FOXPRO:-

The 1st DBMS software for PC was developed by "ASHTON-TATE" in 1979 called dbase 2 . then that company introduced the updated version of d base 3 as d base 2 in 1984 . to get a share of the expanding of DBMS market for the PC segment for software . the come out with fox base + that was compatible with d BASE 3 then it was updated and the new software was called foxpro. Gradually new version of foxpro were released to market earlier version of foxoro are available for dos , Foxpro , 2.5 ,,2.6. it can run by done different operating system as dos, UNIX . in 1992 for software in merge with Microsoft and released a new version of foxpro known as visual foxpro 3.0, ms gradually release advance version of visual foxpro as 4.0 ,5.0 ,6.0, 7.0.

using this we can design a database as per user requirement of any organization . it also help you to enter and manage data in database in database file using this we can edit, view, change, delete, using of data from the database file. We can also create & report using data of the database file. According to user requirement. We can also write program to create customize software in foxpro.

DBMS CONCEPTS

A Database Management System (DBMS) is basically a collection of programs that enables users to store, modify, and extract information from a database as per the requirements. DBMS is an intermediate layer between programs and the data. Programs access the DBMS, which then accesses the data. There are different types of DBMS ranging from small systems that run on personal computers to huge systems that run on mainframes. The following are main examples of database applications:

- Computerized library systems
- Automated teller machines
- Flight reservation systems
- Computerized parts inventory systems

A database management system is a piece of software that provides services for accessing a database, while maintaining all the required features of the data. Commercially available Database management systems in the market are dbase, FoxPro, IMS and Oracle, MySQL, SQL Servers and DB2 etc.

These systems allow users to create update, and extract information from their databases.

Compared to a manual filing system, the biggest advantages to a computerized database

system are speed, accuracy, and' accessibility.

Advantages of DBMS

The database management system has promising potential advantages, which are explained below:

1. Controlling Redundancy: In file system, each application has its own private files, which cannot be shared between multiple applications. 1:his can often lead to considerable redundancy in the stored data, which results in wastage of storage space. By having centralized database most of this can be avoided. It is not possible that all redundancy should be eliminated. Sometimes there are sound business and technical reasons for· maintaining multiple copies of the same data. In a database system, however this redundancy can be controlled.

For example: In case of college database, there may be the number of applications like General Office, Library, Account Office, Hostel etc. Each of these applications may maintain the following information into own private file applications:

General Office	Library	Hostel	Account Office
Roll No Name Class Father_Name Date_of_Birth Address Phone No Previous Record Attendance Marks etc.	Roll No Name Class Address Date at Birth Phone No No of books issued Fine etc	Roll No Name Class Father_Name Date of Birth Address Phone No Mess bill RoomNo etc.	Roll No Name Class Address Phone No Fee Installments Discount Balance Total etc.

It is clear from the above file systems, that there is some common data of the student which has to be mentioned in each application, like Rollno, Name, Class, Phone_No~ Address etc. This will cause the problem of redundancy which results in wastage of storage space and difficult to maintain, but in case of centralized database, data can be shared by number of applications and the whole college can maintain its computerized data with the following database:

General Office	Library	Hostel	Account Office
Rollno Name Class Father_Name Address Phone - No Date_of_birth Previous_Record Attendance Marks etc.	Rollno No_of_books_issued Fine etc.	Rollno RoomNo Mess_Bill etc.	Rollno Fee Installments Discount Balance Total etc.

It is clear in the above database that Rollno, Name, Class, Father_Name, Address,

Phone_No, Date_of_birth which are stored repeatedly in file system in each application, need not be stored repeatedly in case of database, because every other application can access this information by joining of relations on the basis of common column i.e. Rollno. Suppose any user of Library system need the Name, Address of any particular student and by joining of Library and General Office relations on the basis of column Rollno he/she can easily retrieve this information.

Thus, we can say that centralized system of DBMS reduces the redundancy of data to great extent but cannot eliminate the redundancy because RollNo is still repeated in all the relations.

2. Integrity can be enforced: Integrity of data means that data in database is always accurate, such that incorrect information cannot be stored in database. In order to maintain the integrity of data, some integrity constraints are enforced on the database. A DBMS should provide capabilities for defining and enforcing the constraints.

For Example: Let us consider the case of college database and suppose that college having only BTech, MTech, MSc, BCA, BBA and BCOM classes. But if a \.,ser enters the class MCA, then this incorrect information must not be stored in database and must be prompted that this is an invalid data entry. In order to enforce this, the integrity constraint must be applied to the class attribute of the student entity. But, in case of file system tins constraint must be enforced on all the application separately (because all applications have a class field).

In case of DBMS, this integrity constraint is applied only once on the class field of the

General Office (because class field appears only once in the whole database), and all other applications will get the class information about the student from the General Office table so the integrity constraint is applied to the whole database. So, we can conclude that integrity constraint can be easily enforced in centralized DBMS system as compared to file system.

3. Inconsistency can be avoided : When the same data is duplicated and changes are made at one site, which is not propagated to the other site, it gives rise to inconsistency and the two entries regarding the same data will not agree. At such times the data is said to be inconsistent. So, if the redundancy is removed chances of having inconsistent data is also removed.

Let us again, consider the college system and suppose that in case of General_Office file

it is indicated that Roll_Number 5 lives in Amritsar but in library file it is indicated that

Roll_Number 5 lives in Jalandhar. Then, this is a state at which tIle two entries of the same object do not agree with each other (that is one is updated and other is not). At such time the database is said to be inconsistent.

An inconsistent database is capable of supplying incorrect or conflicting information. So there should be no inconsistency in database. It can be clearly shown that inconsistency can be avoided in centralized system very well as compared to file system ..

Let us consider again, the example of college system and suppose that RollNo 5 is .shifted from Amritsar to Jalandhar, then address information of Roll Number 5 must be updated, whenever Roll number and address occurs in the system. In case of file system, the information must be updated separately in each application, but if we make updation only at three places and forget to make updation at fourth application, then the whole system show the inconsistent results about Roll Number 5.

In case of DBMS, Roll number and address occurs together only single time in General_Office table. So, it needs single updation and then an other application retrieve the address information from General_Office which is updated so, all application will get the current and latest information by providing single update operation and this single update operation is propagated to the whole database or all other application automatically, this property is called as Propagation of Update.

We can say the redundancy of data greatly affect the consistency of data. If redundancy is less, it is easy to implement consistency of data. Thus, DBMS system can avoid inconsistency to great extent.

4. Data can be shared: As explained earlier, the data about Name, Class, Father __name etc. of General_Office is shared by multiple applications in centralized DBMS as compared to file system so now applications can be developed to operate against the same stored data. The applications may be developed without having to create any new stored files.

5. Standards can be enforced : Since DBMS is a central system, so standard can be enforced easily may be at Company level, Department level, National level or International level. The standardized data is very helpful during migration or interchanging of data. The file system is an independent system so standard cannot be easily enforced on multiple independent applications.

6. Restricting unauthorized access: When multiple users share a database, it is likely that some users will not be authorized to access all information in the database. For example, account office data is often considered confidential, and hence only authorized persons are allowed to access such data. In addition, some users may be permitted only to retrieve data, whereas other are allowed both to retrieve and to update. Hence, the type of access operation retrieval or update must also be controlled. Typically, users or user groups are given account numbers protected by passwords, which they can use to gain access to the database. A DBMS should provide a security

and authorization subsystem, which the DBA uses to create accounts and to specify account restrictions. The DBMS should then enforce these restrictions automatically.

7. Solving Enterprise Requirement than Individual Requirement: Since many types of users with varying level of technical knowledge use a database, a DBMS should provide a variety of user interface. The overall requirements of the enterprise are more important than the individual user requirements. So, the DBA can structure the database system to provide an overall service that is "best for the enterprise".

For example: A representation can be chosen for the data in storage that gives fast access for the most important application at the cost of poor performance in some other application. But, the file system favors the individual requirements than the enterprise requirements

8. Providing Backup and Recovery: A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery. For example, if the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the .database is restored to the state it was in before the program started executing.

9. Cost of developing and maintaining system is lower: It is much easier to respond to unanticipated requests when data is centralized in a database than when it is stored in a conventional file system. Although the initial cost of setting up of a database can be large, but the cost of developing and maintaining application programs to be far lower than for similar service using conventional systems. The productivity of programmers can be higher in using non-procedural languages that have been developed with DBMS than using procedural languages.

10. Data Model can be developed : The centralized system is able to represent the complex data and interfile relationships, which results better data modeling properties. The data madding properties of relational model is based on Entity and their Relationship, which is discussed in detail in chapter 4 of the book.

11. **Concurrency Control :** DBMS systems provide mechanisms to provide concurrent access of data to multiple users.

Disadvantages of DBMS

The disadvantages of the database approach are summarized as follows:

1. Complexity : The provision of the functionality that is expected of a good DBMS makes the DBMS an extremely complex piece of software. Database designers, developers, database administrators and end-users must understand this functionality to take full advantage of it. Failure to understand the system can lead to bad design decisions, which can have serious consequences for an organization.

2. Size : The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently.

3. Performance: Typically, a File Based system is written for a specific application, such as invoicing. As result, performance is generally very good. However, the DBMS is written to be more general, to cater for many applications rather than just one. The effect is that some applications may not run as fast as they used to.

4. Higher impact of a failure: The centralization of resources increases the vulnerability of the system. Since all users and applications rely on the ~vailabi1ity of the DBMS, the failure of any component can bring operations to a halt.

5. Cost of DBMS: The cost of DBMS varies significantly, depending on the environment and functionality provided. There is also the recurrent annual maintenance cost.

6. Additional Hardware costs: The disk storage requirements for the DBMS and the database may necessitate the purchase of additional storage space. Furthermore, to achieve the required performance it may be necessary to purchase a larger machine, perhaps even a machine dedicated to running the DBMS. The procurement of additional hardware results in further expenditure.

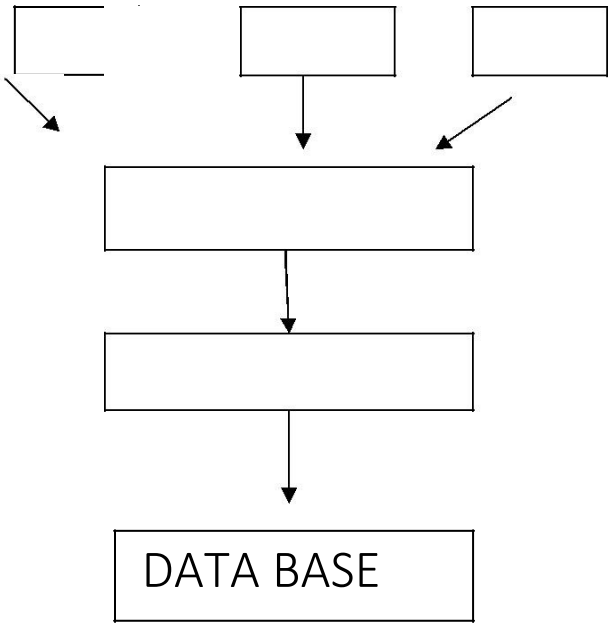
7. Cost of Conversion: In some situations, the cost oftlle DBMS and extra hardware may be insignificant compared with the cost of converting existing applications to run on the new DBMS and hardware. This cost also includes the cost of training staff to use these new systems and possibly the employment of specialist staff to help with conversion and running of the system. This cost is one of the main reasons why some organizations feel tied to their current systems and cannot switch to modern database technology.

Architecture of DBMS:-

External level:-

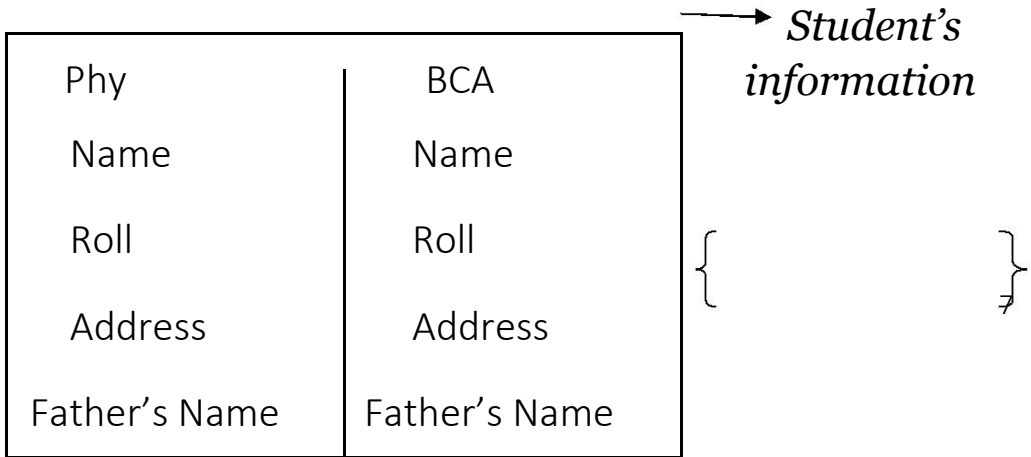
conceptual level:-

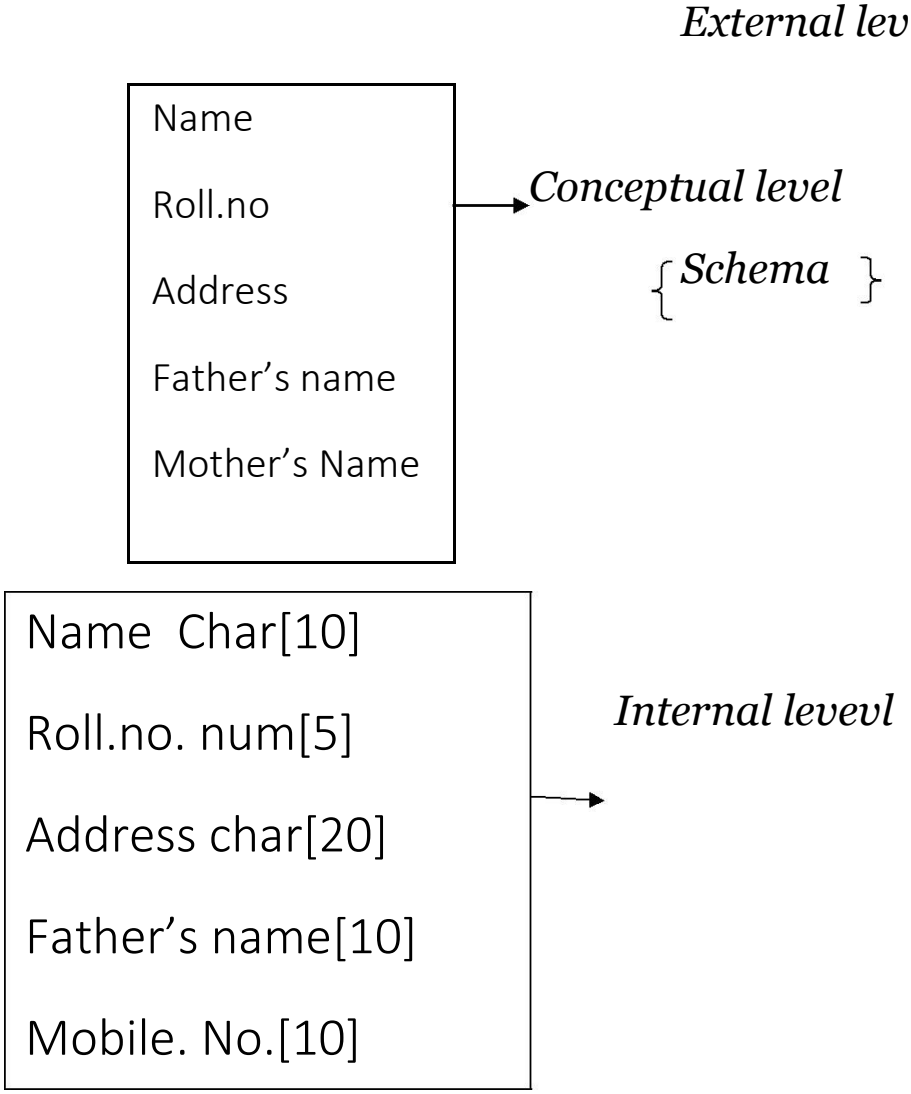
Internal level:-



The database architecture is divided in to three parts:-

- External level:- it is the top level of a database architecture. There each user only consult with it's database using this we can access any information present in it's database . there are more than one external level database present in any database.
- Conceptual level:- the collection of all external level database is called conceptual level database at this level all information regarding the organization for which the data base is created . there are only one conceptual level database present in the database of an organization .
- Internal level :- it's the last level of the database. It shows the physical storage of the database. These we show the data types length the data present in the data base.





Schema :-

Schema is a logical database description and charts of the data that are use. It gives the name of entities and attribute and specify the relation between them. It's frame work in to which the values of the data items can be filled like an information display system such as that giving arrival & departure time and airport & railway station. There are only one schema of any database.

Sub – Schema:-

The sub-schema refers to the data items types and records types which are use in a particulars. Application or by a particular user . there for many different sub-schema can be derived from one schema. If the schema represented in root map of Delhi, showing measure, starical, sides, equations, intiation, railway station, etc. *A sub-schema could* be a similar map showing the root is from the railway station or the airport of the Lalkila.

RDBMS:-

{Relational data base management system}

Relation:-

A={ 1, 2, 3}

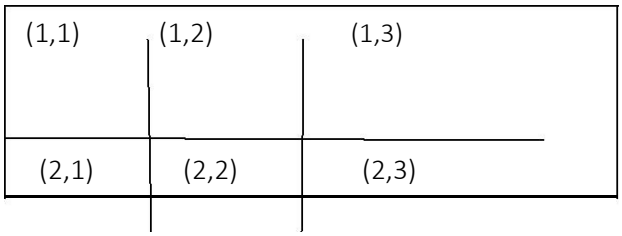
B={2, 4}

A^B={(1,2) (2,4) (2, 2) (1,4) (3,2) (3,4)}

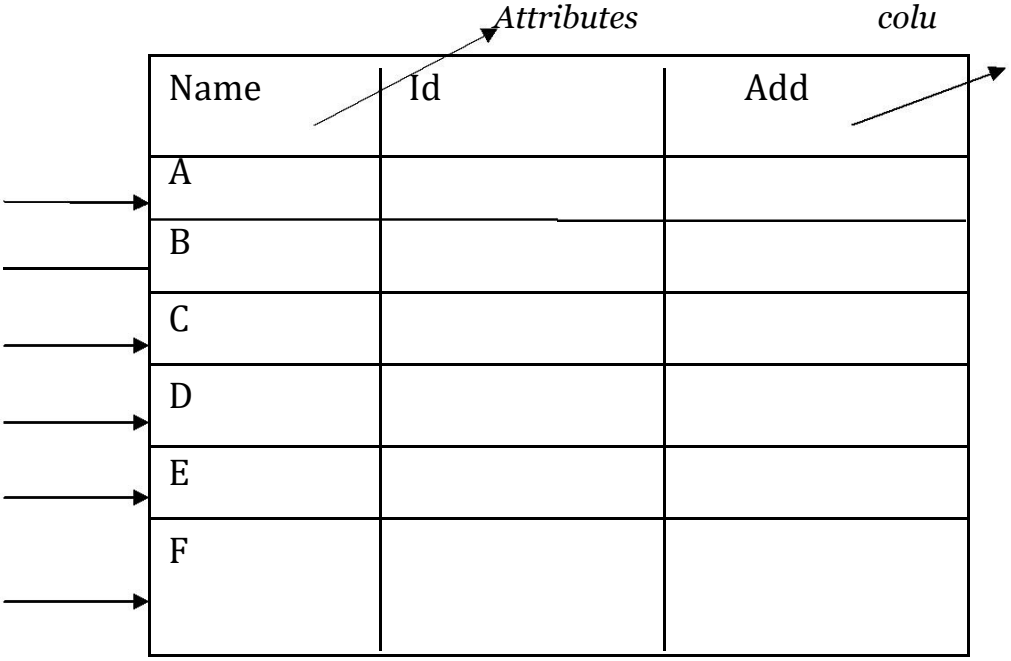
Relation is product of the two sets

1. *2.* *3.*

RDMS:-



It's a data base where we store in table like structure called relation. the column of table is called attributes and the name of table is called relation. The value of each attributes is called domain of the attributes.



Tuples

The row of the table is called tuple relation. The no. of attribute & the no of row (tuple) is called degree & cordinaling respectively.

Example:-

Domain of name (attribute)

A

B

C

D

Degree of relation :-3

Cordinality of relation :- 4

Each relation most follow the following properties:-

- It's column homogeneous in other word in any column of table items are of the same kind .
- All column of a table having different name of each other.
- All row of the tables are distance.

Codd's Rule:-

EDGOR FRANK "Ted" codd of the IBM 1972 designed 12 rules for a database design:-

1. The rule of granted access:- Every item of data must be logically addressable by resorting to combination of table name , primary key , value and column name.

Name	*Id*	*Fee*
A	*1*	*100*
B	*2*	*250*
C	*3*	*350*
D	*4*	*400*
E	*5*	*450*

2. The systematic treatment of null value:-it's fundamental to the DBMS that null value are supported in the representation of missing and in applicable information . this rules supports for null values must be conssistant through the DBMS and independent of data type.

   Null value of numeric data must have only bank we may e.g. define for permisable value for a column sex.

   M:-Male
   F:-Female
   X:-No data available

Y:-Not Application

NULL Value:- it's the value which is missing and does not equal to zero. Null != . zero , it is the missing

Value which not no add the movement of data entered in the data base.

Arithmetic expression using null value

Value + Null = Null

Value –Null = Null

Null + Null= Null

3. Information rule:- the relational database model of all information should be always in form of table data dictionary should be also in form of table.

4. The integrity rule:- the relational model includes two general integrity rule:-

- Entity integrity :- it is concerned with the primary key value. It is defined as follow it's product the insertion or updation of a value to the key . these value can not be duplicate value or a null value. It is used to uniquely identify the records of the database.
- Reference integrity :- it is concerned with the foreign key. It is attribute of relation having domain that are safe of the primary key of another or same relation. It allow to enter a duplicate and null value to the attribute on which the foreign key is

Example:-

Primary key

Name	Id	Fee		Name	ID	Fee
A	1	100				
B	2	21,000				
C	3	200				
D	4	150				
E	5	500				

from data dictionary , using SQL(structure query language).

6. Insertion , deletion and updation :- we can perform insertion , deletion and updation on the existing table. Using insertion , we can insert a new records to the table using deletion . we can delete the existing record of the table. We can also move any type of chances in the existing record , with the help of updation.

7. view updating rule:- It's a view is updated then the based table a view is automatically updated. View is an object of data base . it is realed on any existing table or views. It has not it's own data but it can access the data present in it dose not require physical storage to store it's data.

EMPLOYEE(Based table)

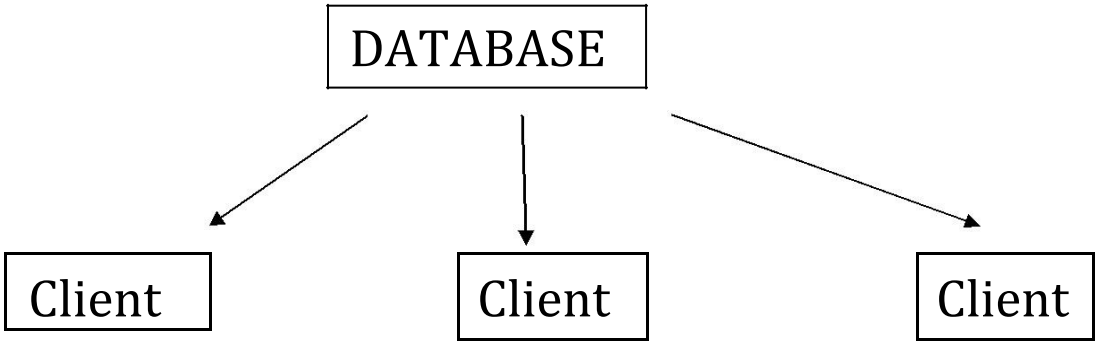
Name	ID	Salary
A	1	10,000
B	2	16,000
C	3	20,000
D	4	25,000
E	5	15,000

Select name , Id, for

Employee

Name	ID
A	1
B	2
C	3
D	4
E	5

8.The distribution rule:- data base system created on the relationship frame work which are well suited in two days client server database desing.



9.The comprechensive rule:- the RDMS most follow at is one language which include statement that can be express as character string, conforming to some will defined syntax that language must support the following function:-

- Data definition
- Data manipulation
- View definition
- Transaction boundary
- Integrity constant
- Authorization

10. logical data independent:- we can change any data present in conceptual schema can be change without effectly the existing external schema the change would be have above by the mapping between external & conceptual level. It include application program from operation such as comparing two records in to two or move records.

11.Physical data independence rule:- it indicates that physical storage structure or devices used for storing the data would be change without be change in the conceptual level or external level data base user access the database remain logically content whenever change to the storage representation of access methods to the data change.

12.Non – subversion rule:- it is a relational database management system supports a lower level language that permits for row at a time processing then this language must not be able to by pass any integrity rule or thus the RDMS is governed by relational rules that must be it's primary rule.

Name	Id	Add
A		
B		

13. All information must be present in a way that is data value.

SQL{Structure Query Language}

It's pronunciation is as "Sequeal" . it is the product of oracle IBM which has capacity to management system using this we can create a data base , manipulated the second in the data base access any information from the database.

These are following Parts of SQL:-

1. Data definition language:- Using this statement we can create a database object (Table , View , index) we can create , drop truncate , Alter, the object of data base.
2. Data manipulation language:- Using this we can manipulated the data of an object in a database we can perform insert a new value , delete & existing value record, modify the existing value, merge the record of an object of the database.
3. Data control language:- Using this we can current & remove the permission given to 1 form any user of an object.