**Human Resource Management System (HRMS)**

**Employee Schema**

**Project Description**

This project is to deliver Employee Schema in the application Database for Itlize Human Resource Management System (HRMS).

**Summary**

Since the explosion of number of employees and consultants, the Human Resource Department wants to automate their daily work into a computerized system. The project was approved by CEO to be implemented in the next 12 months. There are several stages planned to implement to fulfil the final deliverable of this project.

The first stage is done by Business Analysts coordinating with Human Resource Analysts and Experts (and Subject Matter Experts) to gather the requirement, understanding what needs to be done to fulfil the automation process.

Application and Database establishment will be done parallel by SQL Team and .NET Team.

Once the UI/Application/Database is established; the project will move on to the second phase where automation of data cleaning, data modelling and calculation, report development and strategic calibration can be implemented.

This project will be extended to the analytic team where data scientists will dedicate their resources to provide predictive analysis on a range of topics.

**Function Requirement**

For the implementation of Database Establish, you will need to read the technical specification in the next chapter. After you successfully establish your database, and stored procedures for all select statements, you will need to share your work in one file through outlook to your reporting manager and they will upload onto the Enterprise Server to deploy the queries.

**Technical Specification**

1. Create Database, Tables, Structures, and Load Test Data into your Database. Before you work on creating anything, you will need to draw out the ER-Diagram (Entity Relation Diagram) and show it to the Team Lead/Manager (Trainer) to confirm with the design.

* Create a database **EMPLOYEE – YOURNAME**. (Customize it with your own name so that Testers will be able to review your work and after validation they will upload the work onto Production Server for operational usage)
* Create a table **Country** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| Country\_Id | numeric | 10 | No | Primary,Auto Increment |
| Country\_Name | nvarchar | 50 | No | Unique |

* Create a table **State** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| State\_Id | numeric | 10 | No | Primary, Auto Increment |
| Country\_Id | numeric | 10 | No |  |
| State\_Name | nvarchar | 50 | No |  |

* Create a table **City** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| City\_Id | numeric | 10 | No | Primary, Auto Increment |
| State\_Id | numeric | 10 | No | Foreign Key with State table |
| City\_Name | nvarchar | 50 | No |  |

* Create a table **Designation** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| Desig\_Id | numeric | 10 | No | Primary, Auto Increment |
| Desig\_Name | nvarchar | 50 | No | Unique |
| Desig\_Description | nvarchar | 300 | No |  |

* Create a table **Employee\_Details** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| Emp\_Id | numeric | 10 | No | Primary, Auto Increment |
| Emp\_First\_Name | nvarchar | 50 | No |  |
| Emp\_Middle\_Name | nvarchar | 50 | Yes |  |
| Emp\_Last\_Name | nvarchar | 50 | No |  |
| Emp\_Address1 | nvarchar | 100 | No |  |
| Emp\_Address2 | nvarchar | 100 | Yes |  |
| Emp\_Country\_Id | numeric | 10 | No | Foreign Key with Country table. |
| Emp\_State\_Id | numeric | 10 | No | Foreign Key  with State table |
| Emp\_City\_Id | numeric | 10 | No | Foreign Key with City table. |
| Emp\_Zip | numeric | 5 | No |  |
| Emp\_Mobile | numeric | 10 | No |  |
| Emp\_Gender | bit |  | No | 1= Male  0=Female |
| Desig\_Id | numeric | 10 | No |  |
| Emp\_DOB | datetime |  | No |  |
| Emp\_JoinDate | datetime |  | No |  |
| Emp\_Active | Bit |  | No | 1=Active  0=Inactive |

* Create a table **Salary** as below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| Salary\_Id | numeric | 10 | No | Primary, Auto Increment |
| Emp\_Id | numeric | 10 | No | Foreign key with Employee\_Details table |
| Emp\_Salary\_Change\_Year | datetime |  | No |  |
| Emp\_Salary | decimal | 10,2 | No |  |

Emloyee Documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | Allow Nulls | Key |
| Emp\_Doc\_Id | numeric | 10 | No | Primary,Identity |
| Emp\_Id | numeric | 10 | No | Foreign key with  employee\_details table |
| Emp\_Doc\_Name | nvarchar | 30 | No |  |
| Emp\_Doc\_Desc | Nvarchar | 150 | Yes |  |

* Insert values into **Country** table.

|  |  |
| --- | --- |
| Country\_Id | Country\_Name |
| 1 | Afghanistan |
| 2 | Albania |
| 3 | Algeria |
| 4 | Andorra |
| 5 | Australia |
| 6 | Austria |
| 7 | Brazil |
| 8 | Bangladesh |
| 9 | China |
| 10 | Chile |
| 11 | Denmark |
| 12 | Egypt |
| 13 | Ethiopia |
| 14 | France |
| 15 | Finland |
| 16 | Germany |
| 17 | Georgia |
| 18 | Greece |
| 19 | Hong Kong |
| 20 | Hungary |
| 21 | India |
| 22 | Ireland |
| 23 | Japan |
| 24 | Kuwait |
| 25 | Kenya |
| 26 | Luxembourg |
| 27 | Mexico |
| 28 | Nepal |
| 29 | Oman |
| 30 | Poland |
| 31 | Qatar |
| 32 | Russia |
| 33 | South Africa |
| 34 | Switzerland |
| 35 | Syria |
| 36 | Thailand |
| 37 | United States |
| 38 | Uganda |
| 39 | Vietnam |
| 40 | Yemen |
| 41 | Zimbabwe |

* Insert values into **State** table.

|  |  |  |
| --- | --- | --- |
| State\_Id | Country\_Id | State\_Name |
| 1 | 1 | Badakhshan |
| 2 | 1 | Badghis |
| 3 | 1 | Baghlan |
| 4 | 2 | Berat |
| 5 | 2 | Diber |
| 6 | 2 | Durres |
| 7 | 3 | Adrar |
| 8 | 3 | Blida |
| 9 | 3 | Tipaza |
| 10 | 4 | Canillo |
| 11 | 4 | Ordino |
| 12 | 5 | Tasmania |
| 13 | 5 | Victoria |
| 14 | 6 | Burgenland |
| 15 | 6 | Tirol |
| 16 | 7 | Bahia |
| 17 | 7 | Roraima |
| 18 | 8 | Dhaka |
| 19 | 9 | Jilin |
| 20 | 10 | Atacama |
| 21 | 11 | Capital |
| 22 | 12 | Suhaj |
| 23 | 13 | Tigray |
| 24 | 14 | Auvergne |
| 25 | 15 | South Savo |
| 26 | 16 | Bremen |
| 27 | 17 | Ajaria |
| 28 | 18 | Crete |
| 29 | 19 | Eastern District |
| 30 | 20 | BaranyaMegye |
| 31 | 21 | Kerala |
| 32 | 22 | Cavan |
| 33 | 23 | Akita |
| 34 | 24 | Hawalli |
| 35 | 25 | Central |
| 36 | 26 | Diekirch |
| 37 | 27 | Colima |
| 38 | 28 | Eastern |
| 39 | 29 | Dhofar |
| 40 | 30 | Lodzkie |
| 41 | 31 | Al Khawr |
| 42 | 32 | Altay |
| 43 | 33 | Free State |
| 44 | 34 | Bern |
| 45 | 35 | Idlib |
| 46 | 36 | Chai Nat |
| 47 | 37 | Alaska |
| 48 | 37 | Texas |
| 49 | 37 | New York |
| 50 | 37 | California |
| 51 | 38 | Abim |
| 52 | 39 | Bac Can |
| 53 | 40 | Ataq |
| 54 | 41 | Bulawayo |
| 56 | 37 | New Jersey |

* Insert values into **City** table.

|  |  |  |
| --- | --- | --- |
| City\_Id | State\_Id | City\_Name |
| 1 | 1 | Jurm |
| 2 | 2 | Abbakhsh |
| 3 | 3 | Baglan |
| 4 | 4 | Berat |
| 5 | 5 | Burrel |
| 6 | 6 | Durres |
| 7 | 7 | Algiers |
| 8 | 8 | Amsthas |
| 9 | 9 | Aboucha |
| 10 | 10 | Canillo |
| 11 | 11 | Ordino |
| 12 | 12 | Hobart |
| 13 | 13 | Melbourne |
| 14 | 14 | Eisenstadt |
| 15 | 47 | Anchorage |
| 16 | 47 | Fairbanks |
| 17 | 47 | College |
| 18 | 47 | Kodiak |
| 19 | 47 | Meadow Lakes |
| 20 | 48 | Houston |
| 21 | 48 | Dallas |
| 22 | 48 | Austin |
| 24 | 48 | Fort Worth |
| 25 | 49 | New York |
| 26 | 49 | Buffalo |
| 27 | 49 | Rochester |
| 28 | 49 | Syracuse |
| 29 | 49 | Albany |
| 30 | 50 | Los Angeles |
| 31 | 50 | San Diego |
| 32 | 50 | San Francisco |
| 33 | 50 | Oakland |
| 34 | 50 | Sacremento |
| 35 | 56 | Newark |

* Insert values into **Designation** table.

|  |  |  |
| --- | --- | --- |
| Desig\_Id | Designation | Desig\_Description |
| 1 | Project Manager | Project Manager |
| 2 | Software Developer | Software Developer |
| 3 | Team Leader | Team Leader |
| 4 | Business Analyst | Business Analyst |
| 5 | Design Engineer | Design Engineer |
| 6 | Tech Lead | Tech Lead |

* Insert values into **Employee\_Details and Salary** table with data in the flat file.

2. Use the Employee Details.txt as a source file for the last task above (Insert Values into Employee\_Details and Salary). (Hint: Bulk Insert a flat file, txt file, csv file/Import Wizard)

**3. VALIDATE THE DATA YOU INSERTED!!!** Some of rows that you inserted could be wrong according to the relations you built in your tables. (Hint: Since you are using test data, you can directly correct on the data, and you shall provide a plan on how to avoid these problems in the future)

4. Write each of the following SQL Queries for the below requirements in Stored Procedure (Each Procedure shall have a unique name and function according to each requirement). **REMEMBER: ALL QUERIES are for SELECT Statement**, NO INSERT/UPDATE/DELETE shall be used.

1. Select First name, Last name, Address, Country Name (by joining with country table), Sate Name (by joining with state table), City Name (by joining with city table),Mobile, and Designation name (by joining with designation table) of all employees .
2. Select state name and country name and sort the records in ascending order with state name.
3. Select first 3 records from country table in ascending order by country name.
4. Select all employees whose first name starts with letter “a”.
5. Select all employees whose first name ends with letter “a”
6. Select all employees who are inactive.
7. Select first name, last name, middle name of all employees and change the column names to First Name, Last Name, and Middle Name.
8. Print total number of employees.
9. Print total number of employees whose middle name is not null.
10. Select first name, last name, middle name of all employees. If middle name is null change that value to “Not Applicable”.
11. Concatenate first name, middle name, last name of all records in employee table and print.
12. Select all records in employee table and order in ascending direction with country name.
13. Retrieve first 10 records from employee table and order by their first name in ascending direction.
14. Select all record from employee who lives in the cities 🡪Dallas, Algiers.
15. Select all employees who lives in the cities that starts with the letter s between A and D.
16. Fetch First name, last name, middle name, Country name, Designation name and DOB within a range (e.g.: 1/2/1984 to 1/3/1986).
17. Fetch First name, Last name, middle name, Country name, Designation name and highest salary.
18. Fetch First name, Last name, middle name, Country name, Designation name and his current salary.
19. Fetch First name, Last name, middle name, Country name, Designation name and his/her highest salary within a range (e.g. : 50000 to 100000).
20. Select first 3 characters of EMP\_FIRST\_NAME from EMPLOYEE\_DETAILS -- SUBSTRING(COLUMN\_NAME,1,3)
21. If there is any letter “A” in Emp\_First\_Name, replace “A” with “$”.
    1. E.g. Francis 🡪 Fr$ncis
22. Select EMP\_FIRST\_NAME ,Joined year, Joined Month name and joined day from

Employee details table.

E.g.: **First Name Year Moth Day**

**Ayodeji 2014 Jan 5**

1. Get employee details from employee details table whose joining year is “2014”.
2. Get employee details from employee table who joined before January 1st 2014
3. Get Designation, total salary spend by the company with respect to each E.g.

**Designation Name Total Salary**

**Mobile Software Engineer 622100.00**

1. Get designation name, no of employees in a designation with respect to a designation from employee details table.
2. Get designation wise average salaries from employee details table and order by Salary ascending.
3. Select no of employees joined with respect to year and month from

Employee details table.

e.g.:

**Joined Year Joined Month Number of employees**

**2013 1 2**

1. Select employee from employee details table whose salary is not yet in the salary table.
2. Select first name, sum of salary from employee details and salary table for all employees even if they didn't get salary and set salary amount as 0 for those employees who didn't get salary.

Hint: use left outer join

1. Select 10 % of current salary from **Employee1**, 20% of **Employee1**, 15% of **Employee3** and for others 30 % of salary from employee table.
2. Display how many days each employee worked in the company.
3. Select the employee with highest appraisal. (difference of first salary and current salary)
4. Select the employees who have the third highest current salary.
5. Display employee name, and average salary with the help of a user defined function.

4. As a Data Scientist in the HRMS project team, you may have discovered the insufficiency of data to support analytics for human resource management in a lot of aspects – for example: HR team wants to determine how education, job title, and knowledge of certain skills may determine the amount of compensation receive by the employee, but the degree/education, and skillset of employees are not captured in our database.

As a Data Scientist, you will need to list out minimum 10 other variables/column names that are necessary to be recorded in the database system for future analytic purposes.

5. The Data Scientist team will implement certain algorithms to predict compensation level/raise level to employees based on current salary and employee information; the team will also perform prediction of turnover rate (permanent leave vs hiring)/retention rate (contract extend ratio) based on current information provided in the database. As a new Data Scientist in the team, think about another THREE (3) aspects that could be analysed; at the same time, list out all inputs required for such analytics and write down your thinking process about how the analytics could be done based on each analytic requirement (Be prepared to discuss in group presentation).

6. Complete all queries/requirements from Part 1 to Part 4 using Python/Pandas.

Knowledge Reference Link:

<https://msdn.microsoft.com/en-us/dn308572.aspx>

<https://stackoverflow.com/>

<http://learnpython.org/>

SQL Reference Link:

<https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB>

Machine Learning Reference Link:

<https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN>