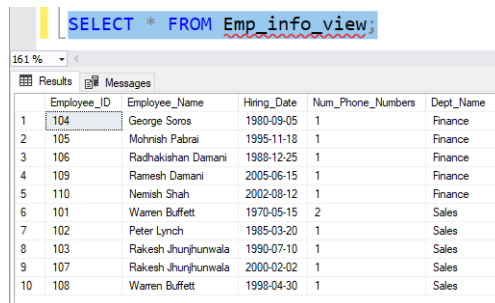


1. SELECT * FROM Emp_info_view;

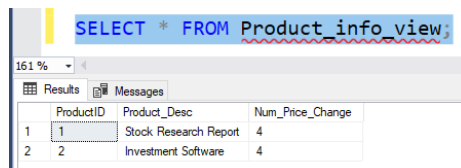


The screenshot shows a SQL query window with the command `SELECT * FROM Emp_info_view;` and its results. The results table has 5 columns: Employee_ID, Employee_Name, Hiring_Date, Num_Phone_Numbers, and Dept_Name. It contains 10 rows of employee data.

Employee_ID	Employee_Name	Hiring_Date	Num_Phone_Numbers	Dept_Name
104	George Soros	1980-09-05	1	Finance
105	Mohnish Pabrai	1995-11-18	1	Finance
106	Radhakishan Damani	1988-12-25	1	Finance
109	Ramesh Damani	2005-06-15	1	Finance
110	Nemish Shah	2002-08-12	1	Finance
101	Warren Buffett	1970-05-15	2	Sales
102	Peter Lynch	1985-03-20	1	Sales
103	Rakesh Jhunjhunwala	1990-07-10	1	Sales
107	Rakesh Jhunjhunwala	2000-02-02	1	Sales
108	Warren Buffett	1998-04-30	1	Sales

This view will give you the list of all employee with it's id, name, hiring date, num of phone numbers they have and also the department name they work for

2. SELECT * FROM Product_info_view;

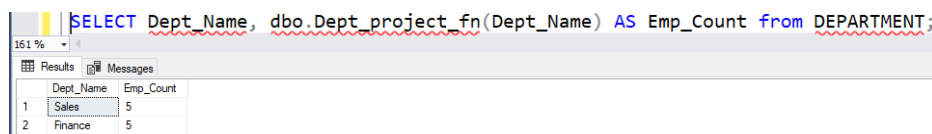


The screenshot shows a SQL query window with the command `SELECT * FROM Product_info_view;` and its results. The results table has 4 columns: ProductID, Product_Desc, and Num_Price_Change. It contains 2 rows of product data.

ProductID	Product_Desc	Num_Price_Change
1	Stock Research Report	4
2	Investment Software	4

This view will give you the list of all Products with it's ID, Desc, Number of time when its price changed

3. SELECT Dept_Name, dbo.Dept_project_fn(Dept_Name) AS Emp_Count from DEPARTMENT;

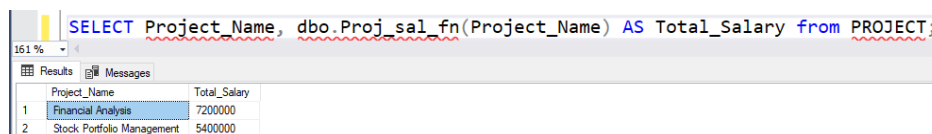


The screenshot shows a SQL query window with the command `SELECT Dept_Name, dbo.Dept_project_fn(Dept_Name) AS Emp_Count from DEPARTMENT;` and its results. The results table has 2 columns: Dept_Name and Emp_Count. It contains 2 rows of department data.

Dept_Name	Emp_Count
Sales	5
Finance	5

This Function will give you the list of all departments along with the total employee count

4. SELECT Project_Name, dbo.Proj_sal_fn(Project_Name) AS Total_Salary from PROJECT;

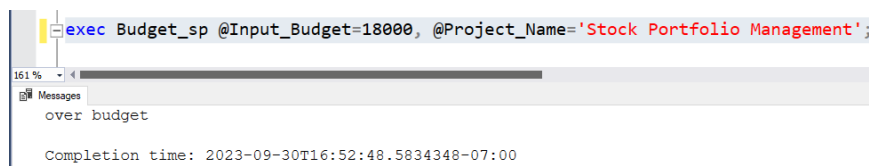


The screenshot shows a SQL query window with the command `SELECT Project_Name, dbo.Proj_sal_fn(Project_Name) AS Total_Salary from PROJECT;` and its results. The results table has 2 columns: Project_Name and Total_Salary. It contains 2 rows of project data.

Project_Name	Total_Salary
Financial Analysis	7200000
Stock Portfolio Management	5400000

It will give you the list of all project with it's employee salary spending in total

5. exec Budget_sp @Input_Budget=18000, @Project_Name='Stock Portfolio Management';



The screenshot shows a SQL query window with the command `exec Budget_sp @Input_Budget=18000, @Project_Name='Stock Portfolio Management';` and its results. The results table has 1 column: over budget. It contains 1 row of data.

over budget
Completion time: 2023-09-30T16:52:48.5834348-07:00

It will show you if the project is within budget or over budget

6. **DECLARE @Hours INT exec emp_hours_sp 2, @Hours OUTPUT PRINT 'Total Hours: ' + CONVERT(VARCHAR(20), @Hours)**

```
DECLARE @Hours INT
exec emp_hours_sp 2, @Hours OUTPUT
PRINT 'Total Hours: ' + CONVERT(VARCHAR(20), @Hours)
```

161 %

Messages

Completion time: 2023-09-30T16:51:11.2875941-07:00

It will give you the total number of hours work done by that employee

7. **exec proj_prod_sp 'Stock Portfolio Management';**

```
exec proj_prod_sp 'Stock Portfolio Management';
```

161 %

Messages

Product ID:1 Product Desc:Stock Research Report

Completion time: 2023-09-30T16:48:46.5382463-07:00

This procedure will list out all the product under this project

8. **exec price_his_sp 57.99,1**

```
exec price_his_sp 57.99,1
```

161 %

Messages

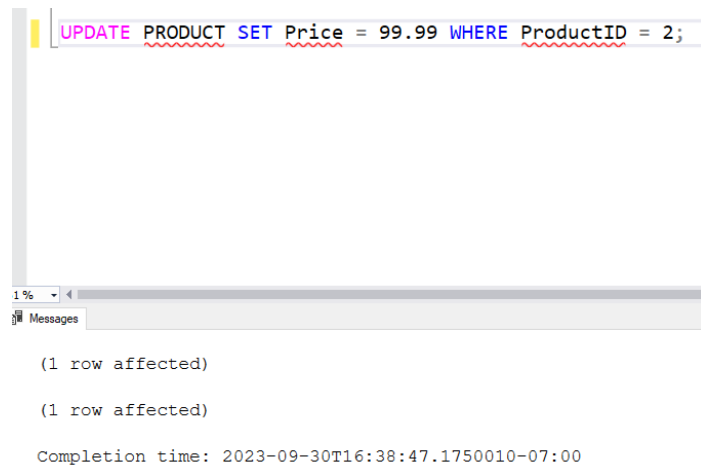
(1 row affected)

(1 row affected)

Completion time: 2023-09-30T16:37:17.7176962-07:00

This Procedure is to update the price_history whenever you are updating the new price for a product. The only condition is it should be differ than the old price

9. **UPDATE PRODUCT SET Price = 99.99 WHERE ProductID = 2;**



```
UPDATE PRODUCT SET Price = 99.99 WHERE ProductID = 2;
```

1% Messages

(1 row affected)

(1 row affected)

Completion time: 2023-09-30T16:38:47.1750010-07:00

The screenshot shows a SQL query execution window. At the top, a SQL statement is entered: `UPDATE PRODUCT SET Price = 99.99 WHERE ProductID = 2;`. Below the query, there is a progress bar and a 'Messages' tab. The 'Messages' tab is active, showing two messages: '(1 row affected)' and '(1 row affected)'. At the bottom, the completion time is displayed as '2023-09-30T16:38:47.1750010-07:00'.

Updating the price of a product will trigger an event and it will automatically update the price_history table