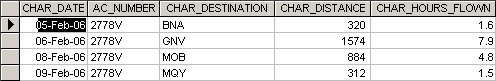
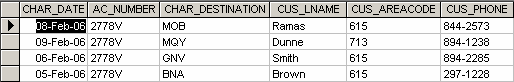
**Load the given Access database into your SQL Server account and finish the following questions. You will define all PKs and FKs. Turn in the following items both in print and electronically on BB:**

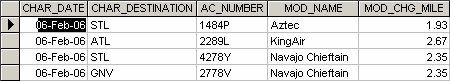
1. **A print out of the ERD from SQL Server.**
2. **A text file containing all the queries.**
3. **Write the SQL code that will list the values for the first four attributes in the CHARTER table.**
4. **Using the contents of the CHARTER table, write the SQL query that will produce the output shown in below. Note that the output is limited to selected attributes for aircraft number 2778V.**



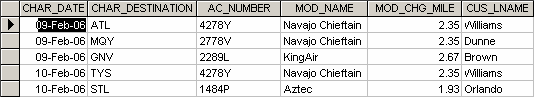
1. **Create a virtual table (named AC2778V) containing the output presented in Problem 2.**
2. **Produce the output shown in below for aircraft 2778V. Note that this output includes data from the CHARTER and CUSTOMER tables. (*Hint*: Use a JOIN in this query.)**



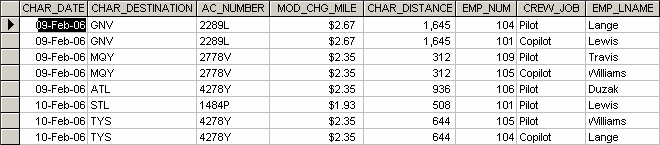
1. **Produce the output shown bewlow. The output, derived from the CHARTER and MODEL tables, is limited to February 6, 2004. (*Hint*: The JOIN passes through another table. Note that the “connection” between CHARTER and MODEL requires the existence of AIRCRAFT, because the CHARTER table does not contain a foreign key to MODEL. However, CHARTER does contain AC\_NUMBER, a foreign key to AIRCRAFT, which contains a foreign key to MODEL.)**



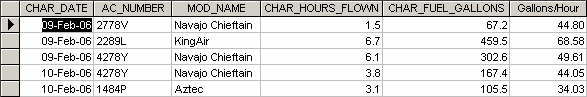
1. **Modify the query in Problem 5 to include data from the CUSTOMER table. This time the output is limited to charter records generated since February 9, 2006. (The query results are shown below.**



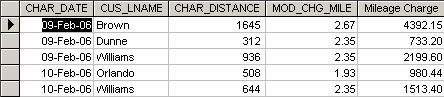
1. **Modify the query in Problem 6 to produce the output shown below. The date limitation in Problem 6 applies to this problem, too. Note that this query includes data from the CREW and EMPLOYEE tables. (*Note:* You may wonder why the date restriction seems to generate more records than it did in Problem 7. Actually, the number of (CHARTER) records is the same, but several records are listed twice to reflect a crew of two: a pilot and a copilot. For example, the record for the 09-Feb-2006 flight to GNV, using aircraft 2289L, required a crew consisting of a pilot (Lange) and a copilot (Lewis).)**



1. **Modify the query in Problem 5 to include the computed (derived) attribute “fuel per hour.” *Hint*: It is possible to use SQL to produce computed “attributes” that are not stored in any table. For example, the following SQL query is perfectly acceptable:**

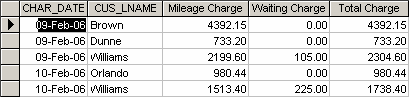


1. **Create a query to produce the output shown below. Note that, in this case, the computed attribute requires data found in two different tables. (*Hint*: The MODEL table contains the charge per mile, and the CHARTER table contains the total miles flown.) Note also that the output is limited to charter records generated since February 9, 2006. In addition, the output is ordered by date and, within the date, by the customer’s last name.**

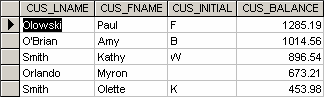


1. **Use the techniques that produced the output in Problem 9 to produce the charges shown below. The total charge to the customer is computed by:**
   1. **Miles flown \* charge per mile**
   2. **Hours waited \* $50 per hour**

**The miles flown (CHAR\_DISTANCE) value is found in the CHARTER table, the charge per mile (MOD\_CHG\_MILE) is found in the MODEL table, and the hours waited (CHAR\_HOURS\_WAIT) are found in the CHARTER table.**



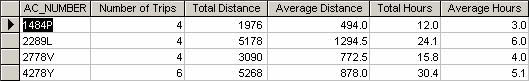
1. **Create the SQL query that will produce a list of customers who have an unpaid balance. The required output is shown below. Note that the balances are listed in descending order.**



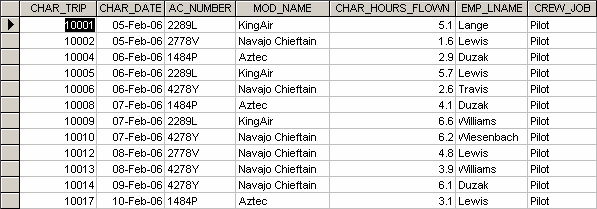
1. **Find the average unpaid customer balance, the minimum balance, the maximum balance, and the total of the unpaid balances. The resulting values are shown below.**

FigP7-12-Customer-Balance-Summary

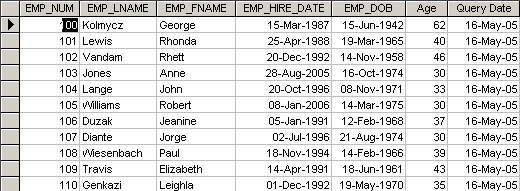
1. **Using the CHARTER table as the source, group the aircraft data. Then use the SQL functions to produce the output shown below. (Utility software was used to modify the headers, so your headers may look different.)**



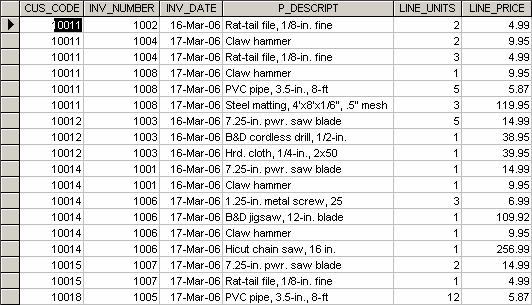
1. **Write the SQL code to generate the output shown below. Note that the listing includes all CHARTER flights that did not include a copilot crew assignment. (*Hint*: The crew assignments are listed in the CREW table. Also note that the pilot’s last name requires access to the EMPLOYEE table, while the MOD\_CODE requires access to the MODEL table.)**



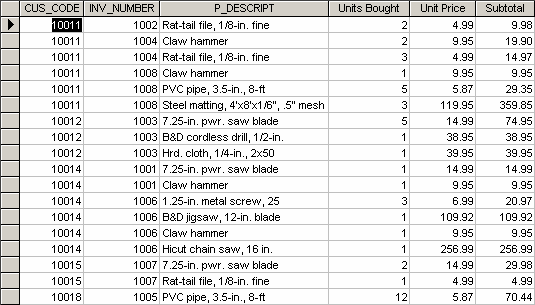
1. **Write a query that will list the ages of the employee and the date the query was run. The required output is shown below. (As you can tell, the query was run on May 16, 2005, so the ages of the employee are current as of that date.)**



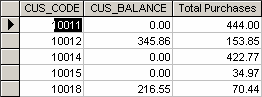
1. **Write a query to count the number of invoices.**
2. **Write a query to count the number of customers with a customer balance over $500.**
3. **Generate a listing of all purchases made by the customers, using the output shown below as your guide. (*Hint*: Use the ORDER BY clause to order the resulting rows as shown below)**



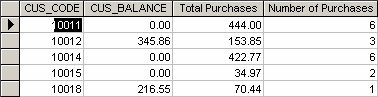
1. **Using the output shown below as your guide, generate the listing of customer purchases, including the subtotals for each of the invoice line numbers. (*Hint*: Modify the query format used to produce the listing of customer purchases in Problem 18, delete the INV\_DATE column, and add the derived (computed) attribute LINE\_UNITS \* LINE\_PRICE to calculate the subtotals.)**



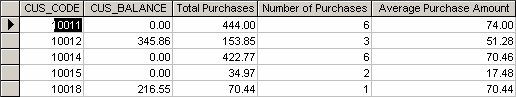
1. **Modify the query used in Problem 19 to produce the summary shown below.**



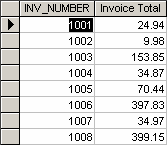
1. **Modify the query in Problem 20 to include the number of individual product purchases made by each customer. (In other words, if the customer’s invoice is based on three products, one per LINE\_NUMBER, you would count three product purchases. If you examine the original invoice data, you will note that customer 10011 generated three invoices, which contained a total of six lines, each representing a product purchase.) Your output values must match those shown below.**



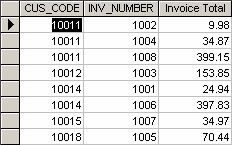
1. **Use a query to compute the average purchase amount per product made by each customer. (*Hint:* Use the results of Problem 21 as the basis for this query.) Your output values must match those shown below. Note that the Average Purchase Amount is equal to the Total Purchases divided by the Number of Purchases.**



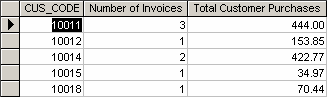
1. **Create a query to produce the total purchase per invoice, generating the results shown below. The Invoice Total is the sum of the product purchases in the LINE that corresponds to the INVOICE.**



1. **Use a query to show the invoices and invoice totals as shown below. (*Hint*: Group by the CUS\_CODE.)**



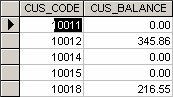
1. **Write a query to produce the number of invoices and the total purchase amounts by customer, using the output shown below as your guide. (Compare this summary to the results shown in Problem 24.)**



1. **Using the query results in Problem 25 as your basis, write a query to generate the total number of invoices, the invoice total for all of the invoices, the smallest invoice amount, the largest invoice amount, and the average of all of the invoices. (*Hint*: Check the figure output in Problem 25.) Your output must match below.**

FigP7-26-No-of-Invoices-Total-Min-Max-and-Ave-Sales

1. **List the balance characteristics of the 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 as shown below.**



1. **Using the results of the query created in Problem 27, provide a summary of customer balance characteristics as shown below.**

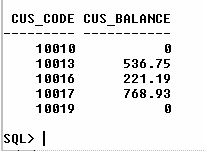
FigP7-28-Balance-Summ-for-Cust-Who-Made-Purchases

1. **Create a query to find the customer balance characteristics for all customers, including the total of the outstanding balances. The results of this query are shown below.**

FigP7-29-Customer-Balance-Summary-for-All-Customers

1. **Find the listing of customers who did not make purchases during the invoicing period. Your output must match the output shown in Figure P7.30.**

**FIGURE P7.30 Customer Balances for Customers Who Did Not Make Purchases**



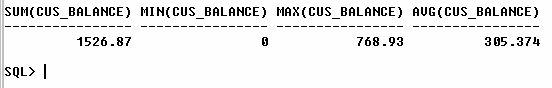
SELECT CUS\_CODE, CUS\_BALANCE

FROM CUSTOMER

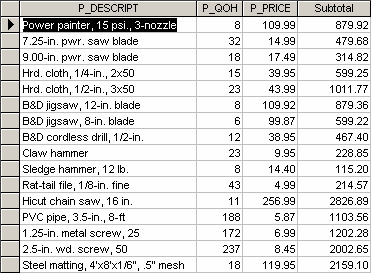
WHERE CUSTOMER.CUS\_CODE NOT IN

(SELECT DISTINCT CUS\_CODE FROM INVOICE);

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



1. **Create a query to produce the summary of the value of products currently in inventory. Note that the value of each product is produced by the multiplication of the units currently in inventory and the unit price. Use the ORDER BY clause to match the order shown below.**



1. **Using the results of the query created in Problem 32, find the total value of the product inventory. The results are shown below.**

FigP7-33-Total-Value-of-all-Products-in-Inventory