

Recreated SQL Project Answers for SQL Server and Oracle

Project 1: Basic Sales Reporting

1. Total sales, total quantity, and total profit for the entire year:

SQL Server:

```
SELECT  
  
    YEAR(order_date) AS order_year,  
  
    SUM(total_sales) AS total_sales,  
  
    SUM(quantity) AS total_quantity,  
  
    SUM(profit) AS total_profit  
  
FROM dbo.DWSales  
  
GROUP BY YEAR(order_date);
```

Oracle:

```
SELECT  
  
    EXTRACT(YEAR FROM order_date) AS order_year,  
  
    SUM(total_sales) AS total_sales,  
  
    SUM(quantity) AS total_quantity,  
  
    SUM(profit) AS total_profit  
  
FROM DWSales  
  
GROUP BY EXTRACT(YEAR FROM order_date);
```

2. Summary table showing total sales by month:

SQL Server:

SELECT

YEAR(order_date) AS order_year,

MONTH(order_date) AS order_month,

SUM(total_sales) AS total_sales

FROM dbo.DWSales

GROUP BY YEAR(order_date), MONTH(order_date)

ORDER BY YEAR(order_date), MONTH(order_date);

Oracle:

SELECT

EXTRACT(YEAR FROM order_date) AS order_year,

EXTRACT(MONTH FROM order_date) AS order_month,

SUM(total_sales) AS total_sales

FROM DWSales

GROUP BY EXTRACT(YEAR FROM order_date), EXTRACT(MONTH FROM order_date)

ORDER BY EXTRACT(YEAR FROM order_date), EXTRACT(MONTH FROM order_date);

3. Breakdown of total sales by product category and sub-category:

SQL Server and Oracle:

SELECT

category,

sub_category,

SUM(total_sales) AS total_sales

FROM DWSales

GROUP BY category, sub_category

ORDER BY total_sales DESC;

Project 2: Customer Segmentation

1. Customer profiles summarizing total sales and profit per customer:

SQL Server and Oracle:

SELECT

customer_id,

customer_name,

SUM(total_sales) AS total_sales,

SUM(profit) AS total_profit

FROM DWSales

GROUP BY customer_id, customer_name;

2. Group customers by market segment:

SQL Server:

SELECT

market_segment,

COUNT(DISTINCT customer_id) AS total_customers,

AVG(total_sales) AS avg_sales_per_customer,

SUM(profit) AS total_profit_per_segment

FROM dbo.DWSales

GROUP BY market_segment;

Oracle:

```
SELECT
    market_segment,
    COUNT(DISTINCT customer_id) AS total_customers,
    AVG(customer_sales) AS avg_sales_per_customer,
    SUM(profit) AS total_profit_per_segment
FROM (
    SELECT
        customer_id,
        market_segment,
        SUM(total_sales) AS customer_sales,
        SUM(profit) AS profit
    FROM DWSales
    GROUP BY customer_id, market_segment
)
GROUP BY market_segment;
```

3. Top 10 customers by total sales and their regions:

SQL Server:

```
SELECT TOP 10
    customer_id,
    customer_name,
    region,
```

```
SUM(total_sales) AS total_sales  
  
FROM dbo.DWSales  
  
GROUP BY customer_id, customer_name, region  
  
ORDER BY total_sales DESC;
```

Oracle:

```
SELECT  
  
    customer_id,  
  
    customer_name,  
  
    region,  
  
    SUM(total_sales) AS total_sales  
  
FROM DWSales  
  
GROUP BY customer_id, customer_name, region  
  
ORDER BY total_sales DESC  
  
FETCH FIRST 10 ROWS ONLY;
```

4. Identify the most popular products in each market segment:

SQL Server and Oracle (with subquery hint):

```
SELECT  
  
    market_segment,  
  
    product_name,  
  
    COUNT(order_id) AS product_count  
  
FROM DWSales  
  
GROUP BY market_segment, product_name  
  
ORDER BY product_count DESC;
```

Project 3: Sales Performance Dashboard

1. Total sales, profit, and quantity sold for the current month compared to the previous month:

SQL Server:

SELECT

YEAR(order_date) AS order_year,
MONTH(order_date) AS order_month,
SUM(total_sales) AS total_sales,
SUM(profit) AS total_profit,
SUM(quantity) AS total_quantity

FROM dbo.DWSales

WHERE order_date BETWEEN DATEADD(MONTH, -1, GETDATE()) AND GETDATE()
GROUP BY YEAR(order_date), MONTH(order_date);

Oracle:

SELECT

EXTRACT(YEAR FROM order_date) AS order_year,
EXTRACT(MONTH FROM order_date) AS order_month,
SUM(total_sales) AS total_sales,
SUM(profit) AS total_profit,
SUM(quantity) AS total_quantity

FROM DWSales

WHERE order_date BETWEEN ADD_MONTHS(SYSDATE, -1) AND SYSDATE

GROUP BY EXTRACT(YEAR FROM order_date), EXTRACT(MONTH FROM order_date);

2. KPIs for AOV, CLV, and profit margin:

SQL Server and Oracle:

- Average Order Value (AOV):

SELECT

AVG(total_sales) AS avg_order_value

FROM DWSales;

- Customer Lifetime Value (CLV):

SELECT

customer_id,

customer_name,

SUM(profit) AS customer_lifetime_value

FROM DWSales

GROUP BY customer_id, customer_name;

- Profit Margin:

SELECT

(SUM(profit) / SUM(total_sales)) AS profit_margin

FROM DWSales;

3. Track sales growth month-over-month:

SQL Server and Oracle (with CTE hint):

WITH MonthlySales AS (

SELECT

YEAR(order_date) AS order_year,

MONTH(order_date) AS order_month,

SUM(total_sales) AS total_sales

FROM DWSales

GROUP BY YEAR(order_date), MONTH(order_date)

)

SELECT

order_year,

order_month,

total_sales,

LAG(total_sales) OVER (ORDER BY order_year, order_month) AS previous_month_sales,

(total_sales - LAG(total_sales) OVER (ORDER BY order_year, order_month)) / LAG(total_sales)

OVER (ORDER BY order_year, order_month) AS month_over_month_growth

FROM MonthlySales;

... More content continues for the remaining projects.