VENDOR INTEGRATION

DEC 2023

Concat, Substring & Order By

LinkedIn Coding Challenge

Intermediate

A company partnered with outside vendors to provide services for their employees. In order to enroll their employees, some information must be shared with the vendors.

Write a query that returns the information in the specified formats.

INPUT FORMAT

The source table is **EMPLOYEES**

Employees		
PK	EmployeeID	int
	FirstName	varchar(255)
	LastName	varchar(255)
	Department	varchar(255)
	Position	varchar(255)
	WeeklyPay	int
	Username	varchar(255)

CONSTRAINTS

- Each employee's ID has a five-digit number (add leading zero's)
- Each employee's Name in the format "Last, First"
- A Login that uses the employee's first initial and then up to 7 characters of their last name (maximum of eight total characters)
- Their work Email, which will be constructed using their Username from the **EMPLOYEES** table and the text "nadias-garden.com"
- Result should be four columns: *ID, Name, Login, Email*. Sort results alphabetically by *Name*.

CODE SOLUTION

```
SELECT

CONCAT(0,0,e.EMPLOYEEID) AS ID,

CONCAT(e.LASTNAME,',',' ',e.FIRSTNAME) AS Name,

LOWER(CONCAT(SUBSTR(e.FIRSTNAME,1,1),SUBSTR(e.LASTNAME,1,7))) AS Login,

CONCAT(e.USERNAME,'@nadias-garden.com') AS EMAIL

FROM Employees e

ORDER BY Name
```

SOLUTION PROCESS

- Concat function: Concatenates two strings next to each other. The use of quotations could be used in the first function as an alternative. However, the use of "00" in that function would add the zeros as a text string rather than keeping the data type uniform in the first column.
- As and renaming functions: The **EMPLOYEES** table is renamed to **e** as best practice because the notation **e.**, when typed into the code space, allows an instant recall of all available columns within the table. The concatenation of various strings creates new columns which are named in compliance with the vendor's request.
- Substr function: This function extracts and returns a specified number of characters from a substring beginning with a specified character placement number.
- Lower function: Formats Login texts to return all lowercase characters because the extraction from the Substr function carried over the proper noun capitalization from the original string.
- Order By function: Results are sorted in alphabetical order based on the Name column.

OUTPUT

```
----------------------------------------------------------------------

| ID    | NAME              | LOGIN    | EMAIL                       |

----------------------------------------------------------------------

| 00100 | Arbore, Nadia     | narbore  | nadia@nadias-garden.com     |

| 00101 | Arbore, Timothy   | tarbore  | tim@nadias-garden.com       |

| 00106 | Bilyard, Gunar    | gbilyard | gbilyard@nadias-garden.com  |

| 00114 | Brophy, Nickola   | nbrophy  | nbrophy@nadias-garden.com   |

| 00109 | Camelin, Jolee    | jcamelin | jcamelin@nadias-garden.com  |

| 00112 | Corde, Charlton   | ccorde   | ccorde@nadias-garden.com    |

| 00111 | Cornelis, Lily    | lcorneli | lcornelis@nadias-garden.com |

| 00110 | Digwood, Earlie   | edigwood | edigwood@nadias-garden.com  |

| 00116 | Dowbekin, Dean    | ddowbeki | ddowbekin@nadias-garden.com |

| 00102 | Galea, Iris       | igalea   | igalea@nadias-garden.com    |

| 00115 | Houliston, Alia   | ahoulist | ahouliston@nadias-garden.com |

| 00107 | Kirwan, Tony      | tkirwan  | tkirwan@nadias-garden.com   |

| 00105 | Lambole, Mel      | mlambole | mlambole@nadias-garden.com  |

| 00113 | Leavesley, Ginnie | gleavesl | gleavesley@nadias-garden.com |

| 00104 | Phear, Witty      | wphear   | wphear@nadias-garden.com    |

| 00108 | Snoday, Fern      | fsnoday  | fsnoday@nadias-garden.com   |

| 00103 | Tschirschky, Mickie | mtschirs | mickie@nadias-garden.com  |

----------------------------------------------------------------------
```