TASK COMPLETION REPORT: Analysis of Employee who got promoted

**Description of Work Done:**

This SQL query identifies employees who have been promoted by comparing their job history records. It uses a self-join on the Job\_History table to match employees' previous jobs with their subsequent jobs, ensuring that the Job\_Id changes and the promotion occurs after the previous job's end date. The query then retrieves and displays the old and new job titles, along with the promotion date, providing a clear record of each promotion event. The result is ordered by employee and promotion date.

Step 1: Database Schema Overview: HR Database

The HR database consists of tables that manage information related to jobs and employees' job history within an organization. The schema is designed to track the different roles an employee has held over time, along with the respective departments and time periods.

**Tables Used:**

1. **Jobs Table:**
   * **Purpose:** Stores information about the different job roles available in the organization, including job titles and salary ranges.
   * **Columns:**
     + Job\_Id (INT, Primary Key): Unique identifier for each job role.
     + Job\_Title (VARCHAR(100)): The title of the job role.
     + Min\_Salary (DECIMAL(10, 2)): The minimum salary for the job role.
     + Max\_Salary (DECIMAL(10, 2)): The maximum salary for the job role.
2. **Job\_History Table:**
   * **Purpose:** Records the job history of employees, including the roles they have held, the departments they worked in, and the dates of their employment in each role.
   * **Columns:**
     + Employee\_Id (INT): Identifier for the employee.
     + Start\_Date (DATE): The date the employee started the job.
     + End\_Date (DATE): The date the employee left the job.
     + Job\_Id (INT, Foreign Key): References Job\_Id from the Jobs table, indicating the job role.
     + Department\_Id (INT): Identifier for the department the employee worked in.

**Relationships:**

* **Foreign Key Relationship:** The Job\_History table has a foreign key (Job\_Id) referencing the Job\_Id column in the Jobs table, linking each job history record to a specific job role in the organization. This relationship enables the tracking of an employee's progression through different job roles within the organization.
* Step 2: SQL Query to Analyze Data

The SQL query identifies and lists employees who have been promoted by comparing their previous and current job roles in the job history records.

SELECT

jh1.Employee\_Id,

jh1.Job\_Id AS Old\_Job\_Id,

j1.Job\_Title AS Old\_Job\_Title,

jh2.Job\_Id AS New\_Job\_Id,

j2.Job\_Title AS New\_Job\_Title,

jh2.Start\_Date AS Promotion\_Date

FROM

Job\_History jh1

JOIN

Job\_History jh2 ON jh1.Employee\_Id = jh2.Employee\_Id

JOIN

Jobs j1 ON jh1.Job\_Id = j1.Job\_Id

JOIN

Jobs j2 ON jh2.Job\_Id = j2.Job\_Id

WHERE

jh1.Job\_Id <> jh2.Job\_Id

AND jh1.End\_Date < jh2.Start\_Date

ORDER BY

jh1.Employee\_Id, jh2.Start\_Date;

Step 3: Query Results

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Employee\_Id** |  | **Old\_Job\_Id** | **Old\_Job\_Title** | **New\_Job\_Id** | **New\_Job\_Title** | **Promotion\_Date** | | 101 |  | 1 | Software Developer | 3 | Project Manager | 2023-01-01 | | 102 |  | 2 | Data Analyst | 5 | Business Analyst | 2023-02-15 | | 103 |  | 3 | Project Manager | 7 | Database Administrator | 2023-03-01 | | 104 |  | 4 | HR Manager | 8 | UX/UI Designer | 2023-04-10 | | 108 |  | 8 | UX/UI Designer | 10 | Sales Manager | 2023-08-08 | | 110 |  | 10 | Sales Manager | 13 | DevOps Engineer | 2023-10-10 | | 112 |  | 12 | Quality Assurance | 15 | Cloud Architect | 2023-12-12 | | 115 |  | 15 | Cloud Architect | 17 | Content Strategist | 2024-03-03 | | 117 |  | 17 | Content Strategist | 18 | Product Manager | 2024-05-05 | | 118 |  | 18 | Product Manager | 19 | Graphic Designer | 2024-06-06 | | 119 |  | 19 | Graphic Designer | 20 | IT Support Specialist | 2024-07-07 | |
| Top of Form |
| Bottom of Form |

Step4: Interpretation of Results

1. **Promotion Details:**
   * The table shows a list of employees who have been promoted from one job role to another within the organization.
2. **Job Role Transition:**
   * Each employee has moved from an "Old Job Title" to a "New Job Title," indicating a promotion in their career.
3. **Promotion Timeline:**
   * The "Promotion Date" column specifies the exact date when each promotion took effect, showing the chronological order of promotions.
4. **Career Progression:**
   * The table highlights the upward career movement for each employee, from roles such as "Software Developer" to "Project Manager," "Data Analyst" to "Business Analyst," and so on.
5. **Employee Development:**
   * This data can be used to analyze employee development and the effectiveness of the organization's talent management and promotion strategies.