# PART 1: Use database Adventurework

# Task 1: Generate invoice reports

Adventure Works Cycles sells directly to retailers, who must be invoiced for their orders. You have been tasked with writing a query to generate a list of invoices to be sent to customers.

### 1.1 Retrieve customer orders

* As an initial step towards generating the invoice report, write a query that returns the company name from the **SalesLT.Customer** table, and the sales order ID and total due from the **SalesLT.SalesOrderHeader** table.

 -- write a query that returns the company name from the SalesLT.Customer table, and the sales order ID and total due from the SalesLT.SalesOrderHeader table.

SELECT CompanyName

     , SalesOrderID

     , TotalDue

FROM SalesLT.SalesOrderHeader AS ord\_head

LEFT JOIN SalesLT.Customer AS cus

ON ord\_head.CustomerID = cus.CustomerID

### 1.2 Retrieve customer orders with addresses

* Extend your customer orders query to include the *Main Office* address for each customer, including the full street address, city, state or province, postal code, and country or region
* **Tip**: Note that each customer can have multiple addressees in the **SalesLT.Address** table, so the database developer has created the **SalesLT.CustomerAddress** table to enable a many-to-many relationship between customers and addresses. Your query will need to include both of these tables, and should filter the results so that only *Main Office* addresses are included.

SELECT AddressLine1, AddressLine2, City, StateProvince, PostalCode, CountryRegion -- including the full street address, city, state or province, postal code, and country or region

FROM SalesLT.Address AS adr

FULL JOIN SalesLT.CustomerAddress AS cus\_adr

ON adr.AddressID = cus\_adr.AddressID

WHERE AddressType = 'Main Office' -- Extend your customer orders query to include the Main Office address for each customer

# Task 2: Retrieve customer data

As you continue to work with the Adventure Works customer, product and sales data, you must create queries for reports that have been requested by the sales team.

### Retrieve a list of products

○ A sales manager needs a **list of ordered products** with more information.

You must write a query that returns a list of product name (is generated by the string preceded by the '-' character (example: HL Road Frame) ), only started selling in 2006, Product model name contains "Road", CategoryName contains "Bikes" and ListPrice value with integer part equal to 2443

SELECT substring(prod.Name,0,CHARINDEX('-', prod.Name)) as [ProductName]-- -- write a query that returns a list of product name (is generated by the string preceded by the '-' character (example: HL Road Frame)

     , SellStartDate

     , prod\_model.Name

     , prod\_cat.Name

     , ListPrice

FROM SalesLT.Product as prod

LEFT JOIN SalesLT.ProductCategory as prod\_cat

ON prod.ProductCategoryID = prod\_cat.ProductCategoryID

LEFT JOIN SalesLT.ProductModel as prod\_model

ON prod.ProductModelID = prod\_model.ProductModelID

WHERE prod\_model.Name LIKE '%Road%' -- Product model name contains "Road"

AND prod\_cat.Name LIKE '%Bikes%' -- CategoryName contains "Bikes"

AND Year(SellStartDate) = 2006  -- only started selling in 2006

AND CONVERT(integer,ListPrice) = 2443 -- ListPrice value with integer part equal to 2443

# PART 2: Use database PayTM

# Description:

# Paytm is an Indian multinational financial technology company specializing in digital payment system, e-commerce and financial services. Paytm wallet is a secure and RBI-approved digital/mobile wallet that you can use for multiple purposes. It is like digital cash that you can utilize for any kind of consumer payment. You can add money to the Paytm wallet through UPI, internet banking, or credit/debit cards. Also, you can send money from a Paytm wallet to a bank account or another person’s Paytm wallet.

# This is a small database of payment transactions from 2019 to 2020 of PayTM. The database includes 6 tables:

# fact\_transaction: Store information of all types of transactions: Payments, Top-up, Transfers, Withdrawals

# dim\_scenario: Detailed description of transaction types

# dim\_payment\_channel: Detailed description of payment methods

# dim\_platform: Detailed description of payment devices

# dim\_status: Detailed description of the results of the transaction

# Task 1: Retrieve reports on transaction scenarios

# 1.1 Retrieve a report that includes the following information: customer\_id, transaction\_id, scenario\_id, transaction\_type, sub\_category, category. These transactions must meet the following conditions:

# Were created in Jan 2019

# Transaction type is not payment

-- Retrieve a report that includes the following information: customer\_id, transaction\_id, scenario\_id, transaction\_type, sub\_category, category.

SELECT transaction\_id

     , customer\_id

     , fact\_2019.scenario\_id

     , transaction\_type

     , sub\_category

     , category

FROM fact\_transaction\_2019 as fact\_2019

LEFT JOIN dim\_scenario as scen

ON fact\_2019.scenario\_id = scen.scenario\_id

WHERE month(transaction\_time) = 1 -- Were created in Jan 2019

AND NOT transaction\_type = 'Payment' -- Transaction type is not payment

# 1.2 Retrieve a report that includes the following information: customer\_id, transaction\_id, scenario\_id, transaction\_type, category, payment\_method. These transactions must meet the following conditions:

# Were created from Jan to June 2019

# Had category type is shopping

# Were paid by Bank account

SELECT transaction\_id

     , customer\_id

     , fact\_2019.scenario\_id

     , transaction\_type

     , category

     , payment\_method

     , transaction\_time

FROM fact\_transaction\_2019 as fact\_2019

LEFT JOIN dim\_scenario as scen

ON fact\_2019.scenario\_id = scen.scenario\_id

LEFT JOIN dim\_payment\_channel as pay\_chan

ON fact\_2019.payment\_channel\_id = pay\_chan.payment\_channel\_id

WHERE (month(transaction\_time) >= 1 AND month(transaction\_time) <= 6) -- Were created from Jan to June 2019

AND category = 'Shopping' -- Had category type is shopping

AND payment\_method = 'Bank account' -- Were paid by Bank account