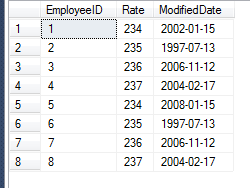
**LAB\_ATKNT003**

1. **Using Pivot Functions**

Write a query to display the sample output using Pivot function

**Sample Input Data:**



**SAMPLE QUERY:**

SELECT \*

INTO #PayHistoryPivotResult

FROM

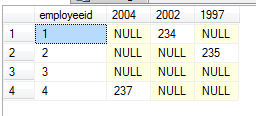
(

SELECT EmployeeID, Rate, year(ModifiedDate) as ModifiedYear FROM EmployeePayHistory where EmployeeID <=4) T PIVOT

(SUM([RATE]) for [ModifiedYear] IN ([2004], [2002], [1997])) AS PivotPayHistory;

select \* from #PayHistoryPivotResult;

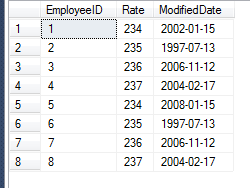
**SAMPLE OUTPUT:**



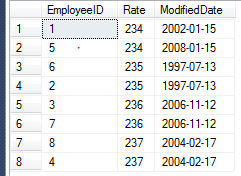
1. **Pagination.**

Write a query to fetch the rows from 4 to 7 in the order of rate.

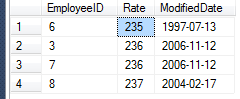
**Sample input table:**



**Sorted table (by rate):**



**Sample output:**



**Query:**

select \* from EmployeePayHistory order by rate offset 3 rows fetch next 4 rows only;

1. **Merge**

**Problem Statement:**

Write a query using MERGE statement and perform delete and update operations in a single query.

Design Rules:

1. Relationship between EmployeeRegion and EmployeeAddress is based on EmployeeID.

2. Target table for Update and Delete operation is EmployeeAddress

3. Perform delete operation in the merge statement when CountryCode is “RUS”.

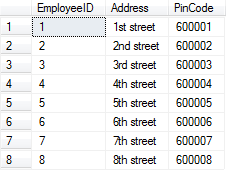
4. Perform update action in the merge statement when CountryCode is “JPN”, update column Address as 'Japan Street

5. Please follow the order of operation as mentioned above. First delete, then update.

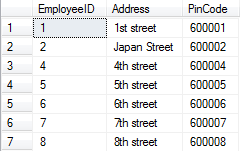
**Sample Input (Source Table):**



**Sample Input (Target Table):**



**Sample Output:**



**Code**:

merge EmployeeAddress as Target

using EmployeeRegion as Source

on Target. EmployeeID = Source. EmployeeID

when matched and Source. CountryCode = 'RUS' then delete

when matched and Source. CountryCode = 'JPN' then update

set Target. Address = 'Japan Street';

1. **Stored Procedure**

User Sored procedure to Fetch employee details, where the employee’s first name is ‘Gopi’

CREATE PROCEDURE GetEmployeeDetails

@name\_parameter nvarchar(20)

AS

BEGIN

Select a.First\_name,a.employee\_id,b.Start\_date from employees as a

join job\_history\_tbl as b

on a.employee\_id = b.employee\_id

where a.First\_name = @name\_parameter

END

EXEC GetEmployeeDetails @name\_parameter = 'Gopi';

drop procedure GetEmployeeDetails;

Note: Only the highlighted code needs to be submitted in exam. The Procedure syntax and the parameter name will be auto populated in the editor. Begin statement or End statement or EXEC or drop should not be saved. You can use EXEC to verify the execution, but should be removed before saving.

1. **Function:**

Use Function to return the values as per the given parameter.

CREATE FUNCTION Employee\_name

(@emp\_id int)

RETURNS table

as return

SELECT a.first\_name +' ' +a.last\_name as employee\_name,b.Start\_Date

FROM employees as a

join job\_history\_tbl as b

on a.employee\_id = b.Employee\_id

WHERE b.Employee\_id = @emp\_id

GO

select \* from employee\_name(202345) ;

drop function Employee\_name;

Note: Only the highlighted part needed to be saved in editor. It should not contain ‘GO’,’DROP’ or function execution statement. The code can be executed by calling the function, but should not be saved at the end.