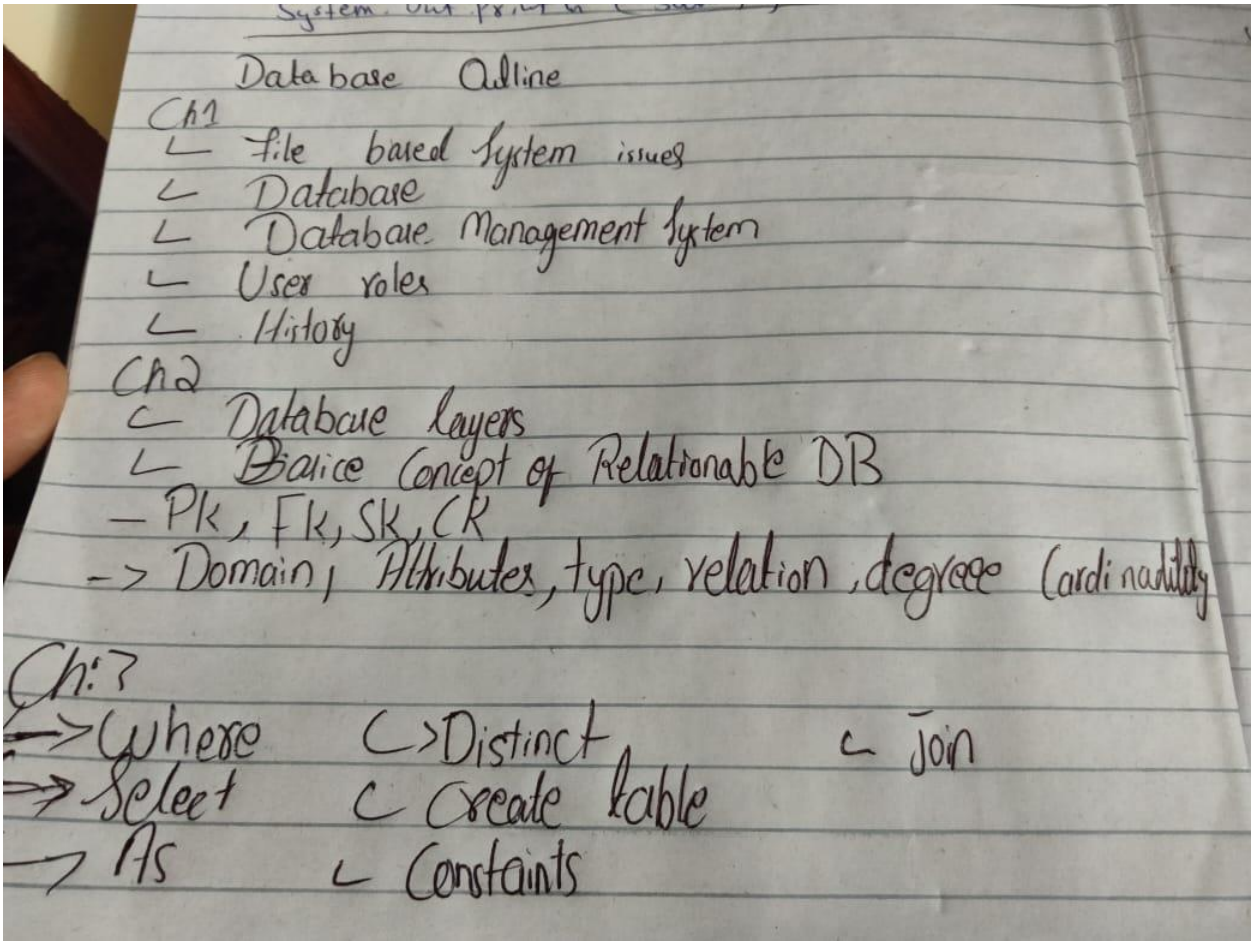
DBMS Outlines



Database Outline

Ch1
- File based system issues
- Database
- Database Management System
- User roles
- History

Ch2
- Database layers
- Basics Concept of Relationable DB
  - Pk, Fk, SK, CK
  - -> Domain, Attributes, type, relation, degree Cardinality

Ch:3
- -> where
- -> Select
- -> As
- -> Distinct
- -> Create table
- -> Constaints
- -> Join

CHAPTER 1

What is Database?

What is Database?

A **database** is an organized collection of data, so that it can be easily accessed and managed.

You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

Database handlers create a database in such a way that only one set of software program provides access of data to all the users.

The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

There are many **dynamic websites** on the World Wide Web nowadays which are handled through databases. For example, a model that checks the availability of rooms in a hotel. It is an example of a dynamic website that uses a database.

There are many **databases available** like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

Modern databases are managed by the database management system (DBMS).

SQL or Structured Query Language is used to operate on the data stored in a database. SQL depends on relational algebra and tuple relational calculus.

A cylindrical structure is used to display the image of a database.

Evolution of Databases

The database has completed more than 50 years of journey of its evolution from flat-file system to relational and objects relational systems. It has gone through several generations.

The Evolution

File-Based

1968 was the year when File-Based database were introduced. In file-based databases, data was maintained in a flat file. Though files have many advantages, there are several limitations.

One of the major advantages is that the file system has various access methods, e.g., sequential, indexed, and random.

It requires extensive programming in a third-generation language such as COBOL, BASIC.

Hierarchical Data Model

1968-1980 was the era of the Hierarchical Database. Prominent hierarchical database model was IBM's first DBMS. It was called IMS (Information Management System).

In this model, files are related in a parent/child manner.

Below diagram represents Hierarchical Data Model. Small circle represents objects.

What is Database Management System (DBMS)?

DBMS (Data Base Management System)

Database management System is software which is used to store and retrieve the database. For example, Oracle, MySQL, etc.; these are some popular DBMS tools.

- DBMS provides the interface to perform the various operations like creation, deletion, modification, etc.
- DBMS allows the user to create their databases as per their requirement.
- DBMS accepts the request from the application and provides specific data through the operating system.
- DBMS contains the group of programs which acts according to the user instruction.
- It provides security to the database.

Advantage of DBMS

Controls redundancy

It stores all the data in a single database file, so it can control data redundancy.

Data sharing

An authorized user can share the data among multiple users.

Backup

It providesBackup and recovery subsystem. This recovery system creates automatic data from system failure and restores data if required.

Multiple user interfaces

It provides a different type of user interfaces like GUI, application interfaces.

Disadvantage of DBMS

Size

It occupies large disk space and large memory to run efficiently.

Cost

DBMS requires a high-speed data processor and larger memory to run DBMS software, so it is costly.

Complexity

DBMS creates additional complexity and requirements.

File Based System Issues

File-based System

One way to keep information on a computer is to store it in permanent files. A company system has a number of application programs; each of them is designed to manipulate data files. These application programs have been written at the request of the users in the organization. New applications are added to the system as the need arises. The system just described is called the *file-based system*.

Consider a traditional banking system that uses the file-based system to manage the organization's data shown in Figure 1.1. As we can see, there are different departments in the bank. Each has its own applications that manage and manipulate different data files. For banking systems, the programs may be used to debit or credit an account, find the balance of an account, add a new mortgage loan and generate monthly statements.

Figure 1.1. Example of a file-based system used by banks to manage data.

Disadvantages of the file-based approach

Using the file-based system to keep organizational information has a number of disadvantages. Listed below are five examples.

Data redundancy

Often, within an organization, files and applications are created by different programmers from various departments over long periods of time. This can lead to *data redundancy*, a situation that occurs in a database when a field needs to be updated in more than one table. This practice can lead to several problems such as:

- Inconsistency in data format
- The same information being kept in several different places (files)
- *Data inconsistency*, a situation where various copies of the same data are conflicting, wastes storage space and duplicates effort

Data isolation

Data isolation is a property that determines when and how changes made by one operation become visible to other concurrent users and systems. This issue occurs in a concurrency situation. This is a problem because:

- It is difficult for new applications to retrieve the appropriate data, which might be stored in various files.

Integrity problems

Problems with *data integrity* is another disadvantage of using a file-based system. It refers to the maintenance and assurance that the data in a database are correct and consistent. Factors to consider when addressing this issue are:

- Data values must satisfy certain consistency constraints that are specified in the application programs.
- It is difficult to make changes to the application programs in order to enforce new constraints.

Security problems

Security can be a problem with a file-based approach because:

- There are constraints regarding accessing privileges.
- Application requirements are added to the system in an ad-hoc manner so it is difficult to enforce constraints.

Concurrency access

Concurrency is the ability of the database to allow multiple users access to the same record without adversely affecting transaction processing. A file-based system must manage, or prevent, concurrency by the application programs. Typically, in a file-based system, when an application opens a file, that file is locked. This means that no one else has access to the file at the same time.

In database systems, concurrency is managed thus allowing multiple users access to the same record. This is an important difference between database and file-based systems.

Roles in DBMS

Privilege and Roles in DBMS

Last Updated : 21 Jan, 2021

Confidentiality, integrity, and availability are the stamps of database security. Authorization is the allowance to the user or process to access the set of objects. The type of access granted can be any like, read-only, read, and write. Privilege means different Data Manipulation Language(DML) operations which can be performed by the user on data like INSERT, UPDATE, SELECT and DELETE, etc.

There are two methods by which access control is performed is done by using the following.

1. Privileges
2. Roles

Let's discuss one by one.

Privileges :

The authority or permission to access a named object as advised manner, for example, permission to access a table. Privileges can allow permitting a particular user to connect to the database. In, other words privileges are the allowance to the database by the database object.

- **Database privileges –**
  A privilege is permission to execute one particular type of SQL statement or access a second persons' object. Database privilege controls the use of computing resources. Database privilege does not apply to the Database administrator of the database.

- **System privileges –**
  A system privilege is the right to perform an activity on a specific type of object. for example, the privilege to delete rows of any table in a database is system privilege. There are a total of 60 different system privileges. System privileges allow users to CREATE, ALTER, or DROP the database objects.

- **Object privilege –**
  An object privilege is a privilege to perform a specific action on a particular table, function, or package. For example, the right to delete rows from a table is an object privilege. For example, let us consider a row of table GEEKSFORGEEKS that contains the name of the employee who is no longer a part of the organization, then deleting that row is considered as an object privilege. Object privilege allows the user to INSERT, DELETE, UPDATE, or SELECT the data in the database object.

Roles :

A role is a mechanism that can be used to allow authorization. A person or a group of people can be allowed a role or group of roles. By many roles, the head can manage access privileges very easily. The roles are provided by the database management system for easy and managed or controlled privilege management.

Properties –

The following are the properties of the roles which allow easy privilege management inside a database:

- **Reduced privilege administration –**
  The user can grant the privilege for a group of users who are related instead of granting the same set of privileges to the users explicitly.
- **Dynamic privilege management –**
  If the privilege of the group changes then, only the right of role needs to be changed.
- **Application-specific security –**
  The user can also protect the use of a role by using a password. Applications can be created to allow a role when entering the correct and best password. Users are not allowed the role if they do not know about the password.

History of DBMS

History of DBMS

Last Updated : 16 Mar, 2020

Data is a collection of facts and figures. The data collection was increasing day to day and they needed to be stored in a device or a software which is safer.

Charles Bachman was the first person to develop the Integrated Data Store (IDS) which was based on network data model for which he was inaugurated with the Turing Award (The most prestigious award which is equivalent to Nobel prize in the field of Computer Science.). **It was developed in early 1960's.**

In the late 1960's, IBM (International Business Machines Corporation) developed the Integrated Management Systems which is the standard database system used till date in many places. It was developed based on the hierarchical database model. It was during the year 1970 that the relational database model was developed by Edgar Codd. Many of the database models we use today are relational based. It was considered the standardized database model from then.

The relational model was still in use by many people in the market.Later during the same decade (1980's), IBM developed the Structured Query Language (SQL) as a part of R project. It was declared as a standard language for the queries by ISO and ANSI. The Transaction Management Systems for processing transactions was also developed by James Gray for which he was felicitated the Turing Award.

Further, there were many other models with rich features like complex queries, datatypes to insert images and many others. The Internet Age has perhaps influenced the data models much more. Data models were developed using object oriented programming features, embedding with scripting languages like Hyper Text Markup Language (HTML) for queries. With humongous data being available online, DBMS is gaining more significance day by day.

CHAPTER 2

Database Layer

Database Layer

Architecturally, databases fall into two fundamental groups: repositories that store Oracle system data; and data sources that are the subject of analysis, presentation, and reporting.

There are two important repositories for information storage:

- **Common repository**—Oracle system data in supported relational database tables
- **Shared Services**—User, security, and project data that can be used across Oracle products

Data Sources:

- Relational data sources, for example, Oracle, IBM DB2, and Microsoft SQL Server
- Multidimensional data sources, for example, Essbase
- Oracle's Oracle applications, for example, Oracle Hyperion Financial Management and Oracle Hyperion Planning
- Data warehouses
- ODBC data sources

Basic Concepts of Relational DB

What is RDBMS?

RDBMS stands for **R**elational **D**atabase **M**anagement **S**ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

What is a table?

The data in an RDBMS is stored in database objects which are called as **tables**. This table is basically a collection of related data entries and it consists of numerous columns and rows.

Remember, a table is the most common and simplest form of data storage in a relational database. The following program is an example of a CUSTOMERS table −

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

What is a field?

Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS table consist of ID, NAME, AGE, ADDRESS and SALARY.

A field is a column in a table that is designed to maintain specific information about every record in the table.

What is a Record or a Row?

A record is also called as a row of data is each individual entry that exists in a table. For example, there are 7 records in the above CUSTOMERS table. Following is a single row of data or record in the CUSTOMERS table −

```
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
+----+----------+-----+-----------+----------+
```

A record is a horizontal entity in a table.

What is a column?

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

For example, a column in the CUSTOMERS table is ADDRESS, which represents location description and would be as shown below −

```
+-----------+
| ADDRESS   |
+-----------+
| Ahmedabad |
| Delhi     |
| Kota      |
| Mumbai    |
| Bhopal    |
| MP        |
| Indore    |
+----+------+
```

What is a NULL value?

A NULL value in a table is a value in a field that appears to be blank, which means a field with a NULL value is a field with no value.

It is very important to understand that a NULL value is different than a zero value or a field that contains spaces. A field with a NULL value is the one that has been left blank during a record creation.

Types of Keys in DB

Types of SQL Keys

We have the following types of keys in SQL which are used to fetch records from tables and to make relationships among tables or views.

01. Super Key

A super key is a set of one or more than one key that can be used to identify a record uniquely in a table. **Example:** Primary key, Unique key, Alternate key are a subset of Super Keys.

02. Candidate Key

A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multiple Candidate Keys in one table. Each Candidate Key can work as a Primary Key.

Example: In the below diagram ID, RollNo and EnrollNo are Candidate Keys since all these three fields can work as Primary Key.

03. Primary Key

A primary key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.

04. Alternate key

An Alternate key is a key that can work as a primary key. Basically, it is a candidate key that currently is not a primary key.

Example: In the below diagram RollNo and EnrollNo become Alternate Keys when we define ID as the Primary Key.

05. Composite/Compound Key

Composite Key is a combination of more than one field/column of a table. It can be a Candidate key, Primary key.

06. Unique Key

A unique key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It is like a Primary key but it can accept only one null value and it can not have duplicate values. For more help refer to the article Difference between primary key and unique key.

07. Foreign Key

Foreign Key is a field in a database table that is the Primary key in another table. It can accept multiple null, duplicate values. For more help refer to the article Difference between primary key and foreign key.

Example: We can have a DeptID column in the Employee table which is pointing to a DeptID column in a department table where it is a primary key.

Defining Keys in SQL Server

```
--Department Table
 CREATE TABLE Department
(
 DeptID int PRIMARY KEY, --primary key
 Name varchar (50) NOT NULL,
 Address varchar (200) NOT NULL
 )
--Student Table
CREATE TABLE Student
(
 ID int PRIMARY KEY, --primary key
 RollNo varchar(10) NOT NULL,
 Name varchar(50) NOT NULL,
 EnrollNo varchar(50) UNIQUE, --unique key
 Address varchar(200) NOT NULL,
 DeptID int FOREIGN KEY REFERENCES Department(DeptID) --foreign key
)
```

Attributes in DB

What Does Attribute Mean?

In general, an attribute is a characteristic. In a database management system (DBMS), an attribute refers to a database component, such as a table.

It also may refer to a database field. Attributes describe the instances in the column of a database.

Techopedia Explains Attribute

In relational databases, attributes are the describing characteristics or properties that define all items pertaining to a certain category applied to all cells of a column.

The rows, instead, are called tuples, and represent data sets applied to a single entity to uniquely identify each item. Attributes are, therefore, the characteristics of every individual tuple that help describe its unique properties.

Think of a table in a relational database as being analogous to an electronic spreadsheet. An attribute is simply one non-null cell in the spreadsheet, or the conjunction of a column and row.

It stores only one piece of data about the object represented by the table in which the attribute belongs. For example, the tuple can be an Invoice entity. The attributes of an invoice might be Price, Number, Date or Paid/unpaid.

Beyond the self-explanatory simple or single-valued attributes, there are several types of attributes available.

- **Composite attribute:** is an attribute composed of several other simple attributes. For example, the Address attribute of an Employee entity could consist of the Street, City, Postal code and Country attributes.

- **Multivalued attribute:** is an attribute where more than one description can be provided. For example, an Employee entity may have more than one Email ID attributes in the same cell.

- **Key attribute or primary attribute:** is an ID, key, letter or number that uniquely identifies that item. For example, it can be the number of a certain invoice (e.g. the individual ID of that invoice). A table that contains a single key attribute is considered a strong entity. However, a table might contain more than one key attribute if it's derived from other tables.

- **Derived attribute**: as the name implies, these are derived from other attributes, either directly or through specific formula results. For example, the Age attribute of an Employee could be derived from the Date of Birth attribute. In other instances, a formula might calculate the VAT of a certain payment, so that whenever the cell with the attribute Payment is filled, the cell with the derived attribute VAT automatically calculates its value.