```
-- START OF FILE ==================================================================
--
--                              ============================================
--
--                   HATFIELD APPLE TREE SUPPLIERS -- ASSIGNMENT 3
--
--                   ============================================
--
-- TASK ONE - Insertion of data ( 3 sub-tasks )
-- TASK TWO - Queries ( 12 in total )
--
-- PLESE READ THIS DOCUMENT CAREFULLY.
-- ----------------------------------
--
-- BEFORE YOU START you must check you have completed the following:
--
--  0 Download HATS_Create_Tables.sql from CANVAS. It is available from the
--    assignment description
--  0 Download HATS_Populate_Tables.sql from CANVAS. It is available from the
--    assignment description
--  0 Run script (F5) on HATS_Create_Tables.sql in an SQL Developer worksheet
--    You should see the following output:
--
--  PL/SQL procedure successfully completed.
--
--  Dropping Tables ...
--    ... A3_ORDER_ITEM    \
--    ... A3_ORDER_FORM     \
--    ... A3_BATCH           \
--    ... A3_TREE             > Only if tables already exist
--    ... A3_CUSTOMER        /
--    ... A3_VARIETY        /
--    ... A3_ROOTSTOCK     /
--  Tables dropped.
--  Create Tables ...
--    ... A3_Rootstock
--    ... A3_Variety
--    ... A3_Tree
--    ... A3_Tree
--    ... A3_Batch
--    ... A3_Order_Form
--    ... A3_Order_Item
--  Tables Created
--
--  0 Run script (F5) on HATS_Populate_Tables.sql in an SQL Developer worksheet
--    You should see the following output:
--
--  PL/SQL procedure successfully completed.
--
--  Populate Tables ...
--    A3_Rootstck |.......| 7 records
--    A3_Variety |.................................| 33 records
--    A3_Tree |..................................// .............| 76 records
--    A3_Customer |..............................// ...........| 63 records
--    A3_Batch |..................................// ..............| 157 records
--    A3_Order_Form |..........................// ............| 70 records
--    A3_Order_Item |..........................// ..............| 118 records
--
--  0 Confirm that 7 new tables exist all starting in A3_ and they contain data
```

```
--
--  0 PLEASE NOTE: all tables referenced in your inserts and queries must start
--    with A3_ . This distinguishes them from previous tables and they all appear
--    in a group on the SQL Developer connections window.
--
--  0 Check each table contains data records. The number of records in each is
--    given above as reported when running HATS_Tables_and_Data.sql.
--
--  0 ONLY once you are happy with the tables and data, should you progress to
--    the two tasks:
--
--
-- TASK ONE - ADDING DATA - 16 marks in total
-- ======================
--
-- You must add the following data to that already inserted into the 7 table
-- schema.
-- DO NOT remove any existing records
--
-- ADD the following:
--
--  1 - Create a new Customer which includes your name and where the Customer_ID
--      is C9xxxx where xxxx are the last 4 numbers of your SID. the remaining
--      attributes can be data you make up
      [ 4 marks ]
--
--  2 - Create two orders with you as the customer.
--
--      Order 1 MUST contain a single order of two trees of your choice from
--      any batch.
--
                  [ 4 marks ]
--
--      Order 2 MUST be an order(_form) which contains the following items:
--         - two trees from one batch, which must be on a Dwarfing rootstock
--         - one tree from a different batch which is an eating apple
--         - one tree from another different batch which has a heavy yield
--
                  [ 8 marks ]
--
--
--  NOTES:
--    Its a good idea to store your final INSERT commands in a script file
--
--    If during this process, you corrupt the dataset, go back and use the
--  script downloaded to reset the original tables and data
--
--    Once you are happy ALL INSERTS are correct, it may be a good idea to
--  run the two scripts (supplied and yours) again to refresh the
--  dataset before starting Task 2
--
-- END OF ASSIGNMENT TASK ONE ----------------------------------------------------
--
--
-- TASK TWO - QUERYING  [ 7 marks per query ] 84 in total
-- ==================
--
-- For this task use SQL Developer to build queries that provide the correct
-- answer to the question asked. Once the query is correct, COPY THE CODE INTO
```

```
-- THE SPACES PROVIDED. Answer as many questions if you can.
--
-- Hints are provided to help you understand what is needed
--
-- Solution Tests indicate how the output should appear if correct.
--
-- Submission instructions are given at the end of this file.
--
--
-- QUESTION 1
-- ==========
--
-- A customer has an existing tree which has a "late" fruiting season, they wish
-- to compliment this with an early fruiting variety.
-- Provide a list of all apple varieties which fruit early. The report should
-- give the name of the Variety, the fruit colour and the yield.
--
-- Solution Test: The list will contain 8 varieties
--
-- Type your query below:

SELECT Name AS "Name",FRUIT_COLOUR AS "Fruit Colour",YIELD AS "Yield"
FROM A3_VARIETY
WHERE FRUITING_SEASON='Early';




-- QUESTION 2
-- ==========
--
-- Provide a customer with a list of all trees HATS have in stock, that grow on
-- M9 rootstock. Order the output starting with the most expensive. The report
-- should provide the Name of the Variety, the price and the Tree_ID.
--
-- Solution Test: Prices range from 95.60 to 17.60
--
-- Type your query below:

SELECT VARIETY AS "Variety list of M9 rootstock",ID AS "ID",TO_CHAR(PRICE,
'FM9999999.90') AS "Price"
FROM A3_TREE
WHERE RootType='M9'
ORDER BY PRICE DESC;



-- QUESTION 3
-- ==========
--
-- Write a query that returns the maximum number of trees placed on any order.
--
-- Hint : Aggregation functions can be nested
--
-- Solution Test: Maximum is 7
--
--
-- Type your query below:
```

```
SELECT MAX(SUM(QUANTITY)) AS "Max no of Trees placed on 1 order"
FROM A3_ORDER_ITEM
GROUP BY ORDER_ID;



-- QUESTION 4
-- ==========
--
-- Use a nested select statement to answer the following:
-- List all the customers with the same Given_Name as the Customer with the
-- Customer_ID 'C11834'. The output should provide a SINGLE column giving the
-- full name of the customers required in a readable form of your choice.
-- The heading for this column should be "Name"
--
-- Hint: Identify the inner and outer queries
--
-- Solution Test: 3 customers, including C11834 share that given name,
--                unless you have the same name of course.
--
-- Type your query below:

SELECT   GIVEN_NAME || ' ' || FAMILY_NAME  AS "Name"
FROM A3_CUSTOMER
WHERE GIVEN_NAME=(SELECT C.GIVEN_NAME FROM A3_CUSTOMER C WHERE
C.CUSTOMER_ID='C11834' );



-- QUESTION 5
-- ==========
--
-- List all orders in which more than one type of tree has been requested.
-- The report should contain the Order_ID and how many different tree types were
-- ordered. Provide meaningful column headings and Order the report by most types
-- first.
--
-- Hint: Think about the Order_Item records as components of an Order_Form
--
-- Solution Test: Including YOUR order 2 with multiple tree types, there should
-- be 28 multi-tree orders.
--
-- Type your query below:

SELECT O.ORDER_ID AS "Order ID",Count(T.VARIETY) AS "Types of Trees requested"
FROM  A3_ORDER_FORM O , A3_ORDER_ITEM OI ,A3_TREE T
WHERE O.ORDER_ID=OI.ORDER_ID and T.ID=OI.TREE_ID
GROUP BY O.ORDER_ID
HAVING COUNT(T.VARIETY)>1
ORDER BY COUNT(T.VARIETY) DESC;



-- QUESTION 6
-- ==========
--
-- List all trees which consist of varieties that have a pollinate
-- group of 1. List the Variety name and tree number.
```

```
--
-- Hint: Which tables do you need to combine to answer this query?
--
-- Solution Test: There are 3 relevant batches.
--
-- Type your query below:

SELECT T.VARIETY AS "Varieties with pollin group 1",T.ID AS "ID"
FROM  A3_TREE T ,A3_VARIETY V
WHERE T.VARIETY=V.NAME and V.POLLINATION_GROUP=1;



-- QUESTION 7
-- ==========
--
-- List all customers who have yet to place an order. The list should be a single
-- column with a meaningful heading and the following format:
--
--    Morris, Paul ( C00000 )
--
-- and ordered alphabetically as presented.
--
-- Hint: You are looking for records containing a PK which is not present in the
--       related FK column, so choose the appropriate type of join.
--       Remember all attributes that are reported un-aggregated in the SELECT
--       Clause must appear in the GROUP BY list.
--
-- Solution Test: There are 22 customers yet to order the last listed is C39202
--
-- Type your query below:

SELECT FAMILY_NAME || ', '|| GIVEN_NAME || '( ' || CUSTOMER_ID|| ' )' AS "Customers
yet to place an order"
FROM A3_CUSTOMER
WHERE CUSTOMER_ID NOT IN (Select O.CUSTOMER_ID from A3_ORDER_FORM O)
ORDER BY FAMILY_NAME,GIVEN_NAME;



-- QUESTION 8
-- ==========
--
-- List the name of all customers who have placed an order for one or more
-- 'Royal Gala' trees. The report should contain their name in full as a single
-- column, their phone number, how many trees where ordered, and the date the
-- order was placed. Repeat orders may be displayed for an individual customer.
--
-- Hint: This will require five (5) tables (including A3_Batch) to be joined.
--       Use the latest E-R diagram to confirm this. Remember, you can ONLY join
--       tables that are directly related.
--       Don't forget to use explicit type conversion.
--
-- Solution Test: 10 orders are placed for this tree (you may also have ordered
--                it additionally). One person placed three separate orders.
--
-- Type your query below:
```

```
SELECT CONCAT(CONCAT( A3_Customer.Family_Name, ' '), A3_Customer.Given_Name) AS
"Full Name", A3_Customer.Phone_No AS "Phone No", SUM(A3_Order_Item.Quantity) AS
"Total Trees Ordered", A3_Order_Form.Order_Date AS "Order Date"
FROM A3_Customer
INNER JOIN A3_Order_Form ON A3_Customer.Customer_ID=A3_Order_Form.Customer_ID
INNER JOIN A3_Order_Item ON A3_Order_Form.Order_ID=A3_Order_Item.Order_ID
INNER JOIN A3_Tree ON A3_Order_Item.Tree_ID=A3_Tree.ID
INNER JOIN A3_Batch ON A3_Tree.ID=A3_Batch.Tree_ID
WHERE A3_Tree.Variety='Royal Gala'
GROUP BY A3_Customer.Family_Name, A3_Customer.Given_Name, A3_Customer.Phone_No,
A3_Order_Item.Quantity, A3_Order_Form.Order_Date, A3_Tree.ID;



-- QUESTION 9
-- ==========
--
-- Report the Variety name, Fruiting Season and Yield for the cooking apple
-- variety which HATS holds the most stock of.
--
-- HINT: DIFFICULT :: Comparison of two aggregation values must be done in the
--                    HAVING clause. This is a nested query as you need to know
--                              the maximum value before answering the main
question.
--
-- Solution Test: This variety is an early fruiter with a heavy yield.
--
-- Type your query below:

Select NAME AS "Name",V.FRUITING_SEASON AS "Fruiting Season",V.YIELD AS "Yield"
FROM A3_VARIETY V
JOIN A3_TREE T  ON (T.VARIETY=V.NAME)
JOIN A3_BATCH B  ON (T.ID=B.TREE_ID)
GROUP BY T.ID,NAME,V.FRUITING_SEASON,V.YIELD,PLANTS_IN_BATCH
HAVING B.PLANTS_IN_BATCH=(Select Max(PLANTS_IN_BATCH) FROM A3_BATCH);



-- QUESTION 10, 11, 12
-- ===================
-- Write three queries to provide information about YOU and YOUR orders from
-- Task 1
--
-- 10
-- ==
--
-- Create a single line report containing YOUR details as entered on the database
-- in the following format:
--
--    Fullname         Address                                Registered for

--    ---------------- -------------------------------------- -----------------
--    Leon Marvin      The Marches, Teal Avenue, Lindon     4y 3m
--
-- Where 4y 3m indicates this customer has been registered for 4 complete years
-- and 3 complete months at the time the query is run
--
-- Hint: You will need one mathematical function not mentioned so far on the
```

```
--       module, see:
--
--
https://docs.oracle.com/cd/E11882_01/server.112/e41084/functions002.htm#SQLRF51178
--
-- Type your query below:

Select FAMILY_NAME || ' ' || GIVEN_NAME AS "Full Name", ADDRESS AS "Address",
TRUNC(months_between (sysdate, REGISTER_DATE)/12)||'y '  ||
TRUNC(MOD( months_between (sysdate, REGISTER_DATE),12)) ||'m' AS "Registered for"
FROM A3_CUSTOMER
WHERE CUSTOMER_ID='C91398';



--
-- 11
-- ==
--
-- Write a query to output the details of YOUR Order 1 using the Order Number
-- in the WHERE clause only.
-- The details needed are your name, delivery address, tree variety, price
-- and quantity.
--
-- The headings and details should be meaningful to any reader of the report.
-- I.e.
--
--  Name               Address                         Tree Variety      Price

--  ------------------ ------------------------------ ---------------- --------
--  Michael Simmonds   28, Highridge Way, Layburn     Feille Morte        £42.60
--
-- Hint: To output the £ symbol, put an L at the front of the format string.
--
-- Type your query below:

Select FAMILY_NAME || ' ' || GIVEN_NAME AS "Name", ADDRESS AS "Address",T.VARIETY
AS "Tree Variety", TO_CHAR(T.PRICE,'L9,999,999.99') AS "Price" ,OI.QUANTITY AS
"Quantity"
FROM A3_CUSTOMER  C
JOIN A3_ORDER_FORM O ON (O.Customer_ID=C.CUSTOMER_ID)
JOIN A3_ORDER_ITEM OI ON (O.ORDER_ID=OI.ORDER_ID)
JOIN A3_TREE T ON (T.ID=OI.TREE_ID)
WHERE O.ORDER_ID='00000071';


--
-- 12
-- ==
--
-- Write a query to output the details of YOUR Order 2
-- The details needed are the Order_ID, the total number of
-- trees on the order and the total cost of the trees ordered.
--
-- The headings and details should be meaningful to any reader of the report. I.e.
--
--  Order_ID  Number of Trees Total Price
--  -------- ---------------- -------------
--  00000024                7       £247.00
```

```
--
-- Hint: Selection should be based on the Order_ID of your order 2
--
-- Type your query below:

SELECT ORDER_ID AS "Order_ID", SUM(QUANTITY) AS "Number of Trees",
TO_CHAR(SUM(QUANTITY * A3_TREE.PRICE), 'L9,999,999.99') AS "Total Price"
FROM A3_ORDER_ITEM
INNER JOIN A3_TREE ON A3_ORDER_ITEM.TREE_ID=A3_TREE.ID
WHERE ORDER_ID = '00000072'
GROUP BY ORDER_ID;



-- END OF ASSIGNMENT TASK TWO ----------------------------------------------------

-- SUBMISSION REQUIREMENTS
-- ======================
--
-- Once your queries are tested and the final code placed in the file above in
-- the appropriate places, the following should be done in order to meet the
-- submission requirements.
--
-- MARKS WILL BE DEDUCTED FOR FAILURE TO FOLLOW THESE INSTRUCTIONS.
--
--  1  Rename this file in the following format:
--
--                aa99aaa-Ass3.sql
--
--     where the aa99aaa is replaced by YOUR Oracle username
--
--  2  Open this file in an SQL Worksheet in SQL Developer, clear the Script
--     Output window using the eraser icon, and ensure your 9 table schema is
--     correct and includes your entered data.
--
--  3  Use the "Run Script (F5)" icon (sheet of paper with small green triangle)
--     to run your script completely. Ensure all commands are run.
--
--  4  Save the Script Output text by clicking on the floppy disk icon, use the
--     popup window to save the file as aa99aaa-Ass3.txt, again relacing aa99aaa
--     with your username.
--
--  5  Double check both the aa99aaa-Ass3.sql and aa99aaa-Ass3.txt files, then
--     upload them onto CANVAS in the Assignment 3 point.
--
--  6  Congratualations, you are finished!
--
-- END ===============================================================================
```