Ch7 Problems

Zachary Sisco

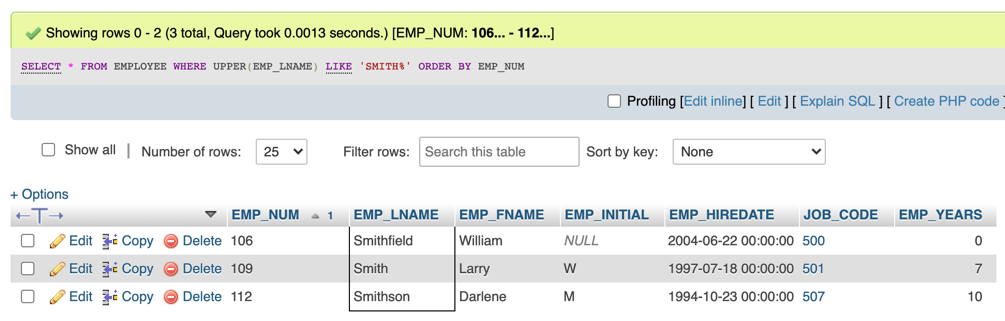
1. Write the SQL code required to list the employee number, last name, first name, and middle initial of all employees whose last names start with *Smith*. In other words, the rows for both Smith and Smithfield should be included in the listing. Sort the results by employee number. Assume case sensitivity.

SELECT \*

FROM EMPLOYEE

WHERE UPPER(EMP\_LNAME) LIKE 'SMITH%'

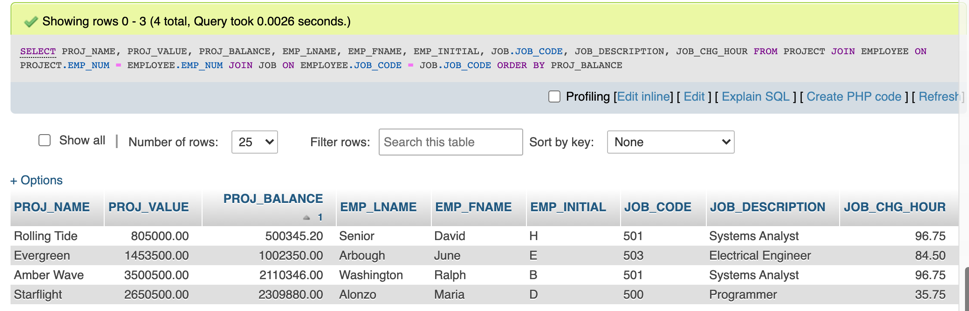
ORDER BY EMP\_NUM;



1. Using the EMPLOYEE, JOB, and PROJECT tables in the Ch07\_ConstructCo data- base, write the SQL code that will join the EMPLOYEE and PROJECT tables using EMP\_NUM as the common attribute. Display the attributes shown in the results presented in Figure P7.2, sorted by project value.

SELECT PROJ\_NAME, PROJ\_VALUE, PROJ\_BALANCE, EMP\_LNAME, EMP\_FNAME, EMP\_INITIAL, JOB.JOB\_CODE, JOB\_DESCRIPTION, JOB\_CHG\_HOUR

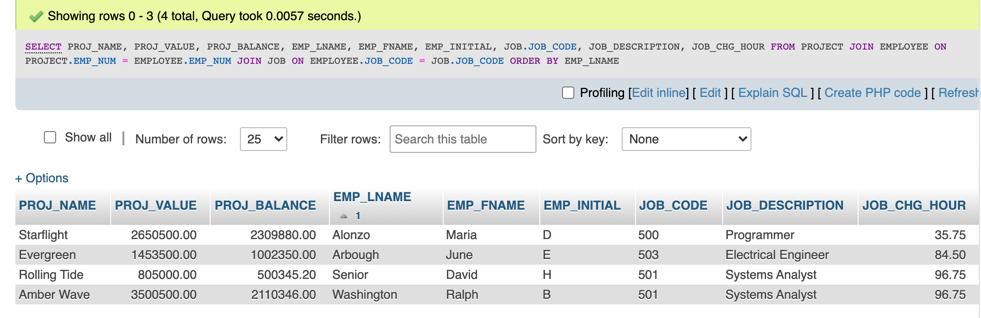
FROM PROJECT JOIN EMPLOYEE ON PROJECT.EMP\_NUM = EMPLOYEE.EMP\_NUM JOIN JOB ON EMPLOYEE.JOB\_CODE = JOB.JOB\_CODE

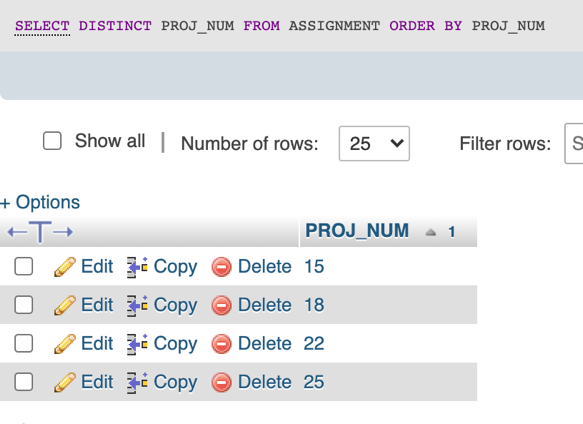
ORDER BY PROJ\_BALANCE;

1. Write the SQL code that will produce the same information that was shown in Problem 2, but sorted by the employee’s last name.

SELECT PROJ\_NAME, PROJ\_VALUE, PROJ\_BALANCE, EMP\_LNAME, EMP\_FNAME, EMP\_INITIAL, JOB.JOB\_CODE, JOB\_DESCRIPTION, JOB\_CHG\_HOUR

FROM PROJECT JOIN EMPLOYEE ON PROJECT.EMP\_NUM = EMPLOYEE.EMP\_NUM JOIN JOB ON EMPLOYEE.JOB\_CODE = JOB.JOB\_CODE

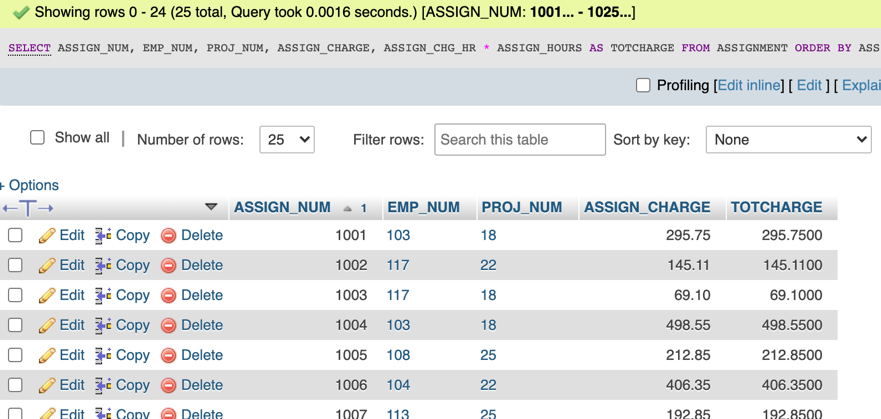
ORDER BY EMP\_LNAME;

1. Write the SQL code that will list only the distinct project numbers in the ASSIGN- MENT table, sorted by project number.

SELECT DISTINCT PROJ\_NUM

FROM ASSIGNMENT

ORDER BY PROJ\_NUM

1. Write the SQL code to validate the ASSIGN\_CHARGE values in the ASSIGN- MENT table. Your query should retrieve the assignment number, employee num- ber, project number, the stored assignment charge (ASSIGN\_CHARGE), and the calculated assignment charge (calculated by multiplying ASSIGN\_CHG\_HR by ASSIGN\_HOURS). Sort the results by the assignment number.

SELECT ASSIGN\_NUM, EMP\_NUM, PROJ\_NUM, ASSIGN\_CHARGE, ASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS TOTCHARGE

FROM ASSIGNMENT

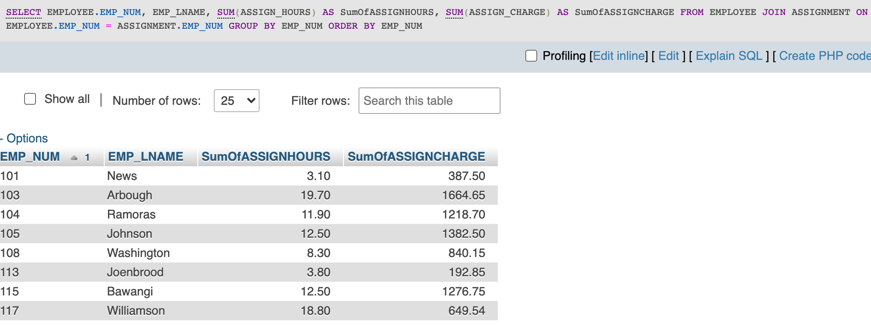
ORDER BY ASSIGN\_NUM;

1. Using the data in the ASSIGNMENT table, write the SQL code that will yield the total number of hours worked for each employee and the total charges stemming from those hours worked, sorted by employee number. The results of running that query are shown in Figure P7.6.

SELECT EMPLOYEE.EMP\_NUM, EMP\_LNAME, SUM(ASSIGN\_HOURS) AS SumOfASSIGNHOURS, SUM(ASSIGN\_CHARGE) AS SumOfASSIGNCHARGE

FROM EMPLOYEE JOIN ASSIGNMENT ON EMPLOYEE.EMP\_NUM = ASSIGNMENT.EMP\_NUM

GROUP BY EMP\_NUM

ORDER BY EMP\_NUM;

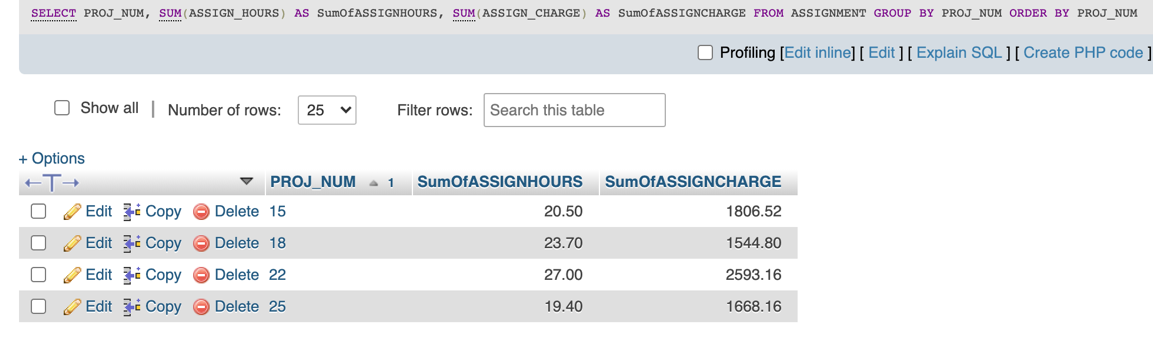
1. Write a query to produce the total number of hours and charges for each of the proj- ects represented in the ASSIGNMENT table, sorted by project number. The output is shown in Figure P7.7.

SELECT PROJ\_NUM, SUM(ASSIGN\_HOURS) AS SumOfASSIGNHOURS, SUM(ASSIGN\_CHARGE) AS SumOfASSIGNCHARGE

FROM ASSIGNMENT

GROUP BY PROJ\_NUM

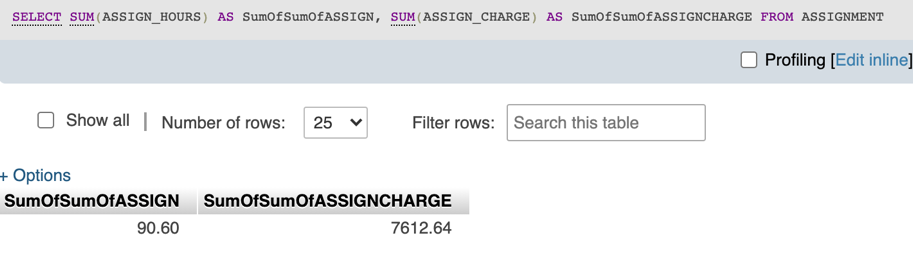
ORDER BY PROJ\_NUM;

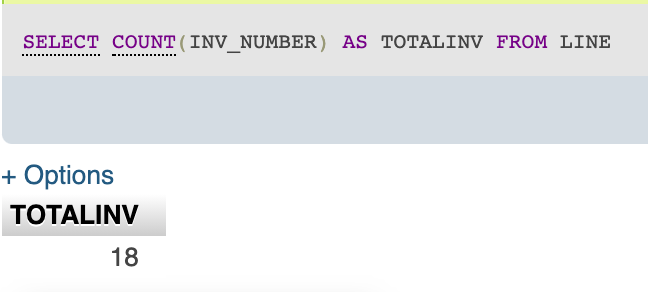


1. Write the SQL code to generate the total hours worked and the total charges made

by all employees. The results are shown in Figure P7.8.

SELECT SUM(ASSIGN\_HOURS) AS SumOfSumOfASSIGN, SUM(ASSIGN\_CHARGE) AS SumOfSumOfASSIGNCHARGE

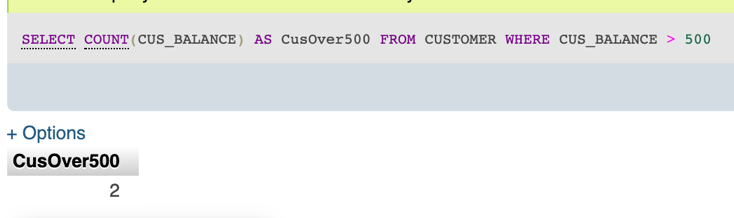
FROM ASSIGNMENT;

1. Write a query to count the number of invoices.

SELECT COUNT(INV\_NUMBER) AS TOTALINV

FROM LINE;

1. Write a query to count the number of customers with a balance of more than $500

SELECT COUNT(CUS\_BALANCE) AS CusOver500

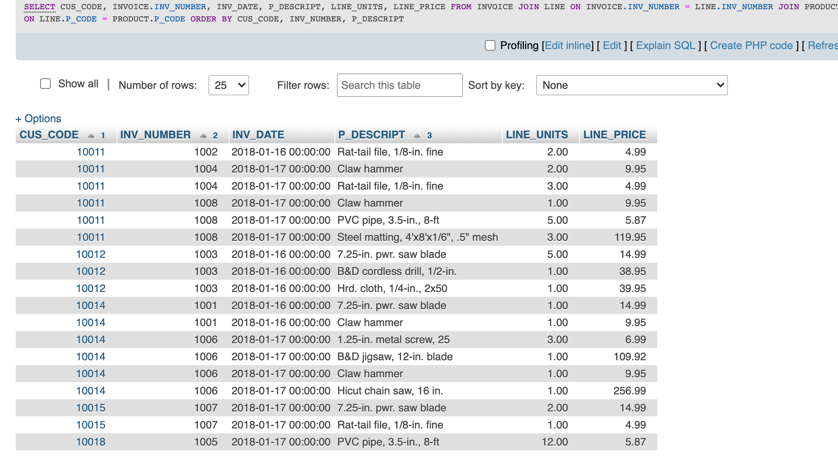
FROM CUSTOMER

WHERE CUS\_BALANCE > 500;

1. Generate a listing of all purchases made by the customers, using the output shown in Figure P7.11 as your guide. Sort the results by customer code, invoice number, and product description.

SELECT CUS\_CODE, INVOICE.INV\_NUMBER, INV\_DATE, P\_DESCRIPT, LINE\_UNITS, LINE\_PRICE

FROM INVOICE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER JOIN PRODUCT ON LINE.P\_CODE = PRODUCT.P\_CODE

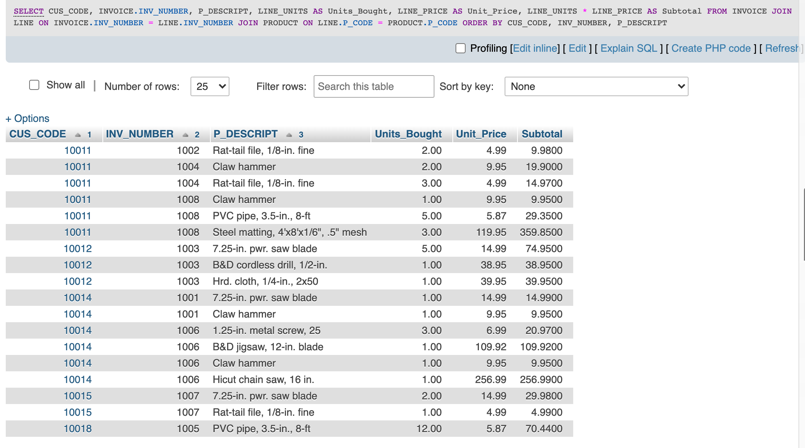
ORDER BY CUS\_CODE, INV\_NUMBER, P\_DESCRIPT;

1. Using the output shown in Figure P7.12 as your guide, generate a list of customer purchases, including the subtotals for each of the invoice line numbers. The subtotal is a derived attribute calculated by multiplying LINE\_UNITS by LINE\_PRICE. Sort the output by customer code, invoice number, and product description. Be certain to use the column aliases as shown in the figure.

SELECT CUS\_CODE, INVOICE.INV\_NUMBER, P\_DESCRIPT, LINE\_UNITS AS Units\_Bought, LINE\_PRICE AS Unit\_Price, LINE\_UNITS \* LINE\_PRICE AS Subtotal

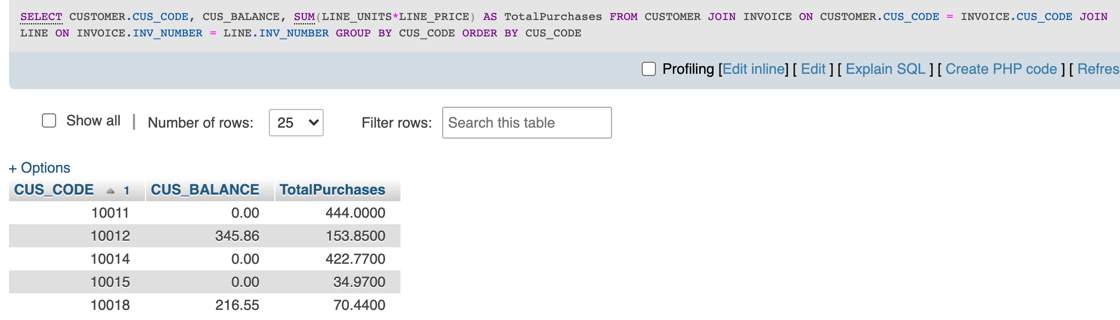
FROM INVOICE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER JOIN PRODUCT ON LINE.P\_CODE = PRODUCT.P\_CODE

ORDER BY CUS\_CODE, INV\_NUMBER, P\_DESCRIPT;



1. Write a query to display the customer code, balance, and total purchases for each customer. Total purchase is calculated by summing the line subtotals (as calculated in Problem 12) for each customer. Sort the results by customer code, and use aliases as shown in Figure P7.13.

SELECT CUSTOMER.CUS\_CODE, CUS\_BALANCE, SUM(LINE\_UNITS\*LINE\_PRICE) AS TotalPurchases

FROM CUSTOMER JOIN INVOICE ON CUSTOMER.CUS\_CODE = INVOICE.CUS\_CODE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY CUS\_CODE

ORDER BY CUS\_CODE;

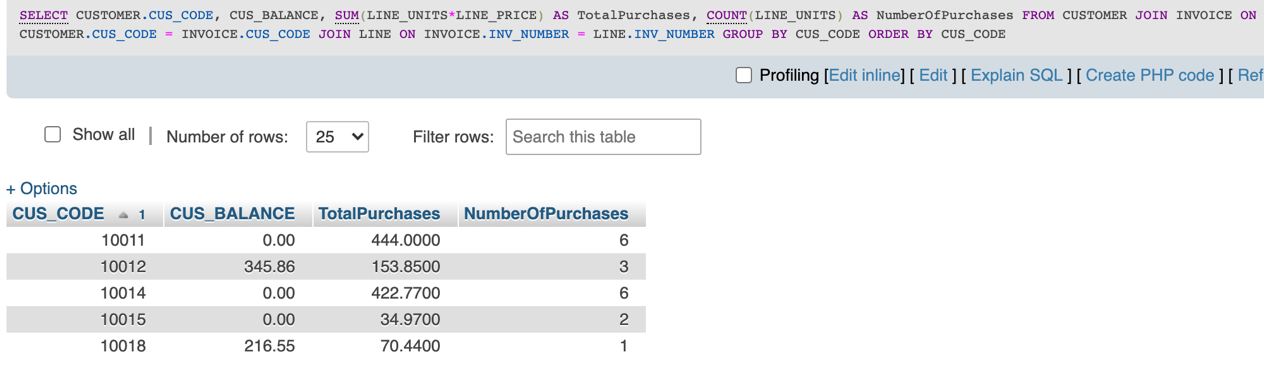
1. Modify the query in Problem 13 to include the number of individual product pur- chases made by each customer. (In other words, if the customer’s invoice is based on three products, one per LINE\_NUMBER, you count three product purchases. Note that in the original invoice data, customer 10011 generated three invoices, which contained a total of six lines, each representing a product purchase.) Your output values must match those shown in Figure P7.14, sorted by customer code.

SELECT CUSTOMER.CUS\_CODE, CUS\_BALANCE, SUM(LINE\_UNITS\*LINE\_PRICE) AS TotalPurchases, COUNT(LINE\_UNITS) AS NumberOfPurchases

FROM CUSTOMER JOIN INVOICE ON CUSTOMER.CUS\_CODE = INVOICE.CUS\_CODE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY CUS\_CODE

ORDER BY CUS\_CODE;

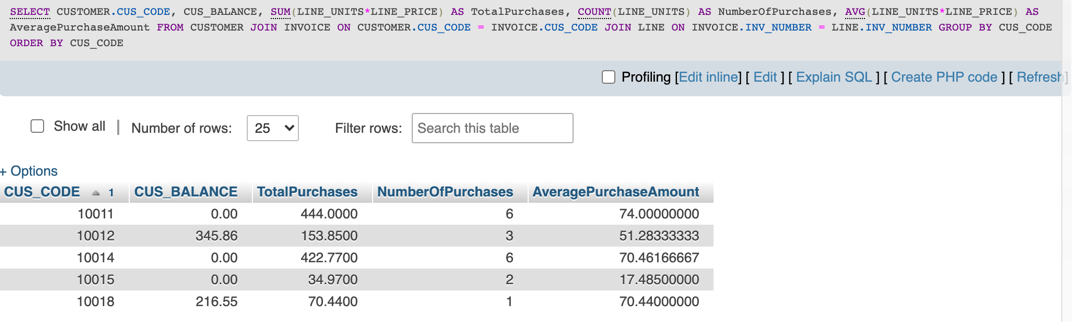


1. Use a query to compute the total of all purchases, the number of purchases, and the average purchase amount made by each customer. Your output values must match those shown in Figure P7.15. Sort the results by customer code.

SELECT CUSTOMER.CUS\_CODE, CUS\_BALANCE, SUM(LINE\_UNITS\*LINE\_PRICE) AS TotalPurchases, COUNT(LINE\_UNITS) AS NumberOfPurchases, AVG(LINE\_UNITS\*LINE\_PRICE) AS AveragePurchaseAmount

FROM CUSTOMER JOIN INVOICE ON CUSTOMER.CUS\_CODE = INVOICE.CUS\_CODE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY CUS\_CODE

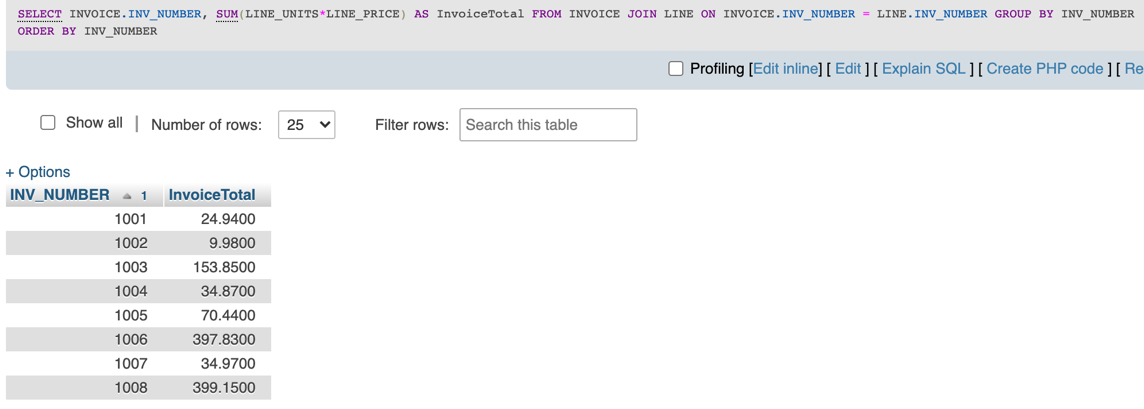
ORDER BY CUS\_CODE;

1. Create a query to produce the total purchase per invoice, generating the results shown in Figure P7.16, sorted by invoice number. The invoice total is the sum of the product purchases in the LINE that corresponds to the INVOICE.

SELECT INVOICE.INV\_NUMBER, SUM(LINE\_UNITS\*LINE\_PRICE) AS InvoiceTotal

FROM INVOICE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY INV\_NUMBER

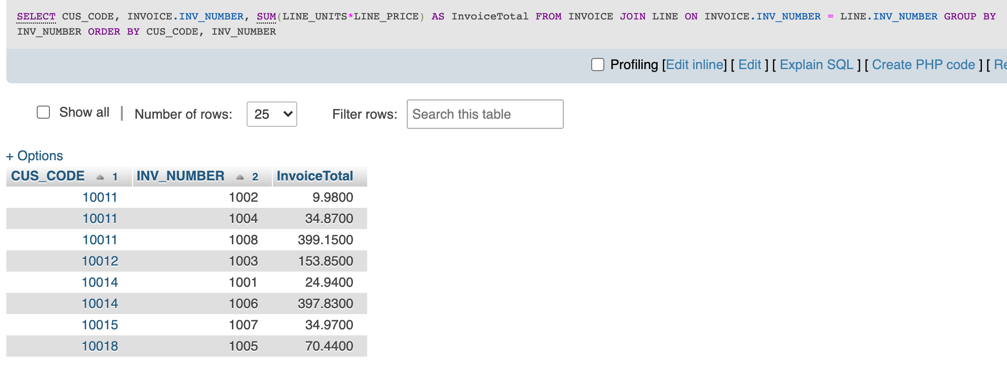
ORDER BY INV\_NUMBER;

1. Use a query to show the invoices and invoice totals in FigureP7.17. Sort the results by customer code and then by invoice number.

SELECT CUS\_CODE, INVOICE.INV\_NUMBER, SUM(LINE\_UNITS\*LINE\_PRICE) AS InvoiceTotal

FROM INVOICE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY INV\_NUMBER

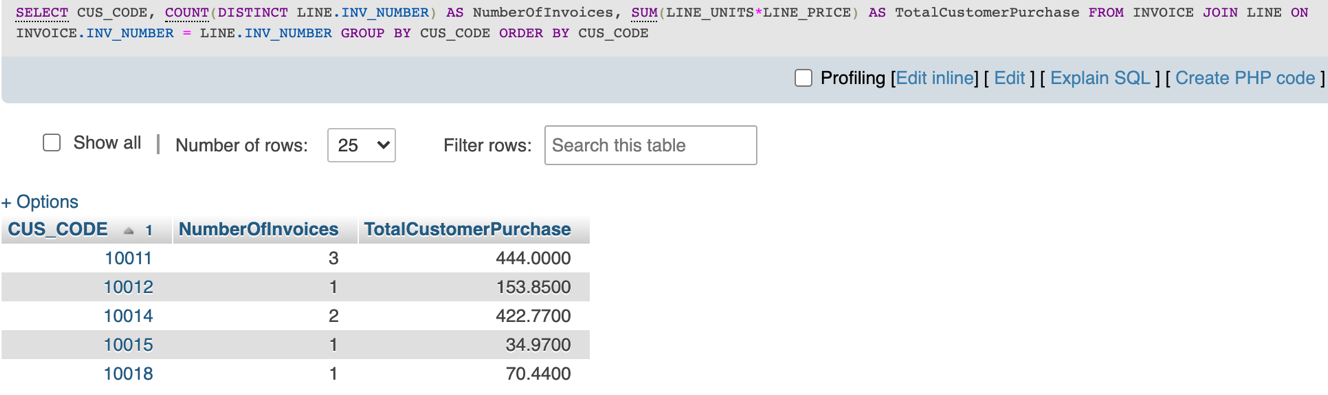
ORDER BY CUS\_CODE, INV\_NUMBER;

1. Write a query to produce the number of invoices and the total purchase amounts by customer, using the output shown in Figure P7.18 as your guide. Note the results are sorted by customer code. (Compare this summary to the results shown in Problem 17.)

SELECT CUS\_CODE, COUNT(DISTINCT LINE.INV\_NUMBER) AS NumberOfInvoices, SUM(LINE\_UNITS\*LINE\_PRICE) AS TotalCustomerPurchase

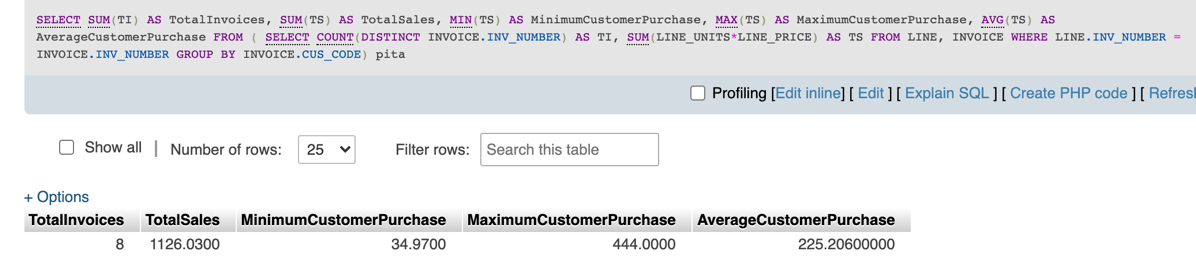
FROM INVOICE JOIN LINE ON INVOICE.INV\_NUMBER = LINE.INV\_NUMBER

GROUP BY CUS\_CODE

ORDER BY CUS\_CODE;

1. Write a query to generate the total number of invoices, the invoice total for all of the invoices, the smallest of the customer purchase amounts, the largest of the customer purchase amounts, and the average of all the customer purchase amounts. Your out- put must match Figure P7.19.

SELECT SUM(TI) AS TotalInvoices, SUM(TS) AS TotalSales, MIN(TS) AS MinimumCustomerPurchase, MAX(TS) AS MaximumCustomerPurchase, AVG(TS) AS AverageCustomerPurchase

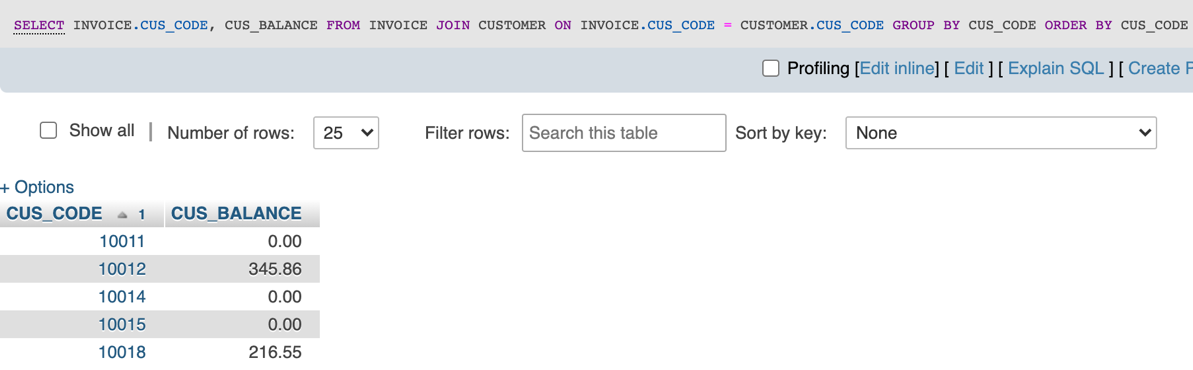
FROM ( SELECT COUNT(DISTINCT INVOICE.INV\_NUMBER) AS TI, SUM(LINE\_UNITS\*LINE\_PRICE) AS TS FROM LINE, INVOICE WHERE LINE.INV\_NUMBER = INVOICE.INV\_NUMBER GROUP BY INVOICE.CUS\_CODE) pita;

1. List the balances of customers who have made purchases during the current invoice cycle—that is, for the customers who appear in the INVOICE table. The results of this query are shown in Figure P7.20, sorted by customer code.

SELECT INVOICE.CUS\_CODE, CUS\_BALANCE

FROM INVOICE JOIN CUSTOMER ON INVOICE.CUS\_CODE = CUSTOMER.CUS\_CODE

GROUP BY CUS\_CODE

ORDER BY CUS\_CODE;