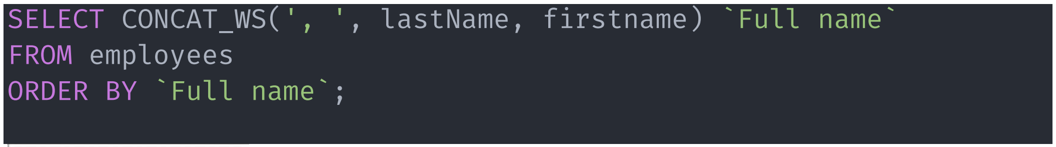
Tugas Nested / Subquery

Nama : Muhammad Nabil Afrizal Rahmadani

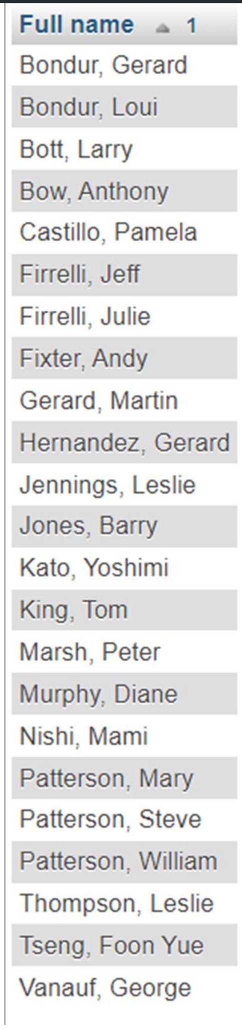
NRP : 5025231014

Kelas : Sistem Basis Data D

1. Section 4: Joining Tables
   A. Table & Column Aliases
   a. Query ini berfungsi untuk menggabungkan dua kolom berbeda yaitu lastName dan firstName, memberikan alias dengan nama Full name dan mengurutkan nya berdasarkan urutan alphabet

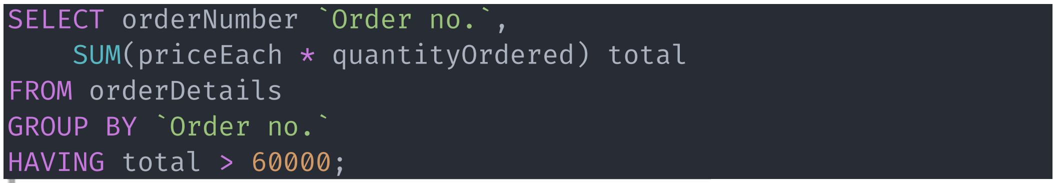


```
SELECT CONCAT_WS(', ', lastName, firstname) `Full name`
FROM employees
ORDER BY `Full name`;
```

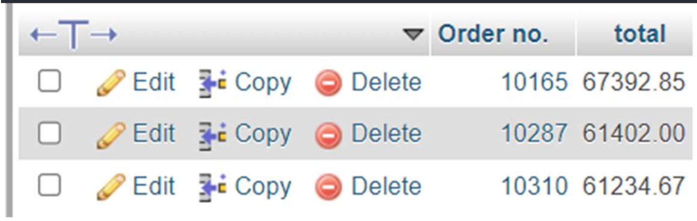


Full name ▲ 1
Bondur, Gerard
Bondur, Loui
Bott, Larry
Bow, Anthony
Castillo, Pamela
Firrelli, Jeff
Firrelli, Julie
Fixter, Andy
Gerard, Martin
Hernandez, Gerard
Jennings, Leslie
Jones, Barry
Kato, Yoshimi
King, Tom
Marsh, Peter
Murphy, Diane
Nishi, Mami
Patterson, Mary
Patterson, Steve
Patterson, William
Thompson, Leslie
Tseng, Foon Yue
Vanauf, George

b. Query ini menampilkan dimana jumlah total dari order lebih dari 6000 dari tabel orderDetails. Query ini juga menggunakan GROUP BY untuk mengelompokkan berdasarkan “Order no.”

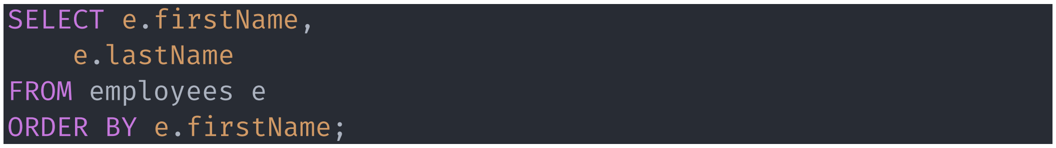


```sql
SELECT orderNumber `Order no.`,
    SUM(priceEach * quantityOrdered) total
FROM orderDetails
GROUP BY `Order no.`
HAVING total > 60000;
```



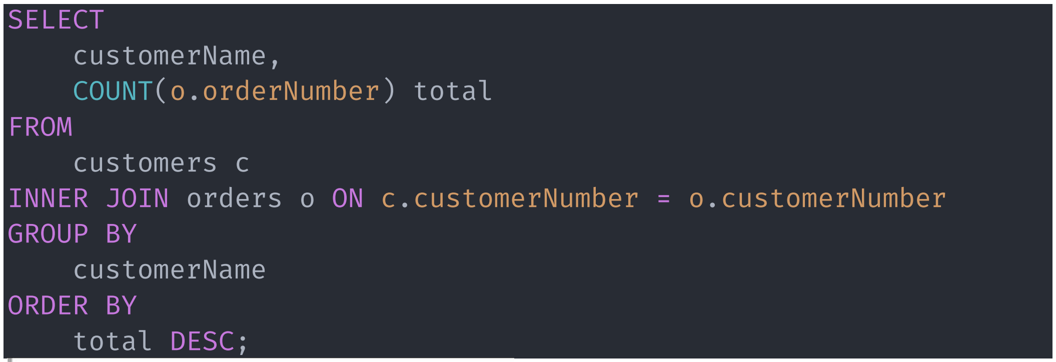
←T→			▼	Order no.	total
□	Edit	Copy	Delete	10165	67392.85
□	Edit	Copy	Delete	10287	61402.00
□	Edit	Copy	Delete	10310	61234.67

c. Query ini memberikan alias kepada tabel employees dengan nama e

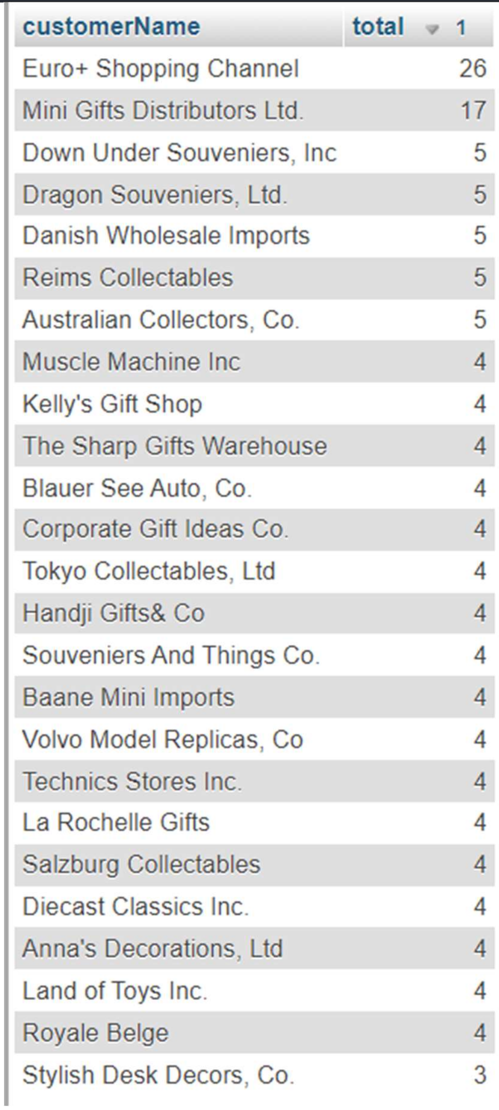


```sql
SELECT e.firstName,
    e.lastName
FROM employees e
ORDER BY e.firstName;
```

d. Query ini menampilkan informasi dari 2 tabel customers yang di aliaskan sebagai c dan tabel orders yang di aliaskan sebagai o, query ini melakukan INNER JOIN sehingga menyambungkan tabel orders dan customers dengan customerNumber yang sama. Query ini akan menampilkan nama customer dan total order yang telah dilakukan oleh customer itu dan diurutkan berdasarkan total.



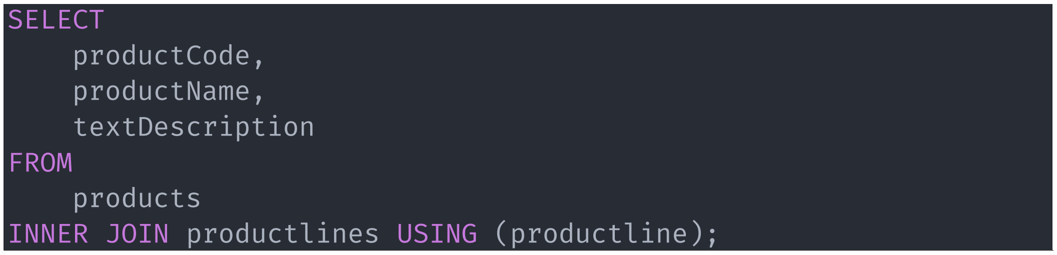
```
SELECT
    customerName,
    COUNT(o.orderNumber) total
FROM
    customers c
INNER JOIN orders o ON c.customerNumber = o.customerNumber
GROUP BY
    customerName
ORDER BY
    total DESC;
```



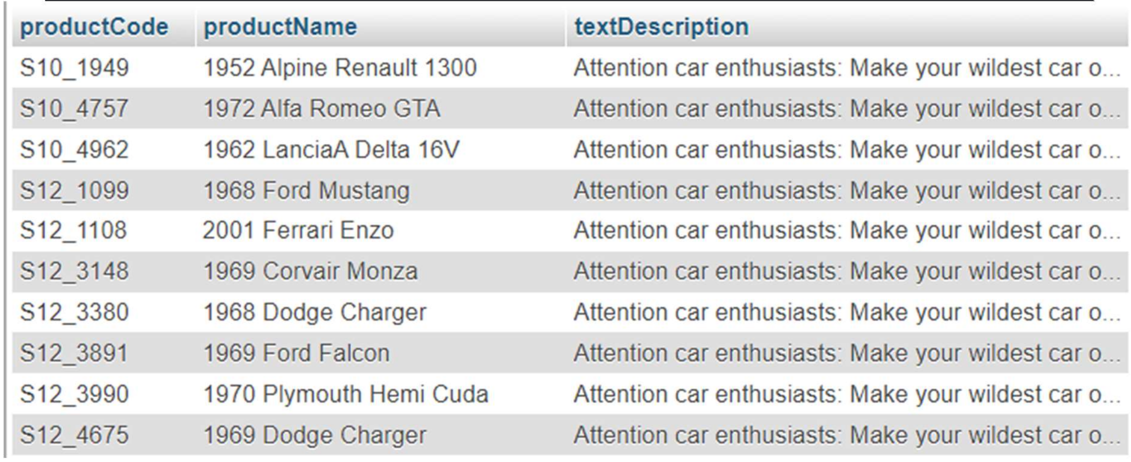
customerName	total ▼ 1
Euro+ Shopping Channel	26
Mini Gifts Distributors Ltd.	17
Down Under Souveniers, Inc	5
Dragon Souveniers, Ltd.	5
Danish Wholesale Imports	5
Reims Collectables	5
Australian Collectors, Co.	5
Muscle Machine Inc	4
Kelly's Gift Shop	4
The Sharp Gifts Warehouse	4
Blauer See Auto, Co.	4
Corporate Gift Ideas Co.	4
Tokyo Collectables, Ltd	4
Handji Gifts& Co	4
Souveniers And Things Co.	4
Baane Mini Imports	4
Volvo Model Replicas, Co	4
Technics Stores Inc.	4
La Rochelle Gifts	4
Salzburg Collectables	4
Diecast Classics Inc.	4
Anna's Decorations, Ltd	4
Land of Toys Inc.	4
Royale Belge	4
Stylish Desk Decors, Co.	3

B. INNER JOINS

a. Query ini berfungsi menampilkan data productCode, productName dan textDesc dari 2 tabel berbeda, karena textDesc berada di tabel berbeda, maka dilakukan INNER JOIN berdasarkan foreign key nya yaitu productLine sehingga dapat ditampilkan data nya.

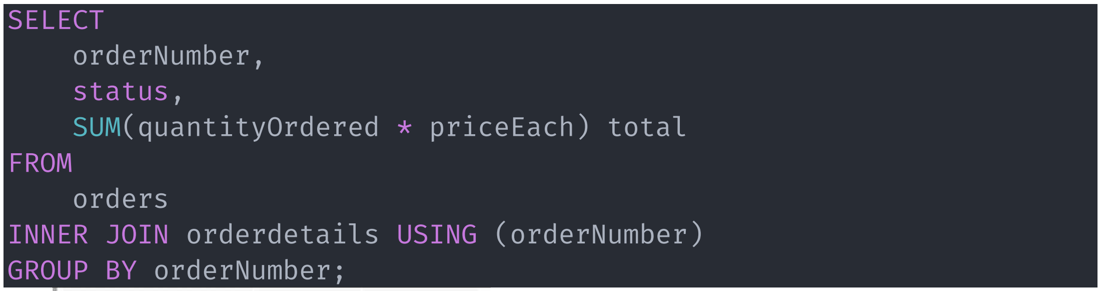


```sql
SELECT
    productCode,
    productName,
    textDescription
FROM
    products
INNER JOIN productlines USING (productline);
```

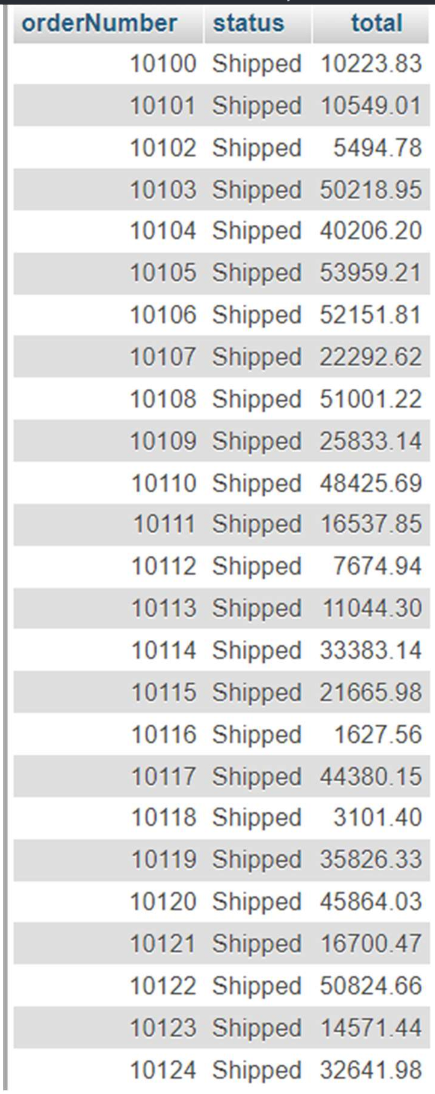


productCode	productName	textDescription
S10_1949	1952 Alpine Renault 1300	Attention car enthusiasts: Make your wildest car o...
S10_4757	1972 Alfa Romeo GTA	Attention car enthusiasts: Make your wildest car o...
S10_4962	1962 LanciaA Delta 16V	Attention car enthusiasts: Make your wildest car o...
S12_1099	1968 Ford Mustang	Attention car enthusiasts: Make your wildest car o...
S12_1108	2001 Ferrari Enzo	Attention car enthusiasts: Make your wildest car o...
S12_3148	1969 Corvair Monza	Attention car enthusiasts: Make your wildest car o...
S12_3380	1968 Dodge Charger	Attention car enthusiasts: Make your wildest car o...
S12_3891	1969 Ford Falcon	Attention car enthusiasts: Make your wildest car o...
S12_3990	1970 Plymouth Hemi Cuda	Attention car enthusiasts: Make your wildest car o...
S12_4675	1969 Dodge Charger	Attention car enthusiasts: Make your wildest car o...

b. Query ini berfungsi menampilkan data orderNumber, status, dan total nya. Karena quantityOrdered dan priceEach berada di tabel yang berbeda, dilakukan INNER JOIN berdasarkan orderNumber sehingga dan data di GROUP BY berdasarkan orderNumber nya lalu ditampilkan

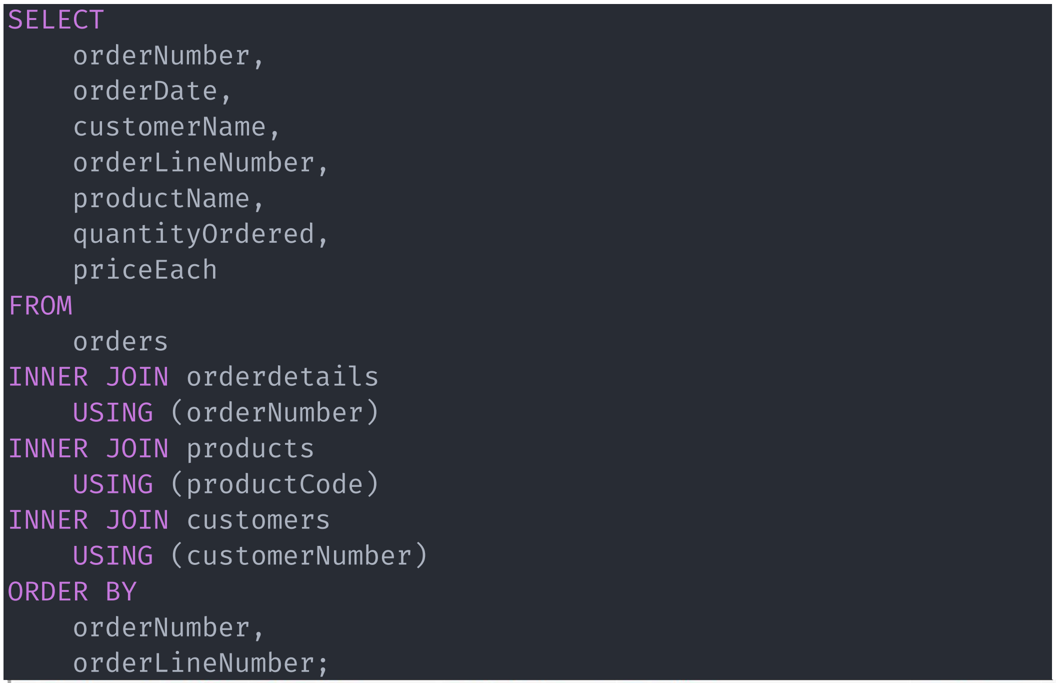


```
SELECT
    orderNumber,
    status,
    SUM(quantityOrdered * priceEach) total
FROM
    orders
INNER JOIN orderdetails USING (orderNumber)
GROUP BY orderNumber;
```

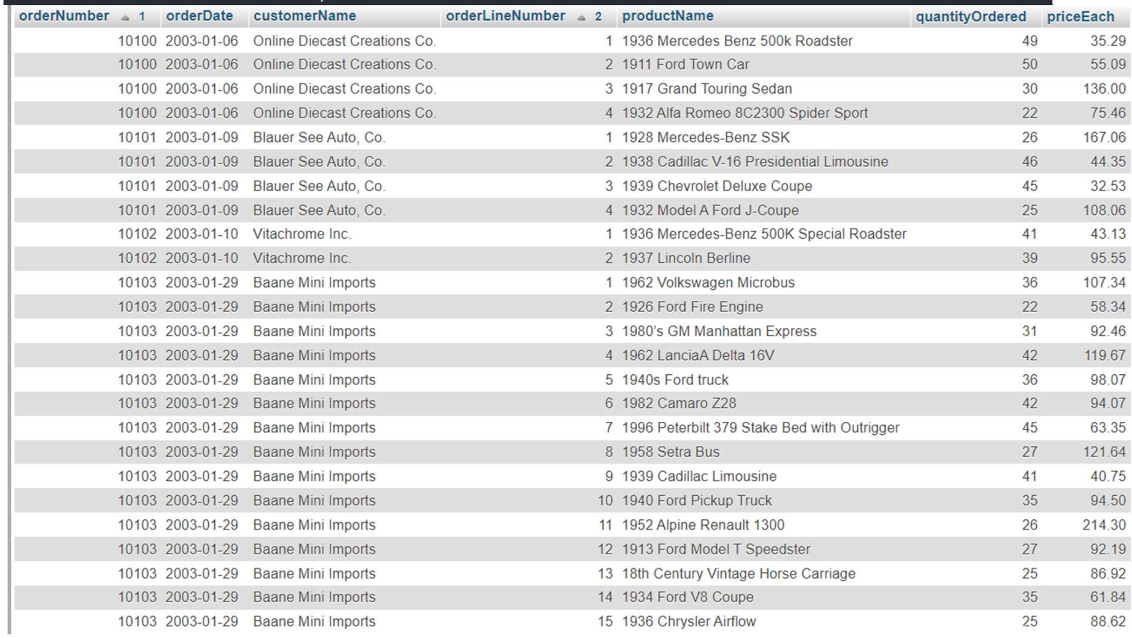


orderNumber	status	total
10100	Shipped	10223.83
10101	Shipped	10549.01
10102	Shipped	5494.78
10103	Shipped	50218.95
10104	Shipped	40206.20
10105	Shipped	53959.21
10106	Shipped	52151.81
10107	Shipped	22292.62
10108	Shipped	51001.22
10109	Shipped	25833.14
10110	Shipped	48425.69
10111	Shipped	16537.85
10112	Shipped	7674.94
10113	Shipped	11044.30
10114	Shipped	33383.14
10115	Shipped	21665.98
10116	Shipped	1627.56
10117	Shipped	44380.15
10118	Shipped	3101.40
10119	Shipped	35826.33
10120	Shipped	45864.03
10121	Shipped	16700.47
10122	Shipped	50824.66
10123	Shipped	14571.44
10124	Shipped	32641.98

c. Query ini menampilkan orderNumber, orderDate, customerName, orderLineNumber, productName, quantityOrdered dan priceEach. Dari tabel order dilakukan INNER JOIN dengan tabel orderdetails menggunakan foreign key nya orderNumber, lalu dilakukan lagi INNER JOIN dengan tabel products menggunakan foreign key nya yaitu productCode, lalu dilakukan INNER JOIN lagi dengan tabel customers menggunakan foreign key nya yaitu customerNumber yang di dapat dari tabel orders

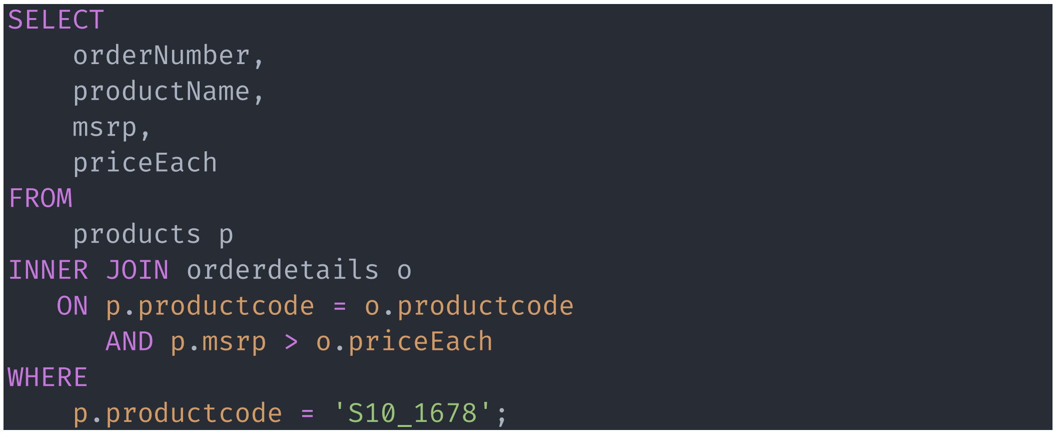


```
SELECT
    orderNumber,
    orderDate,
    customerName,
    orderLineNumber,
    productName,
    quantityOrdered,
    priceEach
FROM
    orders
INNER JOIN orderdetails
    USING (orderNumber)
INNER JOIN products
    USING (productCode)
INNER JOIN customers
    USING (customerNumber)
ORDER BY
    orderNumber,
    orderLineNumber;
```



orderNumber ▲ 1	orderDate	customerName	orderLineNumber ▲ 2	productName	quantityOrdered	priceEach
10100	2003-01-06	Online Diecast Creations Co.	1	1936 Mercedes Benz 500k Roadster	49	35.29
10100	2003-01-06	Online Diecast Creations Co.	2	1911 Ford Town Car	50	55.09
10100	2003-01-06	Online Diecast Creations Co.	3	1917 Grand Touring Sedan	30	136.00
10100	2003-01-06	Online Diecast Creations Co.	4	1932 Alfa Romeo 8C2300 Spider Sport	22	75.46
10101	2003-01-09	Blauer See Auto, Co.	1	1928 Mercedes-Benz SSK	26	167.06
10101	2003-01-09	Blauer See Auto, Co.	2	1938 Cadillac V-16 Presidential Limousine	46	44.35
10101	2003-01-09	Blauer See Auto, Co.	3	1939 Chevrolet Deluxe Coupe	45	32.53
10101	2003-01-09	Blauer See Auto, Co.	4	1932 Model A Ford J-Coupe	25	108.06
10102	2003-01-10	Vitachrome Inc.	1	1936 Mercedes-Benz 500K Special Roadster	41	43.13
10102	2003-01-10	Vitachrome Inc.	2	1937 Lincoln Berline	39	95.55
10103	2003-01-29	Baane Mini Imports	1	1962 Volkswagen Microbus	36	107.34
10103	2003-01-29	Baane Mini Imports	2	1926 Ford Fire Engine	22	58.34
10103	2003-01-29	Baane Mini Imports	3	1980's GM Manhattan Express	31	92.46
10103	2003-01-29	Baane Mini Imports	4	1962 LanciaA Delta 16V	42	119.67
10103	2003-01-29	Baane Mini Imports	5	1940s Ford truck	36	98.07
10103	2003-01-29	Baane Mini Imports	6	1982 Camaro Z28	42	94.07
10103	2003-01-29	Baane Mini Imports	7	1996 Peterbilt 379 Stake Bed with Outrigger	45	63.35
10103	2003-01-29	Baane Mini Imports	8	1958 Setra Bus	27	121.64
10103	2003-01-29	Baane Mini Imports	9	1939 Cadillac Limousine	41	40.75
10103	2003-01-29	Baane Mini Imports	10	1940 Ford Pickup Truck	35	94.50
10103	2003-01-29	Baane Mini Imports	11	1952 Alpine Renault 1300	26	214.30
10103	2003-01-29	Baane Mini Imports	12	1913 Ford Model T Speedster	27	92.19
10103	2003-01-29	Baane Mini Imports	13	18th Century Vintage Horse Carriage	25	86.92
10103	2003-01-29	Baane Mini Imports	14	1934 Ford V8 Coupe	35	61.84
10103	2003-01-29	Baane Mini Imports	15	1936 Chrysler Airflow	25	88.62

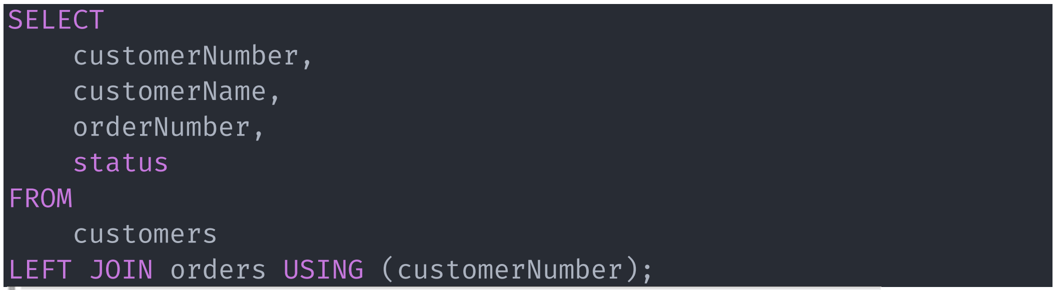
d. Menampilkan data orderNumber, productName, msrp, priceEach dari productCode “S10_1678”. Dari tabel products di INNER JOIN dengan tabel orderdetails, tetapi hanya tampilkan dimana msrp > priceEach



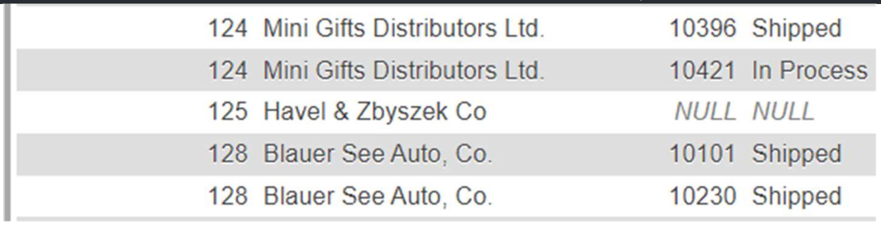
```
SELECT
    orderNumber,
    productName,
    msrp,
    priceEach
FROM
    products p
INNER JOIN orderdetails o
   ON p.productcode = o.productcode
      AND p.msrp > o.priceEach
WHERE
    p.productcode = 'S10_1678';
```

C. LEFT JOIN

a. Query ini menampilkan data customerNumber, customerName, orderNumber dan status dari tabel customers lalu di LEFT JOIN dengan tabel orders menggunakan customerNumber. Query ini akan menampilkan semua data customers, meskipun customer tersebut belum pernah melakukan order.

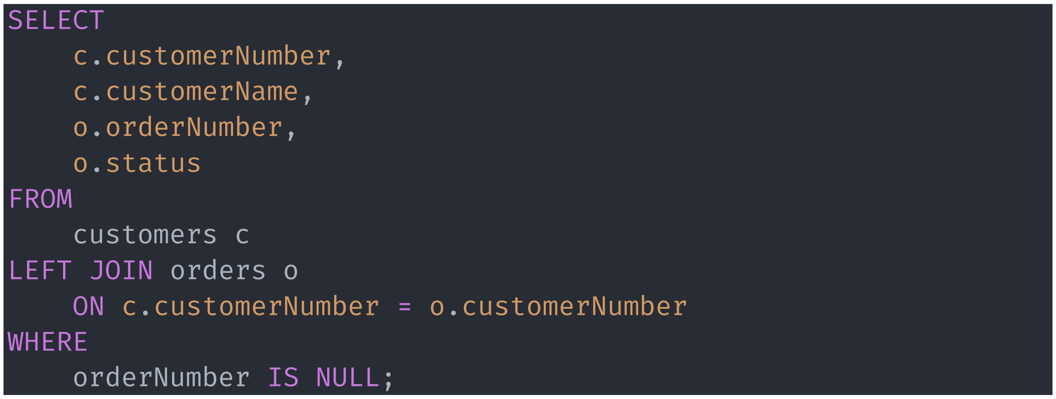


```
SELECT
    customerNumber,
    customerName,
    orderNumber,
    status
FROM
    customers
LEFT JOIN orders USING (customerNumber);
```

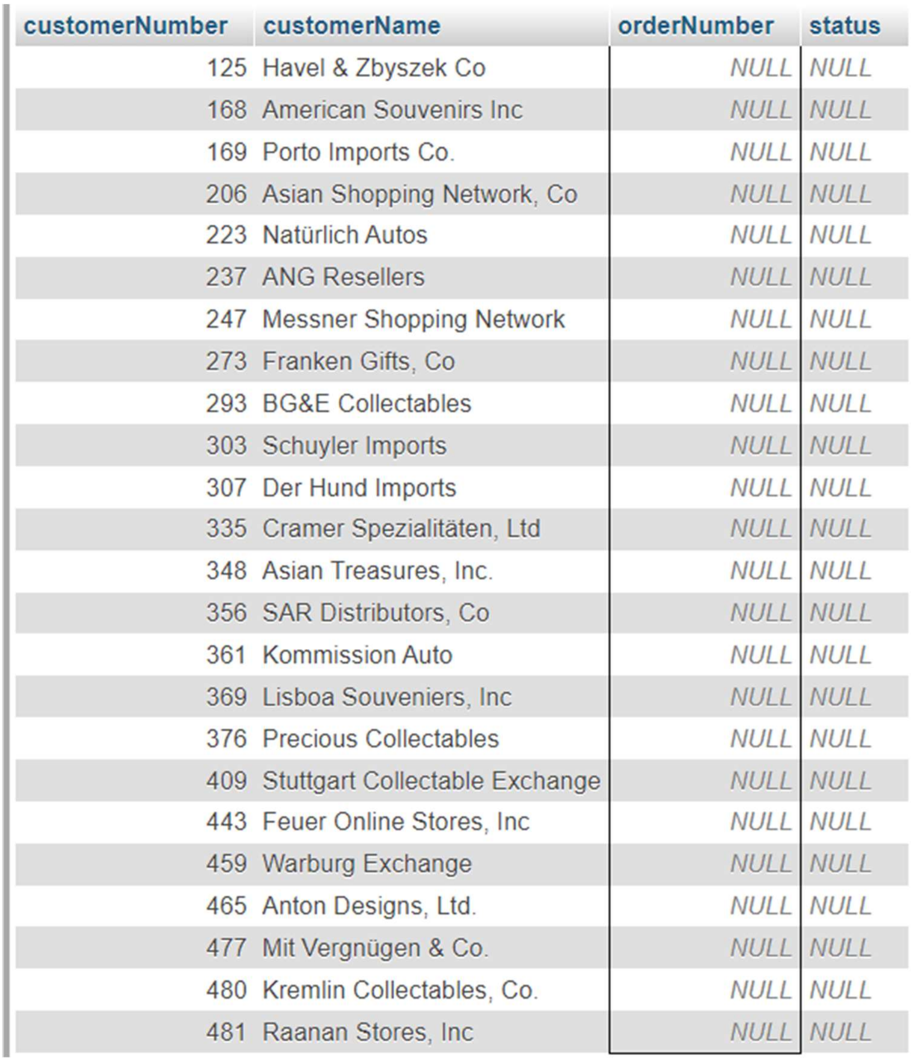


124	Mini Gifts Distributors Ltd.	10396	Shipped
124	Mini Gifts Distributors Ltd.	10421	In Process
125	Havel & Zbyszek Co	*NULL*	*NULL*
128	Blauer See Auto, Co.	10101	Shipped
128	Blauer See Auto, Co.	10230	Shipped

b. Query ini menampilkan data customerNumber, customerName, orderNumber dan status dari tabel customers lalu di LEFT JOIN dengan tabel orders menggunakan customerNumber. Query ini akan menampilkan data dimana customer tersebut tidak pernah melakukan order.



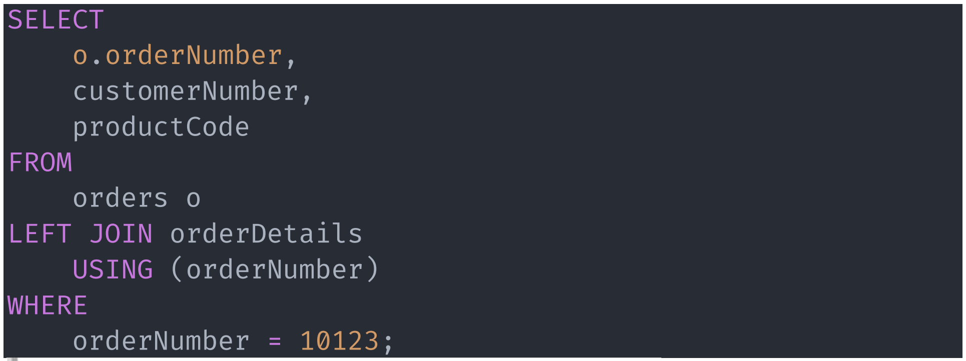
```
SELECT
    c.customerNumber,
    c.customerName,
    o.orderNumber,
    o.status
FROM
    customers c
LEFT JOIN orders o
    ON c.customerNumber = o.customerNumber
WHERE
    orderNumber IS NULL;
```



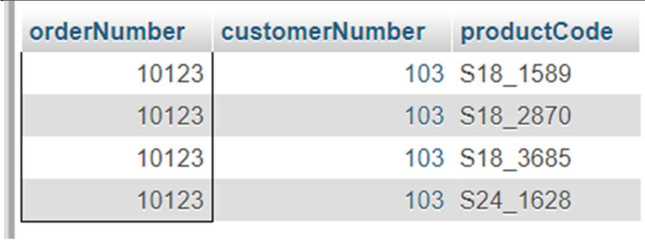
customerNumber	customerName	orderNumber	status
125	Havel & Zbyszek Co	NULL	NULL
168	American Souvenirs Inc	NULL	NULL
169	Porto Imports Co.	NULL	NULL
206	Asian Shopping Network, Co	NULL	NULL
223	Natürlich Autos	NULL	NULL
237	ANG Resellers	NULL	NULL
247	Messner Shopping Network	NULL	NULL
273	Franken Gifts, Co	NULL	NULL
293	BG&E Collectables	NULL	NULL
303	Schuyler Imports	NULL	NULL
307	Der Hund Imports	NULL	NULL
335	Cramer Spezialitäten, Ltd	NULL	NULL
348	Asian Treasures, Inc.	NULL	NULL
356	SAR Distributors, Co	NULL	NULL
361	Kommission Auto	NULL	NULL
369	Lisboa Souveniers, Inc	NULL	NULL
376	Precious Collectables	NULL	NULL
409	Stuttgart Collectable Exchange	NULL	NULL
443	Feuer Online Stores, Inc	NULL	NULL
459	Warburg Exchange	NULL	NULL
465	Anton Designs, Ltd.	NULL	NULL
477	Mit Vergnügen & Co.	NULL	NULL
480	Kremlin Collectables, Co.	NULL	NULL
481	Raanan Stores, Inc	NULL	NULL

c. WHERE vs ON

   a. Pada query ini data dari orders di LEFT JOIN dengan data dari orderDetails dengan orderNumber 10123.

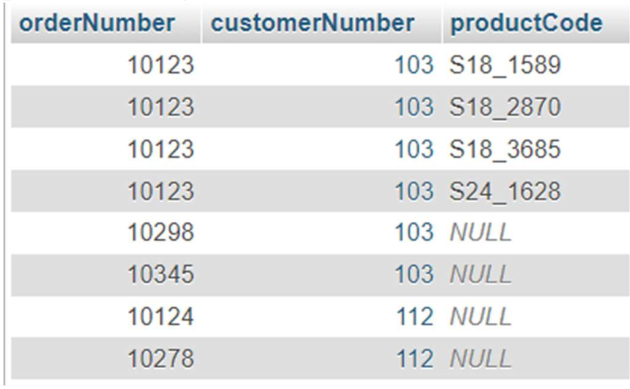


```
SELECT
    o.orderNumber,
    customerNumber,
    productCode
FROM
    orders o
LEFT JOIN orderDetails
    USING (orderNumber)
WHERE
    orderNumber = 10123;
```



orderNumber	customerNumber	productCode
10123	103	S18_1589
10123	103	S18_2870
10123	103	S18_3685
10123	103	S24_1628

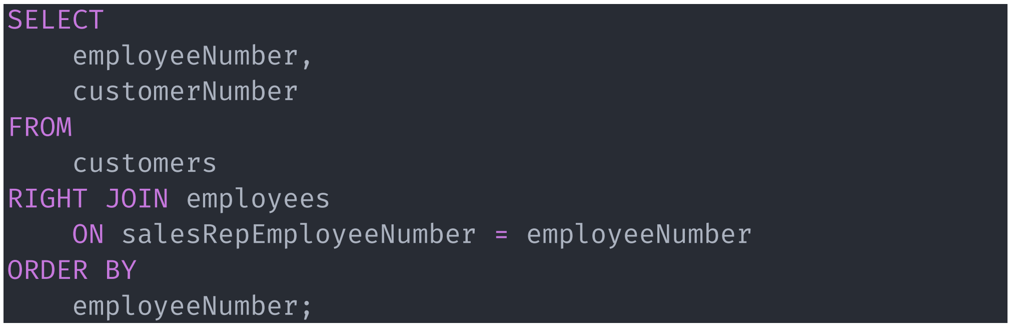
   b. Pada query ini data dari orders di LEFT JOIN dengan data dari orderDetails, tetapi hanya data dengan orderNumber 10123 yang akan menampilkan product code nya.



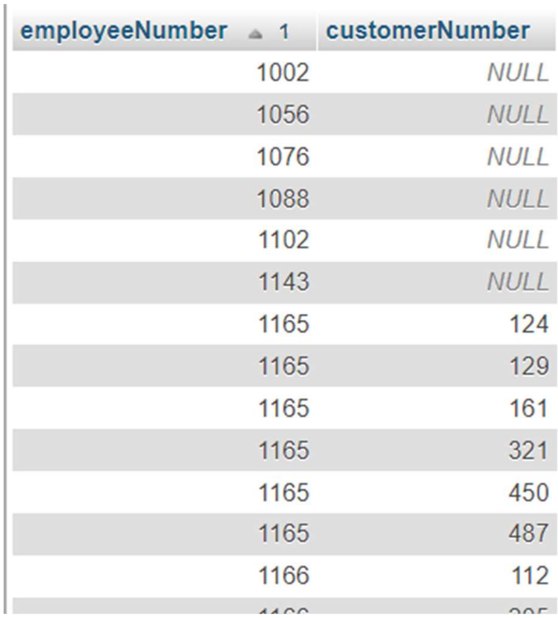
orderNumber	customerNumber	productCode
10123	103	S18_1589
10123	103	S18_2870
10123	103	S18_3685
10123	103	S24_1628
10298	103	*NULL*
10345	103	*NULL*
10124	112	*NULL*
10278	112	*NULL*

D. RIGHT JOIN

a. Query ini menggabungkan table customer dengan employee menggunakan RIGHT JOIN dengan salesRepEmployeeNumber dari tabel customer = employeeNumber dari tabel employee lalu diurutkan berdasarkan employeeNumber.

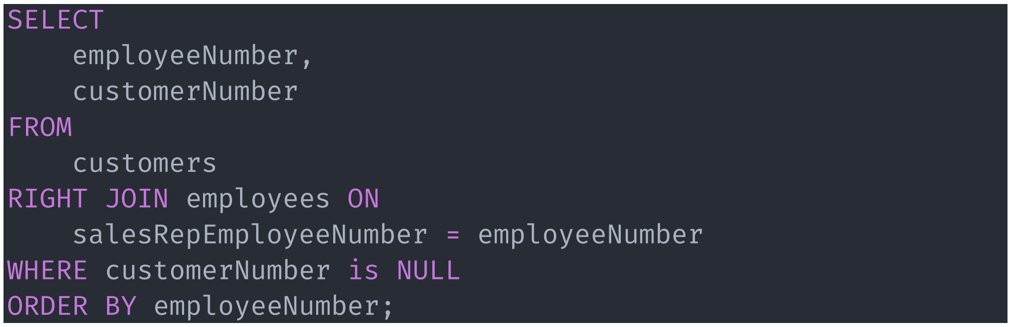


```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees
    ON salesRepEmployeeNumber = employeeNumber
ORDER BY
    employeeNumber;
```

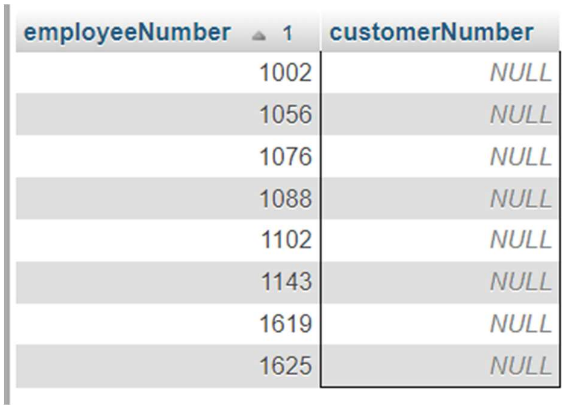


employeeNumber ▲ 1	customerNumber
1002	NULL
1056	NULL
1076	NULL
1088	NULL
1102	NULL
1143	NULL
1165	124
1165	129
1165	161
1165	321
1165	450
1165	487
1166	112
1166	205

b. Query ini menampilkan data employee yang tidak pernah berinteraksi dengan customer



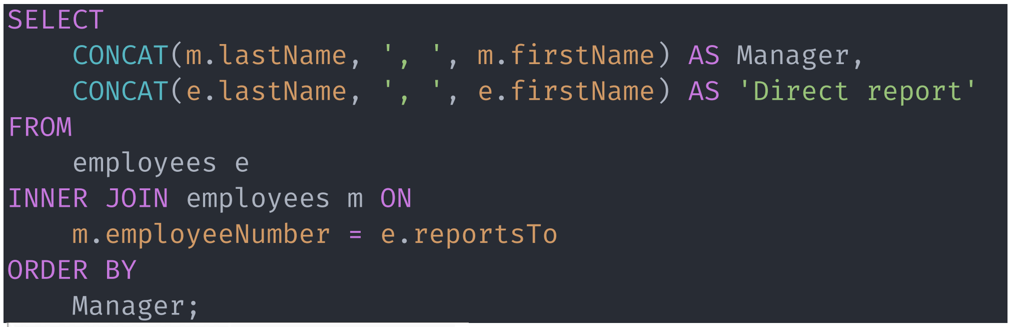
```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees ON
    salesRepEmployeeNumber = employeeNumber
WHERE customerNumber is NULL
ORDER BY employeeNumber;
```



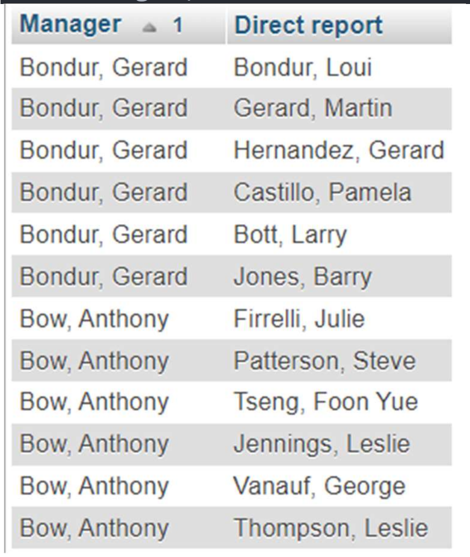
employeeNumber ▲ 1	customerNumber
1002	*NULL*
1056	*NULL*
1076	*NULL*
1088	*NULL*
1102	*NULL*
1143	*NULL*
1619	*NULL*
1625	*NULL*

E. SELF JOIN

a. Query ini menampilkan data manager dan bawahan nya (direct report), menggunakan foreign key reportsTo dengan INNER JOIN

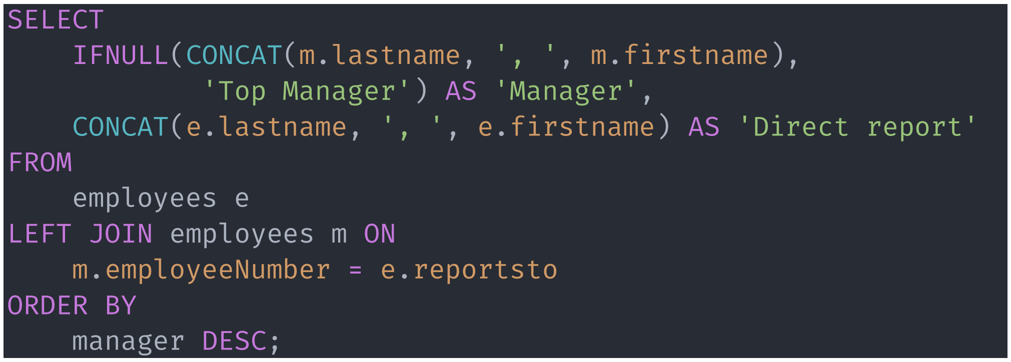


```
SELECT
    CONCAT(m.lastName, ', ', m.firstName) AS Manager,
    CONCAT(e.lastName, ', ', e.firstName) AS 'Direct report'
FROM
    employees e
INNER JOIN employees m ON
    m.employeeNumber = e.reportsTo
ORDER BY
    Manager;
```

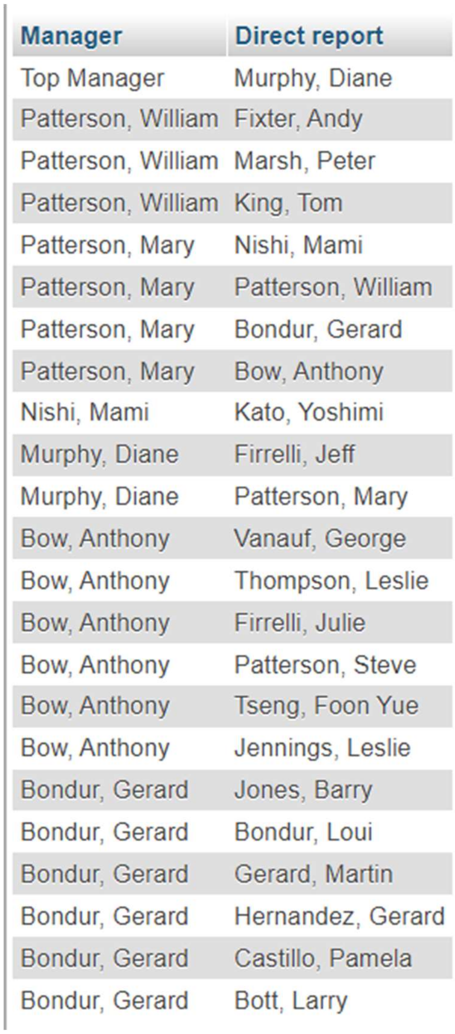


Manager ▲ 1	Direct report
Bondur, Gerard	Bondur, Loui
Bondur, Gerard	Gerard, Martin
Bondur, Gerard	Hernandez, Gerard
Bondur, Gerard	Castillo, Pamela
Bondur, Gerard	Bott, Larry
Bondur, Gerard	Jones, Barry
Bow, Anthony	Firrelli, Julie
Bow, Anthony	Patterson, Steve
Bow, Anthony	Tseng, Foon Yue
Bow, Anthony	Jennings, Leslie
Bow, Anthony	Vanauf, George
Bow, Anthony	Thompson, Leslie

b. Query ini menampilkan data manager dan bawahan nya menggunakan LEFT JOIN jika reportsTo adalah NULL maka manager akan terisi dengan Top Manager.

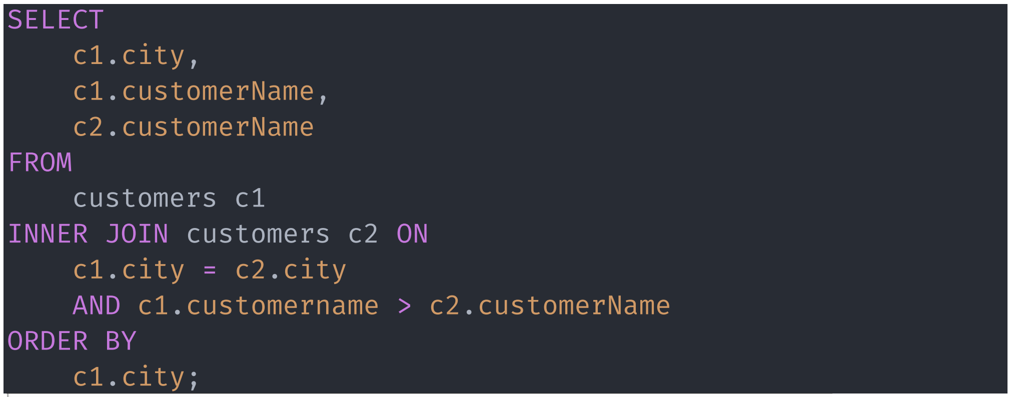


```
SELECT
    IFNULL(CONCAT(m.lastname, ', ', m.firstname),
            'Top Manager') AS 'Manager',
    CONCAT(e.lastname, ', ', e.firstname) AS 'Direct report'
FROM
    employees e
LEFT JOIN employees m ON
    m.employeeNumber = e.reportsto
ORDER BY
    manager DESC;
```

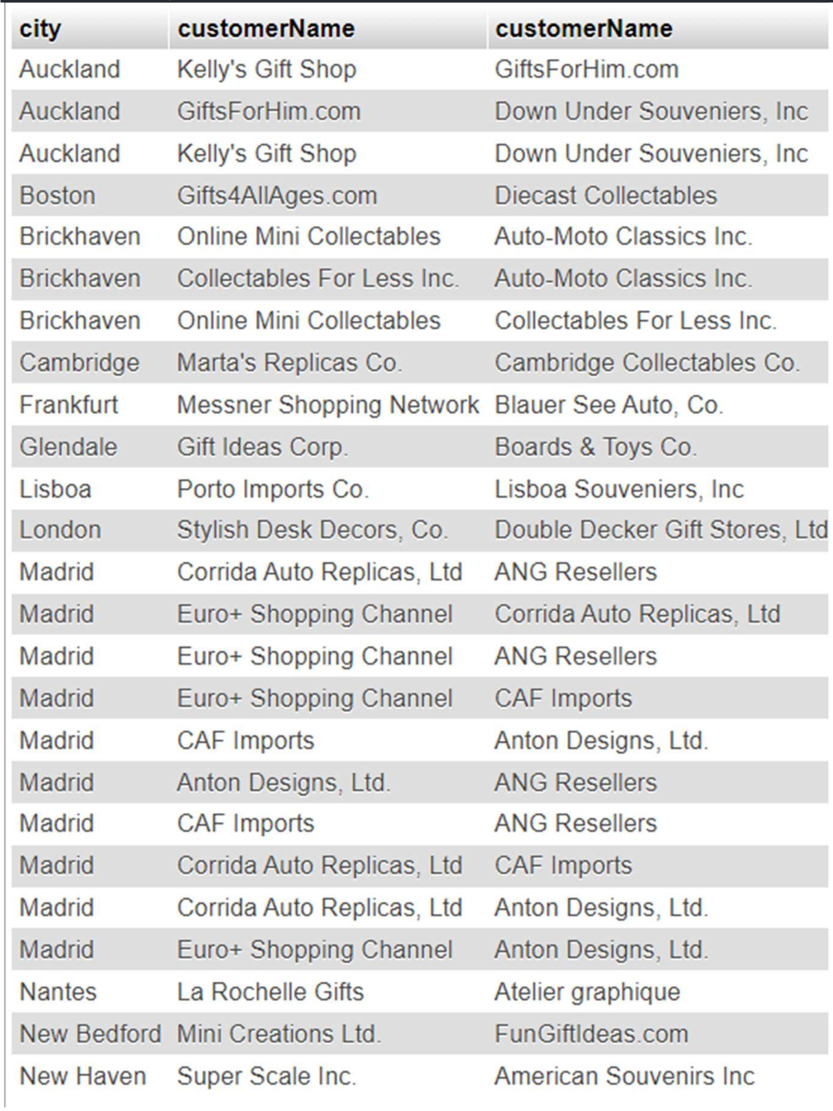


Manager	Direct report
Top Manager	Murphy, Diane
Patterson, William	Fixter, Andy
Patterson, William	Marsh, Peter
Patterson, William	King, Tom
Patterson, Mary	Nishi, Mami
Patterson, Mary	Patterson, William
Patterson, Mary	Bondur, Gerard
Patterson, Mary	Bow, Anthony
Nishi, Mami	Kato, Yoshimi
Murphy, Diane	Firrelli, Jeff
Murphy, Diane	Patterson, Mary
Bow, Anthony	Vanauf, George
Bow, Anthony	Thompson, Leslie
Bow, Anthony	Firrelli, Julie
Bow, Anthony	Patterson, Steve
Bow, Anthony	Tseng, Foon Yue
Bow, Anthony	Jennings, Leslie
Bondur, Gerard	Jones, Barry
Bondur, Gerard	Bondur, Loui
Bondur, Gerard	Gerard, Martin
Bondur, Gerard	Hernandez, Gerard
Bondur, Gerard	Castillo, Pamela
Bondur, Gerard	Bott, Larry

c. Query ini berguna untuk membandingkan row berurutan dari satu tabel yang sama. Dalam konteks ini, jika kota nya sama maka tampilkan lalu di ORDER BY city



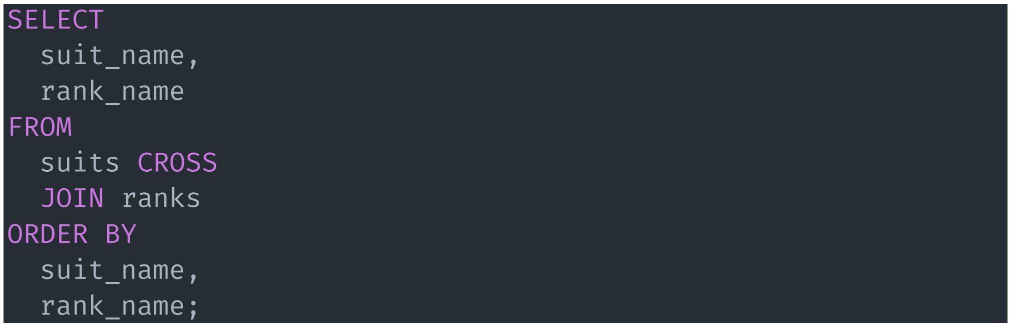
```
SELECT
    c1.city,
    c1.customerName,
    c2.customerName
FROM
    customers c1
INNER JOIN customers c2 ON
    c1.city = c2.city
    AND c1.customername > c2.customerName
ORDER BY
    c1.city;
```



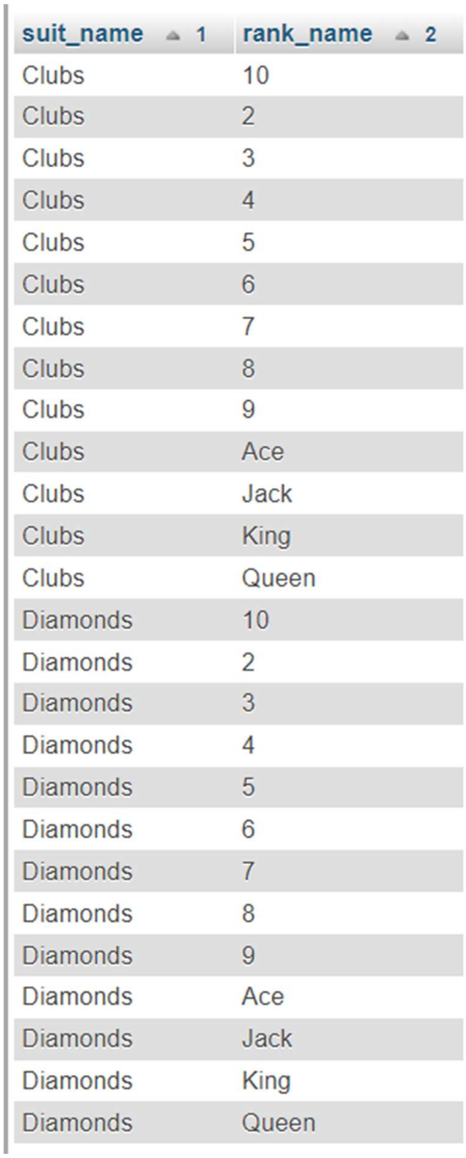
city	customerName	customerName
Auckland	Kelly's Gift Shop	GiftsForHim.com
Auckland	GiftsForHim.com	Down Under Souveniers, Inc
Auckland	Kelly's Gift Shop	Down Under Souveniers, Inc
Boston	Gifts4AllAges.com	Diecast Collectables
Brickhaven	Online Mini Collectables	Auto-Moto Classics Inc.
Brickhaven	Collectables For Less Inc.	Auto-Moto Classics Inc.
Brickhaven	Online Mini Collectables	Collectables For Less Inc.
Cambridge	Marta's Replicas Co.	Cambridge Collectables Co.
Frankfurt	Messner Shopping Network	Blauer See Auto, Co.
Glendale	Gift Ideas Corp.	Boards & Toys Co.
Lisboa	Porto Imports Co.	Lisboa Souveniers, Inc
London	Stylish Desk Decors, Co.	Double Decker Gift Stores, Ltd
Madrid	Corrida Auto Replicas, Ltd	ANG Resellers
Madrid	Euro+ Shopping Channel	Corrida Auto Replicas, Ltd
Madrid	Euro+ Shopping Channel	ANG Resellers
Madrid	Euro+ Shopping Channel	CAF Imports
Madrid	CAF Imports	Anton Designs, Ltd.
Madrid	Anton Designs, Ltd.	ANG Resellers
Madrid	CAF Imports	ANG Resellers
Madrid	Corrida Auto Replicas, Ltd	CAF Imports
Madrid	Corrida Auto Replicas, Ltd	Anton Designs, Ltd.
Madrid	Euro+ Shopping Channel	Anton Designs, Ltd.
Nantes	La Rochelle Gifts	Atelier graphique
New Bedford	Mini Creations Ltd.	FunGiftIdeas.com
New Haven	Super Scale Inc.	American Souvenirs Inc

F. CROSS JOIN

a. Pada tabel ini diberikan sebuah database *playing cards*. Agar semua symbol dapat mendapat semua ranks. Dilakukan CROSS JOIN.

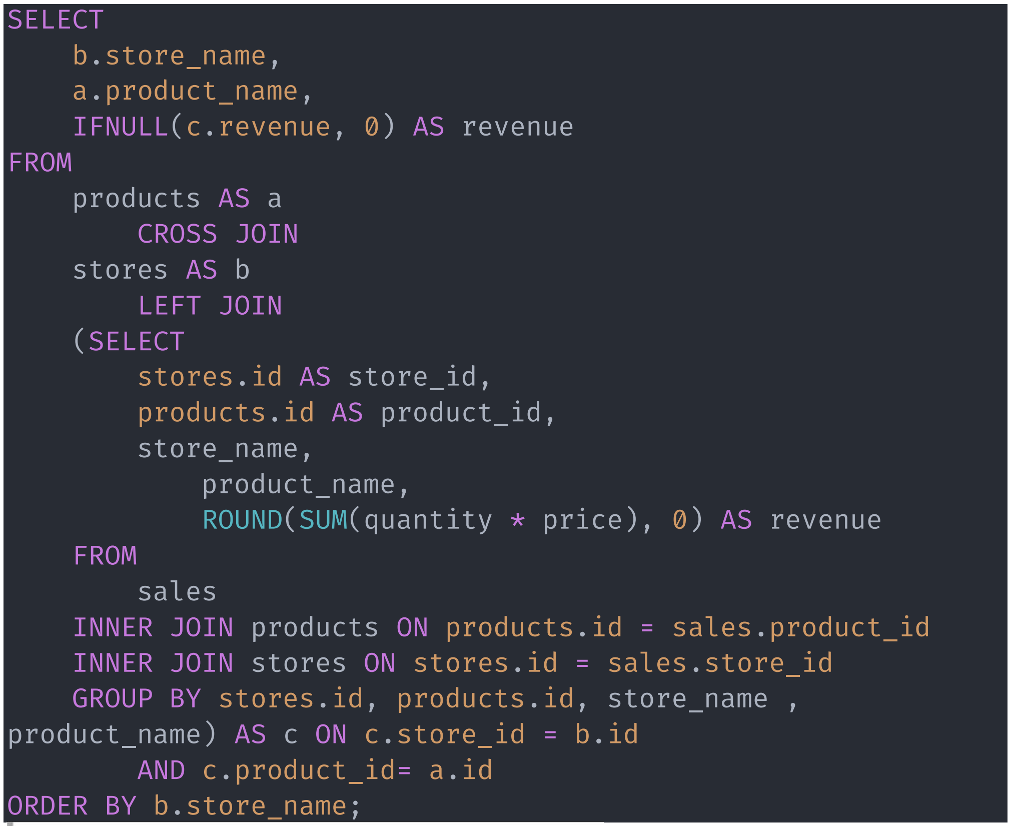


```
SELECT
  suit_name,
  rank_name
FROM
  suits CROSS
  JOIN ranks
ORDER BY
  suit_name,
  rank_name;
```

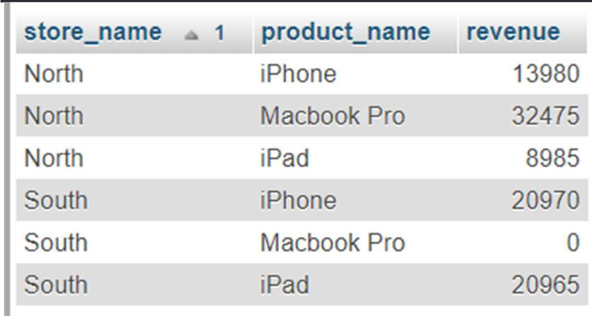


suit_name ▲ 1	rank_name ▲ 2
Clubs	10
Clubs	2
Clubs	3
Clubs	4
Clubs	5
Clubs	6
Clubs	7
Clubs	8
Clubs	9
Clubs	Ace
Clubs	Jack
Clubs	King
Clubs	Queen
Diamonds	10
Diamonds	2
Diamonds	3
Diamonds	4
Diamonds	5
Diamonds	6
Diamonds	7
Diamonds	8
Diamonds	9
Diamonds	Ace
Diamonds	Jack
Diamonds	King
Diamonds	Queen

b. Diberikan sebuah database bernama salesdb, pertama dilakukan INNER JOIN products dengan sales berdasarkan products id nya. Lalu INNER JOIN dilakukan lagi antara stores dengan sales berdasarkan id nya. Lalu di group berdasarkan store_name dan product name. Lalu untuk menemukan toko yang tidak menjual suatu barang dilakukan CROSS JOIN untuk mendapatkan semua data lalu di filter dengan IFNULL.



```
SELECT
    b.store_name,
    a.product_name,
    IFNULL(c.revenue, 0) AS revenue
FROM
    products AS a
        CROSS JOIN
    stores AS b
        LEFT JOIN
    (SELECT
        stores.id AS store_id,
        products.id AS product_id,
        store_name,
            product_name,
            ROUND(SUM(quantity * price), 0) AS revenue
    FROM
        sales
    INNER JOIN products ON products.id = sales.product_id
    INNER JOIN stores ON stores.id = sales.store_id
    GROUP BY stores.id, products.id, store_name ,
product_name) AS c ON c.store_id = b.id
        AND c.product_id= a.id
ORDER BY b.store_name;
```



store_name ▲ 1	product_name	revenue
North	iPhone	13980
North	Macbook Pro	32475
North	iPad	8985
South	iPhone	20970
South	Macbook Pro	0
South	iPad	20965