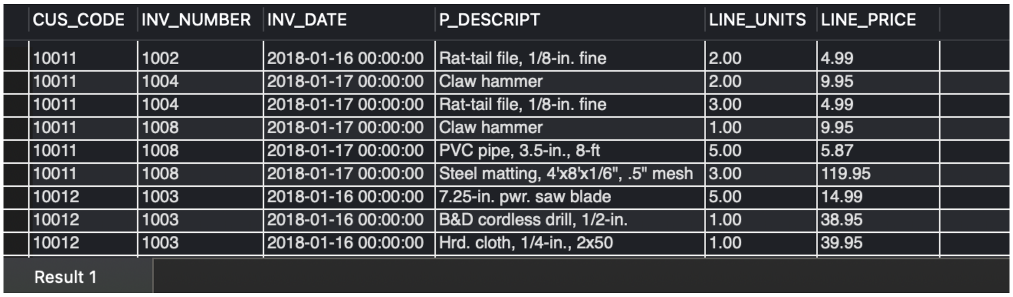
1) List all purchases made by customers. Your query should return the Cus_Code, Inv_Number, Inv_Date, P_Descript, Line_Units and Line_Price. Sort by Customer code, invoice number, and product description in that order.

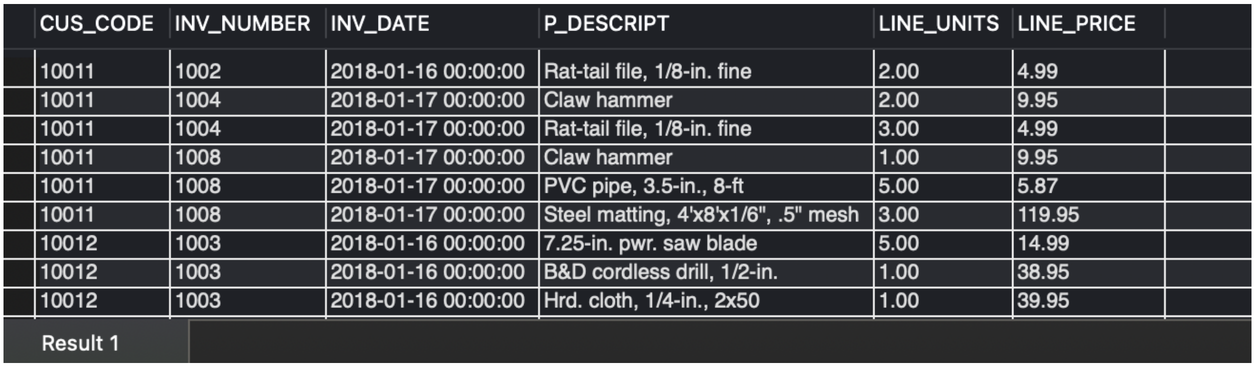
```
SELECT
        CUSTOMER.CUS_CODE,
        INVOICE.INV_NUMBER,
        INVOICE.INV_DATE,
        PRODUCT.P_DESCRIPT,
        LINE.LINE_UNITS,
        LINE.LINE_PRICE
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
JOIN PRODUCT ON LINE.P_CODE = PRODUCT.P_CODE
ORDER BY CUSTOMER.CUS_CODE, INVOICE.INV_NUMBER, PRODUCT.P_DESCRIPT;
```



CUS_CODE	INV_NUMBER	INV_DATE	P_DESCRIPT	LINE_UNITS	LINE_PRICE
10011	1002	2018-01-16 00:00:00	Rat-tail file, 1/8-in. fine	2.00	4.99
10011	1004	2018-01-17 00:00:00	Claw hammer	2.00	9.95
10011	1004	2018-01-17 00:00:00	Rat-tail file, 1/8-in. fine	3.00	4.99
10011	1008	2018-01-17 00:00:00	Claw hammer	1.00	9.95
10011	1008	2018-01-17 00:00:00	PVC pipe, 3.5-in., 8-ft	5.00	5.87
10011	1008	2018-01-17 00:00:00	Steel matting, 4'x8'x1/6", .5" mesh	3.00	119.95
10012	1003	2018-01-16 00:00:00	7.25-in. pwr. saw blade	5.00	14.99
10012	1003	2018-01-16 00:00:00	B&D cordless drill, 1/2-in.	1.00	38.95
10012	1003	2018-01-16 00:00:00	Hrd. cloth, 1/4-in., 2x50	1.00	39.95

Query 2: Generate a list of customer purchases, including subtotals for the invoice line numbers. Subtotals are a derived attribute, calculated by multipying Line_Units by Line_Price. Columns displayed should be Cus_Code, Inv_Number, P_Description, Units Bought (this is an alias), Unit Price (alias) and Subtotal (alias). It is up to you to determine which columns you will use to create the aliases.

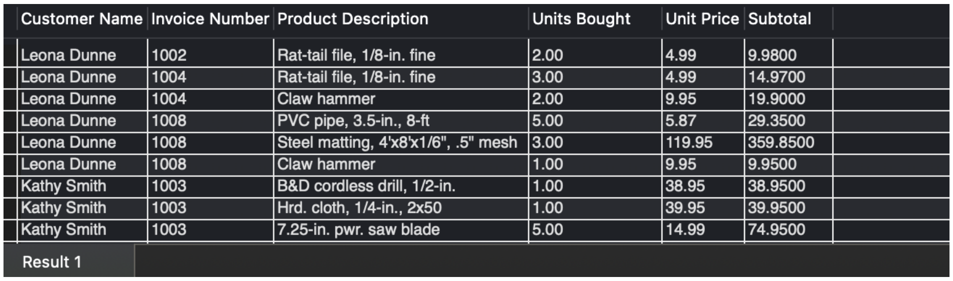
```
SELECT
        CUSTOMER.CUS_CODE, INVOICE.INV_NUMBER, PRODUCT.P_DESCRIPT,
        LINE.LINE_UNITS AS `Units Bought`,
        LINE.LINE_PRICE AS `Unit Price`,
        (LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Subtotal`
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
JOIN PRODUCT ON LINE.P_CODE = PRODUCT.P_CODE;
```



CUS_CODE	INV_NUMBER	INV_DATE	P_DESCRIPT	LINE_UNITS	LINE_PRICE
10011	1002	2018-01-16 00:00:00	Rat-tail file, 1/8-in. fine	2.00	4.99
10011	1004	2018-01-17 00:00:00	Claw hammer	2.00	9.95
10011	1004	2018-01-17 00:00:00	Rat-tail file, 1/8-in. fine	3.00	4.99
10011	1008	2018-01-17 00:00:00	Claw hammer	1.00	9.95
10011	1008	2018-01-17 00:00:00	PVC pipe, 3.5-in., 8-ft	5.00	5.87
10011	1008	2018-01-17 00:00:00	Steel matting, 4'x8'x1/6", .5" mesh	3.00	119.95
10012	1003	2018-01-16 00:00:00	7.25-in. pwr. saw blade	5.00	14.99
10012	1003	2018-01-16 00:00:00	B&D cordless drill, 1/2-in.	1.00	38.95
10012	1003	2018-01-16 00:00:00	Hrd. cloth, 1/4-in., 2x50	1.00	39.95

Query 3: Improve the query in question 2 by displaying Cus_LName, Cus_FName instead of Cus_Code. Use proper, modern JOIN syntax to accomplish this. Add meaningful aliases to all columns.

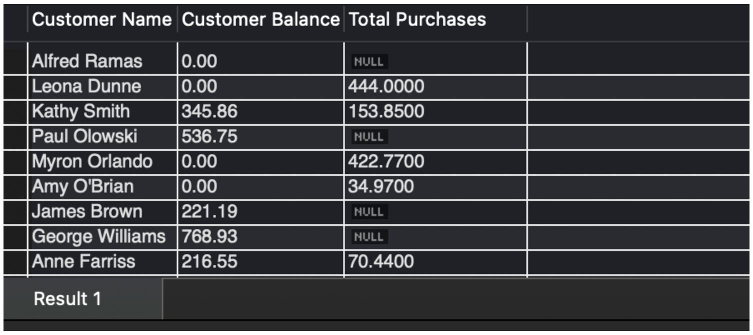
```
SELECT
        CONCAT(CUSTOMER.CUS_FNAME, ' ', CUSTOMER.CUS_LNAME) AS `Customer Name`,
        INVOICE.INV_NUMBER AS `Invoice Number`,
        PRODUCT.P_DESCRIPT AS `Product Description`,
        LINE.LINE_UNITS AS `Units Bought`,
        LINE.LINE_PRICE AS `Unit Price`,
        (LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Subtotal`
FROM CUSTOMER
JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
JOIN PRODUCT ON LINE.P_CODE = PRODUCT.P_CODE;
```



Customer Name	Invoice Number	Product Description	Units Bought	Unit Price	Subtotal
Leona Dunne	1002	Rat-tail file, 1/8-in. fine	2.00	4.99	9.9800
Leona Dunne	1004	Rat-tail file, 1/8-in. fine	3.00	4.99	14.9700
Leona Dunne	1004	Claw hammer	2.00	9.95	19.9000
Leona Dunne	1008	PVC pipe, 3.5-in., 8-ft	5.00	5.87	29.3500
Leona Dunne	1008	Steel matting, 4'x8'x1/6", .5" mesh	3.00	119.95	359.8500
Leona Dunne	1008	Claw hammer	1.00	9.95	9.9500
Kathy Smith	1003	B&D cordless drill, 1/2-in.	1.00	38.95	38.9500
Kathy Smith	1003	Hrd. cloth, 1/4-in., 2x50	1.00	39.95	39.9500
Kathy Smith	1003	7.25-in. pwr. saw blade	5.00	14.99	74.9500

Query 4: Write a query to display the Customer's Name (same format as above), the Cus_Balance and Total purchases for each customer. Total purchases is calculated by summing subtotals from problem 3.

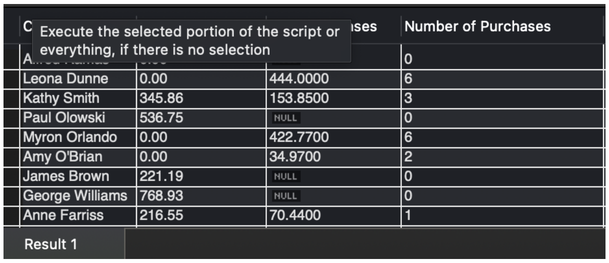
```
SELECT
        CONCAT(CUSTOMER.CUS_FNAME, ' ', CUSTOMER.CUS_LNAME) AS `Customer Name`,
        CUSTOMER.CUS_BALANCE AS `Customer Balance`,
        SUM(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Total Purchases`
FROM CUSTOMER
LEFT JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
LEFT JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
GROUP BY `Customer Name`, `Customer Balance`;
```



Customer Name	Customer Balance	Total Purchases
Alfred Ramas	0.00	NULL
Leona Dunne	0.00	444.0000
Kathy Smith	345.86	153.8500
Paul Olowski	536.75	NULL
Myron Orlando	0.00	422.7700
Amy O'Brian	0.00	34.9700
James Brown	221.19	NULL
George Williams	768.93	NULL
Anne Farriss	216.55	70.4400

Query 5: Add the number of total purchases to query 4.

```
SELECT
        CONCAT(CUSTOMER.CUS_FNAME, ' ', CUSTOMER.CUS_LNAME) AS `Customer Name`,
        CUSTOMER.CUS_BALANCE AS `Customer Balance`,
        SUM(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Total Purchases`,
        COUNT(INVOICE.INV_NUMBER) AS `Number of Purchases`
FROM CUSTOMER
LEFT JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
LEFT JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
GROUP BY `Customer Name`, `Customer Balance`;
```

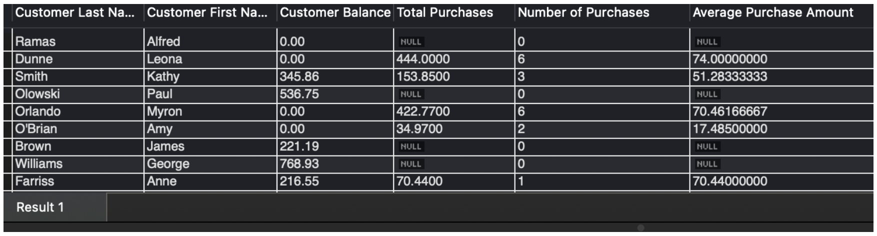


Execute the selected portion of the script or everything, if there is no selection

C		ses	Number of Purchases
A[illegible]	[illegible]	[illegible]	0
Leona Dunne	0.00	444.0000	6
Kathy Smith	345.86	153.8500	3
Paul Olowski	536.75	NULL	0
Myron Orlando	0.00	422.7700	6
Amy O'Brian	0.00	34.9700	2
James Brown	221.19	NULL	0
George Williams	768.93	NULL	0
Anne Farriss	216.55	70.4400	1

Query 6: Calculate the average purchase amount for each customer to query 5. Your column headers should be Customer Last Name, Customer First Name, Total Purchases, Number of Purchases, and Average Purchase Amount. The average purchase amount is calculated for each customer.

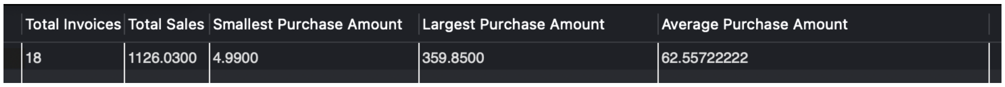
```
SELECT
        CUSTOMER.CUS_LNAME AS `Customer Last Name`,
        CUSTOMER.CUS_FNAME AS `Customer First Name`,
        CUSTOMER.CUS_BALANCE AS `Customer Balance`,
        SUM(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Total Purchases`,
        COUNT(INVOICE.INV_NUMBER) AS `Number of Purchases`,
        SUM(LINE.LINE_UNITS * LINE.LINE_PRICE) / COUNT(INVOICE.INV_NUMBER) AS `Average Purchase Amount`
FROM CUSTOMER
LEFT JOIN INVOICE ON CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
LEFT JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER
GROUP BY `Customer Last Name`, `Customer First Name`, `Customer Balance`;
```



Customer Last Na...	Customer First Na...	Customer Balance	Total Purchases	Number of Purchases	Average Purchase Amount
Ramas	Alfred	0.00	NULL	0	NULL
Dunne	Leona	0.00	444.0000	6	74.00000000
Smith	Kathy	345.86	153.8500	3	51.28333333
Olowski	Paul	536.75	NULL	0	NULL
Orlando	Myron	0.00	422.7700	6	70.46166667
O'Brian	Amy	0.00	34.9700	2	17.48500000
Brown	James	221.19	NULL	0	NULL
Williams	George	768.93	NULL	0	NULL
Farriss	Anne	216.55	70.4400	1	70.44000000

Query 7: Write a query that provides the total number of invoices, the sales total for all invoices, the smallest purchase amount, largest purchase amount, and average purchase amount. You should use aliases to make your output look reasonable for a business user.

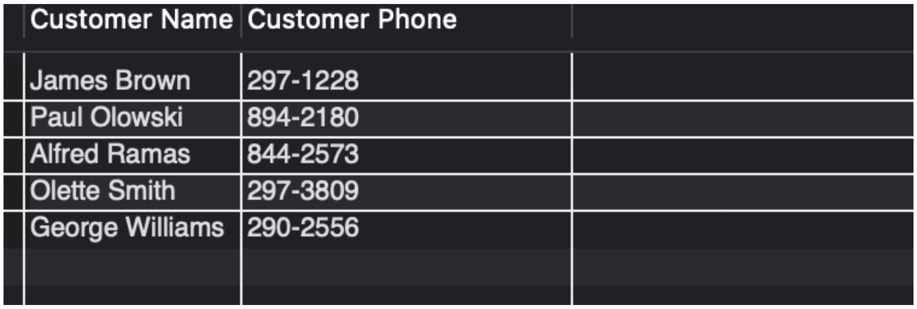
```
SELECT
        COUNT(INVOICE.INV_NUMBER) AS `Total Invoices`,
        SUM(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Total Sales`,
        MIN(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Smallest Purchase Amount`,
        MAX(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Largest Purchase Amount`,
        AVG(LINE.LINE_UNITS * LINE.LINE_PRICE) AS `Average Purchase Amount`
FROM INVOICE
LEFT JOIN LINE ON INVOICE.INV_NUMBER = LINE.INV_NUMBER;
```



Total Invoices	Total Sales	Smallest Purchase Amount	Largest Purchase Amount	Average Purchase Amount
18	1126.0300	4.9900	359.8500	62.55722222

Query 8: Advertising has requested a list of all customers who have never made a purchase, along with the customer's name and phone number. Provide them with a query with this information. Display the name in the style you believe would be most useful - briefly explain your choice in the comments.

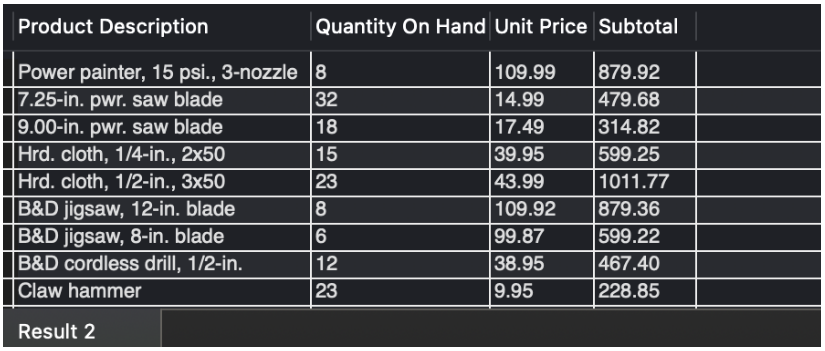
```
SELECT
        CONCAT(CUSTOMER.CUS_FNAME, ' ', CUSTOMER.CUS_LNAME) AS `Customer Name`,
        CUSTOMER.CUS_PHONE AS `Customer Phone`
FROM CUSTOMER
WHERE CUSTOMER.CUS_CODE NOT IN (SELECT DISTINCT CUS_CODE FROM INVOICE);
```



Customer Name	Customer Phone
James Brown	297-1228
Paul Olowski	894-2180
Alfred Ramas	844-2573
Olette Smith	297-3809
George Williams	290-2556

Query 9: Calculate the value of products currently in inventory. The value of the products can be calculated using P_QOH (quantity on hand) and P_Price. Display results with a subtotal for each different product. You should display the P_Descript, P_QOH, P_Price, and Subtotal for each product. Use aliases to improve the display.

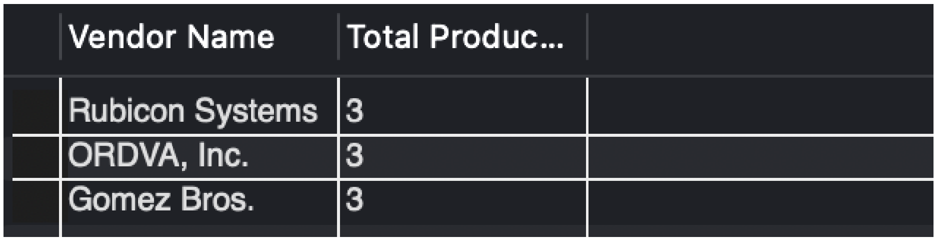
```
SELECT
        P_DESCRIPT AS `Product Description`,
        P_QOH AS `Quantity On Hand`,
        P_PRICE AS `Unit Price`,
        (P_QOH * P_PRICE) AS `Subtotal`
FROM PRODUCT;
```



Product Description	Quantity On Hand	Unit Price	Subtotal
Power painter, 15 psi., 3-nozzle	8	109.99	879.92
7.25-in. pwr. saw blade	32	14.99	479.68
9.00-in. pwr. saw blade	18	17.49	314.82
Hrd. cloth, 1/4-in., 2x50	15	39.95	599.25
Hrd. cloth, 1/2-in., 3x50	23	43.99	1011.77
B&D jigsaw, 12-in. blade	8	109.92	879.36
B&D jigsaw, 8-in. blade	6	99.87	599.22
B&D cordless drill, 1/2-in.	12	38.95	467.40
Claw hammer	23	9.95	228.85

Query 10:Which Vendor does this company use the most frequently to supply products? Display the Vendor's name with the most products currently in the Product Table as well as the total number of products for that particular vendor. This query should display correctly if there is a tie between two vendors and if the data in the product table changes (e.g. we begin ordering more products from a particular vendor).

```
SELECT
        V.V_NAME AS `Vendor Name`,
        COUNT(P.P_CODE) AS `Total Products`
FROM VENDOR V
LEFT JOIN PRODUCT P ON V.V_CODE = P.V_CODE
GROUP BY `Vendor Name`
ORDER BY `Total Products` DESC
LIMIT 3;
```



Vendor Name	Total Produc...
Rubicon Systems	3
ORDVA, Inc.	3
Gomez Bros.	3