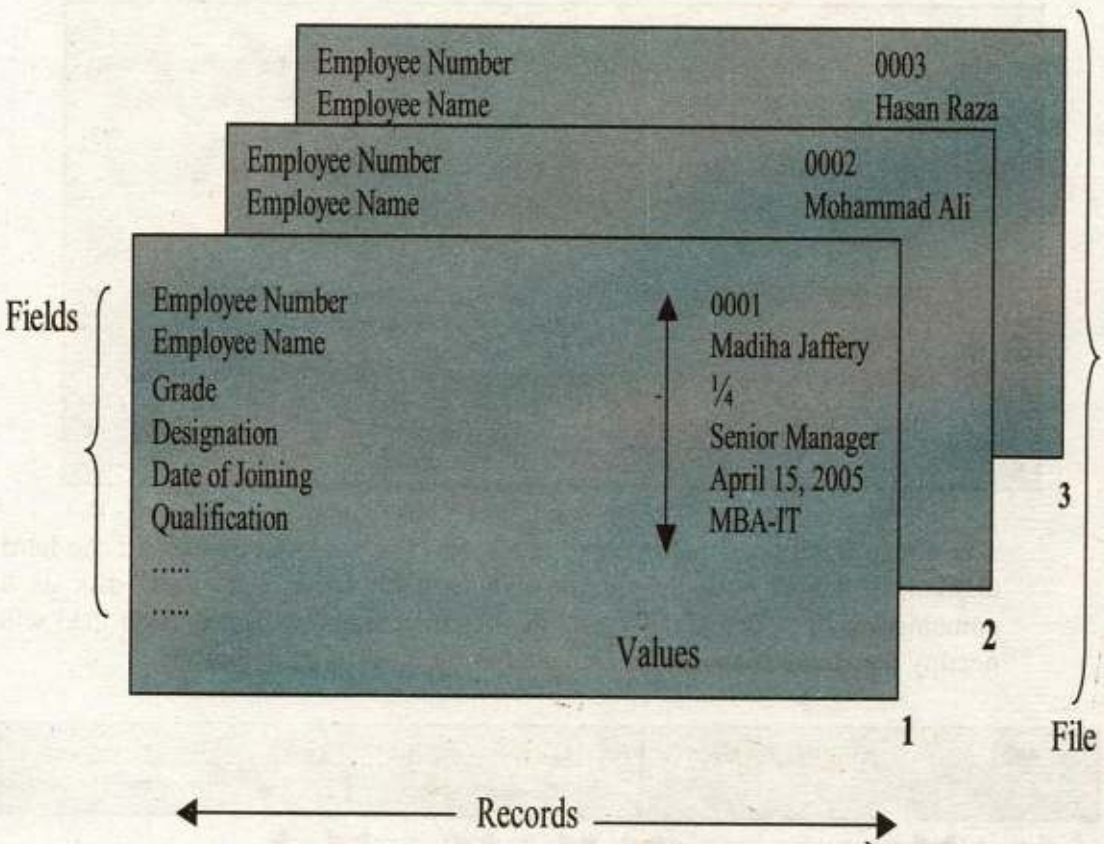
Chapter 2
BASIC CONCEPTS AND TERMINOLOGY OF DATABASES

2.1 OVERVIEW

In the previous chapter, we have discussed Files and Records in details. In fact, the concept of databases evolved from the old, traditional working of *File Management System* (FMS). Let us see how this *evolution* emerged by considering the same diagram of Record and File.



It is worth mentioning here that the file must have a *meaningful* name and all the field names in each record must be *unique* and *meaningful* too. For example, we

can give the following names to the above mentioned *file* and its *Record Fields*:

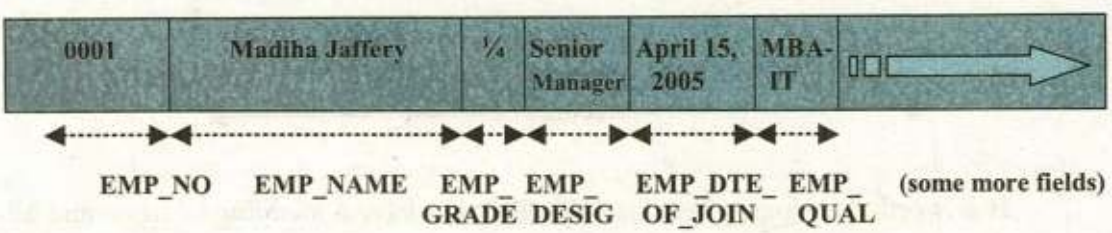
File name : EMP_BIO_INFO (Employee's bio-graphic information)
Employee Number : EMP_NO
Employee Name : EMP_NAME
Grade : EMP_GRADE
Designation : EMP_DESIG
Date of Joining : EMP_DTE_OF_JOIN
Qualification : EMP_QUAL

} *Some more fields*

Basic Definitions

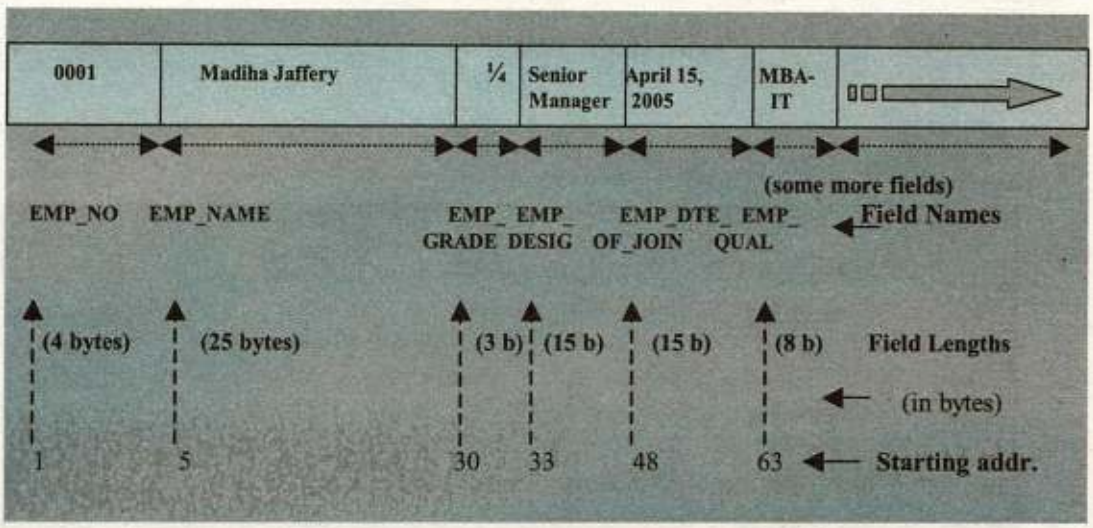
- **Field :** A field is a unit of data consisting of one or more charact i.e., Employee number Employee name or grade of an emplo in a record of the employee.
- **Record :** A collection of related data items treated as a single uni is called a record.
- **File:** A collection of related records treated as a single unit is called a file or a data set.

The above mentioned record can be shown (as it would appear on the hard disk) as follows. Actually, all the data will be saved onto hard disk as a combination of "0"s and "1"s i.e., the machine language form. Each field will occupy the space as would be allocated at the time of its definition.



2.1.1 Data Handling in File Management Systems

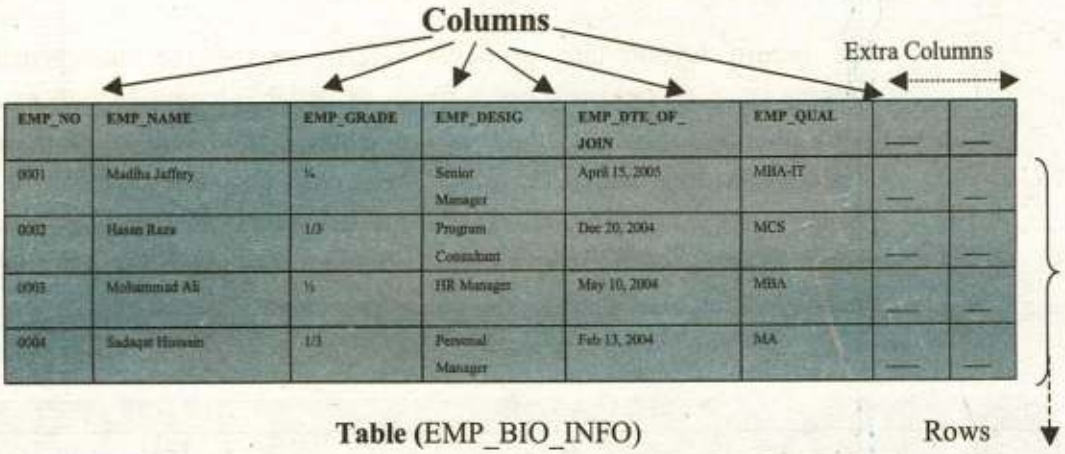
- The record layouts are properly defined in the file management systems i.e. each field is given a fixed- or variable-length sequences of bytes and they are put together contiguously in fixed- or variable-length collections called *records*. Thus, each field corresponds to a proper starting memory address. As the fields have already been given proper lengths, so the values of each field are determined within those memory addresses. See the example given below :



- Thus, the field names are used only as a "names reference" within the programs using them. Their values "flows" with them as the contents of the memory spaces they are occupying.

2.2 ATTRIBUTES, ROWS AND TABLES

In the late 1960s, the researchers came up with an idea of using *Relational Databases* instead of file structures. They gave the idea of defining a file as a "Two Dimensional" array (or *Table* having a unique name), placing all the fields as *columns* (having unique names) of that table and putting each record as a *row* into the table. Each row is also known as a *tuple* or *occurrence* in that table. Following these concepts, we can see the traditional file structure into a new and easy to manage database structure called a *Table* or *Relation*.



EMP_NO	EMP_NAME	EMP_GRADE	EMP_DESIG	EMP_DTE_OF_ JOIN	EMP_QUAL	___	___
0001	Madiha Jaffery	¼	Senior Manager	April 15, 2005	MBA-IT	___	___
0002	Hasan Raza	1/3	Program Consultant	Dec 20, 2004	MCS	___	___
0003	Mohammad Ali	½	HR Manager	May 10, 2004	MBA	___	___
0004	Sadaqat Hussain	1/3	Personal Manager	Feb 13, 2004	MA	___	___

Table (EMP_BIO_INFO)

So, consequently, we summarize the above discussion as follows :

- **Data Elements** : The fields or data items in databases are termed as data items, items, attributes or columns in database structures.
- Records in file management structures are termed *as rows or tuples in* database structures.
- Files or datasets in databases are termed as *tables, relations or data objects* in database structures.
- The collection of tables with some traditional files and some other necessary data objects is termed as a *database*.

2.3 RELATION or TABLE

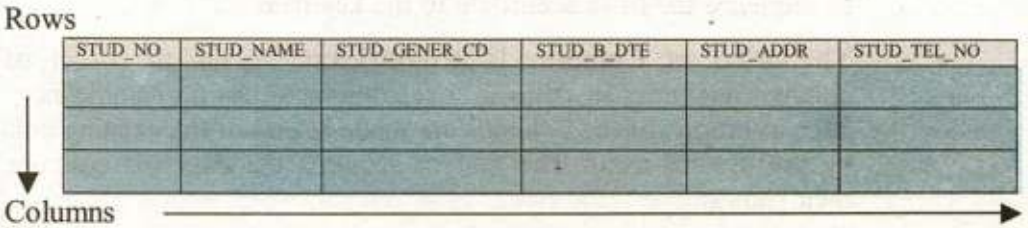
A two dimensional array or table of data containing descriptive information about an entity. The entity must have a unique identifier, which is composed of a combination of one or more attributes, and each attribute must have one and only one value. It is appropriate to define the word *Entity* here.

Entity: An *entity* is any thing about which you want to keep information in the database. Let us consider an example of "Student Information System", which has entities like student, teacher, course list, scholarships, time-tabling

etc. Thus, the entities involved in this case are the same and the entity "student" can be defined in the form of database modeling as follows :

- STUDENT (STUD_NO, STUD_NAME, STUD_GENDER_CD, STUD_B_DTE,
  STUD_ADDR, STUD_TEL_NO)

From the above given definition of entity, we can easily construct a two-dimensional array or a relation by converting all the attributes in the brackets into columns of the array, as follows :



In the above diagram, all the attributes are called columns of the table and all occurrences of the records are called rows.

For example, considering the entity 'STUDENT' as above, following table or relation can be constructed as a part of the database.

STUDENT

STUD_NO	STUD_NAME	STUD_GENER_CD	STUD_B_DTE	STUD_ADDR	STUD_TEL_NO
M001-F05	Ahmad Dar	M	12-10-1980	12, St#5,F10-I	9258481
F009-S04	Saba Afzal	F	31-03-1982	Abcdefgh	9292999
M100-S03	Adil Amin	M	28-02-1980	Gulberg, LHR	5566778

A relation STUDENT having *rows* (records) under *columns* (attributes)

2.3.1 Properties of Relation

A Relation or a Table which is the basis of a Relational DBMS, by definition must have certain inherent characteristics that form the basic for its underlying strength and flexibility. Because of these features, an application implemented by using such a system is much more flexible and can be easily

modified when alterations or enhancements to the underlying data base take place. These characteristics are:

- **No duplicate rows exist:** No two rows can be identical. Why to put two rows (records) for the same entity (e.g., a Person). It will also violate the definition of what a relation represents, as it says by definition that there must be a unique key for each row in a relation/table.
- **The order of Rows is insignificant:** There is no ordering or sequencing of the rows in the tables. The relational implementation of the tables support all required access mechanism i.e., it is not necessary to sequence the rows according to the key field.
- **The order of Columns is insignificant:** Again, the order of the columns/attributes in defining a relation/table has no significance. The later insertions of the columns are made at end of the existing columns by the system itself. The system acquires the data (of columns) by their names.
- **Columns/Attributes are all Elemental or Atomic:** All the intersections of Rows and Columns must have a (single) value. The nulls are inserted by the system at the time of later insertion of a column, which should immediately be replaced by zeros/spaces or valid values for that particular column.

2.4 VIEWS

Views are created by using SQL, which is a powerful database language, used for data definition and data manipulation purposes. The purpose of using views is purely to keep the data safe and secure from un-authorized and illegal users. The views provide the descriptions of relations that are not stored, but constructed as needed from stored relations.

To create a view, normally the following create sql command is used :

```
CREATE VIEW     STUDENT_VIEW_01 AS
    SELECT      STUD_NO, STUD_NAME, STUD_ADDR
    FROM        STUDENT
    WHERE       STUD_GENDER_CD = "M";
```

This will create a *view* from the STUDENT table for only **male** students, which can be used by the users according to the authorization given to them, leaving the original table aside, safe and secure.

2.5 INDEXES

It is another table created by the system developer/DBA containing the key attributes of the table for which the Index is created. It has a very vital role in the data

base management systems, especially in RDBMS. The important associations defined in the system make use of this. It helps the system run smooth and fast.

2.6 KEYS

A key is a single or combination of one or more fields and its purpose is to point/retrieve the data rows from the tables, according to the requirement.

Keys are defined in the relations/tables to access or sequence the stored data fast and smooth or to create the links between them.

Let us assume two relations/tables as follows to define different types of key.

PATIENT

PATNO	PNAME	PHCODE	LOCN	STATUS
191	ABID	ME1000	1000101	01
192	SALIM	ME1001	1000102	01
193	FAROOQ	ME1003	1000103	01
198	KASHIF	SU1008	1000104	01
201	ZAHID	ME1000	1000101	09
202	NAEEM	ME1000	1000105	01
213	KHALID	SU1008	1000107	01

PHYSICIAN

PHCODE	PHNAME
ME1000	Dr. Alamgir
ME1001	Prof. Aslam Naqvi
ME1003	Dr. Aftab Chaudhery
ME1008	Dr. Ehsan Haider

2.6.1 Types of KEYS

- **Primary Key:** In a relation, the attribute (column) or a combination of attributes (columns) that uniquely identifies a row or a record. PATNO is the attribute that uniquely identifies each patient and thus can be used as a Primary key for the above defined table 'PATIENT'. On the other hand, PNAME is normally not unique, so it can not be used as a primary key.
- **Secondary Key:** A secondary key is non-unique field that is used as a secondary(alternate) key. In the above table, by using PHCODE (physician code), we can scan the records from the table. If the physician leaves, we can change it with a new one by using update statement.
- **Candidate Key/Alternate Key:** Sometimes, it is unclear which field to select as the primary key. There might exist some additional field (or combination of fields) that also have the uniqueness property. These keys can be termed as Candidate keys or Alternate keys.
    - In addition for the uniqueness requirement, Candidate keys must possess following two properties:

- **Composite /Concatenate Key:** These keys consists of two or more data elements or attributes. Invariably, these are the same as Candidate/Alternate keys except that of uniqueness requirement. In order to make it unique, assign STATUS or another attribute (say patient's ID number).
- **Sort/Control Key:** A sort/Control key is used to physically sequence the stored date according to our need. Multiple attributes can be used as sort fields.
- **Foreign Key:** A foreign key is an attribute in a table whose values must match a primary key in another table. The table in which the foreign key is found is called as *dependent* table and to which it refers is called as *parent* table.

NOTE : Foreign key relationships are the basis for establishing 1:1 or 1:M relationships across the Relations/Tables in a relational database management system.

2.7 THE USER

The *user* or *end-user* is simply a person who uses the computers for his specific need. He might have a moderate knowledge of computers, computer science and information technology, and his need to use the computers may be entertainment, education, or professional tasks. He does not need to know the in-depth knowledge of the computer systems, but instead, he should be aware of the installed software he intends to use.

2.8 THE DATA ADMINISTRATOR

A *data administrator* (DA) is responsible for the entire data of an organization. He normally develops the overall functional requirements for the databases being used in the office. He shares in developing the logical design for each database. He should control and manage the databases, establish the data standards, supervise the data distribution within the organization and communicate with the users when necessary. He should also participate in developing the data dictionary, prepare documentation and conduct user training where needed. Normally, the Data Administrator serves as a bridge between users and data processing staff.

2.9 THE DATABASE ADMINISTRATOR

A *database administrator* (DBA) is responsible for the design, implementation, operation, management and maintenance of the database. He/She must be technically expert on the overall intricacies of the database and DBMS. He is supposed to plan, coordinate and carry out a variety of jobs during all phases of the database projects. He must possess the technical skills because he has to work on the complex software and hardware issues involved and to solve the problems of the system and application experts in the organization. He is also responsible to make sure the database access rights, to safeguard its security and to maintain and fine-tune the database functionality.

Exercise 2c

1. Fill in the blanks:

(i) A(n) _____ is a two dimensional array containing descriptive information about an entity.

(ii) A(n) _____ is any thing about which the information is kept in the database.

(iii) In a table the order of rows and columns is _____

(iv) In a relation, the attribute or a combination of attributes that uniquely identifies a record is called _____.

(v) A(n) _____ describes the characteristics of an entity.

(vi) A(n) _____ is an attribute in a table whose values must match a primary key in another table.

(vii) A(n) _____ is responsible for the design, implementation, operation, management and maintenance of the database.

(viii) A(n) _____ consists of two or more attributes.

(ix) A(n) _____ is the dynamic result of one or more relational operations on the base relations to produce another relation.

(x) The _____ refers to raw facts and figures.

2. **Select the correct option:**

(i) Insert command is used to insert:

a) a new table b) a new record

c) a view d) dependencies

(ii) CREATE command is used to create a:

a) table b) view

c) report d) query

(iii) SQL is used for:

a) data definition b) data manipulation

c) data definition and manipulation d) searching records

(iv) The foreign key is found in:

a) parent table b) dependant table

c) pivot table d) index table

(v) A table must have:

a) primary key b) secondary key

c) composite key d) sort key

3. Mark as True or False

(i) The view is not stored in the database.

(ii) Two tables can not have the same name in the database.

(iii) Index makes the searching of a record faster.

(iv) Secondary key must be unique.

(v) The primary key can not work as a sort key.

(vi) The DBA is responsible for maintaining the database.

(vii) A file is a collection of related fields.

(viii) DBMS provide more security to protect data than traditional file management systems.

(ix) DBMS is a software used to train database administrator.

(x) A relation is also termed as a tuple in relational database.

4. How the Records and Files are constructed in traditional File Management System ?

5. How the data is stored and retrieved in FMS (file management system)?

6. How the tables/relations are formed up in DBMS?

7. Discuss the data manipulation in DBMS system.

8. Write down the properties of relations in details.

9. What is a VIEW? How do we create it?

10. What is usage of indexes in FMS and DBMS?