Git

To save project online

To store project remotely

We are creating one project i.es website so in that website we will add updates and changes and

Redeploy the code for that we can maintain the copy if the code virtually remote server

Repository is just a project folder

Repo means folder for a project

Remote repository means online folder

remote repository hoisting services

* Github
* Gitlab
* Bitbucket
* Aws
* Micorsoft azure

To store remote repository one of the example GitHub

Why GitHub is famous means it is free of cost

1. Remote repository
2. Local repository
3. Staging area
4. Working directory

Code.py

Example.py

We have two directories

When I enter git commit it commit all directories to the local repository

but to send only one directory (which we want) to local repository

we need to send step by step

from working repository to statging area to local repository to remote repositoty

to send from working directory to staging area we need git add directory name(code.py)

from staging area to local repository we enter git commit -m ”any message”

from local repository to remote repository git push

step 3: after commiting the code we have to push the code to remote repository

for pushing the code to remote repository

we need to create the repository in the github

after creating the repository we have one link command in that then copy and paste it in the command prompt

ex: git remote add origin <https://github.com/sakethbajineni/example.git>

after these we have add the branch of the repository

BRANCH :branch like a folder in the vs code

Code to move to branch

git push -u origin main(branch name)

normally we have main/master branch

to create new branch we need git checkout –b “copy1”

if we have lot of branches then we have to give the branch name while pushing the code

git push –-set-upstream origin copy1

git clone :

it is used to clone the project from git to local it to our laptop

ex: git clone <https://github.com/sakethbajineni/blood/blood-donor>

<https://github.com/sakethbajineni/blood/blood-donor> this we wii get from git hub

in that open code in that HTTPS

|  |  |  |  |
| --- | --- | --- | --- |
| Working Directory | Staging area | Local Repository | Remote Repository |
| On which directoty we are working is called working directory  Directory means like in vs code we create  Files  Code.py  Example.py  Example.txt | I want to send only code.py to local repository |  |  |

React Important questions

* Hooks
* Higher order components
* Life cycle methods of your components
* State management
* Redux/zustand
* Custom hooks (code)
* Local storage
* Lazy loading
* Virtual dom
* SSR (server side rendering ) vs CSR (client side rendering )
* Routing
* RBAC (role based access control)
* Dynamic routing
* Testing
* Async tasks
* Promises

DIFFERENCE BETWEEN THE Let, Var , Const

|  |  |  |  |
| --- | --- | --- | --- |
|  | Var | Let | Const |
| Hoisting | Var is hoisted at the top of the global scope  Can we use it before declaration | Let is hoisted at the top of the private scope  We cant use before declaration | Const is hoisted at the top of the private scope  We cant use before declaration |
| Scope | Global scope normally  Start to end of the function in the function | Block scope always  Start to end of the current scope anywhere | Block scope always  Start to end of the current scope anywhere in the scope |
| Redeclaration | Yes we can redeclare in the same scope | No we cant declare in the same scope | No we cant redeclare or re initialize in the same scope |

Es6 features

Let and const is introduced

Shadow copy:

SQL

Data like contacts in our mobile

In amazon we have different products data

Emails and attachments

Google maps location data

Storing the data in the organized way is data base

Data base management system : a software that is used to easily store and access data from the database in a secured way

Advantages of DBMS :

* Security
* Ease of use
* Durability and availability
* Performance
* Quick accessible

Companies and communities massively investing on database tech

Types of Databases

There are different types of databases based on how we organze the data

1. Relational (table format)
2. Analytical
3. Key value
4. Column family
5. Graph
6. Document

Most commonly used data base is relational data base

## Relational database :

## column names

|  |  |  |
| --- | --- | --- |
| Name | Age | Score |
| Virat | 32 | 50 |
| Dhoni | 39 | 35 |
| Sachin | 47 | 35 |

Column row

Horizontal : row

Column: vertical

In relational database we have to maintain same data types in one column

Like int char string

Non relational database means remaining all like graph , column family , analytical comes under

Non relational database

To manage non relational database:

* Elastic search
* couchDB
* DynamoDB
* MangoDB
* Casandra
* Redis

To manage relational data base management system

* Oracle
* PostgreSQL
* MySQL
* SQLite
* SQL Server
* IBM DB2

Overview of RDBMS

# Rows tables DataBase RDBMS

# SQL :

Structured query language

Structured : close to English but with formal syntax

Query: request

Pronounced as -“sequel”

Sql is declarative language hence easy to learn

Declarative means we say like what we want so and so but we don’t say like go to that server this server like that

Declarative :

User specifies what should be done

Rather than how it should be done

SQL provides various clauses (commands) to perform these operations

Operations operated on

* Create date bases
* Retrieve tables
* Update rows
* Delete

SQL –create operation

Insert clause can be used to create new rows in a table

SQL –Read operation

SELECT clause can be used to access rows in a tabe selectively

UPDATE clause can be used to update existing rows in a table

SQL –Delete

DELETE clause can be used to delete existing rows in a table

SQLite

SQLite Is easy to learn and work with

We will be using SQLite in our program

Different RDBMS supports different flavours of sql

DATA TYPES IN SQL

|  |  |  |
| --- | --- | --- |
| Data type | syntax | example |
| Integer | INTEGER/INT | 0,124 ,-1333 ,etc |
| String | VARCHAR | “BOOK”, “423”,etc |
| Text | TEXT | “we can store large number of characters” |
| Float | FLOAT | 12.22 , 2390.0, etc |

To create table

Code:

CREATE TABLE player(

name VARCHAR(250),

age INT,

score INT

);

Pragma

To get the table info

Code: PRAGMA TABLE\_INFO(college);

Insert Data

Code:

INSERT INTO player (

name, age, score

)

VALUES (‘virat’, 32, 50)

Retrieving the data table

To retrieve the data we use code

Code:

SELECT name, age, score FROM students

And to get total table

We use

SELECT \* FROM students

To select only wanted rows

By using WHERE clause we can select the wanted rows

Code:

SELECT \* FROM students WHERE name= ”sachin”

Here it will select the row which name has Sachin

2. for updating the rows

Code:

UPDATE students

SET score=40

WHERE name=’sachin’

3. for deleting the rows

Query :

DELETE FROM students

WHERE name=’sachin’

4.for deleting the table

Query: DROP TABLE students;

1. For adding extra columns

Query : ALTER students ADD column\_name data\_type

EX; ALTER students ADD subject TEXT

6:for renaming the column name

Ex: ALTER TABLE students RENAME subject TO subjectName

QUERING in SQL

Comparision operators: =,<,>,<=,>=,<>(not equal)

Code:

SELECT \* FROM students

WHERE marks<=100

To work on strings we use LIKE operator

Ex:we search on youtube like movie songs so to get songs which title as movie songs

We cannot use < or > operators so for that we using like operator

Code:

SELECT \*

FROM table\_name

WHERE c1 LIKE pattern;

It retrives all the rows that match the given pattern

Common patterns

* Exact match
* Starts with
* Ends with
* Contains
* Pattern matching

We write patterns using the following wild card patterns

* Percentile ( % )
* Underscore( \_ )

Percentile (%)

% percentile represents zero or more characters

Pattern “shirt%” matches all the strings which have shirt at the beginning followed by 0 or more characters

Underscore :

* Represents exactly one character

Pattern “shirt\_” matches all the strings which have shirt at the beginning followed by exactly one character

For example:

If it has a shirts means it will match this because after shirt it has one character

But if we have

Shirt , black shirt means it doesn’t matches because we have starting word as shirt with another one character so in shirt we don’t have another character after shirt

Starts with means

Syntax :

Where name like “sake%”

We can use like “Sake%” here we are retrieving the rows have words with starting as sake and having one to three character to it

Like Saketh , sakeghj like these values it will select

Code:

Select \* from students

Where name like “sake%”

Ends with means

Syntax:

Where name like “%cakes”

We can use like “% cokes”

Here it starts with some value but need to ends eith cokes like that words only we can select

Code:

Select \* from students

Where name like “%cakes”;

Name contains jeans

Syntax :

Where name LIKE “%jeans%”

It selects the items which contains jeans whether it is starting and ending

Quering with sql – part 2

Logical operators:

* And operator
* Or operator
* Not operator

And operator:

Code:

Select \* from product

Where category =”clothing ” AND price <1000;

In this it will filter the products which are having category as clothes and price <1000

Or operator :

When we want chocolate cake or strawberry cake then we use or operator

Code:

Select \* from product

Where brand =”nike ” OR brand=”puma” OR brand =”Roadster”

Like this we can select the products

Not operator :

Not Is used negate a condition in the where clause

Syntax:

Select \* from table\_name

Where NOT condition;

To ignore all denim brand products we use NOT operator

Use multiple logic operators

AND & OR

Find all products which belongs

* Redmi brand and rating above 4 or
* Oneplus brand

Here the code goes like to use multiple logical operators

Code:

Select \* from products

Where ( brand=”redmi” AND rating>4) OR brand=”oneplus”

Precedence

NOT

AND

OR

NOT having highest precedence means it will excute first

AND has second precedence

OR has 3 rd precedence

It means NOT having highest precedence next AND next OR

Find products

* Belongs to clothing category and
* Name does not contains jeans

Code:

Select \* from products

Where category=”clothing” AND

NOT name LIKE “%jeans%”

QUERING WITH SQL PART-3:

1. Comparision operators contd

* IN
* BETWEEN

1. Pagination

* LIMIT
* OFFSET

1. Ordering

* ASC
* DESC

1. Distint

* Handling data duplication

Comparision operators contd:

To get products whose brand belongs to

Puma or roadstar or,levis or ,nike or,denim brands

Code:

Select \* from products

Where brand=’puma’, OR brand=”roadstar” OR brand=”levis” OR brand=”nike” OR brand=”denim”

Generally we write like this but its too complex

By using IN operator we can write easily

IN Operator :

It checks if a particular value exists in a set of values

We have written code by using the in operator

Code:

Select \* from product

Where brand IN (“puma”,”levis”,”roadstar”,”nike”,”denim”);

Between operator:

Checks if particular value exists in given range

When we have search the products like

Searching products from 1000-500 price range

In this condition we can use

Between operator

Code:

Select \* from products

Where price BETWEEN 600 AND 750;

The values should have same data type

ORDER BY

It sorts the rows based on values of column c1

For

Ascending order ASC

Descending order DESC

Syntax:

Select \* from table\_name

ORDER BY c1 ASC

Ex: like in Ajio app we apply filter like price range low to high this is done by using order by clause

Get products in the increasing order of piece

Syntax: ORDER BY price ASC

Get products in the descending order of rating

Syntax: ORDER BY rating DESC

On multiple columns :

Like we want to filter the products in price from low to high but rating from highest to lowest

For this

Syntax:

Select \* from table\_name

ORDER BY

price ASC

rating DESC

Pagination :

Pagination means in e commerce application like flipkart and amazon having lot of products but we cant load all oproducts in one page so then we use

1 2 3 4 next more like buttons in bottom to show all products page by page

Or else when we scroll up we will get lots of data by scrolling down

OFFSET & LIMIT clauses :

LIMIT : limit clause is used to specify the number of rows

Limit =4;

Means it selects the 4 rows

OFFSET : it is used to specify the position from where the number of rows to be selected

Offset =2

Like from row 2 it start

By default offset value be 0

To get top 5 products

Syntax:

Select \* from product

ORDER BY rating DESC

LIMIT 5;

To get next 5 products after top 5

Syntax:

Select \* from products

ORDER BY rating DESC

LIMIT 50;

DISTINCT :

It can retrieve the unique values

Syntax:

Select DISTINCT brand from product

Order BY brand ;

It helps in finding the brands in e commerce app

Suppose we select brand column means it will select the all values in brand it means from denim we have 50 products from netplya 50 products we have then it selects the denim 50 times amd netplay also so to avoid these

To select the unique values in column we use distinct then it selcts netplay and denim only one time

Distinct on multiple columns

To get unique category and unique brands we use

Select DISTINCT category,brand FROM product

ORDER BY category ;

Aggregations

Combining multiple values in to single value is called aggregation

* Sum
* Avg
* Min
* Max
* Count

Sum :

Returns sum of all values in c1

Syntax:

SELECT SUM(c1)

FROM table\_ name;

To get sum of your favourite cricketer score

Syntax:

select SUM(score) from player\_ details

where name= ’saketh’ ;

to count number of rows

syntax:

SELECT COUNT(\*) FROM player\_ match\_ details

Ex:

To calculate the total number of players in the database

Mainly we have 2 steps

Step1:we select the distint names of player

Step2: count the distint names of player

Code:

Select COUNT(DISTINT names) from player\_table

Aggregations on strings

|  |  |
| --- | --- |
| Aggregate function | Result |
| MIN | Based on lexographic ordering |
| MAX | Based on lexicographic ordering |
| SUM | Depends on DBMS |
| COUNT | Default behavior |
| AVG | Depends on DBMS |

ALIAS

You can provide an alternative temporary name to the columns

Syntax:

Select c1 AS a1

From table\_ name

Group By

Group by means it will combine the same values as a group

Ex:

We want to know the total score of a players. players have played lot of matches in every match we have their name

So first we have to group the players by name then after we have to perform the sum

Like in this situation we need Group By

Code:

Select name,SUM(score) AS total\_score FROM player\_match\_details

GROUP BY name;

Question:

* Calculate total number of sixes and fours scored by each player
* Results should be in the descending order of total number of sixes and fours

Code:

Select

Name,

SUM(fours) AS total\_fours

SUM(sixes) AS total\_sixes

FROM

Player\_match\_details

GROUP BY name,

ORDER BY total\_fours DESC,total\_sixes DESC;

GROUP BY with HAVING

We will go through the some questions

Question 1:

Retrieve top 10 players who played most number of matches in a year

Select name, year

COUNT(\*) AS matches\_ count

FROM player \_match\_ detaills

GROUP BY name, year

ORDER BY matches\_count DESC

LIMT 10;

Question 2:

Calculate the average score of players for all the matches played from the years 2010 till 2014 and bottom 5 players in the average score

Code:

Select name,

AVG(score) AS avg\_score

FROM player\_match\_details

WHERE year BETWEEN 2010 AND 2014

GROUP BY name,

ORDER BY avg\_score ASC

LIMIT 5

HAVING Clause:

Table fitering by WHERE clause next GROUPING next AGGREGTION Next results

To filter results we use HAVING

Question 1 : to get the number of half centuries scored by players

Code:

Select name,

COUNT () AS half\_ centuries

FROM player\_ match\_ details

WHERE

Score>=50

GROUP BY name,

HAVING half\_ centuries>1

Not operator

**SELECT** \* **FROM** Customer **WHERE** Country <>'USA';

NOT EQUAL OPERATOR

***SELECT****\*****FROM****table\_name*

***WHERE****column\_name != value;*

UNION OPERATOR

Union operator is used to combile 2 or more table

Terms and conditions:

* The each table should have same columns
* Each column should have same date type
* The columns in each table must be in same order