1. INTRODUCTION TO RDBMS

- ❑ What is DBMS?
- ❑ Data Models
- ❑ Relational database management system (RDBMS)
- ❑ Relational Algebra
- ❑ Structured query language (SQL)

What Is DBMS?

Data is one of the most important assets of a company. It is very important to make sure data is stored and maintained accurately and quickly. DBMS (**D**ata**b**ase **M**anagement **S**ystem) is a system that is used to store and manage data.

A DBMS is a set of programs that is used to store and manipulation data. Manipulation of data include the following:

- ❑ Adding new data, for example adding details of new student.
- ❑ Deleting unwanted data, for example deleting the details of students who have completed course.
- ❑ Changing existing data, for example modifying the fee paid by the student.

A DBMS provides various functions like data security, data integrity, data sharing, data concurrence, data independence, data recovery etc. However, all database management systems that are now available in the market like Sybase, Oracle, and MS-Access do not provide the same set of functions, though all are meant for data management.

Database managements systems like Oracle, DB2 are more powerful and meant for bigger companies. Whereas, database management systems like MS-Access are meant for small companies. So one has to choose the DBMS depending upon the requirement.

Features of DBMS

The following are main features offered by DBMS. Apart from these features different database management systems may offer different features. For instance, Oracle is increasing being fine-tuned to be the database for Internet applications. This may not be found in other database management systems. These are the general features of database management systems. Each DBMS has its own way of implementing it. A DBMS may have more features the features discussed here and may also enhance these features.

Support for large amount of data

Each DBMS is designed to support large amount of data. They provide special ways and means to store and manipulate large amount of data. Companies are trying to store more and more amount of data. Some of this data will have to be online (available every time).

In most of the cases the amount of data that can be stored is not actually constrained by DBSM and instead constrained by the availability of the hardware. For example, Oracle can store terabytes of data.

Data sharing, concurrency and locking

DBSM also allows data to be shared by two or more users. The same data can be accessed by multiple users at the same time – data concurrency. However when same data is being manipulated at the same time by multiple users certain problems arise. To avoid these problems, DBMS locks data that is being manipulated to avoid two users from modifying the same data at the same time.

The locking mechanism is transparent and automatic. Neither we have to inform to DBMS about locking nor we need to know how and when DBMS is locking the data. However, as a programmer, if we can know intricacies of locking mechanism used by DBMS, we will be better programmers.

Data Security

While DBMS allowing data to be shared, it also ensures that data in only accessed by authorized users. DBMS provides features needed to implement security at the enterprise level. By default, the data of a user cannot be accessed by other users unless the owner gives explicit permissions to other users to do so.

Data Integrity

Maintaining integrity of the data is an import process. If data loses integrity, it becomes unusable and garbage. DBMS provides means to implement rules to maintain integrity of the data. Once we specify which rules are to be implemented, then DBMS can make sure that these rules are implemented always.

Three integrity rules (discussed later in this chapter) – domain, entity and referential are always supported by DBMS.

Fault tolerance and recovery

DBMS provides great deal of fault tolerance. They continue to run in spite of errors, if possible, allowing users to rectify the mistake in the mean time.

DBSM also allows recovery in the event of failure. For instance, if data on the disk is completely lost due to disk failure then also data can be recovered to the point of failure if proper back up of the data is available.

Support for Languages

DBMS supports a data access and manipulation language. The most widely used data access language for RDBMS (relational database management systems) is SQL. We will discuss more about RDBMS and SQL later in this chapter.

DBMS implementation of SQL will be compliant with SQL standards set by ANSI.

Apart from supporting a non-procedural language like SQL to access and manipulate data DBMS now a days also provides a procedural language for data processing. Oracle supports PL/SQL and SQL Server provides T-SQL.

Entity and Attribute

An entity is any object that is stored in the database. Each entity is associated with a collection of attributes. For example, if you take a data of a training institute, student is an entity as we store information about each student in the database. Each student is associated with certain values such as roll number, name, course etc., which are called as attributes of the entity.

There will be relationship among entities. The relationship between entities may be one-to-one, one-to-many or many-to-many.

If you take entities student, batch and subject, the following are the possible relationships.

There is one-to-one relationship between batch and subject. One batch is associated with only one subject.
Three is one-to-many relationship between batch and student entities. One batch may contain many students.

There is many-to-many relationship between student and subject entities. A single student may take many subjects and a single subject may be taken by multiple students.

Data Models

Data model is a way of storing and retrieving the data. There are three different data models. Data models differ in the way they allow users to view and manipulate relationships between entities. Each has its own way of storing the data. The following are the three different data models:

- ❑ Hierarchical
- ❑ Network
- ❑ Relational

Hierarchical

In this model, data is stored in the form of a tree. The data is represented by parent-child relation ship. Each tree contains a single root record and one or more subordinate records. For example, each batch is root and students of the batch will be subordinates.

This model supports only one-to-many relationship between entities.

This was used in IBM's Information *management system,* IMS.

Network

Data is stored along with pointers, which specify the relationship between entities. This was used in Honeywell's *Integrated Data Store*, IDS.

This model is complex. It is difficult to understand both the way data is stored and the way data is manipulated. It is capable of supporting many-to-many relationship between entities, which hierarchical model doesn’t.

Relational

This stores data in the form of a table. Table is a collection of rows and columns. We will discuss more about relational model in the next second.

Relational Database Management System (RDBMS)

A DBMS that is based on *relational model* is called as RDBMS. Relation model is most successful mode of all three models. Designed by E.F. Codd, relational model is based on the theory of sets and relations of mathematics.

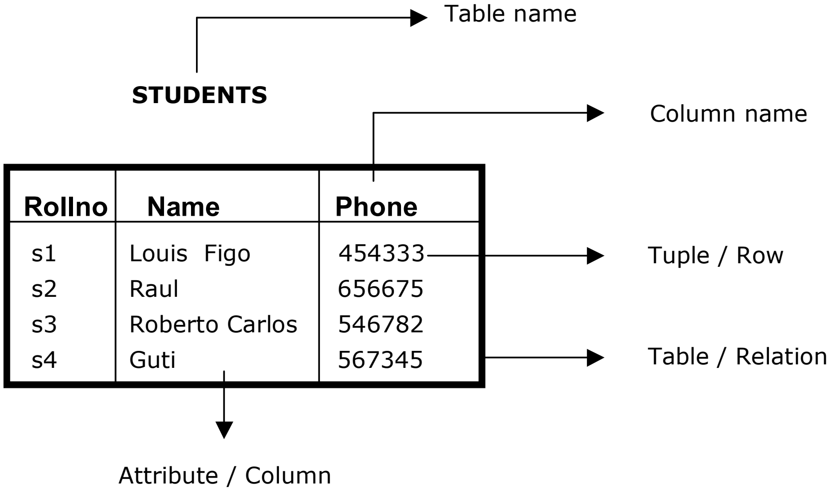
Relational model represents data in the form a table. A table is a two dimensional array containing rows and columns. Each row contains data related to an entity such as a student. Each column contains the data related to a single attribute of the entity such as student name.

One of the reasons behind the success of relational model is its simplicity. It is easy to understand the data and easy to manipulate.

Another important advantage with relational model, compared with remaining two models is, it doesn’t bind data with relationship between data item. Instead it allows you to have dynamic relationship between entities using the values of the columns.

Almost all Database systems that are sold in the market, now- a-days, have either complete or partial implementation of relational model.

Figure 1 shows how data is represented in relational model and what are the terms used to refer to various components of a table. The following are the terms used in relational model.



STUDENTS

Rollno	Name	Phone
s1	Louis Figo	454333
s2	Raul	656675
s3	Roberto Carlos	546782
s4	Guti	567345

Figure 1: A table in relational model.

Tuple / Row

A single row in the table is called as tuple. Each row represents the data of a single entity.

Attribute / Column

A column stores an attribute of the entity. For example, if details of students are stored then student name is an attribute; course is another attribute and so on.

Column Name

Each column in the table is given a name. This name is used to refer to value in the column.

Table Name

Each table is given a name. This is used to refer to the table. The name depicts the content of the table.

The following are two other terms, primary key and foreign key, that are very important in relational model.

Primary Key

A table contains the data related entities. If you take STUDETNS table, it contains data related to students. For each student there will be one row in the table. Each student's data in the table must be uniquely identified. In order to identify each entity uniquely in the table, we use a column in the table. That column, which is used to uniquely identify entities (students) in the table is called as primary key.

In case of STUDENTS table (see figure 1) we can use ROLLNO as the primary key as it in not duplicated.

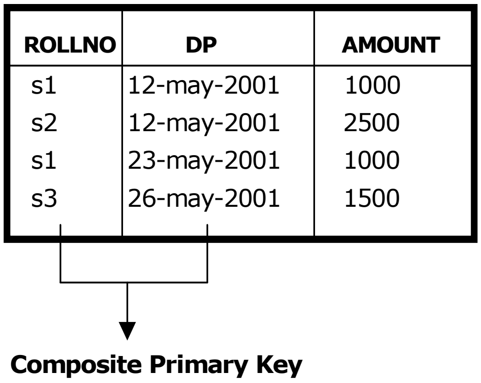
So a primary key can be defined as a **set of columns used to uniquely identify rows of a table.**

Some other examples for primary keys are account number in bank, product code of products, employee number of an employee.

Composite Primary Key

In some tables a single column cannot be used to uniquely identify entities (rows). In that case we have to use two or more columns to uniquely identify rows of the table. When a primary key contains two or more columns it is called as composite primary key.

In figure 2, we have PAYMENTS table, which contains the details of payments made by the students. Each row in the table contains roll number of the student, payment date and amount paid. Neither of the columns can uniquely identify rows. So we have to combine ROLLNO and DP to uniquely identify rows in the table. As primary key is consisting of two columns it is called as composite primary key.



PAYMENTS

ROLLNO	DP	AMOUNT
s1	12-may-2001	1000
s2	12-may-2001	2500
s1	23-may-2001	1000
s3	26-may-2001	1500

Composite Primary Key

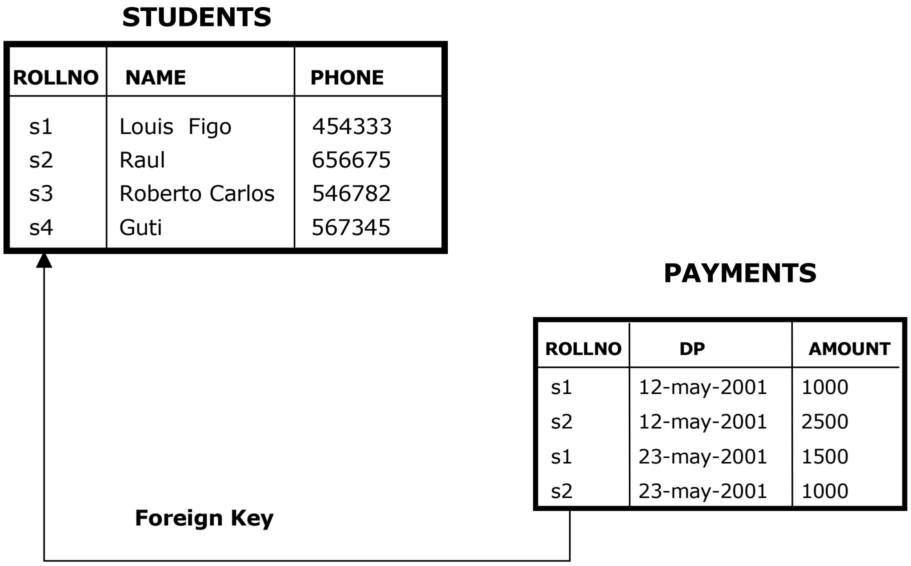
Figure 2: Composite Primary Key

Foreign Key

In relational model, we often store data in different tables and put them together to get complete information. For example, in PAYMENTS table we have only ROLLNO of the student. To get remaining information about the student we have to use STUDETNS table. Roll number in PAYMENTS table can be used to obtain remaining information about the student.

The relationship between entities student and payment is one-to-many. One student may make payment for many times. As we already have ROLLNO column in PAYMENTS table, it is possible to join with STUDENTS table and get information about parent entity (student).

Roll number column of PAYMENTS table is called as *foreign key* as it is used to join PAYMENTS table with STUDENTS table. So foreign key is the key on the many side of the relationship.



STUDENTS

ROLLNO	NAME	PHONE
s1	Louis Figo	454333
s2	Raul	656675
s3	Roberto Carlos	546782
s4	Guti	567345

PAYMENTS

ROLLNO	DP	AMOUNT
s1	12-may-2001	1000
s2	12-may-2001	2500
s1	23-may-2001	1500
s2	23-may-2001	1000

Figure 3: Foreign Key

ROLLNO column of PAYMENTS table must derive its values from ROLLNO column of STUDENTS table.

When a child table contains a row that doesn’t refer to a corresponding parent key, it is called as orphan record. We must not have orphan records, as they are result of lack of data integrity.

Integrity Rules

Data integrity is to be maintained at any cost. If data loses integrity it becomes garbage. So every effort is to be made to ensure data integrity is maintained. The following are the main integrity rules that are to be followed.

Domain integrity

Data is said to contain domain integrity when the value of a column is derived from the domain. Domain is the collection of potential values. For example, column date of joining must be a valid date. All valid dates form one domain. If the value of date of joining is an invalid date, then it is said to violate domain integrity.

Entity integrity

This specifies that all values in primary key must be not null and unique. Each entity that is stored in the table must be uniquely identified. Every table must contain a primary key and primary key must be not null and unique.

Referential Integrity

This specifies that a foreign key must be either null or must have a value that is derived from corresponding parent key. For example, if we have a table called BATCHES, then ROLLNO column of the table will be referencing ROLLNO column of STUDENTS table. All the values of ROLLNO column of BATCHES table must be derived from ROLLNO column of STUDENTS table. This is because of the fact that no student who is not part of STUDENTS table can join a batch

Relational Algebra

A set of operators used to perform operations on tables is called as ***relational algebra***. Operators in relational algebra take one or more tables as parameters and produce one table as the result.

The following are operators in relational algebra:

- ❑ Union
- ❑ Intersect
- ❑ Difference or minus
- ❑ Project
- ❑ Select
- ❑ Join

Union

This takes two tables and returns all rows that are belonging to either first or second table (or both). See figure 4.

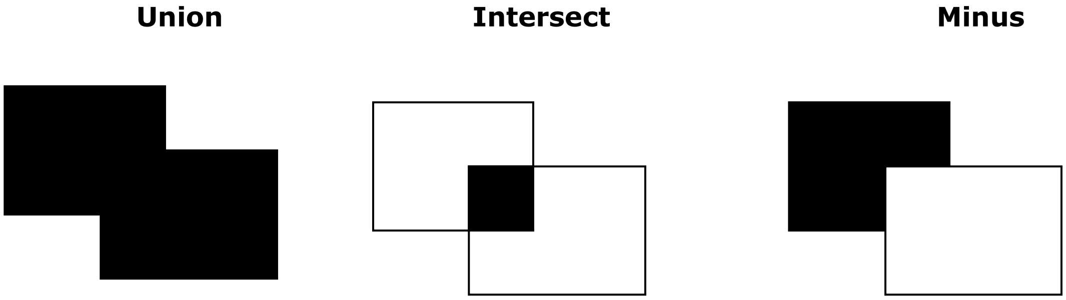


Figure 4: Union, Intersect and Minus

Intersect

This takes two tables and returns all rows that are belonging to first and second table. See figure 4.

Difference or Minus

This takes two tables and returns all rows that exist in the first table and not in the second table. See figure 4.

Project

Takes a single table and returns the vertical subset of the table. See figure 1.5.

Select

Takes a single table and returns a horizontal subset of the table. That means it returns only those rows that satisfy the condition. See figure 1.5.

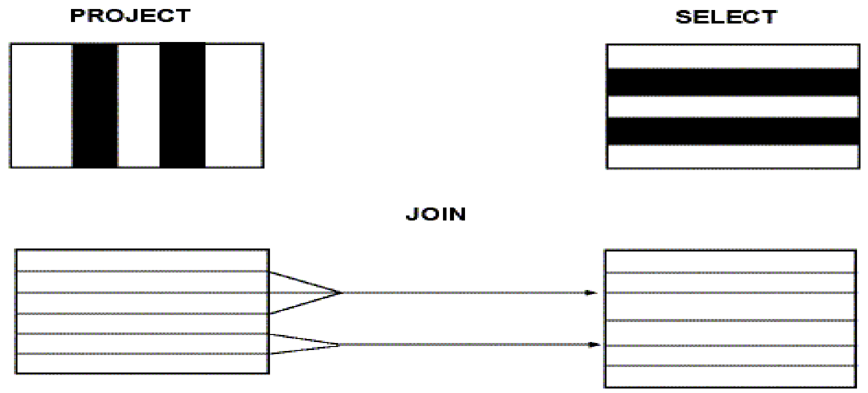


Figure 5: Project, Select and Join

Join

Rows of two table are combined based on the given column(s) values. The tables being joined must have a common column. See figure 5.

***Note**: See chapter 3, for SELECT and PROJECT, chapter 9 for JOIN, UNION, INTERSECT and MINUS.*

Structured Query Language (SQL)

Almost all relational database management systems use SQL (Structured Query Language) for data manipulation and retrieval. SQL is the standard language for relational database systems. SQL is a non-procedural language, where you need to concentrate on what you want, not on how you get it. Put it in other way, you need not be concerned with procedural details.

SQL Commands are divided into four categories, depending upon what they do.

- ❑ DDL (Data Definition Language)
- ❑ DML (Data Manipulation Language)
- ❑ DCL (Data Control Language)
- ❑ Query (Retrieving data)

DDL commands are used to define the data. For example, CREATE TABLE.

DML commands such as, INSERT and DELETE are used to manipulate data.

DCL commands are used to control access to data. For example, GRANT.

Query is used to retrieve data using SELECT.

DML and Query are also collectively called as DML. And DDL and DCL are called as DDL.

Data processing Methods

Data that is stored is processed in three different ways. Processing data means retrieving data and deriving information from data. Depending upon where it is done and how it is done, there are three methods.

- ❑ Centralized data processing
- ❑ De-centralized data processing
- ❑ Distributed data processing

Centralized data processing

In this method the entire data is stored in one place and processed there itself. Mainframe is best example for this kind of processing. The entire data is stored and processed on mainframe. All programs, invoked from clients (dumb terminals), are executed on the mainframe and data is also stored in mainframe.

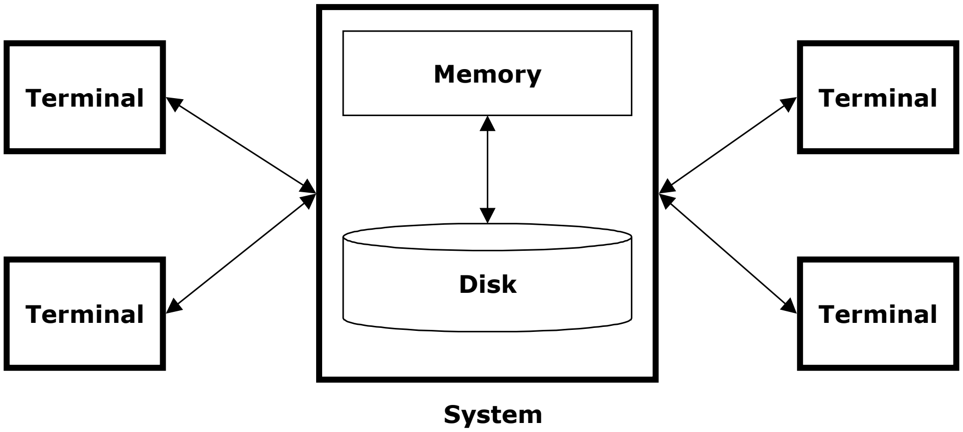


Figure 6: Centralized data processing.

As you can see in figure 6, all terminals are attached to mainframe. Terminals do not have any processing ability. They take input from users and send output to users.

Decentralized data processing

In this data is processed at various places. A typical example is each department containing its own system for its own data processing needs. See figure 7, for an example of decentralized data processing. Each department stores data related to itself and runs all programs that process its data. But the biggest drawback of this type of data processing is that data is to be duplicated. As common data is to be stored in each machine, it is called as ***redundancy***. This redundancy will cause data inconsistency. That means the data stored by two departments will not agree with each other.

Data in this mode is duplicated, as there is no means to store common data in one place and access from all machines.

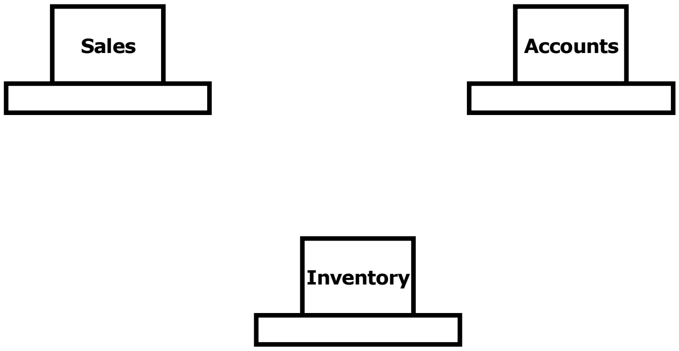


Figure 7: Decentralized Data Processing.

Distributed Data Processing (Client/Server)

In this data processing method, data process is distributed between client and server. Server takes care of managing data. Client interacts with user. For example, if you assume a process where we need to draw a graph to show the number of students in a given month for each subject, the following steps will take place:

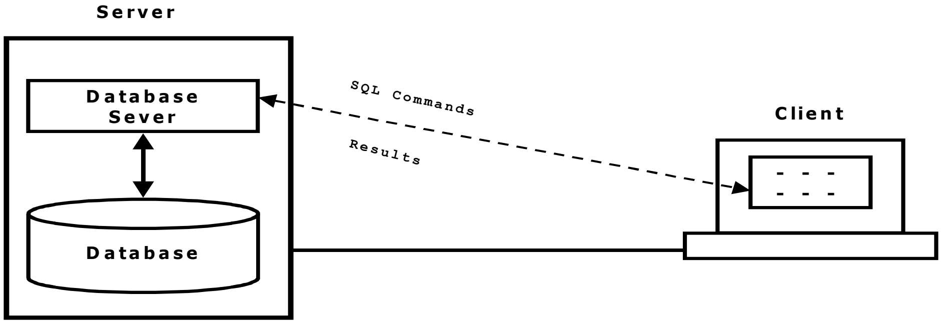


Figure 8: Distributed data processing.

1. First, client interacts with user and takes input (month name) from user and then passes it to server.
2. Server then will query the database to get data related to the month, which is sent to server, and will send data back to client.
3. The client will then use the data retrieved from database to draw a graph.

If you look at the above process, the client and server are equally participating in the process. That is the reason this type of data processing is called as distributed. The process is evenly distributed between client and server. Client is a program written in one of the font-end tools such as Visual basic or Delphi. Server is a database management system such as Oracle, SQL Server etc. The language used to send commands from client to server is SQL (see figure 8).

This is also called as two-tier client/server architecture. In this we have only two tiers (layers) one is server and another is client.

The following is an example of 3-tier client server, where client interacts with user on one side and interacts with application server on another side. Application, which processes and validates data, takes the request from client and sends the request in the language understood by database server. Application servers are generally object oriented. They expose a set of object, whose methods are to be invoked by client to perform the required operation.

Application server takes some burden from database server and some burden from client.

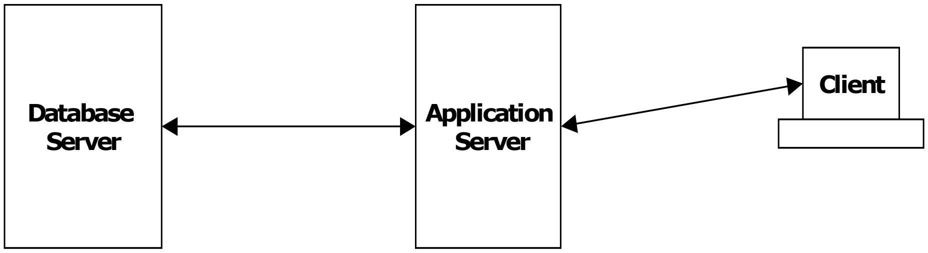


Figure 9: 3-tier client-server architecture.

In 3-tier client/server architecture, database server and application server may reside on different machines or on the same machine. Since the advent of web application we are also seeing more than 3-tiers, which is called as n-tier architecture. For example, the following is the sequence in a typical web application.

1. Client- web browser, sends request to web server.
2. Web server executes the request page, which may be an ASP or JSP.
3. ASP or JSP will access application server.
4. Application server then will access database server.

Summary

A DBMS is used to store and manipulate data. A DBMS based on relational model is RDBMS. Primary key is used for unique identification of rows and foreign key to join tables. Relational algebra is a collection of operators used to operate on tables. We will see how to practically use these operators in later chapter.

SQL is a language commonly used in RDBMS to store and retrieve data. In my opinion, SQL is one of the most important languages if you are dealing with an RDBMS because total data access is done using SQL.

Exercises

1. _________ Designed relational model.
2. Data models are ________, ________ and _________.
3. Composite primary key is ____________.
4. A row is otherwise known as _________.
5. How many tables does SELECT operator take? _____________.
6. _______ is an example for an RDBMS.
7. SQL command used to create table belongs to _______ category.
8. ________ is the key used to join a child table with parent table.

9. _______ is the standard language for RDBMS.

10. Client/server architecture is an example of _____________ data processing method.

11. Centralized database is used in both _____________ and ______________ data processing methods.

12. What is a domain? _________________