

1) 1997년 1월의 각 직원 별 취급 제품 종류 수 (Number_of_Products)와 총 개수(Total_Quantity)를 보이시오.

SQL Statement:

```
SELECT EmployeeID, COUNT(*) Number_of_Product, SUM(Quantity) Total_Quantity
FROM OrderDetails JOIN
(SELECT * FROM Orders JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID
WHERE OrderDate LIKE '1997-01%') AS T1
ON T1.OrderID = OrderDetails.OrderID GROUP BY EmployeeID
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 8

EmployeeID	Number_of_Product	Total_Quantity
1	8	304
2	9	230
3	17	364
4	22	812
6	4	64
7	9	248
8	13	305
9	3	74

2) 당사는 각 직원 별 업무 성과(KPI)를 취급 제품의 (개수) * (가격)으로 계산하고 있다. 직원별 KPI를 KPI가 높은 순으로 보이시오.

SQL Statement:

```
SELECT EmployeeID, SUM(Quantity * Price) AS KPI
FROM Products JOIN (SELECT * FROM OrderDetails JOIN (SELECT * FROM Orders JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID) AS T1
ON T1.OrderID = OrderDetails.OrderID) AS T2
ON T2.ProductID = Products.ProductID GROUP BY EmployeeID
ORDER BY KPI DESC
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 9

EmployeeID	KPI
4	105696.49999999999
1	57690.38999999999
3	42838.350000000006
7	39772.3
8	39309.380000000005
2	32503.16
5	27480.8
6	25399.25
9	15734.099999999999

3) 당사는 KPI가 40000이 넘는 직원에게는 A, 30000이 넘는 직원에게는 B, 나머
지에게는 C를 부여하려 한다. 직원별 부여 점수(Performance)를 보이시오.

SQL Statement:

```
SELECT EmployeeID, CASE WHEN SUM(Quantity * Price) >= 40000 THEN 'A' WHEN SUM(Quantity * Price) >= 30000 THEN 'B' ELSE 'C' END AS KPI
FROM Products JOIN (SELECT * FROM OrderDetails JOIN (SELECT * FROM Orders JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID) AS T1
ON T1.OrderID = OrderDetails.OrderID) AS T2
ON T2.ProductID = Products.ProductID GROUP BY EmployeeID ORDER BY KPI ASC
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 9

EmployeeID	KPI
1	A
3	A
4	A
2	B
7	B
8	B
5	C
6	C
9	C