**1. Display in descending order of seniority the male employees whose net salary (salary + commission) is greater than or equal to 8000.**

**--The resulting table should include the following columns: Employee Number, First Name and Last Name (using LPAD or RPAD for formatting),**

**--Age, and Seniority.**

**--employees order by descending order**

**--filter based on salary >= 8000**

**--select from**

**--employee number**

**--first & last name**

**--age**

**--seniority**

SELECT

employee\_number, RPAD(first\_name, 15, '') AS fname,

RPAD(last\_name, 15, '') AS l\_name,

DATE\_PART('Year', AGE(birth\_date)) AS Age,

salary + commission AS net\_salary,

position

FROM employees

WHERE salary >= 8000

ORDER BY CASE position

WHEN 'Vice President' THEN 1

WHEN 'Sales Manager' THEN 2

WHEN 'Sales Assistant' THEN 3

WHEN 'Sales Representative' THEN 4

END;

2. **Display products that meet the following criteria: (C1) quantity is packaged in bottle(s), (C2) the third character in the product name is 't' or 'T', (C3) supplied by suppliers 1, 2, or 3, (C4) unit price ranges between 70 and 200, and (C5) units ordered are specified (not null). The resulting table should include the following columns: product number, product name, supplier number, units ordered, and unit price.**

**--Display products that meet the following criteria:**

**--(C1) quantity is packaged in bottle(s),**

**--(C2) the third character in the product name is 't' or 'T',**

**--(C3) supplied by suppliers 1, 2, or 3,**

**--(C4) unit price ranges between 70 and 200,**

**--(C5) units ordered are specified (not null).**

**--The resulting table should include the following columns:**

**-- product number,**

**-- product name,**

**--supplier number,**

**--units ordered, and**

**--unit price.**

SELECT

quantity,

product\_name,

supplier\_number,

unit\_price,

units\_on\_order

FROM products

WHERE quantity LIKE '%bottle(s)%'

AND (SUBSTRING(PRODUCT\_NAME, 3, 1) = 't' OR SUBSTRING(PRODUCT\_NAME, 3, 1) = 'T')

AND supplier\_number IN (1,2,3)

AND unit\_price BETWEEN 70 AND 200

AND units\_on\_order IS NOT NULL;

3. **Display customers who reside in the same region as supplier 1, meaning they share the same country, city, and the last three digits of the postal code. The query should utilize a single subquery. The resulting table should include all columns from the customer table.**

**--Display customers who reside in the same region as supplier 1,**

**--meaning they share the same country, city,**

**--and the last three digits of the postal code.**

**--The query should utilize a single subquery.**

**--The resulting table should include all columns from the customer table.**

SELECT \*

FROM CUSTOMERS

WHERE (COUNTRY, CITY, SUBSTRING(POSTAL\_CODE FROM LENGTH(POSTAL\_CODE)-2 FOR 3)) =

(SELECT COUNTRY, CITY, SUBSTRING(POSTAL\_CODE FROM LENGTH(POSTAL\_CODE)-2 FOR 3)

FROM SUPPLIERS

WHERE SUPPLIER\_NUMBER = 1);

4. **For each order number between 10998 and 11003, do the following:   
-Display the new discount rate,**

**which should be 0% if the total order amount before discount (unit price \* quantity) is between 0 and 2000,**

**5% if between 2001 and 10000,**

**10% if between 10001 and 40000,**

**15% if between 40001 and 80000,**

**and 20% otherwise.  
-Display the message "apply old discount rate" if the order number is between 10000 and 10999,**

**and "apply new discount rate" otherwise.**

**The resulting table should display the columns:**

**order number,**

**new discount rate,**

**and discount rate application note.**

WITH NEW\_DISCOUNT\_RATE AS (

SELECT

ORDER\_NUMBER,

-- Use CASE statement instead of IF in standard SQL

CASE

WHEN (UNIT\_PRICE \* QUANTITY) BETWEEN 0 AND 2000 THEN '0%'

WHEN (UNIT\_PRICE \* QUANTITY) BETWEEN 2001 AND 10000 THEN '5%'

WHEN (UNIT\_PRICE \* QUANTITY) BETWEEN 10001 AND 40000 THEN '10%'

WHEN (UNIT\_PRICE \* QUANTITY) BETWEEN 40001 AND 80000 THEN '15%'

ELSE '20%'

END AS NEW\_DISCOUNT\_RATE,

-- Use CASE statement instead of IF in standard SQL

CASE

WHEN ORDER\_NUMBER BETWEEN 10000 AND 10999 THEN 'apply old discount rate'

ELSE 'apply new discount rate'

END AS DISCOUNT\_RATE\_APPLICATION\_NOTE

FROM

ORDER\_DETAILS

WHERE

ORDER\_NUMBER BETWEEN 10998 AND 11003

)

SELECT \* FROM NEW\_DISCOUNT\_RATE;

5. **Display suppliers of beverage products.**

**The resulting table should display the columns:**

**supplier number,**

**company, address,**

**and phone number.**

SELECT

SUPPLIER\_NUMBER,

COMPANY,

ADDRESS,

PHONE

FROM

SUPPLIERS

WHERE

SUPPLIER\_NUMBER IN (

SELECT SUPPLIER\_NUMBER

FROM PRODUCTS

WHERE CATEGORY\_CODE = 1

);

SELECT

s.SUPPLIER\_NUMBER,

s.COMPANY, s.ADDRESS,

s.PHONE

FROM

SUPPLIERS s

JOIN

PRODUCTS p ON s.SUPPLIER\_NUMBER = p.SUPPLIER\_NUMBER

WHERE

p.CATEGORY\_CODE = (SELECT CATEGORY\_CODE FROM CATEGORIES WHERE CATEGORY\_NAME = 'Beverages');

6**. Display customers from Berlin who have ordered at most 1 (0 or 1) dessert product**

**. The resulting table should display the column:**

**customer code**

SELECT DISTINCT c.customer\_code FROM customers c

JOIN orders o ON o.customer\_code = c.customer\_code

JOIN order\_details od ON od.order\_number = o.order\_number

JOIN products p ON p.product\_ref = od.product\_ref

JOIN categories ct ON p.category\_code = p.category\_code

WHERE c.city = 'Berlin'

AND ct.category\_code = 3;

**7.Display customers who reside in France**

**and the total amount of orders they placed every Monday in April 1998 (considering customers who haven't placed any orders yet).**

**The resulting table should display the columns:**

**customer number,**

**company name,**

**phone number,**

**total amount,**

**and country.**

SELECT

c.customer\_code,

c.company,

c.phone,

c.country,

COUNT(o.order\_number) AS total\_orders

FROM customers c

LEFT JOIN orders o ON c.customer\_code = o.customer\_code

AND o.order\_date BETWEEN '1998-04-01' AND '1998-04-30'

AND EXTRACT(DOW FROM o.order\_date) = 1 – retrieves day of the week

WHERE c.country = 'France'

GROUP BY c.customer\_code, c.company, c.phone, c.country;

**8.Display customers who have ordered all products.**

**The resulting table should display the columns:**

**customer code,**

**company name,**

**and telephone number.**

SELECT

c.customer\_code,

c.company,

c.phone

FROM customers c

JOIN orders o ON c.customer\_code = o.customer\_code

JOIN order\_details od ON o.order\_number = od.order\_number

GROUP BY c.customer\_code, c.company, c.phone

HAVING COUNT(DISTINCT od.product\_ref) = (SELECT COUNT(DISTINCT product\_ref) FROM products);

**9. Display for each customer from France the number of orders they have placed.**

**The resulting table should display the columns:**

**customer code**

**and number of orders.**

SELECT c.customer\_code, COUNT(o.order\_number) AS number\_of\_orders

FROM customers c

JOIN orders o ON c.customer\_code = o.customer\_code

WHERE c.country = 'France'

GROUP BY c.customer\_code;

**10. Display the number of orders placed in 1996,**

**the number of orders placed in 1997,**

**and the difference between these two numbers.**

**The resulting table should display the columns:**

**orders in 1996,**

**orders in 1997,**

**and Difference.**