Module

3

Single Table SELECT Statements

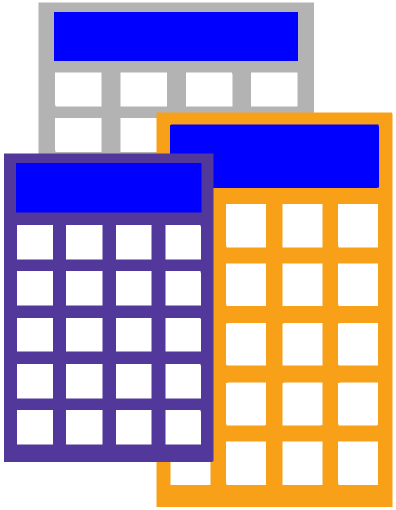
Objectives

At the end of this module, you will be able to:

- Write a single table SELECT statement
- List the optional clauses of a SELECT statement
- Use the optional clauses in a SELECT statement
- Use aggregate functions in a SELECT statement

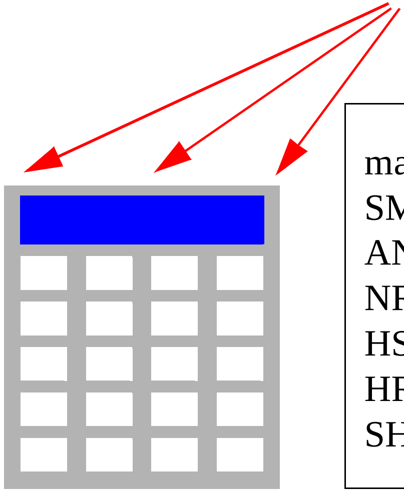
SELECT Statement Clauses

```
SELECT select-list
    FROM table-name
    [WHERE condition]
    [GROUP BY column-list]
    [HAVING condition]
    [ORDER BY column-name]
    [INTO TEMP table-name]
```



Selecting all Columns

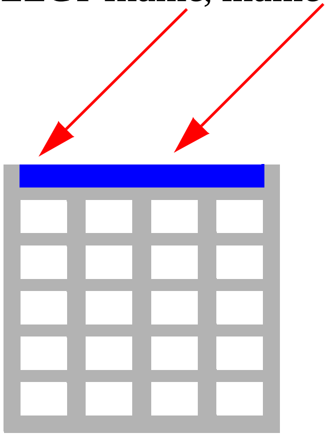
SELECT * FROM manufact



```
manu_code      manu_name lead_time
SMT            Smith            3
ANZ            Anza             5
NRG            Norge            7
HSK            Husky            5
HRO            Hero             4
SHM            Shimara         30
```

Selecting Specific Columns

SELECT **fname**, **lname** FROM customer

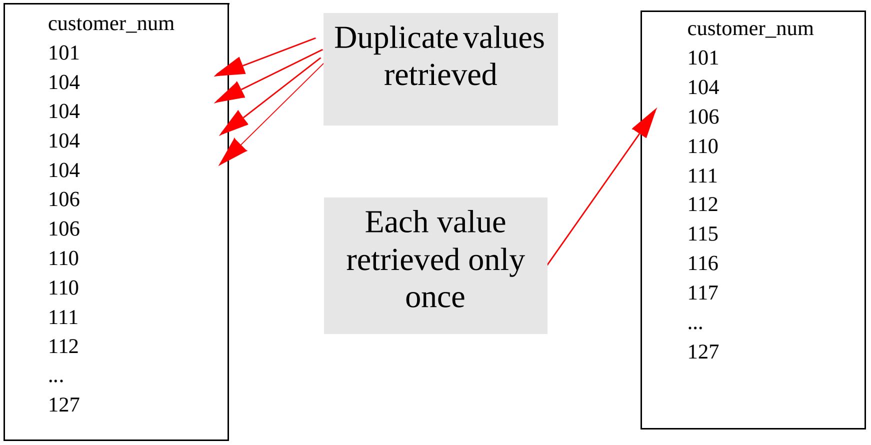


fname	lname
Ludwig	Pauli
Carole	Sadler
Philip	Currie
Anthony	Higgins
Raymond	Vector
George	Watson
Charles	Ream

Selecting Unique Values

```
SELECT customer_num
    FROM orders
```

```
SELECT DISTINCT customer_num
    FROM orders
```



customer_num
101
104
104
104
104
106
106
110
110
111
112
...
127

Duplicate values retrieved

Each value retrieved only once

customer_num
101
104
106
110
111
112
115
116
117
...
127

Executing Multiple Statements

Use a semi-colon (;) to separate multiple SQL statements.

```
SELECT *
      FROM manufact;

SELECT fname,lname,company
      FROM customer;

SELECT phone,customer_num FROM
customer;
```

The WHERE Clause

```
SELECT select-list
FROM table-name
[WHERE condition]
```

Use the WHERE clause to SELECT specific rows

Including or Excluding Rows

```
SELECT    stock_num, manu_code, description, unit
    FROM  stock
    WHERE unit = 'case';
```

stock_num	manu_code	description	unit
1	HRO	baseball gloves	case
1	HSK	baseball gloves	case
...	...	...	...
310	ANZ	kick board	case

```
SELECT    stock_num, manu_code, description, unit
    FROM  stock
    WHERE unit != 'case'
```

stock_num	manu_code	description	unit
5	NRG	tennis racquet	each
5	SMT	tennis racquet	each
...	...	...	...
313	ANZ	swim cap	box

Relational Operators

=	equals
!= or <>	does not equal
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Identifying NULL Values

```
SELECT lname, phone
    FROM customer
    WHERE address2 IS NULL;
```

lname	phone
Pauli	408-789-8075
Sadler	415-822-1289
Vector	415-776-3249
...	...
Neelie	303-936-7731

```
SELECT lname, phone
    FROM customer
    WHERE address2 IS NOT NULL
```

lname	phone
Currie	415-328-4543
Higgins	415-368-1100
Miller	408-723-8789
...	...
Lessor	602-533-1817

WHERE Clause Keywords

AND

OR

[NOT] BETWEEN

[NOT] IN

IS [NOT] NULL

[NOT] LIKE

[NOT] MATCHES

Combining Comparison Conditions

```
SELECT fname, lname
    FROM customer
    WHERE city = "Los Altos" AND state = "CA";
SELECT fname, lname
    FROM customer
    WHERE state = "CA" OR state = "AZ";
SELECT fname, lname
    FROm customer
    WHERE state = "CA"
    AND city = "Los Altos" OR state = "AZ";
```

Finding a Range of Values

```
SELECT stock_num, manu_code, description, unit_price
    FROM stock
    WHERE unit_price BETWEEN 20.00 AND 30.00
```

```
stock_num    manu_code    description          unit_price
        5    NRG          tennis racquet           $28.00
        5    SMT          tennis racquet           $25.00
        9    ANZ          volleyball net           $20.00
      103    PRC          frnt derailleur          $20.00
      106    PRC          bicycle stem             $23.00
      109    PRC          pedal binding            $30.00
```

Finding a Subset of Values

```
SELECT customer_num, lname, fname, company
    FROM customer
    WHERE customer_num IN (118,114,106,101,127)
```

```
customer_num   lname        fname     company
         118   Baxter       Dick      Blue Ribbon Sports
         114   Albertson    Frank     Sporting Place
         106   Watson       George    Watson & Son
         101   Pauli        Ludwig    All Sports Supplies
         127   Satifer      Kim       Big Blue Bike Shop
```

Character Search Operators

LIKE	MATCHES	Meaning
%	*	Evaluates to zero or more characters
_	?	Evaluates to a single character
\	\	Specifies the next character as a literal character
	[ ]	Specifies valid values for a single character

Variable Length Wildcard

```
SELECT customer_num, company
    FROM customer
    WHERE company MATCHES '*Sports'
```

```
customer_num              company
         103              Phil's Sports
         105              Los Altos Sports
         108              Quinn's Sports
         115              Gold Medal Sports
         118              Blue Ribbon Sports
         121              City Sports
         123              Bay Sports
         125              Total Fitness Sports
```

Single Character Wildcard

```
SELECT customer_num, company
    FROM customer
    WHERE company MATCHES '?l*'
```

```
customer_num              company
         101              All Sports Supplies
         104              Play Ball!
         116              Olympic City
         118              Blue Ribbon Sports
```

Restricted Single Character Wildcard

```
SELECT *
    FROM manufact
    WHERE manu_name MATCHES '[A-N]*'
```

manu_code	manu_name	lead_time
ANZ	Anza	5
NRG	Norge	7
HSK	Husky	5
HRO	Hero	4

```
SELECT *
    FROM manufact
    WHERE manu_name MATCHES '[AN]*'
```

manu_code	manu_name	lead_time
ANZ	Anza	5
NRG	Norge	7
NKL	Nikolus	8

Comparing for Special Characters

```
SELECT *
    FROM cust_calls
    WHERE res_descr LIKE '%\%%'
```

The escape character lets the middle **%** be interpreted as a percent sign, not a wildcard.

```
customer_num 116
call_dtime    1990-12-21 11:24
user_id       mannyn
call_code     I
call_descr    Second complaint from this customer! Received two cases
              right-handed outfielder gloves (1 HRO) instead of one
              case lefties.
res_dtime     1990-12-27 08:19
res_descr     Memo to shipping (Ava Brown) to send case of left-
              handed gloves, pick up wrong case; memo to billing
              requesting 5% discount to placate customer due to
              second offense and lateness of resolution because of
              holiday
```

The ORDER BY Clause

```
SELECT select-list
    FROM table-name
    [WHERE condition]
     ...
    [ORDER BY column-name]
```

ORDER BY Example

```
SELECT stock_num, manu_code, description, unit_price
    FROM stock
    ORDER BY description, unit_price DESC
```

stock_num	manu_code	description	unit_price
111	SHM	10-spd, assmbld	$499.99
112	SHM	12-spd, assmbld	$549.00
113	SHM	18-spd, assmbld	$685.90
205	ANZ	3 golf balls	$312.00
205	NKL	3 golf balls	$312.00
205	HRO	3 golf balls	$312.00
2	HRO	baseball	$126.00
3	SHM	baseball bat	$280.00
3	HSK	baseball bat	$240.00
...	...	...	...
311	SHM	water gloves	$48.00

Lab Exercise

Lab Exercise

Exercise 1

What SQL statements would you use to retrieve the information requested below? Use the tool indicated by your instructor to enter and execute the SQL statements against the demonstration database that you created earlier.

1. Sam, the owner of your company, would like a list of all his customers' names and addresses.

2. What if he only wants the customers who live in California?

3. Now he wants a list of the towns in California where his customers live. He only wants each town to appear once.

4. Can you sort the list in reverse alphabetical order (descending order) for him?

5. Shipping wants to know the address of customer number 103.

6. What products from the manufacturer ANZ does Sam stock? Can you give him the list in alphabetical order by description?

7. Sam would like a list of the manufacturer's codes for the items that have been ordered by any customer. He wants it sorted in alphabetical order, and each code should only appear once.

8. A customer left a message and you've lost the note. All you can remember is that the company name had *Medal* in it. Can you find the phone number?

9. One of Sam's customers wants to do a special promotion and giveaway. Do we have any bicycle products in stock that cost between $50 and $75?

10. It appears that there may be a bottleneck in the shipping department. Can you give Sam a list of all the orders that have not been shipped yet?

11. Sam is planning a mass mailing to his customers. He's going to start with those customers whose last names begin with *A* through *G*. Can you prepare a list for him? He'd like it sorted by state and city.

12. Sam is having trouble remembering the company name of one of his customers. He knows that the name has the word *town* in it somewhere, but he is not sure where. He also isn't sure whether the *t* in *town* is upper or lower case.

13. Sam would like to see a list of all of his California and Florida customers whose company name end in *Sports*. He is only interested in the company name, city, state, and zip code. He would like the list sorted alphabetically by state. Within each state, he wants the list sorted by company name in descending order.

Lab
Exercise

Lab Exercise

Challenge exercise:

Roy Jaeger called. He thinks he was charged the wrong amount on order #1008. He plans to call back to discuss the details of his order. Pull up the order's line items and the price charged for each line item, in preparation for the call.

Arithmetic Expressions

```
SELECT stock_num, manu_code, description,
    unit_price, unit_price * 1.05
    FROM stock
    ORDER BY description, unit_price desc
```

```
stock_num   manu_code   description          unit_price   (expression)


      111   SHM         10-spd, assmbld       $499.99      $524.99
      112   SHM         12-spd, assmbld       $549.00      $576.45
      113   SHM         18-spd, assmbld       $685.90      $720.20
      205   ANZ         3 golf balls          $312.00      $327.60
      205   NKL         3 golf balls          $312.00      $327.60
      205   HRO         3 golf balls          $312.00      $327.60
        2   HRO         baseball              $126.00      $132.30
      ...   ...         ...                   ...                         ...
      304   ANZ         watch                 $170.00      $178.50
      311   SHM         water gloves          $48.00       $50.40
```

The ROUND and TRUNC Functions

```
SELECT stock_num, manu_code,
    unit_price * 1.05,
    ROUND (unit_price * 1.05, 1),
    TRUNC (unit_price * 1.05, 1)
    FROM stock
```

```
stock_num  manu_code     (expression) (expression) (expression)

        1  HRO                $262.50        262.5        262.5
        1  HSK                $840.00        840.0        840.0
        1  SMT                $472.50        472.5        472.5
        2  HRO                $132.30        132.3        132.3
        3  HSK                $252.00        252.0        252.0
        3  SHM                $294.00        294.0        294.0
        4  HSK               $1008.00       1008.0       1008.0
      ...  ...                    ...          ...          ...
```

Display Labels

```
SELECT      stock_num, manu_code, description,
            unit_price,unit_price * 1.05 new_price
   FROM     stock
   ORDER BY description, new_price desc
```

```
stock_num    manu_code    description              unit_price        new_price
   111       SHM          10-spd, assmbld              $499.99       $524.9895
   112       SHM          12-spd, assmbld              $549.00       $576.4500
   113       SHM          18-spd, assmbld              $685.90       $720.1950
   205       ANZ          3 golf balls                 $312.00       $327.6000
   205       NKL          3 golf balls                 $312.00       $327.6000
   205       HRO          3 golf balls                 $312.00       $327.6000
    2        HRO          baseball                     $126.00       $132.3000
   ...       ...               ...                           ...     ...
   311       SHM          water gloves                  48.00         $50.4000
```

Aggregate Functions

COUNT (*)
COUNT (DISTINCT column-name)

SUM (column/expression)
SUM (DISTINCT column-name)

AVG (column/expression)
AVG (DISTINCT column-name)

MAX (column/expression)

MIN (column/expression)

The COUNT Function

Given a *hypothetical* subset of the **stock** table:

1	HRO	baseball gloves	$250.00	case	10 gloves/case
1	HSK	baseball gloves	$800.00	case	10 gloves/case
1	SMT	baseball gloves	$450.00	case	10 gloves/case
2	HRO	baseball	$126.00	case	24/case
3	HSK	baseball bat	$240.00	case	12/case

SELECT **COUNT(*)**
    FROM stock;

```
(count(*))
        5
```

SELECT **COUNT(DISTINCT description)**
    FROM stock;

```
(count)
     3
```

The SUM Function

Given a hypothetical subset of the **items** table:

1	1001	1	HRO	1	$250.00
1	1002	4	HSK	1	$960.00
2	1002	3	HSK	1	$240.00
1	1004	1	HRO	1	$250.00
2	1004	2	HRO	1	$126.00
3	1004	3	HSK	1	$240.00
4	1004	1	HSK	1	$800.00

SELECT **SUM(total_price)**
FROM items;

```
(sum)
$2866.00
```

SELECT **SUM(total_price)** total
FROM items;

```
total
$2866.00
```

Additional Examples

SELECT **MAX(unit_price)** FROM stock;

```
(max)

$960.00
```

SELECT **MIN(order_date)** FROM orders;

```
(min)

05/20/1994
```

SELECT **AVG(unit_price)** FROM stock;

```
(avg)

$197.14
```

SELECT Statement Clauses

```
SELECT select-list
    FROM table-name
    [WHERE condition]
    [GROUP BY column-list]
    [HAVING condition]
    [ORDER BY column-name]
```

GROUP BY

```
SELECT        customer_num
      FROM    orders
      GROUP BY customer_num
```

customer_num
101
104
104
104
104
106
106
110
110
111
112
225
...

customer_num
101
104
106
110
111
112
115
116
117
119
...
127

GROUP BY and Aggregates

order_num	total_price
1001	$250.00
1002	**$960.00**
1002	**$240.00**
1003	$20.00
1003	$840.00
1003	$99.00
1004	$250.00
1004	$126.00
1004	$240.00
1004	$800.00
1005	$280.00
1005	$198.00
1005	$36.00
1005	$48.00
...	...
1023	$170.00
1023	$190.00

SELECT order_num,
 sum(total_price) tot
 FROM items
 GROUP BY order_num

Adds for each group

order_num	tot
1001	$250.00
1002	**$1200.00**
1003	$959.00
1004	$1416.00
1005	$562.00
...	...
1023	$824.00

Another Example

SELECT city, state, **COUNT(*)**
    FROM customer
    GROUP BY city, state

city	state	(count(*))
Sunnyvale	CA	3
San Francisco	CA	1
Palo Alto	CA	2
Redwood City	CA	5
Los Altos	CA	2
Mountain View	CA	2
Menlo Park	CA	2
Oakland	CA	1
Cherry Hill	NJ	1
Phoenix	AZ	2
Wilmington	DE	1
Princeton	NJ	1
Jacksonville	FL	1
Bartlesville	OK	1
Brighton	MA	1
Denver	CO	1
Blue Island	NY	1

Ordering Grouped Data

```
SELECT city, state,
    COUNT(*)
    FROM customer
    GROUP BY 1, 2
    ORDER BY 1, 2
```

```
city                 state     (count(*))
Bartlesville         OK                 1
Blue Island          NY                 1
Brighton             MA                 1
Cherry Hill          NJ                 1
Denver               CO                 1
Jacksonville         FL                 1
Los Altos            CA                 2
Menlo Park           CA                 2
Mountain View        CA                 2
Oakland              CA                 1
Palo Alto            CA                 2
Phoenix              AZ                 2
Princeton            NJ                 1
Redwood City         CA                 5
San Francisco        CA                 1
Sunnyvale            CA                 3
Wilmington           DE                 1
```

The HAVING Clause

```
SELECT order_num, SUM(total_price) tot
    FROM items
    GROUP BY order_num
    HAVING COUNT(*) > 2;
```

order_num	tot
1003	$959.00
1004	$1416.00
1005	$562.00
1006	$448.00
1007	$1696.00
1013	$143.80
1016	$654.00
1017	$584.00
1018	$1131.00
1021	$1614.00
1022	$232.00
1023	$824.00

Another Example

```
SELECT    stock_num, description, COUNT(*) ,
          AVG(unit_price) average,
          MAX(unit_price) biggest,
          MIN(unit_price) smallest
     FROM stock
     GROUP BY stock_num, description
     HAVING MIN(unit_price) > 400
```

stock_num	description	(count(*))	average	biggest	smallest
4	football	2	$720.00	$960.00	$480.00
7	basketball	1	$600.00	$600.00	$600.00
8	volleyball	1	$840.00	$840.00	$840.00
111	10-spd, assmbld	1	$499.99	$499.99	$499.99
112	12_spd, assmbld	1	$549.00	$549.00	$549.00
113	18-spd, assmbld	1	$685.90	$685.90	$685.90
203	irons/wedge	1	$670.00	$670.00	$670.00

The INTO TEMP Clause

```
SELECT select-list
    FROM table-name
    [WHERE condition]
    [GROUP BY column-list]
    [HAVING condition]
    [ORDER BY column-name]
    [INTO TEMP table-name [WITH NO LOG]]
```

INTO TEMP Example

```
SELECT stock_num, manu_code,description,
        unit_price * 1.05 final_price
    FROM stock
    INTO TEMP stocktemp WITH NO LOG;
SELECT * FROM stocktemp
```

stocktemp

stock_num	manu_code	description	final_price
1	HRO	baseball gloves	$262.50
1	HSK	baseball gloves	$840.00
1	SMT	baseball gloves	$472.50
2	HRO	baseball	$132.30
3	HSK	baseball bat	$252.00
3	SHM	baseball bat	$294.00
4	HRO	football	$504.00
4	HSK	football	$1008.00
...	...	...	...
313	SHM	swim cap	$75.60

INFORMIX

Lab Exercise

Lab Exercise

Exercise 2:

Sam needs more information about his business to plan for the coming year. What SQL statements would you use to gather the information he needs? Enter and execute the statements against your demonstration database.

1. Sam is thinking about raising prices for all stock from HRO by 15%. Can you give him a report of the old and new prices for each article of stock? He'd like the columns to be headed as follows:
stock_num manu_code old_price new_price

2. How many orders have been placed by Sam's customers?

3. What is the average shipping charge for an order?

4. Sam would like to know the highest and lowest amount he has ever charged for shipping an order.

5. Can you round the results from the query in question 4 to the nearest dollar amount?

6. Sam wants to target his marketing better. He needs to know how many customers he has in each state.

7. Sam wants to know the total number of customers who have actually placed an order.

8. Now Sam wants to know how many articles each manufacturer has in his stock table.

9. Sam would like a list of all the customers by their **customer_num**, together with the total of all shipping charges for that customer. Sort the results by the total in reverse order.

10. Sam would like to run several queries on a subset of his customers, the customers in California. To save time, he wants to create a temporary table that contains only those customers. Also, he wants the results of all of the queries to be sorted alphabetically by company name. Name the temporary table **forsam.** Run a SELECT statement on the temporary table to make sure it contains the correct data.

Challenge exercise:

List each group of customers whose cumulative shipment weight for all of a particular customer's orders exceeds 30 lbs. Sort by the shipping weight, in reverse order.

Solutions

Solution

Solution

SQL statements for Exercise 1:

The SQL statements that you have created may vary slightly, as long as you obtain the desired result.

1.

```
SELECT fname, lname, address1, address2,
    city, state, zipcode
   FROM customer;
```

Results set:

```
fname          Ludwig
lname          Pauli
address1       213 Erstwild Court
address2
city           Sunnyvale
state          CA
zipcode        94086
etc.
```

2.

```
SELECT fname, lname, address1, address2,
    city, state, zipcode
   FROM customer
   WHERE state = "CA";
```

There are 18 customers from California.

3.

```
SELECT DISTINCT city, state
   FROM customer
   WHERE state = "CA";
```

Results set:

```
city              state
Los Altos         CA
Menlo Park        CA
Mountain View     CA
Oakland           CA
Palo Alto         CA
Redwood City      CA
San Francisco     CA
Sunnyvale         CA
```

Solution

Solution

Solutions to Exercise 1, *continued*

4.

```
SELECT DISTINCT city, state
   FROM customer
   WHERE state = "CA"
   ORDER BY city desc;
```

Results set:

```
city            state

Sunnyvale       CA
San Francisco   CA
Redwood City    CA
Palo Alto       CA
Oakland         CA
Mountain View   CA
Menlo Park      CA
Los Altos       CA
```

5.

```
SELECT fname, lname, address1, address2,
    city, state, zipcode
   FROM customer
   WHERE customer_num = 103;
```

The customer is Philip Currie.

Solution

Solution

Solutions to Exercise 1, *continued*

6.

```
SELECT stock_num, manu_code, description
  FROM stock
  WHERE manu_code = "ANZ"
  ORDER BY description;
```

Results set:

```
    stock_num manu_code description
            205 ANZ                3 golf balls
            201  ANZ               golf shoes
            110 ANZ                helmet
            310 ANZ                kick board
            301 ANZ                running shoes
            313 ANZ                swim cap
             6    ANZ              tennis ball
             5    ANZ              tennis racquet
             8    ANZ              volleyball
             9    ANZ              volleyball net
            304 ANZ                watch
```

7.

```
SELECT DISTINCT manu_code
  FROM items
  ORDER by manu_code;
```

Results set:

```
manu_code
ANZ
HRO
HSK
KAR
NKL
NRG
PRC
SHM
SMT
```

8.

```
SELECT fname, lname, company, phone
  FROM customer
  WHERE company MATCHES "*Medal*";
```

The company is Gold Medal Sports.

Solution

Solution

Solutions to Exercise 1, *continued*

9.

```
SELECT stock_num, manu_code, description,
   unit_price
   FROM stock
   WHERE description MATCHES "*bicycle*"
   AND unit_price BETWEEN 50 AND 75;
```

Results set:

```
stock_num manu_code description   unit_price
        101       SHM          bicycle tires    $68.00
        105       PRC          bicycle wheels   $53.00
        107       PRC          bicycle saddle   $70.00
```

10.

```
SELECT order_num, order_date, ship_date
   FROM orders
   WHERE ship_date IS NULL;
```

Order number 1006 is the only order that hasn’t been shipped.

11.

```
SELECT fname, lname, address1, address2,
     city, state, zipcode
   FROM customer
   WHERE lname MATCHES "[A-G]*"
   ORDER by state, city;
```

The customers are:

```
Alfred Grant
Lana Beatty
Dick Baxter
Philip Currie
Frank Albertson
```

12.

```
SELECT company
   FROM customer
   WHERE company MATCHES "*[Tt]own*";
```

The company is Sportstown.

Solution

Solution

Solutions to Exercise 1, *continued*

13.

```
SELECT company, city, state, zipcode
  FROM customer
  WHERE state IN ("CA","FL")
  AND company MATCHES "*Sports"
  ORDER BY state, company DESC;
```

OR

```
SELECT company, city, state, zipcode
  FROM customer
  WHERE (state = "CA" OR state = "FL")
  AND company MATCHES "*Sports"
  ORDER BY state, company DESC;
```

Without the parentheses in the above statement you will not get the correct results.

Results set:

```
company            city           state  zipcode
     Quinn's Sports      Redwood City    CA     94063
     Phil's Sports       Palo Alto       CA     94303
     Los Altos Sports    Los Altos       CA     94022
     Gold Medal Sports   Menlo Park      CA     94025
     Blue Ribbon Sports  Oakland         CA     94609
     Bay Sports          Jacksonville    FL     32256
```

Solution

Solution

Solution to the Challenge Exercise

The following SQL statement can be used to get the information that you need about order number 1008. You can manually do the math by dividing **total_price** by **quantity**, and look in the stock table for the particular **stock_num** and **manu_code** to find out if the **unit_price** matches your calculation.

```
SELECT stock_num, manu_code, quantity, total_price
     FROM items
  WHERE order_num = "1008";
```

Results set:

```
stock_num manu_code quantity total_price
          8    ANZ                    1                  $840.00
          9    ANZ                    5                  $100.00
```

In the next section of the module you will learn that SQL statements can contain math calculations:

```
SELECT stock_num, manu_code, quantity, total_price,
   total_price/quantity perunit
  FROM items
  WHERE order_num = "1008";
```

Solution

Solution

SQL statements for Exercise 2

1.

```
SELECT     stock_num,
           manu_code,
           unit_price old_price,
           unit_price * 1.15 new_price
FROM       stock
WHERE      manu_code = "HRO";
```

There are twelve articles of stock from HRO. Results set:

```
stock_num  manu_code  old_price  new_price
        1  HRO        $250.00    $287.50
        2  HRO        $126.00    $144.90
        4  HRO        $480.00    $552.00
etc.
```

2.

```
SELECT count(*) FROM orders;
```

The count is 23.

3.

```
SELECT AVG(ship_charge)
  FROM orders;
```

The average shipping charge is $13.97.

4.

```
SELECT MAX(ship_charge) highest,
       MIN(ship_charge) lowest
  FROM orders;
```

Results set:

```
highest lowest
$25.20     $5.00
```

Solution

Solution

Solutions to Exercise 2, *continued*

5.

```
SELECT ROUND (MAX(ship_charge),0) highest,
       ROUND (MIN(ship_charge),0) lowest
   FROM orders;
```

OR

```
SELECT ROUND(MAX(ship_charge)) highest,
             ROUND (MIN(ship_charge)) lowest
      FROM orders;
```

If you omit the number of decimal places, the default is 0.

Results set:

```
highest       lowest
     25             5
```

6.

```
SELECT state, count(*)
   FROM customer
   GROUP BY state;
```

Results set:

```
state      (count(*))
OK                   1
CO                   1
NJ                   2
AZ                   2
DE                   1
CA                  18
FL                   1
NY                   1
MA                   1
```

7.

```
SELECT COUNT(DISTINCT customer_num)
   FROM orders;
```

Seventeen customers have placed orders.

Solution

Solution

Solutions to Exercise 2, *continued*:

8.

```
SELECT manu_code, COUNT(*)
  FROM stock
  GROUP BY manu_code;
```

Results set:

```
manu_code  (count(*))
ANZ                    11
HRO                    12
HSK                     4
KAR                     6
NKL                     5
NRG                     1
PRC                    15
SHM                    17
SMT                     3
```

9.

```
SELECT customer_num, SUM(ship_charge) totcharge
  FROM orders
  GROUP BY customer_num
  ORDER BY totcharge DESC;
```

Results set:

```
customer_num totcharge
            117          $39.40
            104          $38.00
            106          $31.50
            122          $23.00
etc.
```

10.

```
SELECT * FROM customer
  WHERE state = "CA"
     ORDER BY company
  INTO TEMP forsam WITH NO LOG;
SELECT * FROM forsam;
```

Solution

Solution

Solution to Challenge exercise

```
SELECT customer_num, SUM(ship_weight) totweight
  FROM orders
  GROUP BY customer_num
  HAVING SUM(ship_weight) > 30
  ORDER BY totweight DESC;
```

Results set:

```
customer_num    totweight
         117       196.70
         106       136.40
         104       127.20
         122        90.00
         116        80.80
         112        70.80
         121        70.50
         110        66.20
         127        60.00
         120        60.00
         101        50.60
         115        40.60
         124        40.00
         119        35.00
```