Setting Up Linked Servers: First, we configured a linked server in SQL Server to establish a connection with the PostgreSQL database. This setup was done with the necessary permissions to allow indirect access.

SQL SERVER

```
EXEC sp_addlinkedserver
   @server = 'POSTGRESQL_SERVER',
   @srvproduct = 'PostgreSQL',
   @provider = 'MSDASQL',
   @datasrc = 'PostgreSQL_DSN';

EXEC sp_addlinkedsrvlogin
   @rmtsrvname = 'POSTGRESQL_SERVER',
   @useself = 'false',
   @rmtuser = 'postgres_user',
   @rmtpassword = 'postgres_password';
```

Writing the OpenQuery: Using OpenQuery, we queried the PostgreSQL database from SQL Server to fetch the necessary sales data.

SQL SERVER

```
-- Query the PostgreSQL database using OpenQuery

SELECT *
FROM OPENQUERY(POSTGRESQL_SERVER,
        'SELECT sales_id, product_name,
         sales_amount, sales_date
 FROM postgres_sales_table');
```

Combining Data: We then combined the data from SQL Server with the data retrieved from the PostgreSQL database to create a comprehensive sales report.

SQL SERVER

```
SELECT
    s.sales_id,
    s.product_name,
    s.sales_amount,
    p.sales_amount AS postgres_sales_amount,
    p.sales_date
FROM
    sql_server_sales_table s
JOIN
    OPENQUERY(POSTGRESQL_SERVER, 'SELECT
sales_id, product_name, sales_amount, sales_date
FROM postgres_sales_table') p
ON
    s.sales_id = p.sales_id;
```