

Q2 (a) Explain the differences between conceptual & external schema.

Ans2 a. Page Number 24 of textbook.

Q2 (b) Describe the four components of a database system.

Answer A database system is composed of four components:

- Data
- Hardware
- Software
- Users

Which coordinate with each other to form an effective database system.

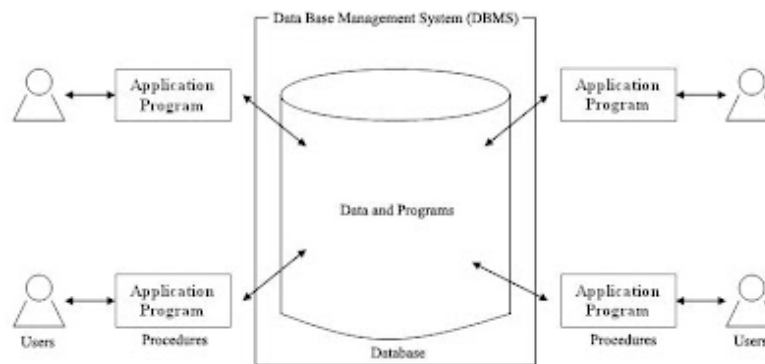


Figure: Data Base System

1. Data - It is a very important component of the database system. Most of the organizations generate, store and process large amount of data. The data acts a bridge between the machine parts i.e. hardware and software and the users which directly access it or access it through some application programs.

Data may be of different types.

- User Data - It consists of a table(s) of data called Relation(s) where Column(s) are called fields of attributes and rows are called Records for tables. A Relation must be structured properly.
- Metadata - A description of the structure of the database is known as Metadata. It basically means "data about data". System Tables store the Metadata which includes.

- Number of Tables and Table Names
- Number of fields and field Names
- Primary Key Fields
- Application Metadata - It stores the structure and format of Queries, reports and other applications components. '

2. Hardware - The hardware consists of the secondary storage devices such as magnetic disks (hard disk, zip disk, floppy disks), optical disks (CD-ROM), magnetic tapes etc. on which data is stored together with the Input/Output devices (mouse, keyboard, printers), processors, main memory etc. which are used for storing and retrieving the data in a fast and efficient manner. Since database can range from those of a single user with a desktop computer to those on mainframe computers with thousand of users, therefore proper care should be taken for choosing appropriate hardware devices for a required database.

3. Software - The Software part consists of DBMS which acts as a bridge between the user and the database or in other words, software that interacts with the users, application programs, and database and files system of a particular storage media (hard disk, magnetic tapes etc.) to insert, update, delete and retrieve data. For performing these operations such as insertion, deletion and updation we can either use the Query Languages like SQL, QUEL, Gupta SQL or application softwares such as Visual Basic, Developer etc.

4. Users - Users are those persons who need the information from the database to carry out their primary business responsibilities i.e. Personnel, Staff, Clerical, Managers, Executives etc. On the basis of the job and requirements made by them they are provided access to the database totally or partially. The various types of users which can access the database are:-

- Database Administrators (DBA)
- Database Designers
- End Users
- Application Programmers

Q2 (c) What are the characteristics of database?

Ans2 c. Page Number 8 of textbook.

Q3 (a) Explain about primary key, super key, candidate key, alternate key using suitable example.

Answer

* **Alternate key** - An alternate key is any candidate key which is not selected to be the primary key

* **Candidate key** - A candidate key is a field or combination of fields that can act as a primary key field for that table to uniquely identify each record in that table.

For Eg: The table:

Employee(Name,Address,Ssn,Employee_Id_{primary_key},Phone_ext)

In the above example Ssn no. and employee identity are candidate keys.

* **Compound key** - compound key (also called a composite key or concatenated key) is a key that consists of 2 or more attributes.

* **Primary key** - a primary key is a value that can be used to identify a unique row in a table. Attributes are associated with it. Examples of primary keys are Social Security numbers (associated to a specific person) or ISBNs (associated to a specific book).

In the relational model of data, a primary key is a candidate key chosen as the main method of uniquely identifying a tuple in a relation.

For Eg:

Employee(Name,Address,Ssn,Employee_Id_{primary_key},Phone_ext)

* **Superkey** - A superkey is defined in the relational model as a set of attributes of a relation variable (relvar) for which it holds that in all relations assigned to that variable there are no two distinct tuples (rows) that have the same values for the attributes in this set. Equivalently a superkey can also be defined as a set of attributes of a relvar upon which all attributes of the relvar are functionally dependent.

For Eg:

Employee(Name,Address,Ssn,Employee_Id_{primary_key},Phone_ext)

<Ssn,Name,Address>

<Ssn,Name>

<Ssn>

All the above are super keys.

* **Foreign key** - a foreign key (FK) is a field or group of fields in a database record that points to a key field or group of fields forming a key of another database record in some (usually different) table. Usually a foreign key in one table refers to the primary key (PK) of another table. This way references can be made to link information together and it is an essential part of database normalization.

For Eg:

For a Student...

School(Name,Address,Phone,School_Reg_no_{primary_key})

Q3 (b) Explain Data definition SQL commands. Give syntax and suitable example.

Answer

The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. Let's take a look at the structure and usage of four basic DDL commands:

The Data Definition Language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. Let's take a look at the structure and usage of four basic DDL commands:

CREATE

Installing a database management system (DBMS) on a computer allows you to create and manage many independent databases. For example, you may want to maintain a database of customer contacts for your sales department and a personnel database for your HR department. The CREATE command can be used to establish each of these databases on your platform. For example, the command:

```
CREATE DATABASE employees
```

creates an empty database named "employees" on your DBMS. After creating the database, your next step is to create [tables](#) that will contain data. Another variant of the CREATE command can be used for this purpose. The command:

```
CREATE TABLE personal_info (first_name char(20) not null, last_name char(20) not null, employee_id int not null)
```

establishes a table titled "personal_info" in the current database. In our example, the table contains three attributes: first_name, last_name and employee_id. Don't worry about the other information included in the command -- we'll cover that in a future article.

USE

The USE command allows you to specify the database you wish to work with within your DBMS. For example, if we're currently working in the sales database and want to issue some commands that will affect the employees database, we would preface them with the following SQL command:

```
USE employees
```

It's important to always be conscious of the database you are working in before issuing SQL commands that manipulate data.

ALTER

Once you've created a table within a database, you may wish to modify the definition of it. The ALTER command allows you to make changes to the structure of a table without deleting and recreating it. Take a look at the following command:

```
ALTER TABLE personal_info  
ADD salary money null
```

This example adds a new attribute to the personal_info table -- an employee's salary. The "money" argument specifies that an employee's salary will be stored using a dollars and cents format. Finally, the "null" keyword tells the database that it's OK for this field to contain no value for any given employee.

DROP

The final command of the Data Definition Language, DROP, allows us to remove entire database objects from our DBMS. For example, if we want to permanently remove the personal_info table that we created, we'd use the following command:

```
DROP TABLE personal_info
```

Similarly, the command below would be used to remove the entire employees database:

```
DROP DATABASE employees
```

Q4 (a) What is the difference between functional and multi valued dependencies?

Answer Page number 502 of Textbook

Q4 (b) Write notes on the following using suitable example ?

- I. Fourth Normal form and its usefulness.**
- II. Loss less join Decomposition into 4NF relation.**

Answer Page number [517 – 518] Third Edition of Text Book

Q5 (a) Specify about the notations used in Entity – Relationship diagrams in DBMS design.

Answer

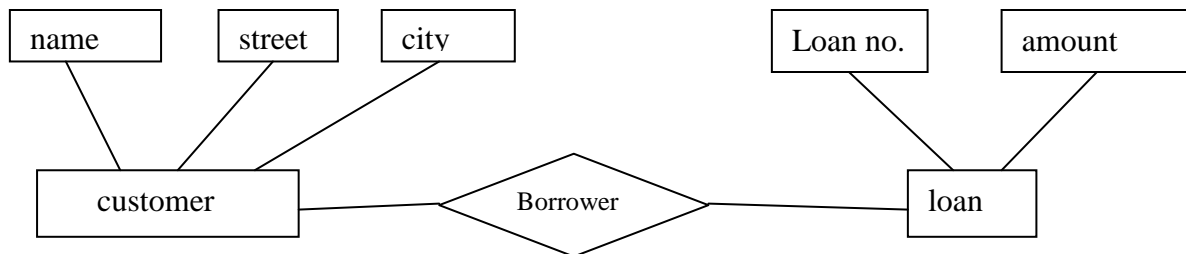
The overall logical structure of a database can be expressed graphically by an E/R diagram. The relative simplicity and pictorial clarity of this diagramming technique may well account in large part of the widespread use of the E/R model.

Such diagram consists of the following major components:-

- Rectangles: they represent entity sets.
- Ellipses: they represent attributes.
- Diamonds: they represent relationship.
- Lines: they link attributes to entity sets and entity sets to relationship.

Again to distinguish between the types of relationship i.e. one to one, one to many, many to many, either a directed line or an undirected line between the relationship and entity sets is drawn.

Example:



Q5 (b) Define Relational algebra. Discuss traditional set operations on relations.

Answer

Relational algebra is a method of specifying “a sequence of operations to be performed on the database” to compute a desired result. It consists of a set of operations that take one or two relations as input and produce a new relation as their result. These set of operations can be categorized.

These set of operations can be categorized in to two types: -

- Traditional set operations
- Special relational operations

Traditional set operations:

These are of two types; unary operations, binary operations.

Unary operations:-

- The SELECT operation : eg. SELECT salary >= 2000 (EMPLOYEE)
- The PROJECT operation: eg. PROJECT empno,empname (EMPLOYEE)

Binary operations:

- The UNION operation

PROJECT empno (SELECT salary >2000 (EMPLOYEE)

- The SET DIFFERENCE operator

PROJECT empno(SELECT salary > 2000 (EMPLOYEE)) – PROJECT empno
(SELECT deptno = “conscience”(EMPLOYEE_DEPTDETAILS))

- The CARTESIAN Product operation

(EMPLOYEE.empno,EMPLOYEE.salary,EMPLOYEE.empname,EMPLOYEE_DEPTDETAILS.empno,EMPLOYEE_DEPTDETAILS.deptname)

Q6 (a) What is a view in SQL and how it is defined? Discuss the problems that may arise when one attempts to update a view.

Answer Page Number 278 –280 of Third Edition of Text Book

Q6 (b) Discuss Codd's rule for relational database to see how relational it is. Enumerate at least six rules of Codd.

Answer The Codd's 12 rules are based on following Rule 0 by which a database could be evaluated to see how relational it is.

The twelve rules of Codd are as follows;

1. All information in a relational database is represented explicitly at the logical level and in exactly one way.
2. Each and every atomic value in a relational database is guaranteed to be logically accessible by resorting to a combination of table name,primary key value and column value.
3. Null values are supported in a fully relational DBMS for representing missing information in a systematic way, independent of data type.
4. Users and programmers need only know one data language to operate the entire database.
5. A relational database must support at least language which should have: data definition,view definition,data manipulation,integrity constraints,authorization,transaction boundaries.

All views that are theoretically updatable are also updatable by the system.

Q7 (a) Describe Heuristics rules used in Query optimization.

Answer

The parser of a high level query first generates an initial internal representation, which is then optimized according to heuristic rules. After that a query execution plan is generated to execute group of operations based on access paths available on the files involved in the query.

One of the main heuristics rule is to apply SELECT and PROJECT operations before applying the JOIN operations. This is because the size of the file resulting from a binary operation like JOIN, is usually a multiplicative function of the size of input files.

Since SELECT and PROJECT operations reduce the size of input file and hence should be applied before the JOIN operation or any other binary operation.

A query tree is used to represent a relation algebra operation or extended relational algebra expression, where a query graph is used to represent a relation calculus expressions.

Q7 (b) What is Normalization? How it play a major role in designing of RDBMS?

Answer

Normalization a database means tendency to simplify the table for easy implementation in large databases. Normalization a logical database design involves organizing the data into more than one table.

Normalization improves the performance by reducing redundancy in database tables.

Benefits of Normalization:-

It reduces redundancy, which means that information to be stored only once in relation. So, it saves wastage of storage space. It also helps to update anomalies. The other benefits are faster sorting, and index creation, few indexes per table, few NULLS and an increase in the compactness of the database. However, complexity increases whenever JOIN operation takes place.

It also helps to simplify the structure of tables. The performance of an application is directly linked to the database design.

Some rules that should be followed to achieve a good database design are:

- Each table should have an identifier.
- Each table should store data for a single type of entity.
- Nullable columns in table should be avoided.

The repetition of values or columns in tables should be avoided.

Q8 (a) Explain the differences between 2NF and 3NF with reference to their definitions. Give suitable example.

Answer

The general definition of 3NF explains that: A relation schema R is in 3NF if every nonprime attribute of R meets both the following terms:

- (a) It is fully functionally dependent to key of R
- (b) It is nontransitively dependent on every key of R.

The BCNF was proposed as a simpler of 3NF, but it was found to be strict than 3NF, because every relation in BCNF is also in 3NF; however a relation in 3NF is not necessarily in BCNF. The formal definition of BCNF differs slightly normal form the definition of 3NF.

A relation schema R is in BCNF if whenever a nontrivial dependency $X \rightarrow A$ holds in R, then X is a superkey of R.

The only difference between the definitions of BCNF and 3NF is that conditions (b) of 3NF, which allows A to be prime attribute, is absent from BCNF.

Q8 (b) With the help of diagram, explain the different steps for processing a high-level query.

Answer Page Number 586 of Textbook

Q9 (a) What do you understand by the term INDEX. Discuss various types of Indexes used for record tables.

Answer

An index is usually defined on a single field of a file, called an indexing Field. The index typically stores each value of the index field along with a list of pointers to all disk blocks that contain a record with that field value. The values in the index are ordered so that we can do a binary search on the index. The index file is much smaller than the data file, so searching the index using binary search is reasonably efficient. Multilevel indexing does away with the need for binary search at the expense of creating indexes to the index itself!

There are several types of indexes. A primary index is an index specified on the ordering key field of an ordered file of records. An ordering key field is used to physically order the file records on disk, and every record has a unique value for that field.

If the ordering field is not a key field that is, several records in the file can have the same value for the ordering field another type of index, called a clustering index, can be used. A file can have at most one physical ordering field, so it can have at most one primary index or one clustering index, but not both.

A third type of index, called a secondary index, can be specified on any non-ordering field of a file. A file can have several secondary indexes in addition to its primary access method. In the next three subsections.

Q9 (b) What is B-tree? What is its advantages for dynamic multiple indexing?

Answer

B-TREES: The B-tree is known as the balanced sort tree, which is useful for external sorting. There are strong uses of B-trees in a database system as pointed "While no single scheme can be optimum for all applications, the technique of organizing a file and its index called the B -tree is, the standard organization for indexes in a database system."

The file is a collection of records. The index refers to a unique key, associated with each record. One application of B- trees is found in Virtual Storage Access Method (VSAM) file organization. Many data manipulation tasks require data storage only in main memory. For applications with a large database running on a system with limited company, the data must be stored as records on secondary memory (disks) and be accessed in pieces. The size of a record can be quite large, so B-TREES are required.

Advantages of B tree indexes:

1. Because there is no overflow problem inherent with this type of organization it is good for dynamic table - those that suffer a great deal of insert / update / delete activity.
2. Because it is to a large extent self-maintaining, it is good in supporting 24-hour operation.
3. As data is retrieved via the index it is always presented in order.
4. 'Get next' queries are efficient because of the inherent ordering of rows with in the index blocks.
5. Btree indexes are good for very large tables because they will need minimal reorganization.
6. There is predictable access time for any retrieval (of the same number of rows of course) because the Btree structure keeps itself balanced, so that there is always the same number of index levels for every retrieval. Bear in mind of course, that the number of index levels does increase both with the number of records and the length of the key value.

Text Book

**Fundamentals of Database Systems, Elmasri, Navathe, Somayajulu, Gupta,
Pearson Education, 2006**