Database Administration Project Documentation

1. Create Database and Use:

- Summary:

  - Creates a new database named "company_database" and sets it as the active database.

- Detailed Explanation:

  - `CREATE DATABASE company_database;` : This statement creates a new database named "company_database" in the database management system.

  - `USE company_database;` : This statement switches the active database context to "company_database", meaning subsequent queries will operate within this database.

2. Create Tables:

- Defines the structure for various tables in the database, including

  - DEPARTMENT
  - EMPLOYEE
  - DEPENDENT
  - DEPT_LOCATIONS
  - PROJECT
  - WORKS_ON

- Each `CREATE TABLE` statement specifies the name of the table and its columns along with their data types and constraints.

Tables :

- DEPARTMENT Table:

  - Attributes :
    - dname (Department Name),
    - dnumber (Department Number)
    - mgrssn (Manager SSN)

  - This table stores information about different departments within the company.

- EMPLOYEE Table:

  - Attributes :
    - fname (First Name),
    - lname (Last Name)
    - ssn (Social Security Number)
    - bdate (Birth Date)

    - address (Address)
    - salary (Salary)
    - superssn (Supervisor SSN)
    - dno (Department Number)

  - Stores details about company employees, including personal information, salary, supervisor, and department affiliation.

- DEPENDENT Table:

    - Attributes:
      - essn (Employee SSN)
      - dependent_name (Dependent Name)
      - sex (Employee Gender)
      - bdate (Dependent Birth Date)
      - relationship (Relationship with Employee)

  - Records information about dependents of company employees.

- DEPT_LOCATIONS Table:

    - Attributes:
      - dnumber (Department Number)
      - dlocation (Department Location)

  - Stores the locations of various departments within the company.

- PROJECT Table:

    - Attributes:
      - pname (Project Name)
      - pnumber (Project Number)
      - plocation (Project Location)
      - dnum (Department Number)

  - Records details about projects undertaken by different departments.

- WORKS_ON Table:

    - Attributes:
      - essn (Employee SSN)
      - pno (Project Number)
      - hours (Hours Worked)

  - Tracks employee participation in various projects and the hours worked on each.

3. Insert Data:

  - Populates the tables with sample data.

- Each `INSERT INTO` statement adds a new row of data into the specified table, providing values for each column according to the table schema.

4. Delete Records:

- Removes all records from the tables.

- Each `DELETE FROM` statement removes all rows from the specified table, effectively clearing the table of its contents while retaining its structure.

- Using to resetting the database to its initial state or removing unnecessary data.

---

1. Query 1 - Retrieve Employee Dependents:

- Purpose: This query retrieves information about employees and their dependents.

- Tables Involved: EMPLOYEE, DEPENDENT

- Columns Selected: Employee's first name, last name, dependent's name, sex, birthdate, and relationship.

- Join Condition: Joins EMPLOYEE and DEPENDENT tables based on the employee's SSN (Social Security Number).

- Ordering: Results are ordered by employee's last name and then first name.

2. Query 2 - Retrieve Employee Projects:

- Purpose: This query retrieves information about employees and the projects they are working on.

- Tables Involved: EMPLOYEE, WORKS_ON, PROJECT

- Columns Selected: Employee's first name, last name, and project name.

- Join Conditions: Joins EMPLOYEE with WORKS_ON using SSN and WORKS_ON with PROJECT using project number.

- Ordering: Results are ordered by project name.

3. Query 3 - Retrieve Top 2 Salaries:

- Purpose: This query retrieves the top 2 salaries from the EMPLOYEE table.

- Table Involved: EMPLOYEE

- Columns Selected: Salary

- Ordering: Salaries are ordered in descending order.

4. Query 4 - Retrieve Employee Salary with Coalesce:

- Purpose: This query retrieves employee names and their salaries, replacing null salaries with a default value of 3000.

- Table Involved: EMPLOYEE

- Columns Selected: Employee's first name, last name, and salary.

- Function Used: COALESCE is used to replace null salary values.

5. Query 5 - Retrieve Employee and Supervisor Names:

- Purpose: This query retrieves employee names along with their supervisor's first and last names.

- Table Involved: EMPLOYEE (twice, for employee and supervisor)

- Columns Selected: Employee's first name, supervisor's first and last names.

- Join Condition: Uses a LEFT JOIN to match employees with their supervisors based on supervisor SSN.

6. Query 6 - Retrieve Second Highest Salary Employee:

- Purpose: This query retrieves the employee with the second-highest salary from the EMPLOYEE table.

- Table Involved: EMPLOYEE

- Columns Selected: All columns

- Subquery: Finds the maximum salary less than the overall maximum salary, then retrieves the employee with that salary.

- Ordering: Results are ordered by descending salary.

7. Query 7 - Retrieve Projects Starting with 'B':

- Purpose: This query retrieves projects whose names start with the letter 'B'.

- Table Involved: PROJECT

- Columns Selected: All columns

- Condition: Uses a LIKE operator to filter projects with names starting with 'B'.

8. Query 8 - Error Handling for Salary Constraint Violation:

- Purpose: This query attempts to select and update employee records with salaries greater than or equal to 6000.

- Table Involved: EMPLOYEE

- Issue: It results in an error as there's an attempt to update salaries violating a predefined constraint (salary < 6000).

- Resolution: The query includes an attempt to update salaries, which triggers an error due to constraint violation. It also adds a new constraint to enforce the salary limit.

9. Query 9 - Error Handling for Address Constraint Violation:

- Purpose: This query attempts to select and update employee records with addresses not in specified values.

- Table Involved: EMPLOYEE

- Issue: It results in an error as there's an attempt to update addresses violating a predefined constraint.

- Resolution: The query includes an attempt to update addresses, which triggers an error due to constraint violation. It also adds a new constraint to enforce valid address values.

10. Query 10 - Create Employee Name Validation Function:

- Purpose: This query creates a user-defined function (CheckEmployeeName) to validate employee names.

- Functionality: Checks if the provided SSN corresponds to an employee with a null first name, last name, both, or neither.

- Input: SSN (Social Security Number)

- Output: Message indicating the status of first name and last name.

11. Query 11 - Create Employee Data Retrieval Function:

- Purpose: This query creates a user-defined function (GetEmployeeData) to retrieve employee data based on input string.

- Functionality: Returns employee's first name, last name, full name, or NULL based on the input string.

- Input: String indicating the desired data ('first name', 'last name', 'full name', or invalid input)

- Output: Employee data based on the input string.

12. Query 12 - Create Project Employee Count View:

- Purpose: This query creates a view (ProjectEmployeeCount) to show the count of employees working on each project.

- View Columns: Project name, Employee count

- Aggregation: Counts the number of employees for each project using the GROUP BY clause.

13. Query 13 - Create Employees in Department View:

- Purpose: This query creates a view (EmployeesInDeptD2) to show employees in a specific department ('d2') whose last name contains the letter 'J'.

- View Columns: Employee number, Employee last name

- Join Condition: Joins EMPLOYEE with DEPARTMENT based on department number and filters records for department 'd2'.

- Filtering: Further filters records to include only employees with last names containing 'J'.

14. Query 14 - Create Employee Update Procedure:

- Purpose: This query creates a stored procedure (UpdateEmployeeInProject) to update employee assignments in a project.

- Functionality: Replaces an old employee's assignment with a new employee's assignment for a specified project.

- Parameters: Old employee number, New employee number, Project number

- Error Handling: Checks if the new employee number exists and handles no-employee-found scenarios.

15. Query 15 - Create Employee Delete Trigger:

- Purpose: This query creates a trigger (EmployeeDeleteTrigger) to prevent deletion of employee records and log deletion attempts.

- Functionality: Instead of actual deletion, it logs deletion attempts in an audit table (EmployeeAudit) along with the user and timestamp.

- Action: Raises an error to prevent actual deletion.