**Database**🡺(it is a storing area)🡺a collection of data and holds this data in the form of tables and it also provides us the capability to access and manipulate the data

**Tables🡺**holds the data in the form of rows and columns

**Sql**🡺for insert,delete,update,get data from database we need to communicate with database using certain language that language is sql(structured query language)🡺sql language is mostly common for all the databases only few key words and syntaxes is different

**Database components**🡺client and server🡺both components will installed while installing the database it self

Client🡺client send the sql commands(request) to perform certain operation(insertion deletion update delete) on the data which is in server🡺client installed in our local system

Server🡺 data is actually stored in database server🡺databases server mostly installed on the remote machine🡺but mostly it will installed on linux or unix but for windows the server installation is very very rare because mostly windows uses client server

If n number of people want to connect the database which in the server those people will able to connect that database and perform any kind of modifications only through client

First person need to install client by using this client the person can able to connect database which is in server and perform operation

While connecting to the client it asks to provide passwords and username , port number and all required by using this credentials our client will connect to the server and it provide command line where we can run commands to perform operations in database

Client is light weight software we can install in our system but server not if we install server it slows down the process of our computer

Clients are 2 types🡺(CLI)command line Interface and GUI clients are there🡺server can have multiple clients🡺people can use any client and all the clients required the connection details at the time of connecting to the server🡺by default when we install database it will install two clients one GUI and one is CLI

CML🡺while working with cml we have to remember all the commands

GUI🡺while working with GUI it provides the syntax for operations which we want to perform

**For oracle database clients are**🡺sql developer(default client)(CLI),sql plus(GUI)🡺 and we even have third party clients those are tools those are toad, squirrel ,aquadatastudio retc

**Mysql clients are🡺** MYSQL workbench(GUI),MYSQL commandLinetool (CLI) 🡺 and we even have third party clients those are tools those are toad, squirrel ,aquadatastudio etc

**The third part clients are no where related to particular database it will remain same for all the datbases based on our requirement we have to install and we have to use**

**4-types of databases:**

**1.Relational database🡺data stores in the form of rows , columns and also tables have relationship between them.**

**Example:**

**Sql server ,postresql, sqlite,marialDB**

**2.No SQL🡺non relation🡺data stored as key value , data stored as document , data stored as graph**

**Example: Hbase , mongodb,Cassandra**

**Difference between sql and mysql**

**mySql is the relational databaseholds the data and sql is the way to communicate with mysql database to perform operations**

**create database:**

1.create database databasename;

2.create schema databasename;

**Drop database(delete):**

1.drop database databasename;

2.drop schema databasename;

**drop database by checking whether it is available or not:**

drop database if not exists databasename

If we ae trying to create database which is already exists then it may throughs error

**Knowing database:**

**Show databases;🡺**it will display all the databases present

**Show tables;🡺** it will display all the tables present

**Select database();🡺**it will display the current database what might you are using

**Use database:**

Use database means if you on any particular database at that time if you want to choose particular database

At that we simply run the bellow command

**Use databasename;**

**Now what ever the table you creating it will stored in the database which you choose using the command use databasename**

**Crud:**

**Create table:**

Create table tablename(field1 datatype(size) not null,field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”);

Inserting data into created table:

Insert into tablename values(filed1value, field2 value,field3 value)==.in this way of insert we need to provide values for all the fields as per the position of fields in table even if you set default value for a particular field which in present in between any field it is not possible to insert the data into the table without providing value for the particular filed which is set with default value.

Way-2:

Insert into tablename (field1name,field2name,field3name) values(field1 value, field2 value,field3value);

If you don’t want to provide a value for a value for any field just remove that field name and that particular field value like

insert into tablename (field1name, field3name) values(field1 value, ,field3value);

Now if we are not providing any value to the field2 now it will store the value of field2 with null but if we provide any default value to that field while creating table it will store that default value instead of null.

**Note: null is the default value for the fields in mysql**

**If we want to insert multiple values into a table in a single statement use ,(comma) and add the another statement of values like the bellow syntax**

Insert into tablename (field1name,field2name,field3name) values(field1 value, field2 value,field3value), (field1 value, field2 value,field3value), (field1 value, field2 value,field3value);

**Data type mismatch**

**Primary key: primary key supports not null and unique key and even it will helpful to identify record uniquely🡺 this is the main purpose**

**Primary key at row level:**

Create table tablename(field1 datatype(size) primary key,field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”);

**Primary key at table level:**

Create table tablename(field1 datatype(size),field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”,primary key(field1));

**Auto insrement:🡺we can apply auto increment to the fields which are primary key or unique key**

The auto increment is a feature that is applied to a field so that it can automatically generate and provide a unique value to every record that you enter into an SQL, auto increment counter starts from 0 by default but if you want the auto increment to start from another number ,use the below syntax

🡺auto increment is good candidate for primary key when increment it will be unique

**While creating table setting auto increment:**

Create table tablename(field1 datatype(size) **AUTO\_INCREMENT**,field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”,primary key(field1));

**Unique key;**

The purpose of unique is to make sure do not duplicate,

unique key holds null values but primary key not

**Note:**

In some other databases **unique key holds only one null** value but **in mysql unique holds multiple null value**

**Example :** for gmail field if we want to restrict duplicates we can use unique key for that field.

As general a table has one primary key but it has multiple unique key(**note:**even a table has more than one primary key but compared no.of unique keys the count of primary key is less in most of the cases)

Example; empid🡺primary key, emp mail🡺unique key

**Unique key at row level:**

Create table tablename(field1 datatype(size) **Unique key**,field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”);

**Unique key at table level:**

Create table tablename(field1 datatype(size),field2 datatype(size) default “default value”,field3 datatype(size) not null default “default value”, **Unique key** (field1));

**Select**

**Syntax to Select all columns**

Select \* from tablename

**Syntax to select only particular columns**

Select field1,field2 from tablename

**Select columns which satisfies condition**

Select name from tablename where name=”sowji”

**Note :**Here in your table if you have name field with value as **SOWJI still it returns that record because by default where clause is case insensitive** **But if you to make it as case sensitive just add binary key word look at bellow syntax**

Select name from tablename where binary name=”sowji”

Output: empty set

Select name from tablename where binary name=”SOWJI”

Output:

Name

SOWJI

**Update**

**If we use where clause it will update only matched records**

Update tablename set location(columnname)=”chitvel” where name(columnname)=”sowji’;

Now it will change the location to chitvel for the records whose name record is sowji

**If we don’t use where clause it will update that particular value for all the records**

Update tablename set location(columnname)=”chitvel”

Now it will set location as chitvel for all the records in the table

**Situation: update salary of all the employees by 5000**

update employee set salary=salary+5000;

**logical oprators and,or =we use these operators when we want to add more than one condition in where clause**

Update tablename set location(columnname)=”chitvel” where firstname(columnname)=”sowji” and lastname=”madineni”;

**Update vs alter🡺**update deals with data manipulation (like changing the value of existing record field value)

Alter deals with structure cahnage like changing the size or datatype or add extra column

Altere

**To add column**

Alter table employee add jobtitle varchar(20);

**To drop column:**

Alter table employee drop jobtitle ;

**Modify: changing size of column from 20 to 30**

Alter table employee add jobtitle varchar(30);

**Delete primary key:** using alter just mention primary key no need to mention field name which is primary key

Alter table employee drop primary key;

**Add primary key**

Alter table employee drop primary key(id);🡺here id is field name

**DDL VS DML:**

**Ddl=🡺data definition language🡺which change the structure of the table they do not deal with data manipulation🡺commands🡺 create , drop, truncate alter, rename**

**Dml🡺data manipulation language🡺here we deal with the data directly🡺commands🡺insert,update,delete**

**Difference between DELETE VS TRUNCATE:**

**Delete 🡺**delete from tablename🡺deletes all the records of table individually and takes more time compared to truncate🡺it is DML Command

**Truncate🡺**truncate table tablename🡺truncate drop the table internally and recreate it 🡺it is DDL command

**Foreign key constraint🡺**the foreign key constraint is used to prevent actions that would destroy links between two tables…

**🡺**Foreign key is a field in one table that refers to the primary key in another table.

🡺Selected\_course is a foreign key in student table which refers to course id(primary key) in course table

🡺The table with foreign key is called child table,the table with primary key is called the parent table(or referenced table)

**Constraints(primary key,foreign key ,not null,unique,check)**🡺 constraint are used to limit (or add rules and conditions) the type of data that can go into a table

🡺constraint ensures reliability and accuracy of the data is maintained

🡺if there any violation against the constraint then action is aborted

**DISTINCT:**

**Order by: for retrieving data based on some asc or desc order on a particular field**

Select firstname from employee order by years\_of\_experience**🡺it will return record in ascending order**

**Note🡺**on the field on which we are applying order by clause it can or cannot be the part of selecting fields there is no such condition that the field in order by clause should present in selecting fields

Select firstname from employee order by years\_of\_experience DESC**🡺it will return record in ascending order**

**We can also done order by based on column numbers**

Select firstname from employee order by 1 DESC🡺it will perform order by based on 1st column

**Note: we can also perform order by on two fields**

Select firstname from employee order by years\_of\_experience,first name;

Here it first order the fields based on years\_of experience then in that ordered fields person who have the same experience in that they will arrange based on first name

Example

Years\_of\_experience firsname

**5 sowji**

**2 anjali**

**5 sai**

**2 siva**

Select years\_of\_experience,first name from employee order by years\_of\_experience,first name;

Output;

Years\_of\_experience firsname

**2 anjali**

**2 siva**

**5 sai**

**5 sowji**

**Limit🡺** to limit the number of records to retrieve

Select \* from students limit 3🡺 It returns three records

**But limit is use full when it use with order by(situation to get the top three records who has high experience )**

Select \* from students order by years\_of\_experience desc limit 3

**Limit starting index, no.of records**🡺limit 0,3🡺it return records from 0th record and returns 3 records

**Limit 2,3🡺** it return records from 2th record and returns 3 records

**LIKE: like for partner matching 🡺 like can used in where clause**

Select \* from students where sname like ‘%sa%’;

%==>is wild card🡺it indicates 0 or more

\_==> is wild card🡺it indicates only 1

**Group by:**

Group the data based on particular field

**Note: on which field we are performing group by that field should be present in select statement**

Select location ,count(\*) from students group by location

**Grouping more than one field:**

Select location , source\_of\_joining ,count(\*) from students group by location , source\_of\_joining

**Note: group by used to group the data based on particular field and we can count no.of records present in each group**

Select location ,count(\*) from students group by location🡺 in this we are knowing from each location how many students present

**Logical operators**

**Situation: get people not from banglore**

Select \* from students location!=’banglore’;

**Situation: get all courses which has the word data**

Select courses from students where course like “%data%”;

**Situation: get all courses which does not have the word data**

Select courses from students where course not like “%data%”;

**Situation : get all the students who joined through linked and having 8 years experience**

Select \* from students experience<8 and source\_of\_joining=’limkedin”;

**Situation : get all the students who do not have experience between 8 and 12**

Select \* from students where experience<8 or experience>12;

Or

**Using between**

Select \* from student where experience not between 8 and 12;

**Situation : select students who are joined from flipcarts,Walmart,linked in**

Way-1: using or

Select \* from students where company= “flipcarts” or company =” Walmart” or company=” linked”;

Way-2 using **in:**

Select \* from students where company in [‘flipcarts’,’Walmart’,’linked’];

**Situation : select students who are not joined from flipcarts,Walmart,linked in**

Select \* from students where company not in [‘flipcarts’,’Walmart’,’linked’];

**\*\*\*\*\*\*situation: adding an extra field in select statement based on certain condition**

**Situation: we have course table with columns coursename,duration,course feee**

**We have to get if a course is more than 4 months we hace to categorize it as masters program else diploma**

**Select coursename,courseduration,cousefee**

**CASE**

**WHEN courseduration>4 then “masters”**

**Else “diploma”**

**End as course\_type**

**From course**

**Having vs group:**

Where clause is used to filter the individual records before group by

Having is used after group by and to perform filtering on aggregated records(on grouping records)

**Situation: count the students who have joined through linkedin**

**Way-1: using having clause**

select source\_of\_joining, count(\*) as total from student group by source\_of\_joining having source\_of\_joining=’linkedin’

here first it will group all the records as per the source of joining later it will performing having(check for linked in)

**Way-1: using where clause**

select source\_of\_joining, count(\*) as total from student where source\_of\_joining=’linkedin’ group by source\_of\_joining ;

here first it will perform where in where it will select all the records which source of joining is linkedin later it will group

**using where here the complexity and time consuming will decrease compared to having clause**

**we can use both group by and having in same query**

when ever we want to process(group by etc) only on particular data we can use

**situation: the locations from which more than 1 student has joined and the students experience is more than 10 years**

**select location,count(\*) as total from students where years\_of\_experience >10 group by location having total>1;**

**row\_num()🡺**to provide numbering for the row (over a particular field(here mostly we use order by))🡺optional

**situation provide row\_num() for the records based on highest salary like 1st highest salary row num=1,next=2 like that etc**

**select firstname, lastname,salary,row\_num() over (order by salary desc) from employees**

joins:

inner join: join only matched records🡺for inner join we use join or inner join key words

left outer join: join all records from left table and only matched records from right table🡺key word used left join

right outer join : join all records from right table and only matched records from left table🡺key word used right join

full outer join : return all records from both table whether records are matched or unmatched🡺no any kind of key word for it but we can implement it by using union in between left join and right join🡺syntax==left join query union right join query

cross join : it will join each record of left table with all the record with right table🡺for this also no key word🡺its returning records🡺no.of records in left table \* no.of records in right table🡺syntax == select \* from left tablename,right table name;

copying one table into another table:

create table new table name as select \* from present table name

present table🡺source table

new table🡺 destination table

Note : here it copies all the data from present table to new table but not constraints(foreign key, primary key,etc) of present table

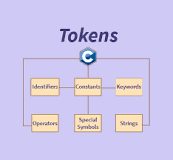
C language

What is meant by derived data types in C?

Last updated on Jul 31, 2023. Derived data types in C language are essentially extensions of the fundamental data types. They do not necessarily invent a new data type but rather add new functionalities to the existing ones.

Why ascii or Unicode

What is tokens in C and its types?



Tokens in C language are the smallest elements or the building blocks used to construct a C program. C Tokens are of 6 types, and they are classified as: Identifiers, Keywords, Constants, Operators, Special Characters and Strings

Infinite key word is macro which is to define infinite loop

Infinite{

Printf(“hello”);

}

Infinite is replacement of for(;;)

1. #include<stdio.h>
2. **void** main ()
3. {
4. **char** s[20];
5. printf("Enter the string?");
6. scanf("%s",s);
7. printf("You entered %s",s);
8. }

**Output**

Enter the string?javatpoint is the best

You entered javatpoint

1. #include<stdio.h>
2. **void** main ()
3. {
4. **char** s[20];
5. printf("Enter the string?");
6. scanf("%[^\n]s",s);
7. printf("You entered %s",s);
8. }

**Output**

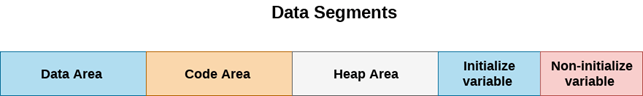
Enter the string?javatpoint is the best

You entered javatpoint is the best

Data Segments

To understand the way our C program works, we need to understand the arrangement of the memory assigned to our program.

All the variables, functions, and data structures are allocated memory into a special memory segment known as Data Segment. The data segment is mainly divided into four different parts which are specifically allocated to different types of data defined in our C program.



## **Why use structure?**

In C, there are cases where we need to store multiple attributes of an entity. It is not necessary that an entity has all the information of one type only. It can have different attributes of different data types. For example, an entity **Student** may have its name (string), roll number (int), marks (float). To store such type of information regarding an entity student, we have the following approaches:

Different data types in c:

Primary,derived,user defined(structure,union),void