--- Write a query to identify the channel that received maximum sales (in dollars) in fiscalyear 2000.

SELECT c.channel\_desc, ROUND(SUM(s.AMOUNT\_SOLD),0) as "Total Sales"

FROM SH.SAlES s JOIN SH.TIMES t ON (s.time\_id = t.time\_id)

    JOIN SH.CHANNELS c ON (c.channel\_id = s.channel\_id)

    WHERE t.fiscal\_year = 2000

    GROUP BY c.channel\_desc

    ORDER BY ROUND(SUM(s.AMOUNT\_SOLD),0) DESC

    FETCH NEXT 1 ROW ONLY;

SELECT c.channel\_desc

FROM SH.SAlES s JOIN SH.TIMES t ON (s.time\_id = t.time\_id)

    JOIN SH.CHANNELS c ON (c.channel\_id = s.channel\_id)

    WHERE t.fiscal\_year = 2000

    GROUP BY c.channel\_desc

    ORDER BY ROUND(SUM(s.AMOUNT\_SOLD),0) DESC

    FETCH NEXT 1 ROW ONLY;

--- Display  the  customer  Id,  first  name,  last  name  and  year  of  birth  of  customers  born

--- between years 1960 and 1970, who live in the city of Barcelona, and belong to the group

--- of customers with the maximum income level.

SELECT c1.CUST\_ID, c1.CUST\_FIRST\_NAME, c1.Cust\_last\_name, c1.cust\_year\_of\_birth

FROM SH.customers c1

WHERE (c1.CUST\_YEAR\_OF\_BIRTH between 1960 and 1970) AND

(c1.cust\_city = 'Barcelona') AND

cust\_income\_level = (

    SELECT MAX(c2.cust\_income\_level)

FROM SH.customers c2

);

--- Return the number of customers that had made at least one purchase with the company.

    --- a. Try executing this query using a single SELECT loop.

    --- b. Try executing this query using the EXISTS function.

SELECT (COUNT(DISTINCT(c.CUST\_ID)))

FROM sh.customers c JOIN sh.sales s ON (c.cust\_id = s.cust\_id);

--- or

SELECT COUNT(DISTINCT(CUST\_ID)) as "Customers that have made purchases"

FROM sh.sales;

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE EXISTS

    (

        SELECT s.cust\_id

        FROM sh.sales s

        where s.cust\_id = c.CUST\_ID

    );

--- Find the number of customers who have not made a single purchase with the company

    --- a. Try executing this query using a single SELECT loop.

    --- b. Try executing this query using the NOT EXISTS function.

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE NOT EXISTS

    (

        SELECT s.cust\_id

        FROM sh.sales s

        where s.cust\_id = c.CUST\_ID

    );

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.CUSTOMERS c

WHERE c.cust\_id NOT IN

    (

        SELECT s.cust\_id

        FROM sh.sales s

    );

--- or

SELECT COUNT(c.CUST\_ID)

FROM sh.sales s RIGHT OUTER JOIN sh.CUSTOMERS c ON(s.cust\_id = c.CUST\_ID)

WHERE s.QUANTITY\_SOLD IS NULL;

--- Which  product  has  the  highest  average  promotion  cost.  Display  the  product  name,

--- product category, and its average promotion cost rounded to two decimals.

SELECT pr.PROD\_NAME, pr.PROD\_CATEGORY, AVG(prom.promo\_cost)

FROM sh.PRODUCTS pr JOIN sh.sales s ON (pr.prod\_id = s.PROD\_ID)

                    JOIN sh.promotions prom ON (prom.promo\_id = s.promo\_id)

GROUP BY pr.PROD\_NAME, pr.PROD\_CATEGORY

ORDER BY AVG(prom.promo\_cost) DESC

FETCH NEXT 1 ROWS ONLY;