ORACLE19c

Introduction of DBMS:

DATA: **IT IS A RAW FACT (i.e. CHARACTERS & NUMBERS)**

EX: EMPID IS DATA, ENAME IS DATA, SALARY IS DATA, DOJ IS DATA……etc.

NOTE: DATA IS NEVER GIVES ACCURATE MEANINGFULL INFORMATION.

INFORMATION: **PROCESSING DATA IS CALLED AS INFORMATIONS.**

EX:

EMPID	ENAME	SALARY	DOJ	DEPTNAME
1021	X	25000.00	05-08-2020	DB
1022	Y	45000.00	24-12-2019	HR

NOTE: INFORMATION IS ALWAYS PROVIDES ACCURATE MEANINGFULL DATA OF PARTICULAR EMPLOYEE, CUSTOMER, STUDENT and PRODUCT……. etc.

DATA STORAGES: **IT A LOCATION WHERE WE CAN STORE DATA / INFORMATION.WE HAS DIFFERENT TYPES OF DATA STORAGES.**

1. **BOOKS & PAPERS**
2. **FLAT FILE (FILE MANAGEMENT SYSTEM)**
3. **DBMS / DATABASE**

1. DISADVANTAGES OF BOOKS & PAPERS:

- **> IT IS COMPLETE MANUAL PROCEE / SYSTEM.**
- **> REQUIRED MORE MAN POWER.**
- **> COSTLY IN MAINTANANCE**
- **> THERE IS NO SECURITY**
- **> HANDLING A VERY SMALL DATA**
- **> RETRIEVING DATA WILL BE TIME CONSUME.**

2. FLAT FILE (FILE MANAGEMENT SYSTEM):

IN FILE MANAGEMENT DATA CAN BE STORED IN FILES.

DISADVANTAGES:

1. **DATA REDUNDANCY & DATA INCONSISTANCY:**
   ==========
   THESE PROBLEMS COMES INTO PICTURE WHEN WE STORE DATA IN MULTIPLE FILES.WHERE THE CHANGES ARE MADE IN ONE FILE WILL NOT BE REFELECTED TO ANOTHER COPY OF FILE .
   BUT IN CASE OF DATABASE WE CAN MAINTAINE NO.OF COPIES OF SAME DATA AND STILL THE CHANGES MADE IN ONE COPY THEN REFELECTED TO OTHER COPY BECAUSE INTERNALLY MAINTAIN ACID PROPERTIES BY DEFAULT IN DATABASE.

2. **DATAINTEGRITY :**
   ==========
   THIS IS ABOUT MAINTAING PROPER DATA IN EVERY ORAGANIZATION IMPOSE SET INTEGRITY RULES ON DATA AND WE WILL CALL THESE RULES ARE BUSINESS RULES.
   DATABASE PROVIDES AN OPTIONS FOR IMPOSING THE BUSINESS RULES WITH THE HELP OF CONSTRAINTS AND TRIGGERS.

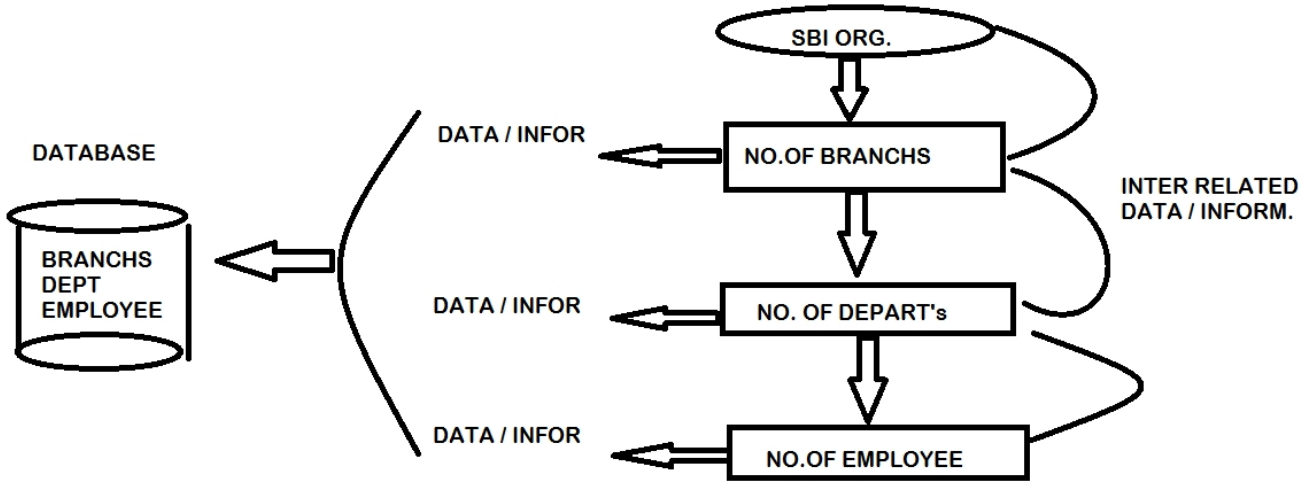
3. **DATARETRIEVE:**
   ==========
   IT IS PROCESS OF DATA RETRIEVING FROM DATA SOURCES.WHICH IS VERY COMPLEX WHILE RETRIEVING DATA FROM FILES WHICH WAS ADDRESSED WITH HIGH LEVEL LANGUAGE.
   WHERE AS IF YOU WANT TO RETRIEVE DATA FROM DATABASE THEN WE ARE USING SQL LANGUAGE.

4. **DATASECURITY:**
   ==========
   DATA IS NEVER SECURE UNDER BOOKS AND FLAT FILE WHERE AS DATABASE ARE PROVIDING AN EXCELLENT CONCEPT IS CALLED AS ROLE BASED SECURITY MECHANISM FOR ACCESSING DATA FROM DATABASE WITH SECURITY MANNER WITH THE HELP OF AUTHENTICATION AND AUTHORIZATION.

5. **DATAINDEXING:**
   ==========
   INDEXES ARE USING FOR ACCESSING DATA MUCH MORE FASTER BUT FLAT FILES DOES NOT PROVIDE ANY INDEX MECHANISM WHERE AS DATABASE WILL PROVIDE INDEXING MECHANISM.

3. DBMS / DATABASE:

DATABASE: IT IS COLLECTION OF INTER RELATED INFORMATION.BY USING DATABASE WE CAN STORE, MODIFY, SELECT AND DELETE DATA FROM DATABASE WITH SECURITY MANNER.



TYPES DATABASES:

1) OLTP (ONLINE TRANSACTION PROCESSING)

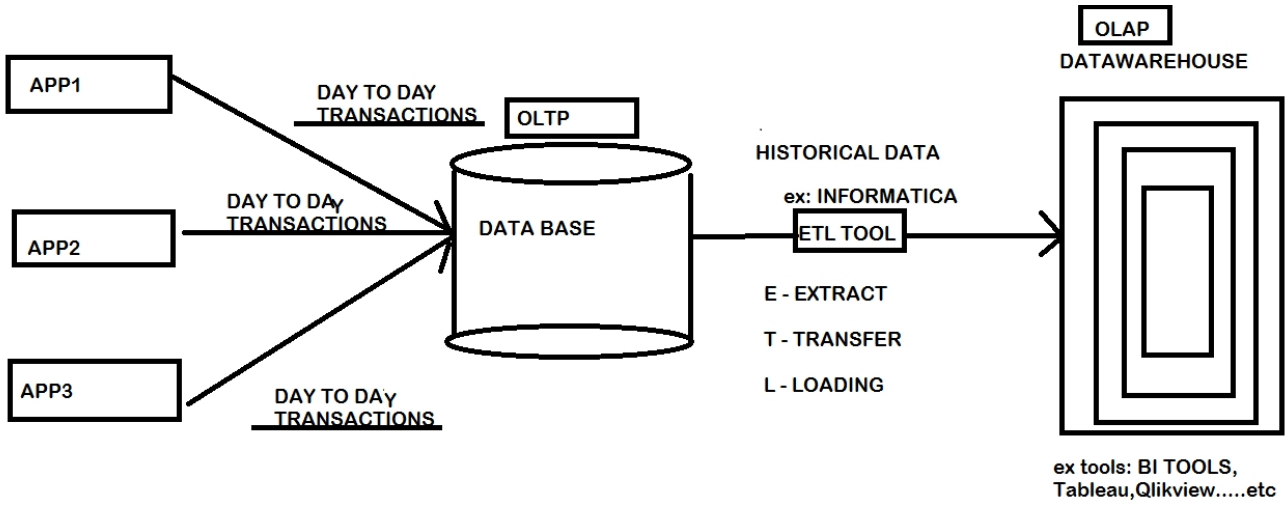
2) OLAP (ONLINE ANALYTICAL PROCESSING)

OLTP: ORGANIZATIONS ARE MAINTAINE OLTP FOR STORING "DAY - TO - DAY TRANSACTIONS INFROMATION ". USING FOR "RUNNING BUSINESS ".

EX: SQLSERVER, ORACLE, MYSQL, ……etc.

OLAP: USED FOR DATA ANALYSIS (OR) DATA SUMMERIZED (OR) HISTORY OF DATA OF PARTICULAR BUSSINESS.

EX: DATAWAREHOUSE.



DBMS: **IT IS A SOFTWARE.WHICH IS USED TO MANAGE DATA IN DATABASE.**

WHY DBMS:

EX:

BUSSINESS ----> (COLLECTION OF ENTITIES)

|

BRANCH | EMPLOYEE | PRODUCTS | CUSTOMERS ----> (ENTITIES)

| < starting >

1 Bran | 10 Emp | 10 Pro | NO Cust. -→(Gen. a very small data)

| < after 5 years >

25 Bran | 500 Emp | 25 Pro | 50000 Cust. ---->(Gen. small data)

| < after 10 years >

100 Bran | 10000 Emp | 100 Pro | 5 Lak's Cust. ------>(Gen. big/large data)

ADVANTAGES OF DBMS:

1. TO REDUCE DATA REDUNDANCY.

2. TO AVOID DATAINCONSISTANCY.

3. EASY TO ACCESS DATA.

4. EASY TO MANIPULATE DATA.

5. MORE SECURITY (AUTHENTICATION & AUTHORIZATION)

6. IT SUPPORTS DATAINTEGRITY RULES.

7. SUPPORTING DATA SHARING

8. SUPPORTS TRANSACTIONS AND "ACID" PROPERTIES.

DBMS MODELS / DATABASE MODELS:

HOW DATA CAN BE ORGANIZED / STORE IN DIFFERENT DATABASE MODELS.THER ARE THREE TYPES OF DATABASE MODELS ARE,

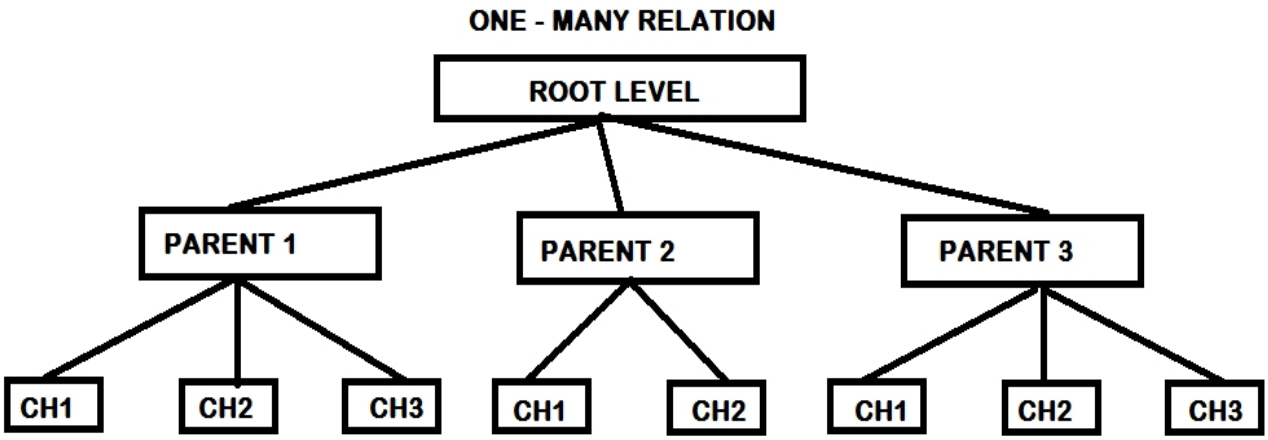
1. **HIERARCHICAL DATABASE MANAGEMENT SYSTEM(HDBMS)**
2. **NETWORK DATABASE MANAGEMENT SYSTEM(NDBMS)**
3. **RELATIONAL DATABASE MANAGEMENT SYSTEM(RDBMS)**
    - **i) OBJECT RELATIONAL DBMS(ORDBMS)**
    - **ii) OBJECT ORIENTED RELATIONAL DBMS(OORDBMS)**

HDBMS:

IT IS A FIRST MODEL OF DATABASE THAT CAME INTO EXISTANCE IN THE 1960's WHICH WILL ORGANIZE THE DATA IN THE FORM OF A "TREE STRUCTURE " AND WHICH WAS DESIGN BASED ON "ONE – MANY RELATION"

IN ONE – MANY RELATION EVERY CHILD IS HAVING ONLY ONE PARENT.THIS TREE IS CONTAINS THE FOLLOWING THREE LEVEL ROOT,PARENT AND CHILD LEVEL.

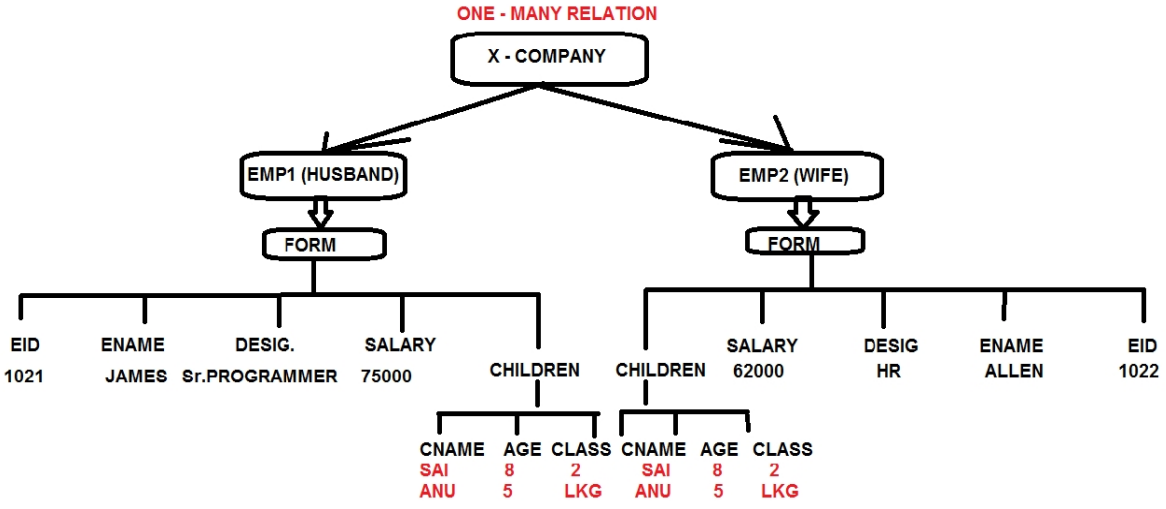
EX: IMS SOFTWARE (INFORMATION MANAGEMENT SYSTEM)



DISADVANTAGES:

1. **WHEN WE WANT TO ADD A NEW LEVEL (PARENT / CHILD) TO AN EXISTING STRCTURE THEN USER HAS TO RE CONSTRCT THE ENTIRE STRCTURE SO THAT IT LEADS TIME CONSUME.**
2. **WHEN WE WANT TO ACCESS DATA FROM THIS MODEL THEN WE NEED TO TRAVEL FROM ROOT LEVEL TO CHILD LEVEL WHICH WILL TIME TAKEN PROCESS.**
3. **THIS MODEL DESIGNED BASED ON ONE – MANY RELATION i.e EVER CHILD IS HAVING ONLY ONE PARENT SO THAT THERE IS A CHANCES TO OCCURE DATA DUPLICATE.**

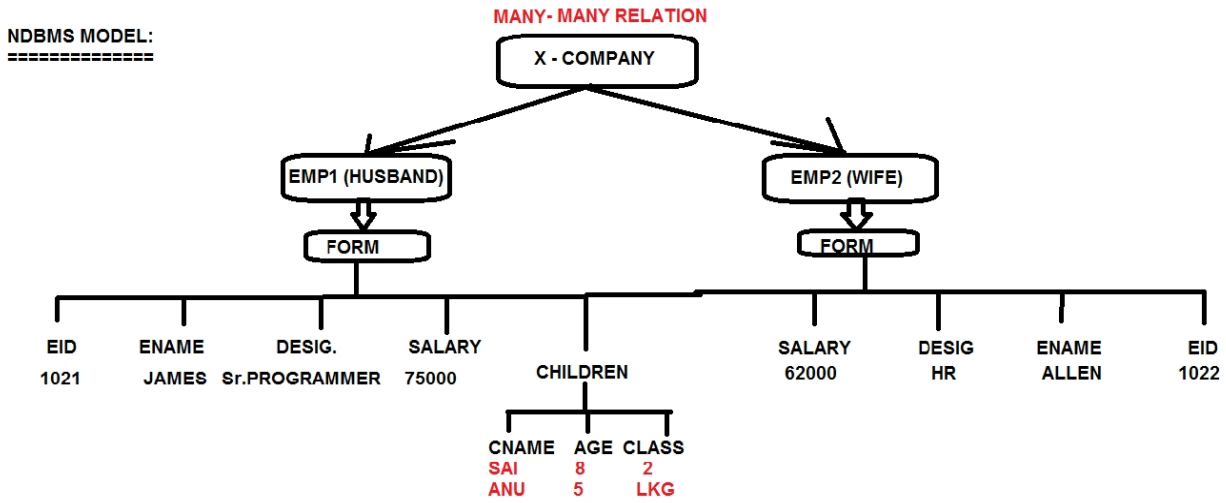
EX:



NDBMS:

THIS MODEL IS A MODIFICATION OF AN EXISTING HIERARCHICAL MODEL BRINGING "MANY – TO – MANY "RELATION SO THAT A CHILD CAN HAVE MORE THAN ONE PARENT WHICH WILL REDUCE DUPLICATE DATA.IN 1969 THE FIRST NDBMS SOFTWARE LAUNCHED WITH THE NAME AS "IDBMS" (INTEGRATED DATABASE MANAGEMENT SYSTEM).

EX:



ADVANTAGES OF NDBMS:

1. **TO REDUCE DUPLICATE DATA BECAUSE SUPPORTING MANY – MANY RELATION (A CHILD CAN HAVE MULTIPLE PARENTS)**
2. **BY USING POINTERS MECHANISM WE CAN ADD NEW LEVEL (PARENT / CHILD) TO AN EXISTING STRUCTURE WITHOUT RECONSTRUCTION.**
3. **ACCESSING DATA IN THIS MODEL IS VAERY FAST BECAUSE IT USES POINTERS.**

DISADVANTAGES OF NDBMS:

1. **WHEN WE USE NUMBER OF POINTERS IN AN APPLICATION THEN IT WILL INCRESE COMPLEXITY(DIFFICULT) TO IDENTIFYING WHICH POINTER IS BELONGS TO WHICH PARENT OR WHICH CHILD AND ALSO DEGRADE PERFORMANCE.**
2. **NDBMS MODEL WAS NOT MORE SUCCESSFUL MODEL IN REAL TIME BECAUSE IMMEDIATE TAKE OVER BY RDBMS MODEL IN 1970's WITH EFFECTIVE FEATURES.**

RDBMS:

IN HDBMS AND NDBMS DATA IS ORGANIZED IN THE FORM OF A TREE STRUCTURE WHICH IS LOOKS COMPLEX TO MANAGE AND UNDERSTAND ALSO SO TO OVERCOME THIS PROBLEM IN 1970's Mr.E.F.CODD FROM IBM CAME WITH A NEW CONCEPT ON STORING DATA IN A TABLE STRUCTURE i.e. ROWS AND COLUMNS FORMAT.

E.F.CODD WITH ALL THESE IDEAS FOR THE NEW MODEL CALLED AS "RELATIONAL MODEL" HAS PUBLISHED AN ARTICLE WITH THE TITLE AS "A RELATIONAL MODEL OF DATA FOR LARGE SHARED DATA BANK".

BASING ON THIS ABOVE ARTICLE MANY COMPANIES CAME FORWARD LIKE IBM, RELATIONAL SOFTWARE INC (PRESENT IT IS ORACLE COR.)……etc. HAS STARTED THE DESIGNED FOR THE NEW DATABASE MODEL i.e. RDBMS.

RDBMS IS MAINLY BASED ON TWO MATHEMATICAL PRINCIPLES ARE "RELATIONAL ALGEBRA" AND "CALCULATIONS".IN THE YEAR 1970's IBM HAS GIVEN THE PROTOTYPE FOR RDBMS KNOWN AS "SYSTEM R".

IN THE YEAR 1974 IBM HAS LAUNCHED A LANGUAGE FOR COMMUNICATION WITH RDBMS KNOWN AS "SEQUEL" AND LATER CHANGED AS "SQL".

FEATURES OF RDBMS:

- DATA CAN BE ORGANIZED IN TABLE FORMAT.

- MORE SECURITY WITH THE HELP OF "AUTHENTICATION & AUTHORIZATION".

- REDUCE DATAREDUNDANCY & DATAINCONSISTANCY USING NORMALIZATION.

- EASY TO MANIPULATION DATA USING DML COMMANDS.

- EASY TO ACCESS DATA FROM DB WITH THE HELP OF "SQL QUERY (SELECT)".

- FASTLY RETRIEVE DATA USING "INDEXES ".

- DATA SHARING USING "VIEWS ".

- SUPPORTING TRANSACTIONS WITH "ACID PROPERTIES".

- SUPPORTING DATATYPES, OPERATORS, FUNCTIONS / PROCEDURE, CLAUSES .etc

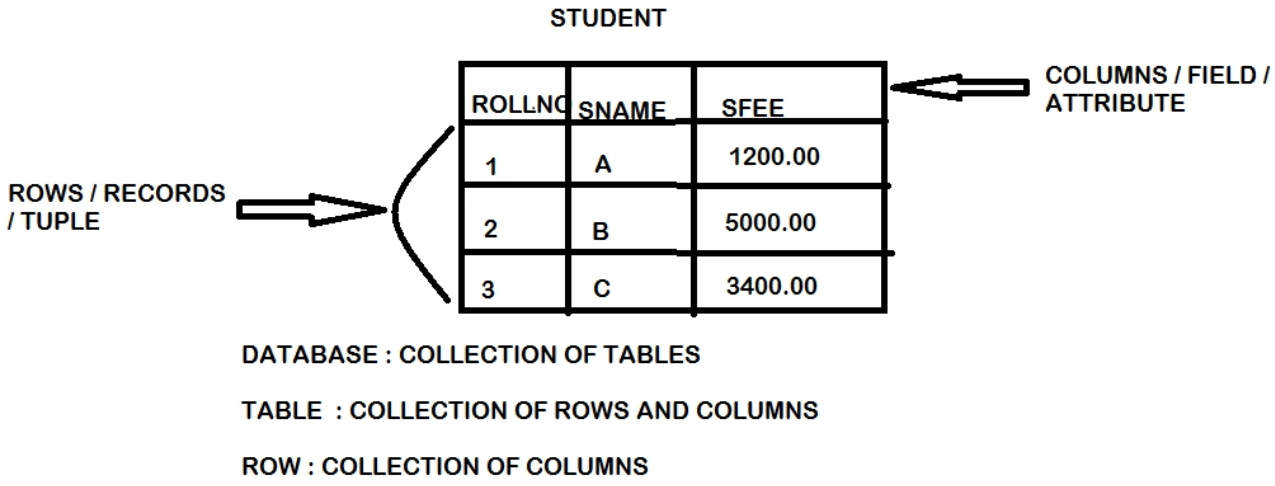
- SUPPORTING ALL RELATIONSHIPS THOSE ARE "ONE – ONE","ONE – MANY /MANY – ONE" AND "MANY-MANY".

- SUPPORTING DATAINTEGRITY RULES WITH CONSTRAINTS & TRIGGERS

- SUPPORTING SQL & PL/SQL LANGUAGES.

EX.OF AN RDBMS PRODUCTS:

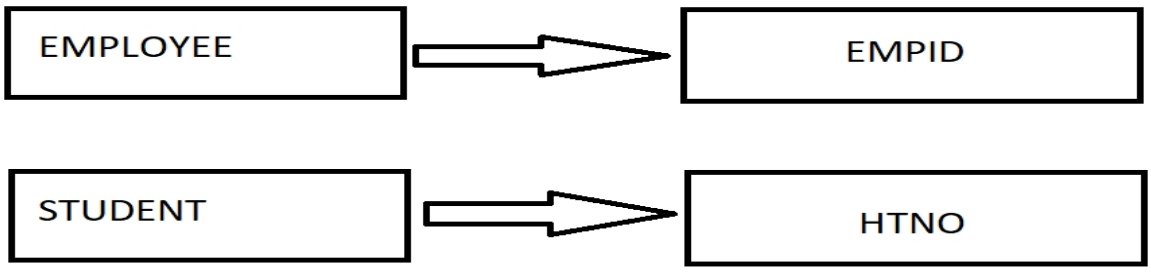
ORACLE, SQL SERVER, MYSQL, DB2, SYBASE, INFORMIX, INGREES, TERADATA, MAXDB, POSTGRESQL ...etc



- **Here relation can be defined as commonness between objects these relations are classified into 3 types**
  - **One to One relation**
  - **One to Many relation / Many to One relation**
  - **Many to Many relation**

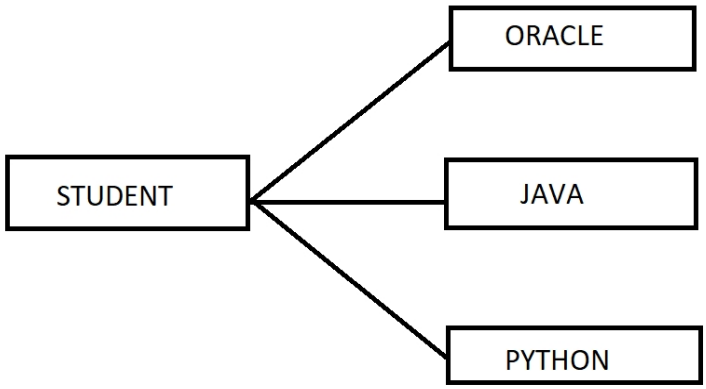
One – One relationship:

- **In this relationship one object can have a relationship with another object**



One – Many / Many – One relationship:

- **In this relationship one object can have a relationship with many objects**



Many – Many relationship:

- **In this relationship many vendors (or) many objects can have the relationship with many other objects**