# Case Study: Bike Store Database Analysis

### Scenario Overview:

You are working as a business analyst for a bike store, and your task is to extract meaningful insights from the corporate database. The project involves several smaller tasks aimed at building foundational SQL skills and solving business problems. The marketing team, sales department, and management rely on your analysis to drive business campaigns and improve customer engagement.

### Tasks and Scenarios:

### Scenario 1: Overview of Customers Table

The marketing team has requested customer information to send invitations for a new campaign.

1. Write a SQL query to return **all rows and columns** from the Sales.Customers table.
2. Write a query to return the following columns from Sales.Customers: contactname, address, postalcode, city, and country.

### Scenario 2: Identifying Customer Countries

The marketing team needs a list of countries where customers are located to tailor campaign strategies.

1. Write a query to return only the **country** column from the Sales.Customers table.
2. Modify the query above to return only **distinct** country names.
3. How many rows are returned by the query in **Task 1**?
4. How many rows are returned by the query in **Task 2**?
5. When do the following queries return the **same result**?

```
SELECT city, region FROM Sales.Customers;
SELECT DISTINCT city, region FROM Sales.Customers;
```

6. Is the DISTINCT clause applied to all specified columns or just the first column? Explain your answer.

## Scenario 3: Human-Friendly Column Titles and Debugging Errors

The marketing department needs reports with customer-friendly column titles.

1. Write a query that selects the contactname and contacttitle columns from Sales.Customers using the alias C for the table and prefix these columns accordingly.
2. Write a query that returns the contactname, contacttitle, and companyname columns from Sales.Customers with the following aliases:
    - **Name** (for contactname)
    - **Title** (for contacttitle)
    - **Company Name** (for companyname)
3. Write a query to select the productname from Production.Products with:
    - **Alias for the table:** P
    - **Alias for the column:** Product Name
4. **Debugging Task:**

- A developer wrote the following query:
  SELECT city country FROM Sales.Customers;
- Identify and correct the error in the query so it returns **two columns**. Explain why the original query failed.

## Scenario 4: Product Categories and Campaign Identification

The marketing team provided a mapping between product category IDs and their names for a campaign.

| Category ID | Category Name |
|---|---|
| 1 | Beverages |
| 2 | Condiments |
| 3 | Confections |
| 4 | Dairy Products |
| 5 | Grains/Cereals |
| 6 | Meat/Poultry |
| 7 | Produce |
| 8 | Seafood |

1. Write a query to display the categoryid and productname columns from the Production.Products table.
2. Modify the query to include a **CASE expression** that assigns the appropriate categoryname based on the supplied mapping. If a category ID is not found, assign the value **"Other"**.
3. Add a new column named iscampaign that returns:
   - **"Campaign Products"** for categories: Beverages, Produce, and Seafood.
   - **"Non-Campaign Products"** for all other categories.
4. You no longer need to use the manual mapping because the Production.Categories table contains the required data.
   - Write a query using an **INNER JOIN** between Production.Products and Production.Categories to retrieve the productname and categoryname.
   - Use table aliases (p for Products and c for Categories).
   - What column should you use in the ON clause of the join? Why?
   - If there is a new category in Production.Categories that has no products associated with it, will it appear in the result? Explain why or why not.

## Scenario 5: Sales Report with Order Details

The sales department wants a report showing customers who placed at least one order, along with detailed order information.

1. A developer wrote the following query:

```
SELECT custid, contactname, orderid
FROM Sales.Customers
INNER JOIN Sales.Orders
ON Customers.custid = Orders.custid;
```

   Execute the query as written and observe the error message. What is the error, and why did it occur?
2. Modify the query to run correctly, ensuring it joins Sales.Customers with both Sales.Orders and Sales.OrderDetails.