UNIVERSITY OF ZIMBABWE

COURSE NAME: **BIG DATA TECHNOLOGIES**

COURSE CODE: **HDS301**

EXAMINER : **MR GODFREY MBIZO**

INSTRUCTION SHEET

Follow the following instructions for this practical examination

---

Duration: 3 hours
Instructions:

1. All queries must run successfully on the AdventureWorks2019 (OLTP) database.
2. Submit the exact SQL code, query execution plans, and result sets.
3. Avoid hardcoding where dynamic solutions are possible.
4. Zip all your scripts with question number and put in a folder with your name and reg number
5. Crying is allowed but do it quietly so that you don't disturb other candidates
6. This is open book exam-you can use all allowed material like books, existing scripts but **use of internet is not allowed**

Question 1: Subqueries (20 Marks)

Write a query to find the ProductID, Name, and ListPrice of the **top 5 most expensive products** in each ProductSubcategoryID.

- Use a subquery to calculate the rank of each product based on ListPrice within its subcategory.
- Exclude subcategories that have fewer than 5 products.

Display the ProductSubcategoryID and the average ListPrice of products in each subcategory alongside the results.

Question 2: Complex Aggregation and Grouping (20 Marks)

Using the Sales.SalesOrderDetail and Production.Product tables, write a query to calculate the following for each ProductModelID (from Production.Product):

- Total quantity sold (SUM(OrderQty))
- Total revenue (SUM(LineTotal))
- The percentage of total revenue contributed by each ProductModelID to the overall revenue across all models.

Conditions:

- Include only products with more than 1000 total sales (SUM(OrderQty) > 1000).
- Exclude any products that belong to subcategories with the word "Bike" in their name (use ProductSubcategoryID and Production.ProductSubcategory).

Question 3: Joins with Four Tables and Filtering (20 Marks)

Write a query to identify **sales representatives** who have handled at least 50 orders in 2013 and 2014 combined, but have no orders recorded in 2015.

- Display the following:
  - Sales Rep FullName (combine FirstName and LastName from Person.Person)
  - Total orders handled in 2013-2014
  - Total revenue generated in 2013-2014
  - Hire date (HireDate)

You will need to join the following tables:

- Sales.SalesOrderHeader
- Sales.SalesPerson
- Person.Person
- HumanResources.Employee

Order the results by total revenue in descending order.

Question 4: Advanced Window Functions (20 Marks)

Write a query using **window functions** to analyze sales trends. The query should return the following for each sales representative in Sales.SalesPerson:

- Sales Rep FullName (from Person.Person)
- Yearly sales total (SUM(LineTotal) grouped by year)
- Rolling 3-year average of sales totals, starting in 2012.

Conditions:

- Use Sales.SalesOrderHeader and Sales.SalesOrderDetail.
- Include only sales representatives who have been active for at least 3 years (i.e., their first sale occurred in or before 2012).

Question 5: Ranking and Analytics (20 Marks)

Using the Sales.SalesOrderDetail and Production.Product tables, write a query to rank products within each ProductSubcategoryID based on their total revenue (SUM(LineTotal)). The output should include:

- ProductID
- Name
- Total revenue (SUM(LineTotal))
- Rank within the subcategory (ties should have the same rank).

Bonus: Include a column that calculates the percentage difference in revenue between the current product and the top-ranked product in its subcategory.

--**THE END!**!--