CRM SALES OPPORTUNITIES INSIGHTS USING SQL

```sql
-- 1. Accounts table

CREATE TABLE accounts (

    account VARCHAR(150) PRIMARY KEY,

    sector VARCHAR(100),

    year_established INT,

    revenue NUMERIC(15,2),

    employees INT,

    office_location VARCHAR(100),

    subsidiary_of VARCHAR(150)  -- could reference another account

);


-- 2. Products table

CREATE TABLE products (

    product VARCHAR(100) PRIMARY KEY,

    series VARCHAR(50),

    sales_price NUMERIC(15,2)

);


-- 3. Sales Teams table

CREATE TABLE sales_teams (

    sales_agent VARCHAR(100) PRIMARY KEY,

    manager VARCHAR(100),

    regional_office VARCHAR(100)

);


-- 4. Sales Pipeline table

CREATE TABLE salespipeline (

    opportunity_id VARCHAR(20) PRIMARY KEY,  -- was text ID in CSV

    sales_agent VARCHAR(100) REFERENCES sales_teams(sales_agent),
    product VARCHAR(100) REFERENCES products(product),
    account VARCHAR(150) REFERENCES accounts(account),
    deal_stage VARCHAR(50),
    engage_date DATE,
    close_date DATE,
    close_value NUMERIC(15,2)
);

-- 5. Data Dictionary (optional metadata table)
CREATE TABLE data_dictionary (
    table_name VARCHAR(100),
    field VARCHAR(100),
    description TEXT
);
```

-- BEGINNER SQL QUESTIONS

--Q1. Find all opportunities in the salespipeline that are still open

```
SELECT opportunity_id FROM salespipeline
WHERE close_date IS NULL;
```

--Q2. Display the first 10 rows of the salespipeline.

```
SELECT *  FROM salespipeline LIMIT 10;
```

-- INTERMIDIATE SQL QUESTIONS

--Q3 Find the total deal value by each sales agent.

```
SELECT sales_agent, SUM(close_value) AS Deal_Value
FROM salespipeline
GROUP BY sales_agent;
```

-- Q4. Count how many opportunities each account has in the pipeline.

```
SELECT account,COUNT(opportunity_id)

FROM salespipeline

GROUP BY account;
```

-- Q5. Find the top 5 products by total deal value closed.

```
SELECT product, SUM(close_value) As Deal_Value

FROM salespipeline

GROUP BY product

ORDER BY Deal_Value DESC

LIMIT 5;
```

--Q6. List all opportunities that took more than 60 days to close.

```
SELECT opportunity_id FROM salespipeline

WHERE close_date - engage_date>60;
```

--Q7. Show all sales agents who are managed by “Dustin Brinkmann”.

```
SELECT sales_agent FROM sales_teams

WHERE manager = 'Dustin Brinkmann';
```

-- ADVANCE SQL QUESTIONS

-- Q8. Find the average deal value per sector

```
SELECT a.sector , AVG(close_value) AS Avg_deal_value

FROM salespipeline sp

JOIN accounts a

on sp.account = a.account

WHERE sp.close_value IS NOT NULL

GROUP BY a.sector

ORDER BY Avg_deal_value DESC;
```

-- Q9. Which regional office generated the highest total sales?

```
SELECT st.regional_office,SUM(close_value) AS total_sales

FROM salespipeline sp

JOIN sales_teams st

ON sp.sales_agent = st.sales_agent

WHERE sp.close_value IS NOT NULL

GROUP BY regional_office

ORDER BY total_sales DESC;
```

-- Q10. For each year, show how many deals were Won vs Lost vs Still Open.

```
SELECT

   EXTRACT(YEAR FROM engage_date) AS year,

   SUM(CASE WHEN deal_stage = 'Won' THEN 1 ELSE 0 END) AS won_deals,

   SUM(CASE WHEN deal_stage = 'Lost' THEN 1 ELSE 0 END) AS lost_deals,

   SUM(CASE WHEN close_date IS NULL THEN 1 ELSE 0 END) AS still_open

FROM salespipeline

GROUP BY EXTRACT(YEAR FROM engage_date)

ORDER BY year;
```

-- Q11. Find the top 3 accounts that generated the highest revenue in closed deals.

```
SELECT a.account,SUM(sp.close_value) AS Total_deal_revenue

FROM accounts a

JOIN salespipeline sp

ON a.account = sp.account

WHERE sp.close_value IS NOT NULL

GROUP BY a.account

ORDER BY Total_deal_revenue DESC

LIMIT 3;
```

-- Q12. Show the sales agent with the shortest average sales cycle (time from engage to close).

```
SELECT sales_agent, AVG(close_date - engage_date) AS avg_sales_cycle

FROM salespipeline

WHERE close_date IS NOT NULL

GROUP BY sales_agent

ORDER BY avg_sales_cycle ASC

LIMIT 1;
```

-- Analytical Questions

--Q13. Show the product series with the highest average deal value.

```
SELECT p.series,AVG(sp.close_value) AS deal_value

FROM salespipeline sp

JOIN products p

ON sp.product = p.product

WHERE sp.close_value IS NOT NULL

GROUP BY p.series

ORDER BY deal_value DESC

LIMIT 1;
```

-- Q14. Rank sales agents by total revenue

```
SELECT

   sales_agent,

   SUM(sp.close_value) AS total_revenue,

   RANK() OVER (ORDER BY SUM(sp.close_value) DESC) AS revenue_rank

FROM salespipeline sp

JOIN accounts a

   ON sp.account = a.account

WHERE sp.close_value IS NOT NULL

GROUP BY sales_agent

ORDER BY total_revenue DESC;
```

```
-- Q15. Calculate the win rate per sales agent
SELECT
   sales_agent,
   COUNT(*) AS total_deals,
   SUM(CASE WHEN deal_stage = 'Won' THEN 1 ELSE 0 END) AS won_deals,
   ROUND(
      (SUM(CASE WHEN deal_stage = 'Won' THEN 1 ELSE 0 END)::numeric
       / COUNT(*)) * 100, 2
   ) AS win_rate_percentage
FROM salespipeline
GROUP BY sales_agent
ORDER BY win_rate_percentage DESC;
```

```
-- Q16. Find accounts that are subsidiaries of other accounts and their total closed deal value.
SELECT a.account, a.subsidiary_of AS parent_account,
SUM(sp.close_value) AS total_closed_value
FROM accounts a
JOIN salespipeline sp
ON a.account = sp.account
WHERE a.subsidiary_of IS NOT NULL
AND sp.close_value IS NOT NULL
GROUP BY a.account,a.subsidiary_of
ORDER BY total_closed_value DESC;
```

```
-- Q17. Create a monthly sales trend report
SELECT
   DATE_TRUNC('month', close_date) AS month,
   COUNT(*) AS total_deals,
   SUM(close_value) AS total_revenue
FROM salespipeline
```

```
WHERE close_date IS NOT NULL   -- only closed deals

GROUP BY DATE_TRUNC('month', close_date)

ORDER BY month;
```