databricks07 - Higher Order Functions

Create Tables

Run the cell below to create tables for the questions in this notebook.

```
%run ../Utilities/07-CreateTables
```

Declared the following table:
- **finances**

Declared the following table:
- **charges**

Question 1: Transform

Summary

Use the `TRANSFORM` function and the table `finances` to calculate `interest` for all cards issued to each user.

Steps to complete

Write a SQL query that achieves the following:

- Displays cardholder's `firstName`, `lastName`, and a new column named `interest`
- Uses `TRANSFORM` to extract charges for each card in the expenses column and calculates interest owed assuming a rate of 6.25%.
- Stores the new values as an array in the interest column
- Stores results in a temporary table named `q1Results`

A properly completed solution should return a view that looks similar to this:

firstName	lastName	interest

```
CREATE OR REPLACE TEMPORARY VIEW q1Results AS
SELECT
firstName,
lastName,
TRANSFORM(expenses, i -> i["charges"]*0.0625) interest
FROM finances;

SELECT * FROM q1Results
```

	firstName	lastName	interest
1	Lance	Da Costa	▸ [138.90375, 373.5475, 158.971875]
2	Emilie	Newlove	▸ [396.520625]
3	Alvy	Records	▸ [573.176875, 200.104375, 380.475625, 212.038125]
4	Jena	Fairley	▸ [607.885, 161.13625]
5	Klarika	Pady	▸ [528.77875, 535.8575]

Showing all 5 rows.

Question 2: Exists

Summary

Use the table from Question 1, `finances`, to flag users whose records indicate that they made a late payment.

Steps to complete

Write a SQL query that achieves the following:

- Displays cardholder's `firstName`, `lastName`, and a new column named `lateFee`
- Uses the EXISTS function to flag customers who have made been charged a late payment fee.
- Store the results in a temporary view named `q2Results`

A properly completed solution should return a DataFrame that looks similar to this:

firstName	lastName	lateFee
Lance	DaCosta	true

```
CREATE OR REPLACE TEMPORARY VIEW q2Results AS
SELECT
firstName,
lastName,
EXISTS(expenses, i -> i["charges"]>0) lateFee
FROM finances;

SELECT * FROM q2Results
```

	firstName	lastName	lateFee
1	Lance	Da Costa	true
2	Emilie	Newlove	true
3	Alvy	Records	true
4	Jena	Fairley	true
5	Klarika	Pady	true

Showing all 5 rows.

Question 3: Reduce

Summary

Use the `REDUCE` function to produce a query on the table `charges` that calculates total charges in dollars and total charges in Japanese Yen.

Steps to complete

Write a SQL query that achieves the following:

- Uses the `REDUCE` function to calculate the total charges in US Dollars (given)
- Uses the `REDUCE` function to convert the total charges to Japanese Yen using a conversion rate where 1 USD = 107.26 JPY
- Stores the results in a temporary table named `q3Results`

NOTE: In the `REDUCE` function, the accumulator must be of the same type as the input. You will have to `CAST` the accumulator as a `DOUBLE` to use this function with this data. example: `CAST (0 AS DOUBLE`)

A properly completed solution should return a DataFrame that looks similar to this:

firstName	lastName	allCharges	totalDollars	totalYen
Lance	DaCosta	["2222.46", "5976.76", "2543.55"]	10742.77	1152269.51

```
--CREATE OR REPLACE TEMPORARY VIEW q3Results AS
SELECT
  firstName,
  lastName,
  REDUCE(allCharges, CAST(0 AS double), (c, acc) -> c + acc, acc -> (acc))  AS
totalDollars,
  REDUCE(allCharges, CAST(0 AS double), (c, acc) -> c + acc, acc ->
(acc*107.26)) AS totalYen
FROM
(
  SELECT
    firstName,
    lastName,
    TRANSFORM (allCharges, t -> CAST(t AS double)) allCharges
  FROM charges
);
--SELECT * FROM q3Results
```

	firstName	lastName	totalDollars	totalYen
1	Lance	Da Costa	10742.77	1152269.5102000001
2	Emilie	Newlove	6344.33	680492.8358
3	Alvy	Records	21852.72	2343922.7472

4	Jena	Fairley	12304.34	1319763.5084000002
5	Klarika	Pady	17034.18	1827086.1468000002

Showing all 5 rows.

```
CREATE OR REPLACE TEMPORARY VIEW q3Results AS
SELECT
  firstName,
  lastName,
  REDUCE(allCharges, CAST(0 AS double), (c, acc) -> c + acc, acc -> CAST(acc AS double))  AS totalDollars,
  REDUCE(allCharges, CAST(0 AS double), (c, acc) -> c + acc, acc -> (CAST(acc AS double)*107.26)) AS totalYen
FROM
charges;
```

```
SELECT * FROM q3Results
```

	firstName	lastName	totalDollars	totalYen
1	Lance	Da Costa	10742.77	1152269.5102000001
2	Emilie	Newlove	6344.33	680492.8358
3	Alvy	Records	21852.72	2343922.7472
4	Jena	Fairley	12304.34	1319763.5084000002
5	Klarika	Pady	17034.18	1827086.1468000002

Showing all 5 rows.