**Lab 3**

1. Create a table with two columns: FirstName, Father.
   * Insert the data of your family members (at least 5 members): you, your father, grandfather, uncle, siblings and cousins
   * Write a query to retrieve the first names of your family members and their place in the family hierarchy, starting at your grandfather with level 1.
   * Write the query once using WITH Recursive and once using Oracle Start with .. Connect By

CREATE TABLE Family (

FirstName VARCHAR2(50),

Father VARCHAR2(50)

);

INSERT INTO Family (FirstName, Father) VALUES ('Ahmed', NULL);

INSERT INTO Family (FirstName, Father) VALUES ('Ali', 'Ahmed');

INSERT INTO Family (FirstName, Father) VALUES ('Israa', 'Ali');

INSERT INTO Family (FirstName, Father) VALUES ('Mohamed', 'Ahmed');

INSERT INTO Family (FirstName, Father) VALUES ('Hasnaa', 'Ali');

INSERT INTO Family (FirstName, Father) VALUES ('Akram', 'Mohamed');

INSERT INTO Family (FirstName, Father) VALUES ('Amr', 'Mohamed');

WITH FamilyTree(FirstName, Father, familyHierarchy) as (

SELECT FirstName, Father, 1 AS familyHierarchy

FROM Family

WHERE Father IS NULL

UNION ALL

SELECT f.FirstName, f.Father, ft.familyHierarchy + 1

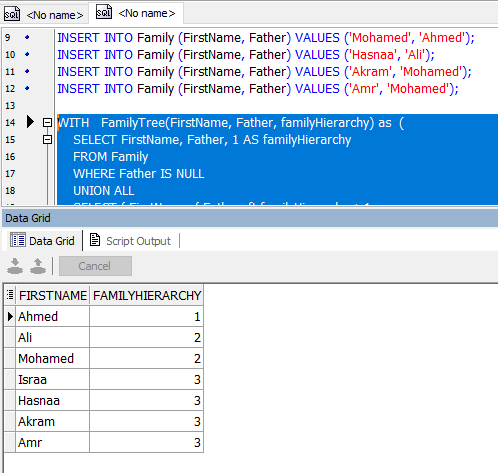
FROM Family f

INNER JOIN FamilyTree ft ON f.Father = ft.FirstName

)

SELECT FirstName, familyHierarchy

FROM FamilyTree;

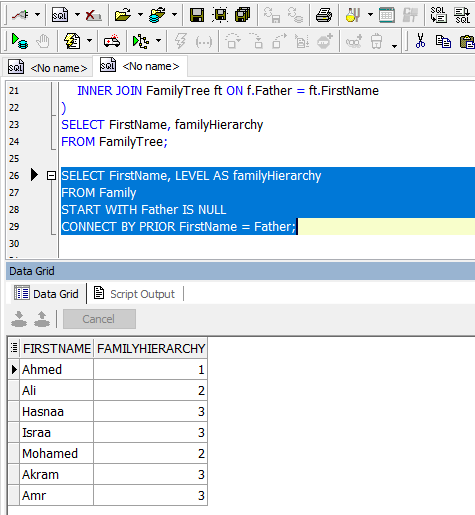


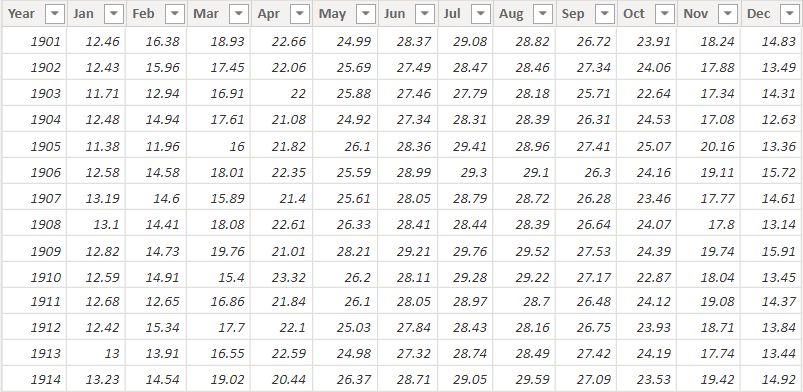
SELECT FirstName, LEVEL AS familyHierarchy

FROM Family

START WITH Father IS NULL

CONNECT BY PRIOR FirstName = Father;



1. The following dataset contains the monthly avg temperature in Egypt in the last 100 years, We want to create a line chart to visualize how the temperature has changed over the years, however, the current table structure is not suitable for visualization, derive a new table that can be used to create a line chart with two columns, Date and Temperature.
   * Before 

* After



CREATE TABLE visualization AS

SELECT Temperature, TO\_CHAR(TO\_DATE(month || ' ' || year, 'Mon YYYY'), 'MM YYYY') AS "date"

FROM (

SELECT year,

jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

FROM temp

) UNPIVOT (

Temperature FOR month IN (

jan as 'JAN', feb as 'FEB', mar as 'MAR',

apr as 'APR', may as 'MAY', jun as 'JUN',

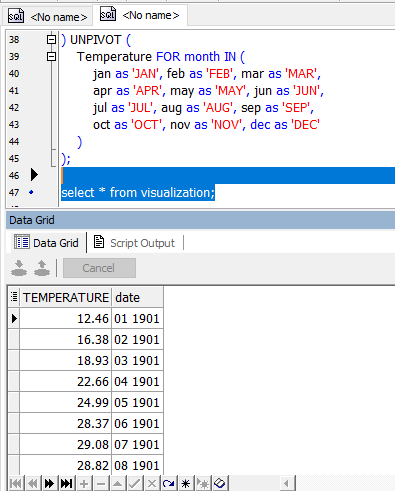
jul as 'JUL', aug as 'AUG', sep as 'SEP',

oct as 'OCT', nov as 'NOV', dec as 'DEC'

)

);

select \* from visualization;



1. Choose any 2 of the queries you answered in previous labs, and show how you would improve them using CTEs. Make sure to include the before and after in this file.
2. We need to categorize our customers based on how much sales they make, divide the customers into 3 equal groups, and give them labels from lowest to highest sales as follows

* Customers Needing Attention
* Promising Customers
* High-Value Customers

select

customername as customer\_name,total\_sales,

case

when category = 1 then 'Customers Needing Attention'

when category = 2 then 'Promising Customers'

when category = 3 then 'High-Value Customers'

end as category

from (select CustomerName,sum(sales) as total\_sales,ntile(3) over (order by sum(sales)) as category

from sales

group by customername

);

WITH CustomerSales AS (

SELECT

CustomerName,

SUM(sales) AS total\_sales

FROM

sales

GROUP BY

customername

),

CategorizedCustomers AS (

SELECT

CustomerName,

total\_sales,

NTILE(3) OVER (ORDER BY total\_sales) AS category

FROM

CustomerSales

)

SELECT

customername,

total\_sales,

CASE

WHEN category = 1 THEN 'Customers Needing Attention'

WHEN category = 2 THEN 'Promising Customers'

WHEN category = 3 THEN 'High-Value Customers'

END AS category

FROM

CategorizedCustomers;

…………………………………………………………………………………………………………………….

1. The company wants to analyze the purchase behavior of its customers over time to identify trends and patterns.

One question the company wants to answer is: "For each customer, what was their first and last purchase date?"

select distinct CustomerName,

first\_value(OrderDate) over (partition by CustomerName order by to\_date(orderdate, 'MM/DD/YYYY')) as first\_purchase,

last\_value(OrderDate) over (partition by CustomerName order by to\_date(orderdate, 'MM/DD/YYYY') rows between unbounded preceding and unbounded following) as last\_purchase

from sales

order by customername;

WITH OrderedSales AS (

SELECT

CustomerName,

OrderDate,

ROW\_NUMBER() OVER (PARTITION BY CustomerName ORDER BY TO\_DATE(OrderDate, 'MM/DD/YYYY')) AS rn\_asc,

ROW\_NUMBER() OVER (PARTITION BY CustomerName ORDER BY TO\_DATE(OrderDate, 'MM/DD/YYYY') DESC) AS rn\_desc

FROM

sales

),

FirstLastPurchase AS (

SELECT

CustomerName,

MAX(CASE WHEN rn\_asc = 1 THEN OrderDate END) AS first\_purchase,

MAX(CASE WHEN rn\_desc = 1 THEN OrderDate END) AS last\_purchase

FROM

OrderedSales

GROUP BY

CustomerName

)

SELECT

CustomerName,

first\_purchase,

last\_purchase

FROM

FirstLastPurchase

ORDER BY

CustomerName;