Solution for 1st case study

Step 1: Data Cleaning

To clean up the data, you can use SQL queries to correct inconsistencies and errors. Here are a few examples:

1. Update product names to correct typos:

sql

UPDATE sales\_data

SET product = 'Corrected Product Name'

WHERE product = 'Incorrect Product Name';

2. Update misspelled regions:

sql

UPDATE sales\_data

SET region = 'Corrected Region Name'

WHERE region = 'Misspelled Region Name';

Step 2: Data Aggregation

To aggregate the data and get a summary of the sales by product and region, you can use SQL's GROUP BY clause. Here's an example query:

sql

SELECT product, region, SUM(sales) AS total\_sales

FROM sales\_data

GROUP BY product, region;

Step 3: Data Analysis

To analyze the sales data and gain insights, you can use various SQL queries. Here are a few examples:

1. Identify top-selling products:

sql

SELECT product, SUM(sales) AS total\_sales

FROM sales\_data

GROUP BY product

ORDER BY total\_sales DESC

LIMIT 5;

2. Identify top-selling regions:

sql

SELECT region, SUM(sales) AS total\_sales

FROM sales\_data

GROUP BY region

ORDER BY total\_sales DESC

LIMIT 5;

3. Look for trends in the data:

sql

SELECT MONTH(sale\_date) AS month, SUM(sales) AS total\_sales

FROM sales\_data

GROUP BY month

ORDER BY month;

Step 4: Recommendations

Based on the analysis, you can make recommendations using SQL queries. Here's an example:

sql

SELECT product, region, SUM(sales) AS total\_sales

FROM sales\_data

GROUP BY product, region

HAVING total\_sales > (SELECT AVG(sales) FROM sales\_data)

ORDER BY total\_sales DESC;

This query identifies products and regions with sales above the average and can be used to recommend focusing on selling those products in the best-performing regions.

Remember to adjust the table and column names in the queries to match your actual database schema.

Solution for 2nd case study

The tasks provided in the case study involve data management and calculations that are typically better suited for spreadsheet software like Excel rather than SQL. However, if you have the data stored in a database and want to perform these tasks using SQL, here's how you can approach it:

Task 1: Creating an Inventory Sheet

To create an inventory sheet using SQL, you can create a table to store the inventory data. Here's an example query:

sql

CREATE TABLE Inventory (

ProductName VARCHAR(255),

QuantityInStock INT,

CostPrice DECIMAL(10, 2)

);

You can then insert the relevant data into the Inventory table using INSERT statements.

Task 2: Creating a Sales Sheet

Similarly, you can create a table for the sales data:

sql

CREATE TABLE Sales (

ProductName VARCHAR(255),

QuantitySold INT,

SalePrice DECIMAL(10, 2)

);

Insert the relevant sales data into the Sales table.

Task 3: Calculating Profit per Sale

To calculate the profit per sale using SQL, you can add a computed column to the Sales table:

sql

ALTER TABLE Sales

ADD Profit DECIMAL(10, 2) AS (SalePrice - CostPrice);

This assumes that you have a CostPrice column in the Inventory table that corresponds to each product.

Task 4: Total Profit Calculation

To calculate the total profit made on each product, you can use SQL to join the Inventory and Sales tables and calculate the sum of profits for each product:

sql

SELECT i.ProductName, SUM(s.Profit) AS TotalProfit

FROM Inventory i

JOIN Sales s ON i.ProductName = s.ProductName

GROUP BY i.ProductName;

Task 5: Filtering and Sorting Data

Filtering and sorting can be done using the WHERE and ORDER BY clauses in SQL queries. For example, to filter sales data by a specific product, you can use:

sql

SELECT \*

FROM Sales

WHERE ProductName = 'Product X';

To sort sales data by sale price in descending order, you can use:

sql

SELECT \*

FROM Sales

ORDER BY SalePrice DESC;

Task 6: Creating Visualizations

SQL is not typically used for creating visualizations. You may want to export the data to a spreadsheet software like Excel or use a data visualization tool like Tableau to create charts and graphs.

Task 7: Updating Inventory

To update the quantity of each product in the inventory, you can use SQL's UPDATE statement. For example, to increase the quantity of Product X by 10:

sql

UPDATE Inventory

SET QuantityInStock = QuantityInStock + 10

WHERE ProductName = 'Product X';

Task 8: Generating Reports

SQL can be used to generate reports by combining various SELECT statements and aggregations. You can create a query that summarizes the profits made on each product and the total profit made by the business. The exact query would depend on the specific report requirements.

While SQL can handle basic data management and calculations, Excel is generally a more suitable tool for the tasks described in this case study.