Andrei Alin-Ionuț 344C3

Laborator 7-8 BD2

- Laborator 7

  - EX1

    Folosim N’ deoarece reprezinta un standard SQL-92 si este folosit pentru reprezentarea caracterelor Unicode(NCHAR, NVCHAR, NTEXT), in mare daca nu il folosim avem sanse sa scriem unele caractere gresit.

  - EX2

    SELECT E.[last_name] AS ‘Nu mai avem caractere non-ASCII. ’,

    E.[first_name] AS ‘Exista decat un tip varchar.’

    FROM [dbo].[employees] E;

  - EX3

    Lipseste Steven King deoarce nu are un manager_id
    Folosim LEFT JOIN

    ```
    SELECT E.[first_name] + ‘ ‘ + E.[last_name] [NumeCompletAngajat]
        ,M.[first_name] + ‘ ‘ + E.[last_name] [NumeCompletManager]
    FROM [dbo].[emplopyees] E
    LEFT JOIN [dbo].[employees] M ON E.[manager_id] = M.[employee_id];
    ```

  - EX4

    Din cate am observat da, da trebuie sa folosim alt tip de JOIN + inca un JOIN pentru a scapa de EMP_CTE

    ```
    SELECT E.[employee_id],
      E.[first_name] + ' ' + E.[last_name] AS [NumeCompletAngajat],
    ```

```
    DE.[department_name] AS [NumeDepartament],
    E.[manager_id]
FROM [dbo].[employees] E
LEFT JOIN [dbo].[employees] M ON E.[manager_id] = M.[employee_id]
LEFT JOIN [dbo].[department] DE ON E.[department_id] = DE.[department_id]
```

- EX5

```
SELECT E.[first_name] + ‘ ‘ + E.[last_name] [NumeCompletManager]
FROM [dbo].[emplopyees] E
WHERE E.[employee_id] IN (SELECT M.[manager_id] FROM
[dbo].[emplopyees] M )
```

- EX6

```
SELECT D.[department_name] NUME_DEP
     ,COUNT(D.[department_name]) NUMAR_ANG
     ,STRING_AGG(E.[first_name] + ' ' + E.[last_name], ' | ') NUME_ANG
    ,STRING_AGG(E2.[first_name] + ' ' + E2.[last_name], ' | ') NUME_MANAGER
FROM [dbo].[departments] D
INNER JOIN [dbo].[emplopyees] E ON D.[department_id] = E.[department_id]
INNER JOIN [dbo].[emplopyees] E2 ON E.[manager_id] = E2.[employee_id]
    AND E.[manager_id] IS NOT NULL
GROUP BY D.[department_name]
HAVING COUNT(D.[department_name]) % 2 = 1;
```

- EX7

```
(
    SELECT 'Ana are Mere' + CHAR(10) + CHAR(13) + 'Ana are Mere'
    UNION
```

```
    SELECT 'Ana are MERE'
)
UNION ALL
(
    SELECT 'Ana are MERE'
    UNION
    SELECT 'Ana are Mere' + CHAR(10) + CHAR(13) + 'Ana are Mere'
)
```

- EX8

```
SELECT DEP.[department_name] [NumeDepartament],
   STRING_AGG(EMP.[last_name] + ' ' + EMP.[first_name]) [NumeAngajat],
   STRING_AGG(MAN.[last_name] + ' ' + MAN.[first_name]) [NumeManager],
   MIN(STDEV(EMP.[salary])) [GrilaMinima],
   MAX(STDEV(EMP.[salary])) [GrilaMaxima]
FROM
   [dbo].[employees] EMP
   ,[dbo].[emplopyees] MAN
   ,[dbo].[departments] DEP
WHERE
   EMP.[manager_id] = MAN.[employee_id] AND
   DEP.[department_id] = EMP.[department_id] AND
   STDEV(EMP.[salary]) > 1 AND
   STDEV(EMP.[salary]) < 1.5
```

- EX9

    1. Afiseaza pentru un departmanet angajatii care au managerii in acel departament.
    2. Am putea avea o eroare de parsare deoarece unele coloane ar fi pozitionate altfel.

- Laborator 8

    - EX1

```
INSERT INTO [dbo].[EMPLOYEE_CLONE]
SELECT *
FROM [dbo].[EMPLOYEES] E
WHERE E.[employee_id] NOT IN (
    SELECT EE.[employee_id]
    FROM [dbo].[employee_clone] EE
)
```

    - EX2

```
WITH dep AS
 (
    SELECT COUNT(*) [NR], E.[department_id]
    FROM [dbo].[employees] E
    GROUP BY E.[department_id]
)

UPDATE E
SET [salary] =[salary] * 1.15
FROM
```

```
        [dbo].[EMPLOYEES] E,
        dep D
WHERE E.[department_id] = D.[department_id] AND d.[NR] % 2 = 0
```

- EX3

```
DELETE FROM
    emp
FROM
    departments dep,
     locations loc,
     [dbo].[EMPLOYEE_CLONE] emp
WHERE
    dep.location_id = loc.location_id AND
     emp.department_id = dep.department_id AND
     loc.country_id != 'US'
```

- EX4

```
SELECT *
INTO [dbo].[clone]
FROM [dbo].[departments] d
WHERE  d.[department_name] LIKE ‘%E%’ OR
    d.[department_name] LIKE ‘%e%’;

TRUNCATE TABLE [dbo].[clone];
```

- EX5

```
MERGE INTO employees AS [Target]
```

```
USING employee_clone AS [Source]
        ON [Target].employee_id = [Source].employee_id
    WHEN MATCHED AND [Target].job_id IN (
    SELECT job.[job_id]
    FROM [dbo].[jobs] job
     WHERE job.[job_title] LIKE '%A%' OR job.[job_title] LIKE '%a%')
        THEN
            UPDATE
            SET     [Target].[salary] = [Source].[salary],
            [Target].[commission_pct] = [Source].[commission_pct];
```

- EX6

```
DECLARE @SQL NVARCHAR(500)
DECLARE @Cursor CURSOR

SET @Cursor = CURSOR FAST_FORWARD FOR SELECT  'DROP TABLE [' +
TABLE_SCHEMA + '].[' + TABLE_NAME + ']' FROM
INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME LIKE '%CLONE'

OPEN @Cursor FETCH NEXT FROM @Cursor INTO @SQL


WHILE (@@FETCH_STATUS = 0)

BEGIN

    print(@SQL)
    EXEC sp_executesql @SQL
    FETCH NEXT FROM @Cursor INTO @SQL
END

CLOSE @Cursor
DEALLOCATE @Cursor
GO
```

- EX7

```
CREATE OR ALTER FUNCTION [dbo].[GET_SALARY]
(
   @EmpId INT
)
RETURNS REAL AS
BEGIN
    DECLARE @salary REAL;

    SELECT @salary = e.[salary]
    FROM [dbo].[employees] e
    WHERE e.[employee_id] = @EmpId;

    RETURN @salary;
END
GO

CREATE OR ALTER FUNCTION [dbo].[GET_SUBORDINATES]
(
   @MgrId INT,
   @Salary REAL = NULL
)
RETURNS INT AS
BEGIN
   IF @Salary IS NULL
      EXEC @Salary = [dbo].[GET_SALARY] @EmpId = @MgrId;

   DECLARE @result int;

   SELECT @result = COUNT(*)
```

```
    FROM [dbo].[employees] EMP
    LEFT OUTER JOIN [dbo].[employees] MGR ON EMP.[manager_id] =
MGR.[employee_id]
    WHERE EMP.[manager_id] = @MgrId AND EMP.[salary] <= @Salary;


     RETURN @result;
END
GO
```

- EX8

  Deoarece ne ajuta sa pastram doar procedurile de care avem nevoie fara a le stoca pe toate, astfel putem modifica moi usor anumite parti ale lor.

- EX9

  Putem atasa si tabelea departments pentru observa ce angajatii sunt manageri de department.