# Individual Assignment 5

DB 설계 및 디자인 2, 2023학년도 2학기

* 제출시한: 12월 12일 (월요일), 11:59 PM, (LMS를 통해 제출).
* 제출하는 파일명은 “숙제5\_자기이름.docx”와 같이 자신의 이름을 마지막에 붙이시오.
* 질문 있을 시 ijkim@mju.ac.kr으로 질문하시오.

Weights (비중): 5% of total grades.

Write a SQL query for each of the following data retrieval requirements based on the database used in class. 각 문제에 결과테이블을 제공하였으니, SQL문의 결과가 동일하게 작성하시오.

1. List ID, Name, and Price for all products with price greater than the average product price. (6 points) – (Hint: subquery can be used with the operator ‘>’, not necessarily just with ‘IN’)

PRODUCT\_ID PRODUCT\_NAME PRODUCT\_PRICE

-------------------- -------------------------- -------------

101 Dell E5300 Laptop 489.98

102 Apple Laptop 988.72

Answer:

select product\_id, product\_name, product\_price

from product

where product\_price > (select AVG(product\_price) from product);

1. For each product, list its ID and total quantity ordered. Products are listed in ascending order of product\_id. (6 points)

PRODUCT\_ID TOTAL\_QUANTITY

-------------------- -------------

101 2

102 4

103 11

104 8

105 2

106 2

107 5

108 15

Answer:

select product\_id, SUM(quantity)As total\_quantity

from orderline

group by product\_id

order by product\_id ASC;

1. For each product, list its ID and total quantity ordered. Products are listed in ascending order of total quantity ordered. (6 points)

PRODUCT\_ID TOTAL\_QUANTITY

---------- --------------

105 2

101 2

106 2

102 4

107 5

104 8

103 11

108 15

Answer:

select product\_id, SUM(quantity)As total\_quantity

from orderline

group by product\_id

order by total\_quantity ASC;

1. For each product, list its ID, Name and total quantity ordered. Products are listed in ascending order of product\_id. (6 points)

PRODUCT\_ID PRODUCT\_NAME Total\_ORDER\_QUANTITY

---------- -------------------------------------------------- -----------------------

101 Dell E5300 Laptop 2

102 Apple Laptop 4

103 Printer 11

104 Desk 8

105 Office Chair 2

106 Stapler 2

107 Index Divider 5

108 Shredder 15

Answer:

select product\_id, product\_name, SUM(quantity) AS total\_ORDER\_Quantity

from product NATURAL JOIN orderline

group by product\_id, product\_name

Order by product\_id ASC;

1. List name for all customers, who have placed order(s) after 23-OCT-2008. Each customer name appears exactly once. Customer names are sorted in ascending alphabetical order. Use equal join for this query. (6 points)

CUST\_NAME

-------------------------

Bill Doll

Jennifer English

John Doe

Josh Turburn

Joyce French

Mary Lee

Answer:

select distinct cust\_name

from customer,ordertable

where order\_date>'2008/10/23' AND customer.cust\_id = ordertable.cust\_id

order by cust\_name ASC;

1. Implement query 5 using IN and subquery. (6 points)

CUST\_NAME

-------------------------

Bill Doll

Jennifer English

John Doe

Josh Turburn

Joyce French

Mary Lee

Answer:

Select distinct cust\_name

from customer

where cust\_id IN (select cust\_id from ordertable where order\_date>'2008/10/23')

order by cust\_name ASC;

1. For each city, list number of customers from the city, who have placed order(s). Cities are listed in ascending alphabetical order. Use equal join for this query. (6 points) – (Hint: make sure your query does not count the same customer more than once.)

CITY NUMBER\_OF\_CUSTOMER

-------------------- -------------------------------

SF 1

SLC 6

Tucson 2

Answer:

select city, count( distinct ordertable.cust\_id) as number\_of\_customer from customer, ordertable where customer.cust\_id = ordertable.cust\_id group by city order by city ASC;

1. Implement query 7 using natural join. (6 points)

CITY NUMBER\_OF\_CUSTOMER

-------------------- ----------------------

SF 1

SLC 6

Tucson 2

Answer:

select city, count( distinct ordertable.cust\_id) AS number\_of\_customer from customer, ordertable where customer.cust\_id = ordertable.cust\_id group by city order by city ASC;

1. Implement query 7 using IN and subquery. (6 points) – (Hint: Do we have to take care of printing out each customer once? Think about it and try it.)

CITY NUMBER\_OF\_CUSTOMER

-------------------- --------------

SF 1

SLC 6

Tucson 2

Answer:

select city, count( distinct customer.cust\_id) as number\_of\_customer

from customer, ordertable

where customer.cust\_id IN (select cust\_id from ordertable)

group by city order by city ASC;

1. List ID for all products, which have NOT been ordered since 28-OCT-2008. Sort your results by product\_id in ascending order. Use Minus for this query. (6 points)

PRODUCT\_ID

-------------------

101

102

105

106

Answer:

select product\_id from orderline minus select product\_id from orderline, ordertable where orderline.order\_id= ordertable.order\_id and order\_date > '2008/10/28' order by product\_id ASC;

1. List ID for all Arizona customers, who have placed order(s) since 28-OCT-2008. Sort your results by cust\_id in ascending order. Use Intersect for this query. (6 points)

CUST\_ID

----------

12

Answer:

select cust\_id from customer where state='AZ' intersect select cust\_id from ordertable where order\_date > '2008.10/28' group by cust\_id Order by cust\_id ASC ;

1. Implement query 11 using IN and subquery. (6 points)

CUST\_ID

----------

12

Answer:

select cust\_id from customer where state='AZ' and cust\_id IN ( select cust\_id from ordertable where order\_date > '2008.10/28') Order by cust\_id ASC ;

1. List total dollar amount ordered and ordered date for each order that has been placed since 28-OCT-2008, where for each product ordered its amount equals to quantity times price. Change the column title for the total dollar amount in the output as ‘Total\_Amount’. (6 points)

ORDER\_ID TOTAL\_AMOUNT ORDER\_DAT

---------------- -------------------------- -----------------

1010 749.9 05-NOV-08

1009 189.93 01-NOV-08

1008 401.97 30-OCT-08

Answer:

select orderline.order\_id, SUM(quantity\*product\_price) AS total\_amount, order\_date from orderline,product,ordertable where orderline.order\_id = ordertable.order\_id and product.product\_id = orderline.product\_id and order\_date > '2008/10/28' group by orderline.order\_id, ordertable.order\_date;

1. List ID for all California customers and all customers who have placed order(s) since 28-OCT-2008. Sort your results by cust\_id in ascending order. Use Union for this query. (6 points)

CUST\_ID

----------------

1

4

12

13

14

15

Answer:

select cust\_id from customer where State='CA' union select cust\_id from ordertable where order\_date > '2008/10/28' order by cust\_id ASC;

1. List ID, Name and total quantity ordered for all products with total quantity ordered greater than 10. (8 points)

PRODUCT\_ID PRODUCT\_NAME TOTAL\_ORDER\_QUANTITY

------------------- ---------------------------- --------------------------

103 Printer 11

108 Shredder 15

Answer:

select product\_id, product\_name, SUM(quantity) as total\_order\_quantity from product natural join orderline group by product\_id, product\_name Having SUM(quantity)>10;

1. List ID, Name and total quantity ordered for all products, which have been ordered more than 6 by Utah customers. (8 points)

PRODUCT\_ID PRODUCT\_NAME TOTAL\_ORDER\_QUANTITY

------------------- -------------------------- --------------------------

104 Desk 8

108 Shredder 12

Answer:

select product.product\_id, product\_name,SUM(quantity) as total\_order\_quantity from product,orderline,customer,ordertable where product.product\_id= orderline.product\_id and ordertable.cust\_id = customer.cust\_id and ordertable.order\_id = orderline.order\_id and state='UT' group by product.product\_id, product\_name having SUM(orderline.quantity) > 6;