SQL Exercise 1 – Solution

SELECT QUERY:

1.

SELECT PRODUCT\_ID, PRODUCT\_NAME, UNIT\_PRICE

FROM CO.PRODUCTS

ORDER BY PRODUCT\_NAME ASC

2.

SELECT FIRST\_NAME||' '||LAST\_NAME FULL\_EMPLOYEE\_NAME, PHONE\_NUMBER, DATEDIFF(YEAR, HIRE\_DATE, GETDATE()) YEARS\_WORKING\_AT\_THE\_COMPANY

FROM HR.EMPLOYEES

**BUT Live SQL doesn’t recognize the DATEDIFF, GETDATE Functions so:**

SELECT FIRST\_NAME||' '||LAST\_NAME FULL\_EMPLOYEE\_NAME, PHONE\_NUMBER, 2023 - EXTRACT(YEAR FROM HIRE\_DATE) YEARS\_WORKING\_AT\_THE\_COMPANY

FROM HR.EMPLOYEES

3.

SELECT FIRST\_NAME, EMAIL, SALARY

FROM HR.EMPLOYEES

WHERE EXTRACT(YEAR FROM HIRE\_DATE) >= 2005

ORDER BY EMAIL ASC

4.

SELECT PRODUCT\_NAME

FROM CO.PRODUCTS

WHERE UNIT\_PRICE - (

SELECT MIN(UNIT\_PRICE)

FROM CO.PRODUCTS

) > 100

ORDER BY PRODUCT\_NAME

JOIN TABLES:

1.

SELECT CUSTOMERS.CUSTOMER\_ID CUSTOMER\_ID, CUSTOMERS.CUST\_FIRST\_NAME||' '||CUSTOMERS.CUST\_LAST\_NAME CUSTOMER\_FULL\_NAME, ORDERS.ORDER\_ID ORDER\_ID

FROM OE.CUSTOMERS CUSTOMERS, OE.ORDERS ORDERS

WHERE (

CUSTOMERS.CUSTOMER\_ID = ORDERS.CUSTOMER\_ID

AND

EXTRACT (YEAR FROM ORDERS.ORDER\_DATE) = 2007

AND

CUSTOMERS.CREDIT\_LIMIT >= 200

AND

CUSTOMERS.CREDIT\_LIMIT <= 700

)

ORDER BY CUSTOMER\_FULL\_NAME, ORDER\_ID

2.

SELECT PRUDOCTS.PRODUCT\_NAME, PRUDOCTS.LIST\_PRICE, CATEGORIES.CATEGORY\_NAME

FROM OE.PRODUCT\_INFORMATION PRUDOCTS, OE.CATEGORIES\_TAB CATEGORIES

WHERE (

CATEGORIES.CATEGORY\_ID = PRUDOCTS.CATEGORY\_ID

)

ORDER BY CATEGORIES.CATEGORY\_NAME ASC

3.

SELECT CUST.CUST\_FIRST\_NAME||' '||CUST.CUST\_LAST\_NAME AS CUSTOMER\_FULL\_NAME, PROD.PRODUCT\_NAME

FROM (

(

(

OE.CUSTOMERS CUST

INNER JOIN OE.ORDERS ORD

ON ORD.CUSTOMER\_ID = CUST.CUSTOMER\_ID

)

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEMS ON ORD\_ITEMS.ORDER\_ID = ORD.ORDER\_ID

)

INNER JOIN OE.PRODUCT\_INFORMATION PROD ON PROD.PRODUCT\_ID = ORD\_ITEMS.PRODUCT\_ID

)

WHERE (

EXTRACT(YEAR from ORD.ORDER\_DATE) = 2008

AND

ORD\_ITEMS.UNIT\_PRICE > 10

)

4.

SELECT ORD.ORDER\_ID AS ORDER\_ID, PROD\_INFO.PRODUCT\_NAME AS PRODUCT\_NAME, ORD\_ITEMS.QUANTITY \* ORD\_ITEMS.UNIT\_PRICE AS TOTAL\_ITEM\_PRICE

FROM (

(

OE.ORDERS ORD

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEMS

ON ORD\_ITEMS.ORDER\_ID = ORD.ORDER\_ID

)

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON PROD\_INFO.PRODUCT\_ID = ORD\_ITEMS.PRODUCT\_ID

)

WHERE (

EXTRACT (YEAR from ORD.ORDER\_DATE) = 2007

)

5.

SELECT EMPLOYEE.FIRST\_NAME || ' ' || EMPLOYEE.LAST\_NAME AS EMPLOYEE\_NAME, MANAGER.FIRST\_NAME || ' ' || MANAGER.LAST\_NAME AS MANAGER\_NAME

FROM (

HR.EMPLOYEES EMPLOYEE

INNER JOIN HR.EMPLOYEES MANAGER ON EMPLOYEE.MANAGER\_ID = MANAGER.EMPLOYEE\_ID

)

Aggregation Functions:

1.

SELECT AVG(2023 - (EXTRACT (year from EMPLOYEE.HIRE\_DATE))) AS AVERAGE\_SENIORITY

FROM HR.EMPLOYEES EMPLOYEE

2.

SELECT COUNTRY, NUM\_CUSTOMERS

FROM (

SELECT CUST.NLS\_TERRITORY AS COUNTRY, COUNT(CUST.CUSTOMER\_ID) AS NUM\_CUSTOMERS

FROM OE.CUSTOMERS CUST

GROUP BY CUST.NLS\_TERRITORY

)

WHERE NUM\_CUSTOMERS > 1

3.

SELECT PROD\_INFO.SUPPLIER\_ID AS SUPPLIER\_ID, COUNT(PROD\_INFO.PRODUCT\_ID) AS NUM\_PRODUCTS

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

GROUP BY PROD\_INFO.SUPPLIER\_ID

4.

SELECT PROD\_QUNANTITY\_ORDERED.PRODUCT\_ID, PROD\_QUNANTITY\_ORDERED.NUM\_UNITS\_ORDERED, PROD\_INFO.PRODUCT\_NAME

FROM (

SELECT PROD\_INFO.PRODUCT\_ID AS PRODUCT\_ID, SUM(ORD\_ITEM.QUANTITY) AS NUM\_UNITS\_ORDERED

FROM (

(

OE.ORDERS ORD

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEM

ON ORD\_ITEM.ORDER\_ID = ORD.ORDER\_ID

)

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON PROD\_INFO.PRODUCT\_ID = ORD\_ITEM.PRODUCT\_ID

)

GROUP BY PROD\_INFO.PRODUCT\_ID

) PROD\_QUNANTITY\_ORDERED

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON PROD\_INFO.PRODUCT\_ID = PROD\_QUNANTITY\_ORDERED.PRODUCT\_ID

Or (without sub-query):

SELECT PROD\_INFO.PRODUCT\_ID AS PRODUCT\_ID, SUM(ORD\_ITEM.QUANTITY) AS NUM\_UNITS\_ORDERED, MIN(PROD\_INFO.PRODUCT\_NAME) AS PROD\_NAME

FROM (

(

OE.ORDERS ORD

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEM

ON ORD\_ITEM.ORDER\_ID = ORD.ORDER\_ID

)

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON PROD\_INFO.PRODUCT\_ID = ORD\_ITEM.PRODUCT\_ID

)

GROUP BY PROD\_INFO.PRODUCT\_ID

5.

SELECT EXTRACT(YEAR from ORD.ORDER\_DATE) AS YEAR, COUNT(ORD.ORDER\_ID) AS NUM\_ORDERS, MIN(ORD.ORDER\_DATE) AS MIN\_ORDER\_DATE, MAX(ORD.ORDER\_DATE) AS MAX\_ORDER\_DATE

FROM OE.ORDERS ORD

GROUP BY EXTRACT(YEAR from ORD.ORDER\_DATE)

6.

SELECT TO\_CHAR(ORD.ORDER\_DATE, 'DAY') AS DAT\_OF\_WEEK, COUNT(ORD.ORDER\_ID) AS NUM\_ORDERS, SUM(ORD.ORDER\_TOTAL) AS ORDERS\_TOTAL\_SUM

FROM OE.ORDERS ORD

GROUP BY TO\_CHAR(ORD.ORDER\_DATE, 'DAY')

7.

SELECT EMPLOYEE.EMPLOYEE\_ID AS EMPLOYEE\_ID, EMPLOYEE.FIRST\_NAME||' '||EMPLOYEE.LAST\_NAME AS EMPLOYEE\_NAME, FILTERED\_SALES\_REP\_SUM.TOTAL\_ORDERS\_SUM AS TOTAL\_ORDERS\_SUM

FROM (

SELECT \*

FROM (

SELECT ORD.SALES\_REP\_ID, SUM(ORD.ORDER\_TOTAL) AS TOTAL\_ORDERS\_SUM

FROM OE.ORDERS ORD

WHERE EXTRACT(year FROM ORD.ORDER\_DATE) = 2007

GROUP BY ORD.SALES\_REP\_ID

) SALES\_REP\_SUM

WHERE SALES\_REP\_SUM.TOTAL\_ORDERS\_SUM > 20000

) FILTERED\_SALES\_REP\_SUM INNER JOIN HR.EMPLOYEES EMPLOYEE

ON FILTERED\_SALES\_REP\_SUM.SALES\_REP\_ID = EMPLOYEE.EMPLOYEE\_ID

8.

SELECT \*

FROM (

SELECT CUSTOMER\_ID, SUM(ITEM\_COST) / 1000 AS ORACLE\_POINTS

FROM (

SELECT CUST.CUSTOMER\_ID AS CUSTOMER\_ID, ORD.ORDER\_ID AS ORDER\_ID, CUST.CUST\_FIRST\_NAME||' '||CUST.CUST\_LAST\_NAME AS CUSTOMER\_NAME, ORD\_ITEM.UNIT\_PRICE \* ORD\_ITEM.QUANTITY AS ITEM\_COST

FROM (

(

OE.CUSTOMERS CUST

INNER JOIN OE.ORDERS ORD

ON CUST.CUSTOMER\_ID = ORD.CUSTOMER\_ID

)

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEM

ON ORD\_ITEM.ORDER\_ID = ORD.ORDER\_ID

)

)

GROUP BY CUSTOMER\_ID

) CUSTOMER\_POINTS

INNER JOIN OE.CUSTOMERS CUST2

ON CUSTOMER\_POINTS.CUSTOMER\_ID = CUST2.CUSTOMER\_ID

Sub Queries:

1.

SELECT PROD\_INFO.PRODUCT\_NAME

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

WHERE PROD\_INFO.LIST\_PRICE = (

SELECT MAX(LIST\_PRICE) FROM OE.PRODUCT\_INFORMATION

)

2.

SELECT PROD\_INFO.PRODUCT\_NAME

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

WHERE PROD\_INFO.LIST\_PRICE = (

SELECT MAX(LIST\_PRICE) FROM (

(

SELECT \*

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

WHERE PROD\_INFO.PRODUCT\_ID NOT IN (

SELECT PROD\_INFO.PRODUCT\_ID

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

INNER JOIN OE.ORDER\_ITEMS ORD\_ITEM

ON PROD\_INFO.PRODUCT\_ID = ORD\_ITEM.PRODUCT\_ID

)

)

)

)

3.

SELECT \*

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

WHERE PROD\_INFO.LIST\_PRICE > (

SELECT AVG(LIST\_PRICE) FROM OE.PRODUCT\_INFORMATION

)

4.

At first we create a view:

CREATE OR REPLACE VIEW PRODUCT\_SALE\_REP\_SENIORITY AS

SELECT SYSDATE - HIRE\_DATE AS SALE\_REP\_SENIORITY, PRODUCT\_ID

FROM (

SELECT \* FROM

OE.ORDERS ORD

INNER JOIN HR.EMPLOYEES EMPLOYEE

ON ORD.SALES\_REP\_ID = EMPLOYEE.EMPLOYEE\_ID

INNER JOIN OE.ORDER\_ITEMS ORDER\_ITEM

ON ORD.ORDER\_ID = ORDER\_ITEM.ORDER\_ID

)

Then perform a query on it:

SELECT \*

FROM (

SELECT PRODUCT\_ID

FROM PRODUCT\_SALE\_REP\_SENIORITY

WHERE (

SALE\_REP\_SENIORITY = (

SELECT MAX(SALE\_REP\_SENIORITY) FROM PRODUCT\_SALE\_REP\_SENIORITY

)

)

) FILTERED\_PRODUCT\_ID

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON FILTERED\_PRODUCT\_ID.PRODUCT\_ID = PROD\_INFO.PRODUCT\_ID

5.

SELECT \*

FROM (

SELECT DISTINCT PRODUCT\_ID

FROM OE.ORDER\_ITEMS

WHERE ORDER\_ID IN (

SELECT ORDER\_ID

FROM OE.ORDER\_ITEMS

WHERE UNIT\_PRICE = (

SELECT MAX(UNIT\_PRICE) FROM OE.ORDER\_ITEMS

)

)

) FILTERED\_PROD\_ID

INNER JOIN OE.PRODUCT\_INFORMATION PROD\_INFO

ON FILTERED\_PROD\_ID.PRODUCT\_ID = PROD\_INFO.PRODUCT\_ID

Analytic Functions:

1.

SELECT \*

FROM (

SELECT ROW\_NUMBER() OVER (PARTITION BY CATEGORY\_ID Order by LIST\_PRICE DESC) AS RANK\_IN\_CATEGORY , PROD\_INFO.\*

FROM OE.PRODUCT\_INFORMATION PROD\_INFO

WHERE LIST\_PRICE IS NOT NULL -- WE DO NOT WANT TO COMPATE PRICES TO THINGS THAT ARE NULL

)

WHERE RANK\_IN\_CATEGORY <= 3

2.

SELECT

year\_total.\*,

year\_total - LAG(year\_total) OVER (ORDER BY year) AS LAST\_YEAR\_DIFF

FROM (

SELECT

EXTRACT(YEAR FROM order\_date) AS year,

SUM(order\_total) AS year\_total

FROM OE.ORDERS ORD

GROUP BY EXTRACT(YEAR FROM order\_date)

) year\_total

ORDER BY year

3.

SELECT

YEAR,

MONTH,

MONTH\_TOTAL,

SUM (month\_total) OVER (PARTITION BY year ORDER BY month) AS RUNNING\_SUM\_FROM\_YEAR\_BEGGINING

FROM

(

SELECT

MIN(EXTRACT(YEAR FROM order\_date)) AS year,

EXTRACT(MONTH FROM order\_date) AS month,

SUM(order\_total) AS month\_total

FROM OE.ORDERS ORD

GROUP BY EXTRACT(MONTH FROM order\_date)

)

Views:

**The following view should be created for this section:**

CREATE VIEW yearly\_best\_seller\_quantity AS

SELECT

year,

sales\_rep\_id as best\_seller\_id,

quantity\_sold

FROM (

SELECT

year,

sales\_rep\_id,

quantity\_sold,

ROW\_NUMBER() OVER (PARTITION BY year ORDER BY quantity\_sold DESC) AS yearly\_rank

FROM (

SELECT

EXTRACT(YEAR FROM order\_date) AS YEAR,

sales\_rep\_id,

SUM(quantity) AS quantity\_sold

FROM (

SELECT

ord.order\_date,

ord.sales\_rep\_id,

ord\_item.\*

FROM oe.orders ord

INNER JOIN oe.order\_items ord\_item

ON (

ord.sales\_rep\_id IS NOT NULL

AND

ord.order\_id = ord\_item.order\_id

)

)

GROUP BY sales\_rep\_id, EXTRACT(YEAR FROM order\_date)

ORDER BY year ASC, sales\_rep\_id ASC

)

)

WHERE yearly\_rank = 1

ORDER BY year

1.

CREATE VIEW best\_yearly\_seller\_sold\_quantity AS

SELECT

year,

quantity\_sold

FROM yearly\_best\_seller\_quantity

2.

CREATE VIEW best\_yearly\_seller AS

SELECT

yearly\_best\_seller\_quantity.year,

employee.\*

FROM yearly\_best\_seller\_quantity

INNER JOIN hr.employees employee

ON yearly\_best\_seller\_quantity.best\_seller\_id = employee.employee\_id

3.

CREATE VIEW product\_times\_ordered AS

SELECT

product\_id,

sum(quantity) as times\_ordered

FROM oe.order\_items ord\_item

GROUP BY product\_id

CREATE VIEW most\_ordered\_product\_id AS

SELECT

product\_id

FROM product\_times\_ordered

WHERE (

times\_ordered = (

SELECT MAX(times\_ordered) FROM product\_times\_ordered

)

)

4.

CREATE VIEW product\_income AS

SELECT

product\_id,

SUM(unit\_price \* quantity) AS income

FROM oe.order\_items

GROUP BY product\_id

CREATE VIEW most\_income\_product AS

SELECT

product\_id

FROM product\_income

WHERE income = (

SELECT MAX(income) FROM product\_income

)