SQL Assignment

In [1]:
```
import sqlite3 as sql
import pandas as pd
```

Database # 1

In [2]:
```
database1 = 'ConstructCo.db'
conn = sql.connect(database1)
```

In [3]:
```
def read_query(q):
    return pd.read_sql_query(q, conn)
```

In [4]:
```
read_query('''select * from sqlite_master where type == "table"''')
```

Out[4]:

	type	name	tbl_name	rootpage	sql
0	table	JOB	JOB	2	CREATE TABLE JOB (\n JOB_CODE VARCHAR(...
1	table	EMPLOYEE	EMPLOYEE	4	CREATE TABLE EMPLOYEE (\n EMP_NUM VARCH...
2	table	PROJECT	PROJECT	6	CREATE TABLE PROJECT (\n PROJ_NUM VARCHA...
3	table	ASSIGNMENT	ASSIGNMENT	8	CREATE TABLE ASSIGNMENT (\n ASSIGN_NUM I...
4	table	EMP1	EMP1	10	CREATE TABLE EMP1 (\n EMP_NUM VARCHAR(3) PR...
5	table	EMP2	EMP2	12	CREATE TABLE EMP2(\n EMP_NUM TEXT,\n EMP_LNA...

In [5]:
```
read_query('select * from JOB').head()
```

Out[5]:

	JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR	JOB_LAST_UPDATE
0	500	Programmer	35.75	2021-11-20
1	501	Systems Analyst	96.75	2021-11-20
2	502	Database Designer	125.00	2022-3-24
3	503	Electrical Engineer	84.50	2021-11-20
4	504	Mechanical Engineer	67.90	2021-11-20

In [6]:
```
1 read_query('select * from EMPLOYEE').head()
```

Out[6]:

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE	EMP_Y
0	101	News	John	G	2004-11-8	502	
1	102	Senior	David	H	1993-7-12	501	
2	103	Arbough	June	E	2000-12-1	500	
3	104	Ramoras	Anne	K	1991-11-15	501	
4	105	Johnson	Alice	K	1997-2-1	502	

In [7]:
```
1 read_query('select * from PROJECT').head()
```

Out[7]:

	PROJ_NUM	PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_NUM
0	15	Evergreen	1453500	1002350.0	103
1	18	Amber Wave	3500500	2110346.0	108
2	22	Rolling Tide	805000	500345.2	102
3	25	Starflight	2650500	2309880.0	107

In [8]:
```
1 read_query('select * from ASSIGNMENT').head()
```

Out[8]:

	ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_JOB	ASSIGN_CHG_HR	AS
0	1001	2022-3-22	18	103	503	84.50	
1	1002	2022-3-22	22	117	509	34.55	
2	1003	2022-3-22	18	117	509	34.55	
3	1004	2022-3-22	18	103	503	84.50	
4	1005	2022-3-22	25	108	501	96.75	

P1.

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

In [9]:
```
1 read_query("""SELECT EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL
2 from EMPLOYEE
3 where EMP_LNAME LIKE 'Smith%'
4 order by EMP_NUM
5 """)
```

Out[9]:

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL
0	106	Smithfield	William	None
1	109	Smith	Larry	W
2	112	Smithson	Darlene	M

P2.

Using the EMPLOYEE, JOB, and PROJECT tables in the ConstructCo database, 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 2, sorted by project value.

In [114]:
```
read_query("""select p.PROJ_NAME, p.PROJ_VALUE, p.PROJ_BALANCE, e.EMP_L
from JOB j inner join EMPLOYEE e
on j.JOB_CODE = e.JOB_CODE
inner join PROJECT p
on e.EMP_NUM = p.EMP_NUM
order by p.PROJ_VALUE
""")
```

Out[114]:

	PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	J
0	Rolling Tide	805000	500345.2	Senior	David	H	
1	Evergreen	1453500	1002350.0	Arbough	June	E	
2	Starflight	2650500	2309880.0	Alonzo	Maria	D	
3	Amber Wave	3500500	2110346.0	Washington	Ralph	B	

P3.

Write the SQL code that will produce the same information that was shown in Problem P2., but sorted by the employee's last name.

In [11]:
```
read_query("""select p.PROJ_VALUE, p.PROJ_BALANCE, e.EMP_LNAME, e.EMP_F
from JOB j inner join EMPLOYEE e
on j.JOB_CODE = e.JOB_CODE
inner join PROJECT p
on e.EMP_NUM = p.EMP_NUM
order by e.EMP_LNAME
""")
```

Out[11]:

	PROJ_VALUE	PROJ_BALANCE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	JOB_CODE	JO
0	2650500	2309880.0	Alonzo	Maria	D	500	
1	1453500	1002350.0	Arbough	June	E	500	
2	805000	500345.2	Senior	David	H	501	
3	3500500	2110346.0	Washington	Ralph	B	501	

P4.

Write the SQL code that will list only the distinct project numbers in the ASSIGNMENT table, sorted by project number.

In [12]:
```
read_query("""select DISTINCT PROJ_NAME
from PROJECT
order by PROJ_NUM
""")
```

Out[12]:

	PROJ_NAME
0	Evergreen
1	Amber Wave
2	Rolling Tide
3	Starflight

P5.

Write the SQL code to validate the ASSIGN CHARGE values in the ASSIGNMENT table. Your query should retrieve the assignment number, employee number, project number, the stored assign□ment charge (ASSIGN CHARGE), and the calculated assignment charge (calculated by multiplying ASSIGN CHG HR by ASSIGN HOURS). Sort the results by the assignment number.

In [13]:
```
read_query("""select a.ASSIGN_NUM, e.EMP_NUM, p.PROJ_NUM, a.ASSIGN_CHAR
from EMPlOYEE e inner join PROJECT p
on e.EMP_NUM = p.EMP_NUM
inner join ASSIGNMENT a
on p.EMP_NUM = a.EMP_NUM
order by ASSIGN_NUM
""")
```

Out[13]:

	ASSIGN_NUM	EMP_NUM	PROJ_NUM	ASSIGN_CHARGE	CALCULATED ASSIGNMENT CHARGE
0	1001	103	15	295.75	295.75
1	1004	103	15	498.55	498.55
2	1005	108	18	212.85	212.85
3	1008	103	15	76.05	76.05
4	1012	108	18	328.95	328.95
5	1015	103	15	515.45	515.45
6	1021	108	18	298.35	298.35
7	1024	103	15	278.85	278.85

P6.

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 3.

In [115]:
```
read_query("""select DISTINCT a.EMP_NUM, e.EMP_LNAME, sum(a.ASSIGN_HOUR
from EMPLOYEE e inner join ASSIGNMENT a
on e.EMP_NUM = a.EMP_NUM
group by a.EMP_NUM
order by a.EMP_NUM
""")
```

Out[115]:

	EMP_NUM	EMP_LNAME	Sum of ASSIGN_HOURS	Sum of ASSIGN_CHARGE
0	101	News	3.1	387.50
1	103	Arbough	19.7	1664.65
2	104	Ramoras	11.9	1218.70
3	105	Johnson	12.5	1382.50
4	108	Washington	8.3	840.15
5	113	Joenbrood	3.8	192.85
6	115	Bawangi	12.5	1276.75
7	117	Williamson	18.8	649.54

P7.

Write a query to produce the total number of hours and charges for each of the projects represented in the ASSIGNMENT table, sorted by project number. The output is shown in Figure 4.

In [116]:
```
read_query("""select PROJ_NUM, sum(ASSIGN_HOURS) 'Sum of ASSIGN_HOURS',
from ASSIGNMENT
group by PROJ_NUM
order by PROJ_NUM
""")
```

Out[116]:

	PROJ_NUM	Sum of ASSIGN_HOURS	Sum of ASSIGN_CHARGE
0	15	20.5	1806.52
1	18	23.7	1544.80
2	22	27.0	2593.16
3	25	19.4	1668.16

P8.

Write the SQL code to generate the total hours worked and the total charges made by all employees. The results are shown in Figure 5.

In [117]:
```
read_query("""select sum(ASSIGN_HOURS) 'Sum of Sum of ASSIGN_HOURS', su
from ASSIGNMENT
""")
```

Out[117]:

	Sum of Sum of ASSIGN_HOURS	Sum of Sum of ASSIGN_CHARGE
0	90.6	7612.64

Database # 2

In [17]:
```
database2 = 'SaleCo.db'
conn2 = sql.connect(database2)
```

In [18]:
```
def read_query2(q):
    return pd.read_sql_query(q, conn2)
```

In [19]:
```
read_query2('''select * from sqlite_master where type == "table"''')
```

Out[19]:

	type	name	tbl_name	rootpage	sql
0	table	VENDOR	VENDOR	2	CREATE TABLE VENDOR (\nV_CODE \t\tINTEGER,\nV_...
1	table	PRODUCT	PRODUCT	3	CREATE TABLE PRODUCT (\nP_CODE \tVARCHAR(10) P...
2	table	CUSTOMER	CUSTOMER	5	CREATE TABLE CUSTOMER (\nCUS_CODE\tINTEGER PRI...
3	table	INVOICE	INVOICE	7	CREATE TABLE INVOICE (\nINV_NUMBER \tINTEG...
4	table	LINE	LINE	8	CREATE TABLE LINE (\nINV_NUMBER \tINTEGER NOT ...
5	table	EMPLOYEE	EMPLOYEE	11	CREATE TABLE EMPLOYEE (\nEMP_NUM\t\tINTEGER\tP...
6	table	EMP	EMP	12	CREATE TABLE EMP (\nEMP_NUM\t\tINTEGER\tPRIMAR...

In [20]:
```
read_query2('select * from VENDOR').head()
```

Out[20]:

	V_CODE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE	V_STATE	V_ORDER
0	21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
1	21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
2	21231	D&E Supply	Singh	615	228-3245	TN	Y
3	21344	Gomez Bros.	Ortega	615	889-2546	KY	N
4	22567	Dome Supply	Smith	901	678-1419	GA	N

In [21]:
```
read_query2('select * from PRODUCT').head()
```

Out[21]:

	P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
0	11QER/31	Power painter, 15 psi., 3-nozzle	2021-11-03	8	5	109.99	0.00	25595.0
1	13-Q2/P2	7.25-in. pwr. saw blade	2021-12-13	32	15	14.99	0.05	21344.0
2	14-Q1/L3	9.00-in. pwr. saw blade	2021-11-13	18	12	17.49	0.00	21344.0
3	1546-QQ2	Hrd. cloth, 1/4-in., 2x50	2022-01-15	15	8	39.95	0.00	23119.0
4	1558-QW1	Hrd. cloth, 1/2-in., 3x50	2022-01-15	23	5	43.99	0.00	23119.0

In [22]:
```
1 read_query2('select * from CUSTOMER')
```

Out[22]:

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CU
0	10010	Ramas	Alfred	A	615	844-2573	
1	10011	Dunne	Leona	K	713	894-1238	
2	10012	Smith	Kathy	W	615	894-2285	
3	10013	Olowski	Paul	F	615	894-2180	
4	10014	Orlando	Myron	None	615	222-1672	
5	10015	O'Brian	Amy	B	713	442-3381	
6	10016	Brown	James	G	615	297-1228	
7	10017	Williams	George	None	615	290-2556	
8	10018	Farriss	Anne	G	713	382-7185	
9	10019	Smith	Olette	K	615	297-3809	

In [23]:
```
1 read_query2('select * from INVOICE')
```

Out[23]:

	INV_NUMBER	CUS_CODE	INV_DATE
0	1001	10014	2022-01-16
1	1002	10011	2022-01-16
2	1003	10012	2022-01-16
3	1004	10011	2022-01-17
4	1005	10018	2022-01-17
5	1006	10014	2022-01-17
6	1007	10015	2022-01-17
7	1008	10011	2022-01-17

In [24]:
```
1 read_query2('select * from LINE')
```

Out[24]:

	INV_NUMBER	LINE_NUMBER	P_CODE	LINE_UNITS	LINE_PRICE
0	1001	1	13-Q2/P2	1	14.99
1	1001	2	23109-HB	1	9.95
2	1002	1	54778-2T	2	4.99
3	1003	1	2238/QPD	1	38.95
4	1003	2	1546-QQ2	1	39.95
5	1003	3	13-Q2/P2	5	14.99
6	1004	1	54778-2T	3	4.99
7	1004	2	23109-HB	2	9.95
8	1005	1	PVC23DRT	12	5.87
9	1006	1	SM-18277	3	6.99
10	1006	2	2232/QTY	1	109.92
11	1006	3	23109-HB	1	9.95
12	1006	4	89-WRE-Q	1	256.99
13	1007	1	13-Q2/P2	2	14.99
14	1007	2	54778-2T	1	4.99
15	1008	1	PVC23DRT	5	5.87
16	1008	2	WR3/TT3	3	119.95
17	1008	3	23109-HB	1	9.95

In [25]:
```
1 read_query2('select * from EMPLOYEE').head()
```

Out[25]:

	EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_D
0	100	Mr.	Kolmycz	George	D	1967-06-15	2010-0
1	101	Ms.	Lewis	Rhonda	G	1990-03-19	2011-0
2	102	Mr.	Vandam	Rhett	None	1983-11-14	2015-1
3	103	Ms.	Jones	Anne	M	1999-10-16	2019-0
4	104	Mr.	Lange	John	P	1996-11-08	2019-1

In [26]:
```
1 read_query2('select * from EMP').head()
```

Out[26]:

	EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_D
0	100	Mr.	Kolmycz	George	D	1967-06-15	2010-0
1	101	Ms.	Lewis	Rhonda	G	1990-03-19	2011-0
2	102	Mr.	Vandam	Rhett	None	1983-11-14	2015-1
3	103	Ms.	Jones	Anne	M	1999-10-16	2019-0
4	104	Mr.	Lange	John	P	1996-11-08	2019-1

P9.

Write a query to count the number of invoices.

In [27]:
```
1 read_query2('select count(INV_NUMBER) from INVOICE')
```

Out[27]:

	count(INV_NUMBER)
0	8

P10.

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

In [28]:
```
1 read_query2("select count(CUS_CODE) 'Number of Customers' from CUSTOMER
```

Out[28]:

	Number of Customers
0	2

P11.

Using the output shown in Figure 7 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.

In [119]:

```
read_query2("""select c.CUS_CODE, i.INV_NUMBER, p.P_DESCRIPT, l.LINE_UN
from CUSTOMER c inner join INVOICE i
on c.CUS_CODE = i.CUS_CODE
inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
inner join PRODUCT p
on l.P_CODE = p.P_CODE
order by c.CUS_CODE, i.INV_NUMBER, p.P_DESCRIPT
""")
```

Out[119]:

	CUS_CODE	INV_NUMBER	P_DESCRIPT	Units Bought	Units Price	Subtotal
0	10011	1002	Rat-tail file, 1/8-in. fine	2	4.99	9.98
1	10011	1004	Claw hammer	2	9.95	19.90
2	10011	1004	Rat-tail file, 1/8-in. fine	3	4.99	14.97
3	10011	1008	Claw hammer	1	9.95	9.95
4	10011	1008	PVC pipe, 3.5-in., 8-ft	5	5.87	29.35
5	10011	1008	Steel matting, 4'x8'x1/6", .5" mesh	3	119.95	359.85
6	10012	1003	7.25-in. pwr. saw blade	5	14.99	74.95
7	10012	1003	B&D cordless drill, 1/2-in.	1	38.95	38.95
8	10012	1003	Hrd. cloth, 1/4-in., 2x50	1	39.95	39.95
9	10014	1001	7.25-in. pwr. saw blade	1	14.99	14.99
10	10014	1001	Claw hammer	1	9.95	9.95
11	10014	1006	1.25-in. metal screw, 25	3	6.99	20.97
12	10014	1006	B&D jigsaw, 12-in. blade	1	109.92	109.92
13	10014	1006	Claw hammer	1	9.95	9.95
14	10014	1006	Hicut chain saw, 16 in.	1	256.99	256.99
15	10015	1007	7.25-in. pwr. saw blade	2	14.99	29.98
16	10015	1007	Rat-tail file, 1/8-in. fine	1	4.99	4.99
17	10018	1005	PVC pipe, 3.5-in., 8-ft	12	5.87	70.44

P12.

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 P11.) for each customer. Sort the results by customer code, and use aliases as shown in Figure 8.

In [30]:
```
read_query2("""select c.CUS_CODE, c.CUS_BALANCE, sum(l.LINE_PRICE * LIN
from CUSTOMER c inner join INVOICE i
on c.CUS_CODE = i.CUS_CODE
inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by c.CUS_CODE
order by c.CUS_CODE
""")
```

Out[30]:

	CUS_CODE	CUS_BALANCE	Total Purchase
0	10011	0.00	444.00
1	10012	345.86	153.85
2	10014	0.00	422.77
3	10015	0.00	34.97
4	10018	216.55	70.44

P13.

Modify the query in Problem P12. to include the number of individual product purchases 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 9, sorted by customer code.

In [31]:
```
read_query2("""select c.CUS_CODE, c.CUS_BALANCE, sum(l.LINE_PRICE * LIN
from CUSTOMER c inner join INVOICE i
on c.CUS_CODE = i.CUS_CODE
inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by c.CUS_CODE
order by c.CUS_CODE
""")
```

Out[31]:

	CUS_CODE	CUS_BALANCE	Total Purchase	Number of Purchase
0	10011	0.00	444.00	6
1	10012	345.86	153.85	3
2	10014	0.00	422.77	6
3	10015	0.00	34.97	2
4	10018	216.55	70.44	1

P14.

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 10. Sort the results by customer code.

In [32]:
```
read_query2("""select c.CUS_CODE, c.CUS_BALANCE, sum(l.LINE_PRICE * LIN
from CUSTOMER c inner join INVOICE i
on c.CUS_CODE = i.CUS_CODE
inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by c.CUS_CODE
order by c.CUS_CODE
""")
```

Out[32]:

	CUS_CODE	CUS_BALANCE	Total Purchase	Number of Purchase	Average Purchase Amount
0	10011	0.00	444.00	6	74.00
1	10012	345.86	153.85	3	51.28
2	10014	0.00	422.77	6	70.46
3	10015	0.00	34.97	2	17.49
4	10018	216.55	70.44	1	70.44

P15.

Create a query to produce the total purchase per invoice, generating the results shown in Figure 11, sorted by invoice number. The invoice total is the sum of the product purchases in the LINE that corresponds to the INVOICE.

In [120]:
```
read_query2("""select INV_NUMBER, sum(LINE_PRICE * lINE_UNITS) 'Invoice
from LINE
group by INV_NUMBER
order by INV_NUMBER
""")
```

Out[120]:

	INV_NUMBER	Invoice Total
0	1001	24.94
1	1002	9.98
2	1003	153.85
3	1004	34.87
4	1005	70.44
5	1006	397.83
6	1007	34.97
7	1008	399.15

P16.

Use a query to show the invoices and invoice totals in Figure 12. Sort the results by customer code and then by invoice number.

In [34]:
```
read_query2("""select i.CUS_CODE, l.INV_NUMBER, sum(l.LINE_PRICE * LINE
from INVOICE i inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by l.INV_NUMBER
order by 1, 2
""")
```

Out[34]:

	CUS_CODE	INV_NUMBER	INVOICE TOTAL
0	10011	1002	9.98
1	10011	1004	34.87
2	10011	1008	399.15
3	10012	1003	153.85
4	10014	1001	24.94
5	10014	1006	397.83
6	10015	1007	34.97
7	10018	1005	70.44

P17.

Write a query to produce the number of invoices and the total purchase amounts by customer, using the output shown in Figure 13 as your guide. Note the results are sorted by customer code. (Compare this summary to the results shown in Problem P16..)

In [122]:
```
read_query2("""select i.CUS_CODE, count(l.INV_NUMBER) 'No of Invoices',
from INVOICE i inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by i.CUS_CODE
order by i.CUS_CODE
""")
```

Out[122]:

	CUS_CODE	No of Invoices	Total Customer Purchase
0	10011	6	444.00
1	10012	3	153.85
2	10014	6	422.77
3	10015	2	34.97
4	10018	1	70.44

P18.

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 output must match Figure 14.

In [130]:
```
read_query2('''select
count(INV_NUMBER) 'Total Invoice',
sum(TOTAL_SALES) 'Total Sales',
min(TOTAL_SALES) 'Min Customer Purchase',
max(TOTAL_SALES) 'Largest Customer Purchase',
avg(TOTAL_SALES) 'Avg Customer Purchase'
from (select l.INV_NUMBER, sum(l.LINE_UNITS * l.LINE_PRICE) 'TOTAL_SALE
from CUSTOMER c inner join INVOICE i
on i.CUS_CODE = c.CUS_CODE
inner join LINE l
on i.INV_NUMBER = l.INV_NUMBER
group by c.CUS_CODE) 'InvoiceSales' ''')
```

Out[130]:

	Total Invoice	Total Sales	Min Customer Purchase	Largest Customer Purchase	Avg Customer Purchase
0	5	1126.03	34.97	444.0	225.206

P19.

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 15, sorted by customer code.

In [43]:
```
read_query2("""select DISTINCT i.CUS_CODE, c.CUS_BALANCE
from CUSTOMER c inner join INVOICE i
on c.CUS_CODE = i.CUS_CODE
order by 1
""")
```

Out[43]:

	CUS_CODE	CUS_BALANCE
0	10011	0.00
1	10012	345.86
2	10014	0.00
3	10015	0.00
4	10018	216.55

P20.

Provide a summary of customer balance characteristics for customers who made purchases. Include the minimum balance, maximum balance, and average balance, as shown in Figure 16.

In [110]:
```
read_query2('''select min(c.CUS_BALANCE) 'Minimum Balance',
max(c.CUS_BALANCE) 'Maximum Balance',
(sum(c.CUS_BALANCE)/COUNT(DISTINCT c.CUS_CODE)) 'Avg Balance'
from CUSTOMER c inner join INVOICE i
ON i.CUS_CODE = c.CUS_CODE
order by c.CUS_CODE''')
```

Out[110]:

	Minimum Balance	Maximum Balance	Avg Balance
0	0	345.86	112.482

P21.

Create a query to find the balance characteristics for all customers, including the total of the outstanding balances. The results of this query are shown in Figure 17.

In [56]:
```
read_query2("""select sum(CUS_BALANCE) 'Total Balance', min(CUS_BALANCE
from CUSTOMER
""")
```

Out[56]:

	Total Balance	Minimum Balance	Maximum Balance	Avg Balance
0	2089.28	0	768.93	208.928

P22.

Find the listing of customers who did not make purchases during the invoicing period. Sort the results by customer code. Your output must match the output shown in Figure 18.

In [112]:
```
read_query2('''select DISTINCT c.CUS_CODE, c.CUS_BALANCE
from CUSTOMER c LEFT JOIN INVOICE i
on c.CUS_CODE = i.CUS_CODE
where i.CUS_CODE is null
order by c.CUS_CODE''')
```

Out[112]:

	CUS_CODE	CUS_BALANCE
0	10010	0.00
1	10013	536.75
2	10016	221.19
3	10017	768.93
4	10019	0.00

P23.

Find the customer balance summary for all customers who have not made purchases during the current invoicing period. The results are shown in Figure 19.

In [113]:
```
read_query2('''select sum(c.CUS_BALANCE) 'Total Balance',
min(c.CUS_BALANCE) 'Minimum Balance' ,
max(c.CUS_BALANCE) 'Maximum Balance',
avg(c.CUS_BALANCE) 'Avg Balance'
from CUSTOMER c left join INVOICE i
on c.CUS_CODE = i.CUS_CODE
where i.CUS_CODE is null
order by c.CUS_CODE''')
```

Out[113]:

	Total Balance	Minimum Balance	Maximum Balance	Avg Balance
0	1526.87	0	768.93	305.374

P24.

Create a query that summarizes the value of products currently in inventory. Note that the value of each product is a result of multiplying the units currently in inventory by the unit price. Sort the results in descending order by subtotal, as shown in Figure 20.

In [95]:
```
read_query2("""select P_DESCRIPT, P_QOH, P_PRICE, P_QOH * P_PRICE 'Sub
from PRODUCT
order by 4 DESC
""")
```

Out[95]:

	P_DESCRIPT	P_QOH	P_PRICE	Sub Total
0	Hicut chain saw, 16 in.	11	256.99	2826.89
1	Steel matting, 4'x8'x1/6", .5" mesh	18	119.95	2159.10
2	2.5-in. wd. screw, 50	237	8.45	2002.65
3	1.25-in. metal screw, 25	172	6.99	1202.28
4	PVC pipe, 3.5-in., 8-ft	188	5.87	1103.56
5	Hrd. cloth, 1/2-in., 3x50	23	43.99	1011.77
6	Power painter, 15 psi., 3-nozzle	8	109.99	879.92
7	B&D jigsaw, 12-in. blade	8	109.92	879.36
8	Hrd. cloth, 1/4-in., 2x50	15	39.95	599.25
9	B&D jigsaw, 8-in. blade	6	99.87	599.22
10	7.25-in. pwr. saw blade	32	14.99	479.68
11	B&D cordless drill, 1/2-in.	12	38.95	467.40
12	9.00-in. pwr. saw blade	18	17.49	314.82
13	Claw hammer	23	9.95	228.85
14	Rat-tail file, 1/8-in. fine	43	4.99	214.57
15	Sledge hammer, 12 lb.	8	14.40	115.20

P25.

Find the total value of the product inventory. The results are shown in Figure 21.

In [98]:

```
read_query2("""select sum(P_QOH * P_PRICE) 'Total value of Inventory'
from PRODUCT
""")
```

Out[98]:

	Total value of Inventory
0	15084.52