

(tree structure)

Hierarchical Model in DBMS

Hierarchical model is a data model which uses the tree as its basic structure. So, let's define the basics of the tree.

Basics of Tree :

- A tree is a data structure that consists of hierarchy of nodes with a single node, called the **root at highest level**.
- A node may have any number of children, but each child node may have only one parent node on which it is dependent. Thus the parent to child relationship in a tree is one to many relationship whereas child to parent relationship in a tree is one to one.

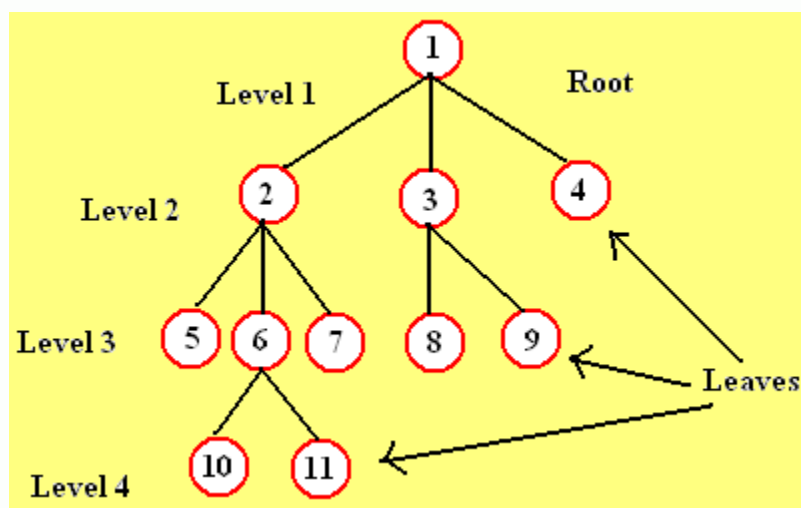


Figure 1

- In figure 1, the node at level 1 is called the root node and the nodes at that has no children are called **leaves**. For example, node 4, 5, 7, 8, 9, 10 and 11.
- Nodes that are children of the same parent are called **siblings**. For example, nodes 2, 3, 4 are siblings.
- For any node there is a single path called the **hierarchical path from the root node**. The nodes along this path are called that nodes **ancestors**.
- Similarly for a given node, any node along a path from that node to leaf is called its **descendent**.
- For example, suppose we have to find out the hierarchical path of node 10, then it will be 1→2→6→10 and the ancestors of node 10 are 1, 2 and 6.
- The **height of tree** is the number of levels on the longest hierarchical path from the root to a leaf. The above tree has a height= 4.
- A **tree is said to be balanced** if every path from the root node to a leaf has the same length.

Figure 2 shows a **balanced and an unbalanced tree**.

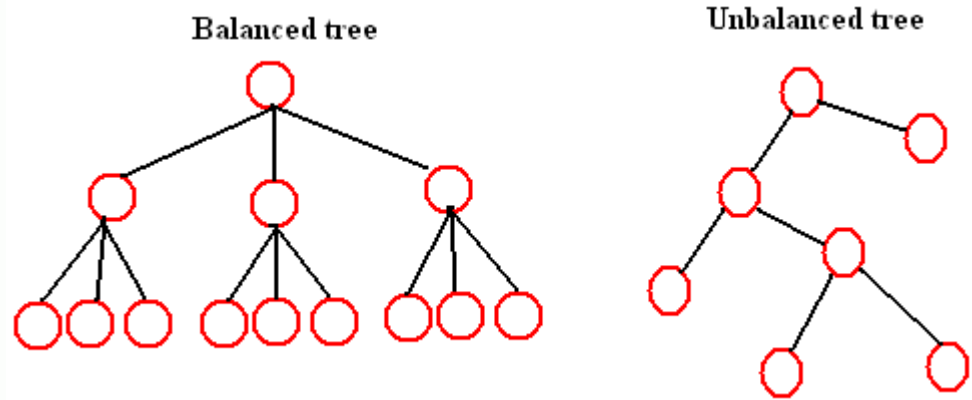


Figure 2

A **binary tree** is one in which each node has not more than two children. Figure 3 shows a binary tree

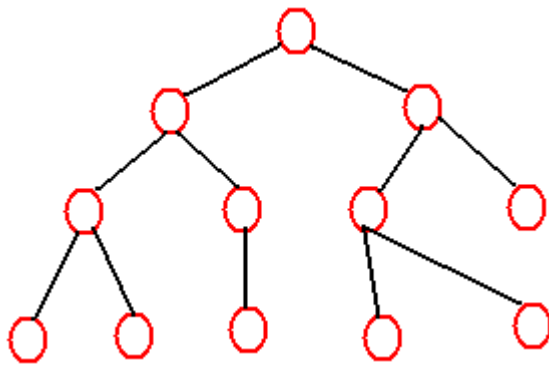


Figure 3

Example of Hierarchical Model :

- **Figure 4 shows** a data structure diagram for a tree representing the STUDENT, FACULTY and CLASS.
- The root node chosen is faculty, CLASS as a child of faculty and STUDENT as a child of class.
- The cardinality between CLASS and FACULTY is one to many cardinality as a FACULTY teaches one or more CLASS.
- The cardinality between a CLASS and a STUDENT is also one to many cardinality because a CLASS has many STUDENTS.

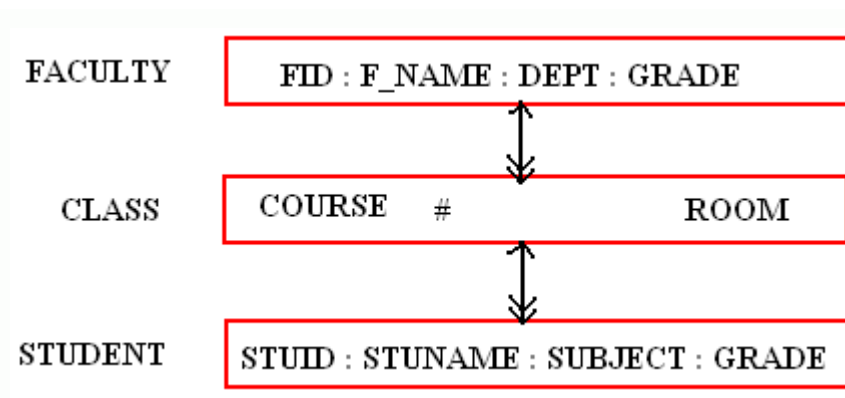
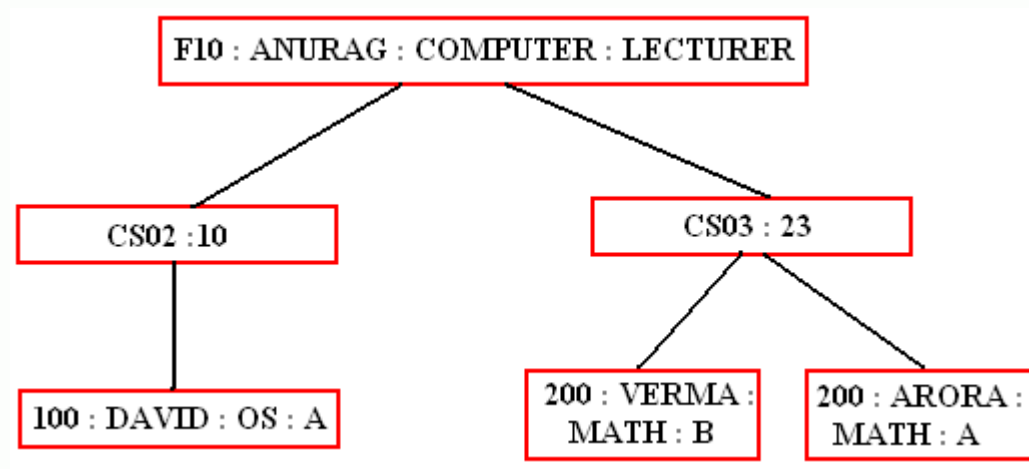


Figure 4

Figure 5 shows an occurrence of the FACULTY-CLASS-STUDENT.



Plex structure

Network Database or Network Model in DBMS

The network database or network model uses the plex structure as its basic data structure. A network is a directed graph consisting of nodes connected by links or directed arcs. The nodes corresponds to record types and the links to pointers or relationships. All the relationship are hardwired or pre-computed and build into structure of database itself because they are very efficient in space utilization and query execution time. The network data structure looks like a tree structure except that a dependent node which is called a child or member, may have more than one parent or owner node. All figure shows the network model –

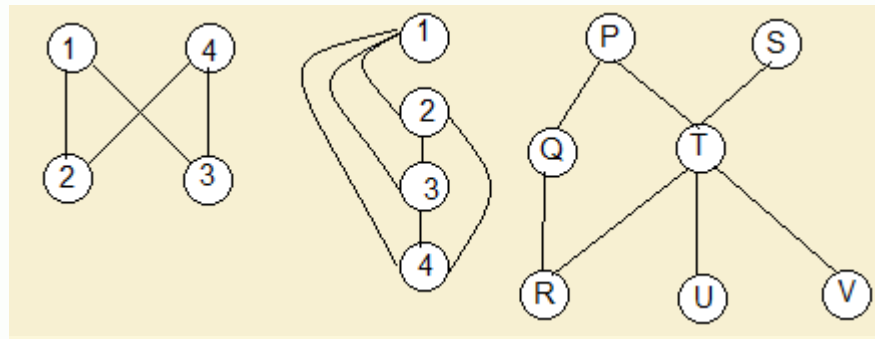


Figure 1

A diagram called as **Bachman Diagram** is used to represent a network data structure. The nodes in the network are replaced by rectangles that represent records and links are shown by lines connecting the rectangles.

A plex structure with two record types is shown / Example of Network Database :

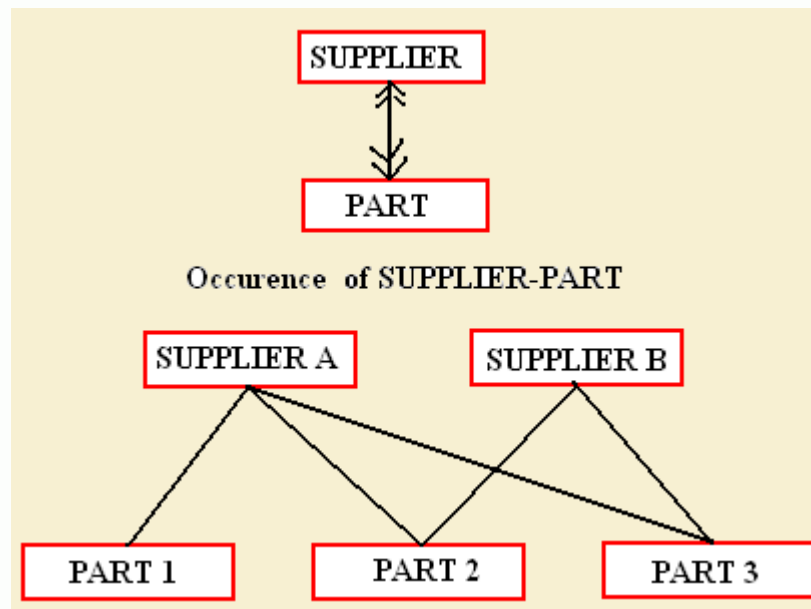
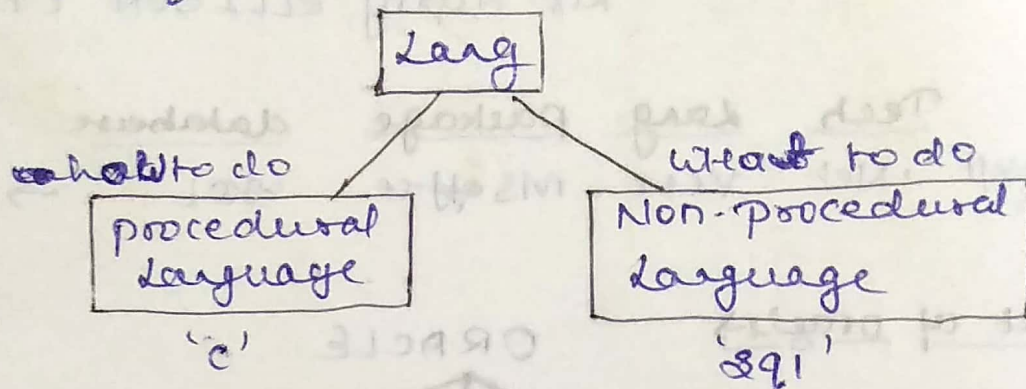


Figure 2

SQL (structured - query - Language) →

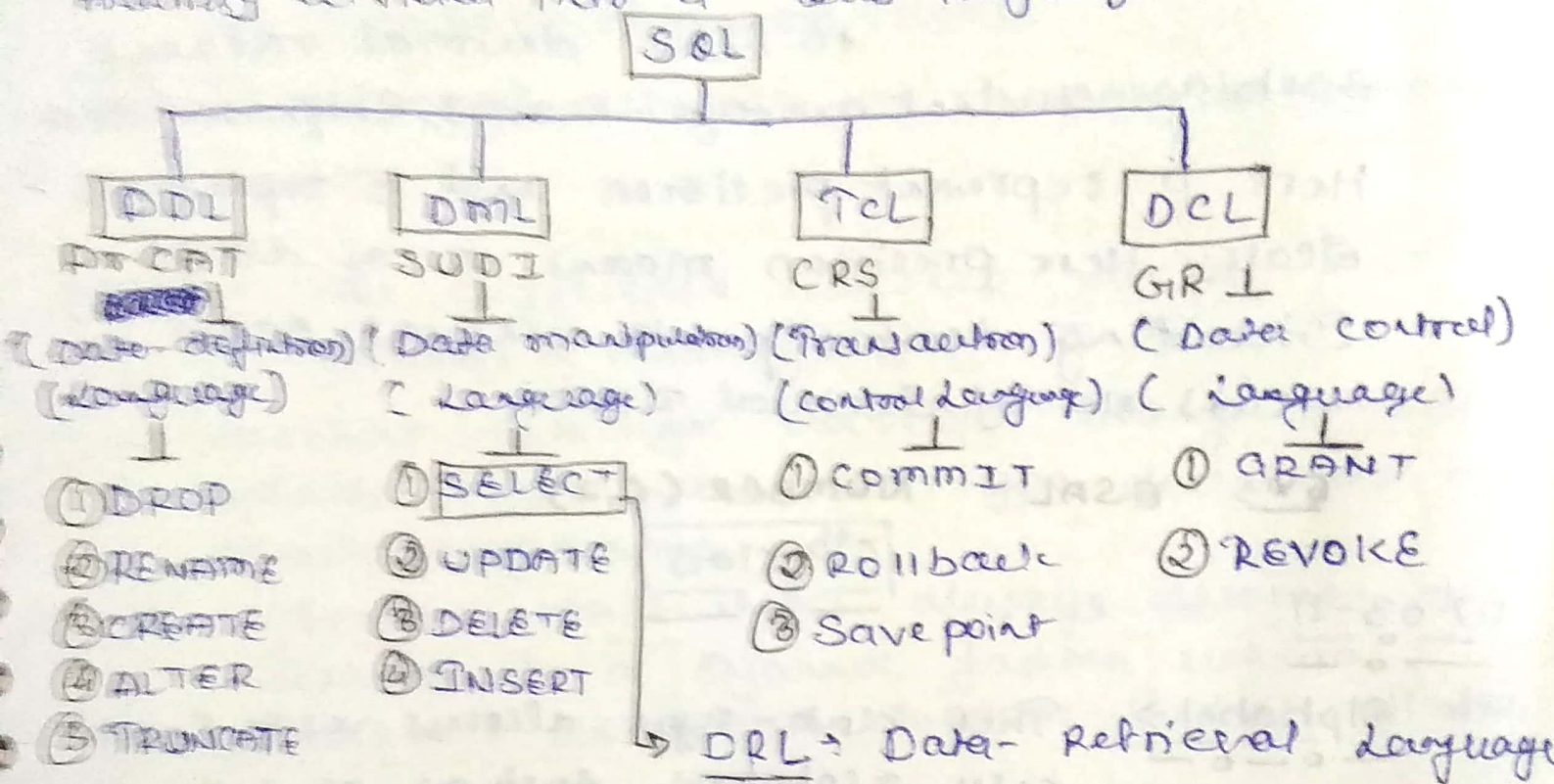
It is developed by Mr. E.F. Codd a German Scientist in 1968. In 1970 ANSI approved this Language and 1972 SQL was released in the Market.



features of SQL →

- ① SQL is not a case-sensitive language. It means that we can write the SQL commands either in capital letters or small letters and or both in also. It is also can be called case-Insensitive language.
- ② Every command of SQL should ends with semicolon(;) .
- ③ SQL can be called as common language Interface. Because this is only language to communicate with Any type of database such as Oracle, SQL-server, Db2, Teradata.
- ④ SQL can also pronounced as SEQUEL stands for Structured English query Language.

SQL can also be called as Natural Language Interface because every command of SQL are almost similar to Normal English Language. Structured query language is mainly divided into 4 sub-languages:-



Data - Types in Oracle 10g

① Number → This data-type allows us to enter ~~any~~ Numerical values such as either integer or decimal values.
This data-type are further divided into 2 types :-

- (i) NUMBER (SIZE)
- (ii) NUMBER (P, S)

* NUMBER (SIZE) → This data-type allows us to enter only Integer values such as :- Student roll no's, Emp-ID's etc.

Here size represents No. of digits.

Ex:-

EMPLOYEE NUMBER(3)

• NUMBER(P,S) → This data-type allows us to enter decimal values such as:- student average marks, Emp. salary etc.

Here P represent precision and S represents scale. Here precision means no. of digits

Including decimal points whereas scale means no. of decimal places.

Ex:-

ESAL

NUMBER(6,2)

6th-class

07-03-11

* Alphabet → This data-type allows us to enter only alphabets such as student names, Employee names etc. This data-type are further divided into 3 types:-

(1) char (size)

(2) varchar (size)

(3) varchar2 (size)

* varchar2(size) → This data-type allows us to enter only alphabets and the maximum length of this data-type is 1 to 2000 bytes.

⇒ It always allocates a memory instance for it.

⇒ Varchar(size) and Varchar2(size)

- ① These two data-types allows us to enter alphabets, special-characters, alphanumeric values etc. The maximum length of this data-type is 1 to 4000 characters (Bytes).
- ② It always allocates a memory in dynamic fashion.

Note:- The difference between varchar and Varchar2 data-type is

- ⇒ varchar data-type develops along with SQL whereas Varchar2 data-type developed by Oracle-corporation.
- ⇒ varchar data-types always allocate a memory in a dynamic fashion whereas Varchar2 data-type supports Garbage collection.

* Data-definition-language → using this language

we can define the data-base objects such as Table, Synonym, Sequence etc.

2) This language contains 5 commands:-

- (1) Create (2) Alter (3) Rename (4) Truncate
- (5) DROP.

* create using this command we can create any database objects such as table, view, synonym etc.

2) Syntax → to create a Table-object.

```
CREATE TABLE  
( COL1 DATA TYPE (Table-name)  
  COL2 DATA-TYPE (size),  
  ⋮  
  COLN DATA-TYPE (size));
```

Ex

```
CREATE TABLE EMP  
( EID NUMBER(3),  
  ENAME VARCHAR2(15),  
  SAL NUMBER(6,2));
```

* rules for Naming a Table

- ① Table-name should start with an Alphabet and it contains minimum 1 character and maximum 30 characters and should not allow special character, space except underscore (_), #, \$ and olog.
- ② The table contains minimum 1 column and maximum 1000 columns (from oracle 9i onwards). upto oracle 8i it allows only 256 columns.
- ③ The table contains minimum of zero records maximum 'n' no. of records up to hard-disk capacity.
- ④ Whatever rules will follow for the table the same rules will be applicable for column-names.

ALTER → By using this command we can modify the structure of the Table.

→ Using this command we can perform 4 operations:-

① These operations are performed through following commands:-

- (i) ALTER-MODIFY (ii) ALTER-ADD (iii) ALTER-RENAME
- (iv) ALTER-DROP

• ALTER-MODIFY → This command is used to ~~create~~ or decrease the size of the data-Type and also change the data-Type from old - data-Type to new-data-Type.

Syntax → (to modify only one column)

ALTER TABLE <Table-Name> MODIFY
COLUMN NAME DATATYPE(size);

Ex. → ALTER TABLE <MORN 1130> MODIFY
ENAME VARCHAR2(20);
TABLE ALTERED

Diagram →

for

Alter-Modify

(15) to (20)

EMPID	ENAME	Size	Dept no

Syntax (to modify more than one column) :-

Ex 1

Syntax :- ALTER TABLE <table-name> MODIFY
(COLUMN NAME 1 Datatype (size),
COLUMN NAME 2 Data Type Size)
!
!
COLUMN NAME N Data-Type (size));

Ex 2

ALTER TABLE MORN1130 MODIFY
(EID NUMBER(5),
ENAME ~~varchar~~ (15));
Table-ALTERED.

- ALTER-ADD :- Using this command we can add one or more than one columns to the existing table.

Syntax :-

ALTER-TABLE <table name> ADD
COLUMN NAME DATATYPE (size);

Ex 3

ALTER-TABLE MORN 1130 ADD
DEPT NO NUMBER(5);
Table-ALTERED.

Syntax → to ADD more than one column :-

ALTER TABLE <Table-Name> ADD
(COLUMN NAME 1 DATATYPE 1 (size),
COLUMN NAME 2 DATATYPE 2 (size),
!
!
COLUMN NAME N DATATYPE N (size));

Ex 1 :-
 ALTER TABLE MOON 1130 ADD
 (MIBNO NUMBER(10),
 ADDRESS VARCHAR2(15));
TABLE ALTERED.

NOTE 3
 ① Whenever we add a new column to the table the new columns always added to the end of the Table only.

- ALTER-RENAME This command is used to change the column name from old column name to new column name.

Syntax → ALTER TABLE <Table Name> RENAME
 COLUMN <old column name> TO <new column name>

Ex 1 → ALTER TABLE MOON 1130 RENAME
 COLUMN SAL TO SALARY;
Table Altered.

Imp-note ② we can't change more than one column name at the same time.

Syntax → to change the table name using ALTER-RENAME

Syntax → ALTER table <old table name> rename to new <table name>

Ex 1 → Alter table moon 1130 rename to moon 1131
Table - Altered.

- ALTER DROP → This command is used to remove one or more than one columns from the existing table.

Syntax → ALTER ~~TABLE~~ TABLE <Table name> DROP COLUMN column name;

Ex:- ALTER TABLE MORN1130 DROP COLUMN deptno;

Syntax to drop more than one columns

Syntax → alter table <table name> drop (list of columns)

Ex:- alter table morn1115 drop (mbno, address).
Table altered.

9th day

11-03-11

- ALTER RENAME → using this command we can change the table-name from old-table name to new-table name

Syntax → RENAME <O.T.N> TO <N.T.N>

Ex → RENAME MORN1130 TO MORN1130;

- TRUNCATE → This command is used for to remove the records or to delete the record from existing-table.

Syntax → Truncate Table <Table name>

Ex:- Truncate Table MORN1130

DROP -> using Drop command we can remove the entire table from the database.

Syntax drop Table <table-name>.

eg drop Table mom1130

FLASHBACK -> This command is used to ~~revert~~ retrieve ~~back~~ the dropped table from the recycle-bin.

Syntax Flashback table <table-name> to before drop.

eg flashback table mom1130 to before drop.

Syntax to drop the table permanently using ~~purge~~ purge.

~~drop~~ drop table mom1130 purge.

Syntax Drop table <tablename> purge.

NOTE -> ① All the ^{DDL} ~~fact~~ ~~query~~ commands auto committed (permanently using).