification

The DBA carries out changes to the existing schema and physical organization.

Granting authorization for data access

The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data, while you go up in the hierarchy to the administrator ,you will get more access rights.

Routine Maintenance: Some of the routine maintenance activities of a DBA is given below.

- Taking backup of database periodically
- Ensuring enough disk space is available all the time.
- Monitoring jobs running on the database.
- Ensure that performance is not degraded by some expensive task submitted by some users.
- Performance Tuning.

Q:Explain Role and Resposibilities of DBA?

Ans: The success of a database environment depends on central control of database design, implementation, and use. This central control and coordination is the role of the database administrator (DBA). The DBA is a single person; however, large organizations may divide DBA responsibilities among a team of personnel, each with specific skills and areas of responsibility such as database design, tuning, or problem resolution.

A **database administrator** (**DBA**) is a person responsible for the design, implementation, maintenance and repair of an organization's database. They are also known by the titles *Database Coordinator* or *Database Programmer*, and is closely related to the *Database Analyst*, *Database Modeler*, *Programmer Analyst*, and *Systems Manager*. The role includes the development and design of database strategies, monitoring and improving database performance and capacity, and planning for future expansion requirements. They may also plan, co-ordinate and implement security measures to safeguard the database. Employing organizations may require that a database administrator have a certification or degree for database systems (for example, the Microsoft Certified Database Administrator). Some organizations have a hierarchical level of database administrators, generally:

- Data Analysts/Query designers
- Junior DBAs
- Midlevel DBAs
- DBA consultants

Personal Characteristics/Skills

1. Strong organizational skills
2. Strong logical and analytical thinker
3. Ability to concentrate and pay close attention to detail
4. Strong written and verbal communication skills
5. Willing to pursue education throughout your career.

Role of DBA:

Schema Definition:

The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database table(Relation) like data types of attributes,length of attributes,integrity constraints etc.

Storage structure and access method definition

Database tables or indexes are stored in the following ways: Flat files,Heaps,B+ Tree etc..

Schema and physical organization modification

The DBA carries out changes to the existing schema and physical organization.

Granting authorization for data access

The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data, while you go up in the hierarchy to the administrator ,you will get more access rights.

Routine Maintenance:

Some of the routine maintenance activities of a DBA is given below.

- Taking backup of database periodically
- Ensuring enough disk space is available all the time.
- Monitoring jobs running on the database.
- Ensure that performance is not degraded by some expensive task submitted by some users.
- Performance Tuning.

Q: What is the use of group by clause?

Group by clause is used to apply aggregate functions to a set of tuples. The attributes given in the group by clause are used to form groups. Tuples with the same value on all attributes in the group by clause are placed in one group.