Program listing:

# The SQLar Federation  
# Emmanuel Diaz, Sarah Ewing, Melissa Lawrence, Edgar Rosales  
# Outdoor Enthusiasts case  
# 05/11/2024  
# Module 11.1 Assignment  
  
import mysql.connector  
from mysql.connector import Error  
from tabulate import tabulate  
from datetime import datetime, timedelta  
  
config = {  
 "user": "root",  
 "password": "G0dzilla#007",  
 "host": "localhost",  
 "database": "outdoor",  
 "raise\_on\_warnings": True  
}  
  
# Initialize variables  
db = None  
cursor = None  
  
try:  
 # Establish connection to the database  
 db = mysql.connector.connect(\*\*config)  
 cursor = db.cursor()  
  
 # Query to get total sales count  
 query = "SELECT COUNT(\*) AS Total\_Sales FROM equipment\_transactions WHERE IsRental = FALSE;"  
 cursor.execute(query)  
 total\_sales = cursor.fetchone()[0]  
  
 # Query to get total available sales count  
 query = "SELECT COUNT(\*) AS Total\_Available\_Sales FROM equipment WHERE ForSale = True;"  
 cursor.execute(query)  
 total\_available\_sales = cursor.fetchone()[0]  
  
 query = """SELECT CASE et.IsRental WHEN 1 THEN 'Rental' ELSE 'Purchase' END AS 'Sales Method',  
 SUM(CASE et.IsRental WHEN 1 THEN DATEDIFF(trip.EndDate, trip.StartDate) \* et.Cost  
 ELSE et.Cost END) AS 'Sales Total' FROM equipment\_transactions AS et  
 JOIN trip ON et.TripID = trip.TripID GROUP BY et.IsRental;"""  
 cursor.execute(query)  
 sales\_info = cursor.fetchall()  
 totRental = sales\_info[0][1]  
 totPurchase = sales\_info[1][1]  
 formatted\_sales\_info = [(method, "${:,.2f}".format(sales)) for method, sales in sales\_info]  
  
 # Query to get total bookings by continent and country  
 query = """  
 SELECT d.Continent, d.Country, COUNT(b.BookingID) AS Total\_Bookings  
 FROM booking b  
 JOIN trip t ON b.TripID = t.TripID  
 JOIN destinations d ON t.DestinationID = d.DestinationID  
 WHERE d.Continent IN ('Africa', 'Asia', 'Europe')  
 GROUP BY d.Continent, d.Country;  
 """  
 cursor.execute(query)  
 booking\_results = cursor.fetchall()  
  
 # Query to get inventory items over five years old  
 five\_years\_ago = datetime.now() - timedelta(days=1825) # Five years ago from today  
 query = "SELECT COUNT(\*) AS old\_inventory FROM equipment WHERE PurchaseDate < %s;"  
 cursor.execute(query, (five\_years\_ago,))  
 old\_inventory = cursor.fetchone()[0]  
  
 # Query to get details of inventory items over five years old  
 query = "SELECT EquipmentID AS ID, Name AS Item, PurchaseDate AS Date FROM equipment WHERE PurchaseDate < %s;"  
 cursor.execute(query, (five\_years\_ago,))  
 old\_items = cursor.fetchall()  
  
 # Query to get total sales from equipment\_transactions  
 query = "SELECT SUM(Cost) AS Total\_Equipment\_Transactions\_Cost FROM equipment\_transactions WHERE IsRental = FALSE;"  
 cursor.execute(query)  
 total\_equipment\_transactions\_cost = cursor.fetchone()[0]  
  
 # Query to get total sales from booking  
 query = """  
 SELECT SUM(b.Cost) AS Total\_Booking\_Cost  
 FROM booking b  
 JOIN trip t ON b.TripID = t.TripID  
 """  
 cursor.execute(query)  
 total\_booking\_cost = cursor.fetchone()[0]  
  
 # Calculate average profit per sale  
 average\_profit\_per\_sale = total\_equipment\_transactions\_cost / total\_sales if total\_sales > 0 else 0  
  
 # Calculate profit margin  
 query = "SELECT SUM(Cost) AS Total\_Cost FROM equipment\_transactions;"  
 cursor.execute(query)  
 total\_cost = cursor.fetchone()[0]  
  
 profit\_margin = (total\_equipment\_transactions\_cost / total\_cost) \* 100 if total\_cost > 0 else 0  
  
 # Additional query to get income sources  
 query = """  
 SELECT 'Rental', SUM(DATEDIFF(t.EndDate, t.StartDate) \* et.Cost)   
 FROM equipment\_transactions AS et  
 JOIN trip t ON et.TripID = t.TripID  
 WHERE IsRental = 1   
 UNION ALL   
 SELECT 'Purchase', SUM(Cost)   
 FROM equipment\_transactions   
 WHERE IsRental = 0   
 UNION ALL   
 SELECT 'Booking', SUM(b.Cost)   
 FROM booking b   
 JOIN trip t ON b.TripID = t.TripID;  
 """  
 cursor.execute(query)  
 income\_sources = cursor.fetchall()  
  
 # Calculate the sum of all values from the 2nd column  
 total\_sum = sum(row[1] for row in income\_sources)  
  
 # Add a third column with the percentage  
 formatted\_income\_sources = [  
 (type, "${:,.2f}".format(income), "{:,.2f}%".format(income / total\_sum \* 100))  
 for type, income in income\_sources  
 ]  
  
 # Display results  
 print("Question 1: Is the volume of equipment purchased by customers"  
 "\nsufficient to sustain the equipment sales segment of the business,"  
 "\nbut also profitable?")  
 print()  
 print("----- Outdoor Equipment Report -----")  
 print()  
 print("Total equipment sales:", total\_sales)  
 print("Total items available for sale:", total\_available\_sales)  
 print()  
 # Sales info formated  
 print(tabulate(formatted\_sales\_info, headers=["Sales Method", "Sales Total"], tablefmt="tabular"))  
 print()  
  
 # Display profit comparison  
 print("----- Profit Comparison -----")  
 print("Total take from rentals: ${:,.2f}".format(totRental))  
 print("Total take from sales: ${:,.2f}".format(totPurchase))  
 print("Total take from equipment transactions: ${:,.2f}".format(totRental + totPurchase))  
 print("Total take from booking: ${:,.2f}".format(total\_booking\_cost))  
 print("Average profit per sale: ${:,.2f}".format(average\_profit\_per\_sale))  
 print("Profit margin: {:.2f}%".format(profit\_margin))  
 print()  
  
 # Display income sources and their percentages  
 print("----- Income Sources -----")  
 print(tabulate(formatted\_income\_sources, headers=["Income Type", "Income from Type",  
 "Percentage of Total"]))  
 print()  
  
 # Display booking results  
 print("Question 2: Among the locations where trips are conducted - Africa, \nAsia, and Southern Europe -"  
 "is there any location experiencing a \ndecline in booking rates?")  
 print()  
 print(tabulate(booking\_results, headers=["Continent", "Country", "Total Bookings"]))  
 print("\n")  
  
 # Display inventory items over five years old  
 print("Question 3: Are there any items in the inventory that have been\nin stock for more than five years, "  
 "considering equipment\ndegradation over time?")  
 print()  
 print("Inventory items over five years old:", old\_inventory)  
 if old\_inventory > 0:  
 print(tabulate(old\_items, headers=["ID", "Item", "Date"]))  
 else:  
 print("No inventory items over five years old.")  
 print()  
  
except Error as e:  
 print("Error:", e)  
  
finally:  
 if cursor:  
 cursor.close()  
 if db:  
 db.close()

Results: A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated