

Day93 - May 8th 2024

1. Started my day as usual
2. Learning how to build binary search trees

The screenshot displays a video player window titled "Media Player" showing a tutorial on building a Binary Search Tree (BST). The video content includes a code editor with Python code for building a BST and a diagram of a BST structure.

Python Code (main.py):

```
1 root = None
2 return Node(ele)
3 if ele < root.data:
4     root.left = self.buildBst(root.left, ele)
5 else:
6     root.right = self.buildBst(root.right, ele)
7 return root
8
9 def countNodes(self, root):
10     if root is None:
11         return 0
12     return 1 + self.countNodes(root.left) + self.countNodes(root.right)
13
14 def leafCount(self, root):
15     if root is None:
16         return 0
17     if root.left is None and root.right is None:
18         return 1
19     else:
20         return self.leafCount(root.left) + self.leafCount(root.right)
```

BST Diagram:

The diagram illustrates a Binary Search Tree (BST) structure. The root node is 2. Its left child is 1, and its right child is 3. Node 1 has a left child 0. Node 3 has a left child 2.5 and a right child 4. Node 4 has a left child 7 and a right child 12. Node 12 has a right child 13. Node 36 is also shown as a leaf node. The diagram is annotated with "L R" for left and right, and "Pre" for pre-order traversal.

The video player interface includes a progress bar, play/pause button, and volume control. The system tray at the bottom shows the date and time as 7:11 AM on 5/8/2024.

3. Ended my day by solving a complex SQL question from TechTFQ

Video #5 - Generate salary report

PROBLEM STATEMENT: Using the given Salary, Income and Deduction tables, first write an sql query to populate the Emp_Transaction table as shown below and generate a salary report as shown.

SALARY			EXPECTED OUTPUT - EMP_TRANSACTION			
EMP_ID	EMP_NAME	BASE_SALARY	EMP_ID	EMP_NAME	TRNS_TYPE	AMOUNT
1	Rohan	5000	1	Rohan	Insurance	250
2	Alex	6000	2	Alex	Insurance	300
3	Maryam	7000	3	Maryam	Insurance	350
			1	Rohan	House	200
			2	Alex	House	240
			3	Maryam	House	280
			1	Rohan	Basic	5000
			2	Alex	Basic	6000
			3	Maryam	Basic	7000
			1	Rohan	Health	300
			2	Alex	Health	360
			3	Maryam	Health	420
			1	Rohan	Allowance	200
			2	Alex	Allowance	240
			3	Maryam	Allowance	280
			1	Rohan	Others	300
			2	Alex	Others	360
			3	Maryam	Others	420

INCOME		
ID	INCOME	PERCENTAGE
1	Basic	100
2	Allowance	4
3	Others	6

DEDUCTION		
ID	DEDUCTION	PERCENTAGE
1	Insurance	5
2	Health	6
3	House	4

SQLQuery1.sql - KAUSHI\SQLSERVER-master (KAUSHI\jamka (64)) - Microsoft SQL Server Management Studio

```

select emp_id,emp_name,trns_type,
case when trns_type = 'Allowance' then round(base_salary*(percentage/100),2)
when trns_type = 'Basic' then round(base_salary*(percentage/100),2)
when trns_type = 'Others' then round(base_salary*(percentage/100),2)
when trns_type = 'House' then round(base_salary*(percentage/100),2)
when trns_type = 'Health' then round(base_salary*(percentage/100),2)
when trns_type = 'Insurance' then round(base_salary*(percentage/100),2)
end as amount
from salary
cross join (select income as trns_type, cast(percent as decimal) as percentage from income
union
select deduction, cast(percent as decimal) as percentage from deduction) a

```

Explanation:

Step1 : To generate the reports as per the given question first we have used union to combine the incomes and deduction info

Step2 : Then next we have used cross join to join the salary table with every row of our subquery which has incomes and deductables

Step3 : At last we have used case when and calculated the amount of deductions from the employee salary

emp_id	emp_name	trns_type	amount
1	Rohan	Allowance	200.000000
2	Alex	Allowance	240.000000
3	Maryam	Allowance	280.000000
1	Rohan	Basic	5000.000000
2	Alex	Basic	6000.000000
3	Maryam	Basic	7000.000000
1	Rohan	Health	300.000000
2	Alex	Health	360.000000
3	Maryam	Health	420.000000
1	Rohan	House	200.000000
2	Alex	House	240.000000
3	Maryam	House	280.000000
1	Rohan	Insurance	250.000000
2	Alex	Insurance	300.000000
3	Maryam	Insurance	350.000000
1	Rohan	Others	300.000000
2	Alex	Others	360.000000
3	Maryam	Others	420.000000

Query executed successfully.