CS 309A- Database Management Systems

GROUP BY

- If you want to group table rows, you could use GROUP BY clause.
- Syntax:

```
 SELECT     columnlist
FROM       tablelist
[WHERE     conditionlist]
 [GROUP BY columnlist]
 [HAVING   conditionlist]
[ORDER BY columnlist [ASC | DESC]];
```

- GROUP BY is generally used when you have columns combined with aggregate functions in the SELECT statement.

GROUP BY

- What is the price of cheapest product from each vendor?

```
SELECT   V_CODE, MIN(P_PRICE)
FROM     PRODUCT
GROUP BY V_CODE
ORDER BY V_CODE;
```

```
+--------+--------------+
| V_CODE | MIN(P_PRICE) |
+--------+--------------+
|   NULL |         5.87 |
|  21225 |         6.99 |
|  21231 |         8.45 |
|  21344 |         4.99 |
|  23119 |        39.95 |
|  24288 |        99.87 |
|  25595 |        38.95 |
+--------+--------------+
7 rows in set (0.00 sec)
```

GROUP BY

- What is the cheapest product from each vendor? Suppose we want to know product code, description, price and its vendor (code).

```
SELECT   P_CODE, P_DESCRIPT, MIN(P_PRICE), V_CODE
FROM     PRODUCT
GROUP BY V_CODE;
```

```
+----------+-----------------------------------+--------------+---------+
| P_CODE   | P_DESCRIPT                        | MIN(P_PRICE) | V_CODE  |
+----------+-----------------------------------+--------------+---------+
| 23114-AA | Sledge hammer, 12 lb.             |         5.87 |    NULL |
| 23109-HB | Claw hammer                       |         6.99 |   21225 |
| SW-23116 | 2.5-in. wd. screw, 50             |         8.45 |   21231 |
| 13-Q2/P2 | 7.25-in. pwr. saw blade           |         4.99 |   21344 |
| 1546-QQ2 | Hrd. cloth, 1/4-in., 2x50         |        39.95 |   23119 |
| 2232/QTY | B&D jigsaw, 12-in. blade          |        99.87 |   24288 |
| 11QER/31 | Power painter, 15 psi., 3-nozzle  |        38.95 |   25595 |
+----------+-----------------------------------+--------------+---------+
7 rows in set (0.00 sec)
```

GROUP BY

Another way to get the cheapest product from each vendor?

```
SELECT   P_CODE, P_DESCRIPT, P_PRICE, V_CODE
FROM     PRODUCT
WHERE    P_PRICE IN ( SELECT MIN(P_PRICE) FROM PRODUCT GROUP BY V_CODE );
```

```
+----------+----------------------------+---------+--------+
| P_CODE   | P_DESCRIPT                 | P_PRICE | V_CODE |
+----------+----------------------------+---------+--------+
| 1546-QQ2 | Hrd. cloth, 1/4-in., 2x50  |   39.95 |  23119 |
| 2232/QWE | B&D jigsaw, 8-in. blade    |   99.87 |  24288 |
| 2238/QPD | B&D cordless drill, 1/2-in. |   38.95 |  25595 |
| 54778-2T | Rat-tail file, 1/8-in. fine |    4.99 |  21344 |
| PVC23DRT | PVC pipe, 3.5-in., 8-ft    |    5.87 |   NULL |
| SM-18277 | 1.25-in. metal screw, 25   |    6.99 |  21225 |
| SW-23116 | 2.5-in. wd. screw, 50      |    8.45 |  21231 |
+----------+----------------------------+---------+--------+
7 rows in set (0.00 sec)
```

GROUP BY

- Suppose we want to know the number of products supplied by each vendor and the price of cheapest product from each vendor.

```
SELECT   COUNT(P_CODE),
MIN(P_PRICE), V_CODE
FROM     PRODUCT
GROUP BY V_CODE;
```

```
+---------------+--------------+--------+
| COUNT(P_CODE) | MIN(P_PRICE) | V_CODE |
+---------------+--------------+--------+
|             2 |         5.87 |   NULL |
|             2 |         6.99 |  21225 |
|             1 |         8.45 |  21231 |
|             3 |         4.99 |  21344 |
|             2 |        39.95 |  23119 |
|             3 |        99.87 |  24288 |
|             3 |        38.95 |  25595 |
+---------------+--------------+--------+
7 rows in set (0.00 sec)
```

Practice 7:

- What is the average price of products from each vendor?
- Find how many invoices each customer have?
- What is the total number of available quantity of products for each vendor?

- What is the average price of products from each vendor?

```
mysql> select avg(p_price), v_code
    -> from product
    -> group by v_code;
+--------------+--------+
| avg(p_price) | v_code |
+--------------+--------+
|    10.135000 |   NULL |
|     8.470000 |  21225 |
|     8.450000 |  21231 |
|    12.490000 |  21344 |
|    41.970000 |  23119 |
|   155.593333 |  24288 |
|    89.630000 |  25595 |
+--------------+--------+
7 rows in set (0.04 sec)
```

- Find how many invoices each customer have?

```
mysql> select count(inv_number), cus_code
    -> from invoice
    -> group by cus_code;
+-------------------+----------+
| count(inv_number) | cus_code |
+-------------------+----------+
|                 3 |    10011 |
|                 1 |    10012 |
|                 2 |    10014 |
|                 1 |    10015 |
|                 1 |    10018 |
+-------------------+----------+
5 rows in set (0.00 sec)
```

- What is the total number of available quantity of products for each vendor?

```
mysql> select sum(p_qoh), min(p_price), v_code
    -> from product
    -> group by v_code;
+------------+--------------+--------+
| sum(p_qoh) | min(p_price) | v_code |
+------------+--------------+--------+
|        196 |         5.87 |   NULL |
|        195 |         6.99 |  21225 |
|        237 |         8.45 |  21231 |
|         93 |         4.99 |  21344 |
|         38 |        39.95 |  23119 |
|         25 |        99.87 |  24288 |
|         38 |        38.95 |  25595 |
+------------+--------------+--------+
7 rows in set (0.00 sec)
```

HAVING

- Operates like the WHERE clause.
- The WHERE applies to columns and expressions for individual rows, while the HAVING is applied to the output of a GROUP BY operation.

- If we want to get the number of products supplied by each vendor whose prices average less than $10.

HAVING

Step 1: Get the number products supplied by each vendor and their average

```
SELECT   V_CODE, COUNT(P_CODE), AVG(P_PRICE)
FROM     PRODUCT
GROUP BY V_CODE;
```

V_CODE	COUNT(P_CODE)	AVG(P_PRICE)
NULL	2	10.135000
21225	2	8.470000
21231	1	8.450000
21344	3	12.490000
23119	2	41.970000
24288	3	155.593333
25595	3	89.630000

Step 2: Select rows which have average price less than 10

```
SELECT   V_CODE, COUNT(P_CODE), AVG(P_PRICE)
FROM     PRODUCT
GROUP BY V_CODE
HAVING   AVG(P_PRICE) < 10;
```

V_CODE	COUNT(P_CODE)	AVG(P_PRICE)
21225	2	8.470000
21231	1	8.450000

Another example

Suppose we want to get the total value of the products in inventory supplied by each vendor which are over $500. List the contents by total value in descending.

```
SELECT   V_CODE, SUM(P_QOH*P_PRICE) AS TOTALVALUE
FROM     PRODUCT
GROUP BY V_CODE
HAVING   (SUM(P_QOH*P_PRICE) > 500)
ORDER BY SUM(P_QOH*P_PRICE) DESC;
```

V_CODE	TOTALVALUE
24288	4305.47
25595	3506.42
21231	2002.65
23119	1611.02
21225	1431.13
NULL	1218.76
21344	1009.07

Practice 8:

- Find the invoices which have total purchase amount over $100. Show the invoice number and its total purchase amount. List the contents by total purchase amount in ascending.
  - SELECT from LINE
  - LINE_UNITS*LINE_PRICE gives the purchase amount of one product in one invoice
  - Use SUM to get the total purchase amount

- Find the invoices which have total purchase amount over $100. Show the invoice number and its total purchase amount. List the contents by total purchase amount in ascending.

```
mysql> select inv_number, sum(line_units * line_price)
    -> from line
    -> group by inv_number
    -> having sum(line_units * line_price) > 100
    -> order by sum(line_units * line_price);
+------------+-------------------------------+
| inv_number | sum(line_units * line_price)  |
+------------+-------------------------------+
|       1003 |                      153.8500 |
|       1006 |                      397.8300 |
|       1008 |                      399.1500 |
+------------+-------------------------------+
3 rows in set (0.05 sec)
```

Join database tables

- Recall join:
  - Combines information from two or more tables.
  - A **natural join** links tables by selecting only the rows with common values in their common attributes.
    - Create a PRODUCT of the tables
    - Yield the rows that the **join columns** values are equal.
    - Remove duplicate columns.

R

ColA	*ColB*
A	1
B	2
D	3
F	4
E	5

S

SColA	*SColB*
A	1
C	2
D	3
E	4

R JOIN $_{R.ColA = S.SColA}$ S

A	1	A	1
D	3	D	3
E	5	E	4

R JOIN $_{R.ColB = S.SColB}$ S

A	1	A	1
B	2	C	2
D	3	D	3
F	4	E	4

How to use SQL to join tables

- Create the Cartesian product

  -- List the tables in the FROM clause

- Select only the rows in which the common attribute values match

  -- Use the WHERE clause to indicate the join condition

  -- The join condition is generally an equality comparison between the *foreign key* and the *primary key* of the related tables.

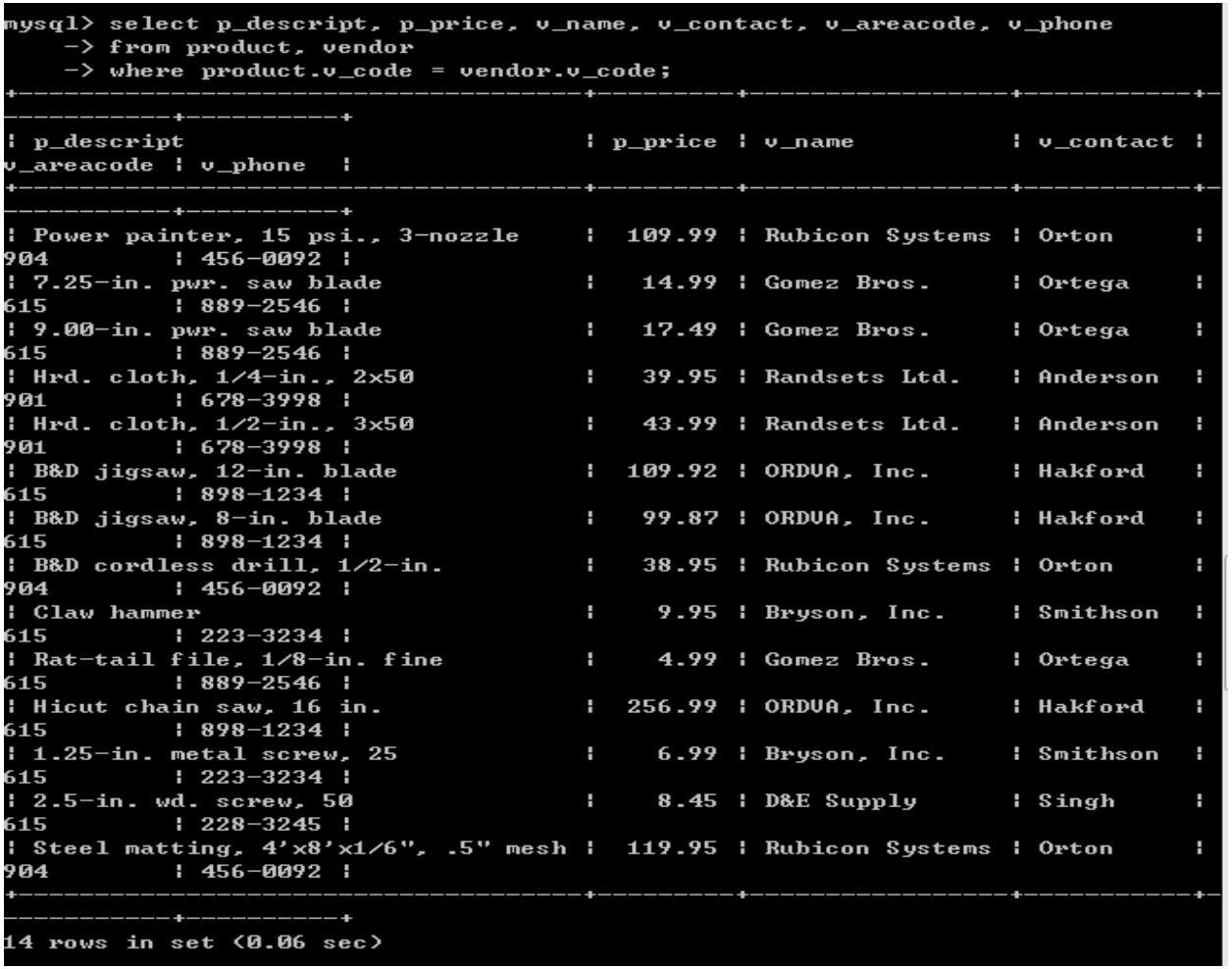
How to use SQL to join tables

- How to get all the products with their vendors information?
  - Join VENDOR and PRODUCT
  - Common attributes: V_CODE

TABLE 7.9 Creating Links Through Foreign Keys

TABLE	ATTRIBUTES TO BE SHOWN	LINKING ATTRIBUTE
PRODUCT	P_DESCRIPT, P_PRICE	V_CODE
VENDOR	V_NAME, V_CONTACT, V_AREACODE, V_PHONE	V_CODE

```
SELECT P_DESCRIPT, P_PRICE, V_NAME, V_CONTACT, V_AREACODE, V_PHONE
FROM   PRODUCT, VENDOR
WHERE  PRODUCT.V_CODE = VENDOR.V_CODE;
```



```
mysql> select p_descript, p_price, v_name, v_contact, v_areacode, v_phone
    -> from product, vendor
    -> where product.v_code = vendor.v_code;
+-----------------------------------+---------+-----------------+-----------+-
-----------+----------+
| p_descript                        | p_price | v_name          | v_contact |
v_areacode | v_phone  |
+-----------------------------------+---------+-----------------+-----------+-
-----------+----------+
| Power painter, 15 psi., 3-nozzle  |  109.99 | Rubicon Systems | Orton     |
904        | 456-0092 |
| 7.25-in. pwr. saw blade           |   14.99 | Gomez Bros.     | Ortega    |
615        | 889-2546 |
| 9.00-in. pwr. saw blade           |   17.49 | Gomez Bros.     | Ortega    |
615        | 889-2546 |
| Hrd. cloth, 1/4-in., 2x50         |   39.95 | Randsets Ltd.   | Anderson  |
901        | 678-3998 |
| Hrd. cloth, 1/2-in., 3x50         |   43.99 | Randsets Ltd.   | Anderson  |
901        | 678-3998 |
| B&D jigsaw, 12-in. blade          |  109.92 | ORDVA, Inc.     | Hakford   |
615        | 898-1234 |
| B&D jigsaw, 8-in. blade           |   99.87 | ORDVA, Inc.     | Hakford   |
615        | 898-1234 |
| B&D cordless drill, 1/2-in.       |   38.95 | Rubicon Systems | Orton     |
904        | 456-0092 |
| Claw hammer                       |    9.95 | Bryson, Inc.    | Smithson  |
615        | 223-3234 |
| Rat-tail file, 1/8-in. fine       |    4.99 | Gomez Bros.     | Ortega    |
615        | 889-2546 |
| Hicut chain saw, 16 in.           |  256.99 | ORDVA, Inc.     | Hakford   |
615        | 898-1234 |
| 1.25-in. metal screw, 25          |    6.99 | Bryson, Inc.    | Smithson  |
615        | 223-3234 |
| 2.5-in. wd. screw, 50             |    8.45 | D&E Supply      | Singh     |
615        | 228-3245 |
| Steel matting, 4'x8'x1/6", .5" mesh |  119.95 | Rubicon Systems | Orton     |
904        | 456-0092 |
+-----------------------------------+---------+-----------------+-----------+-
-----------+----------+
14 rows in set (0.06 sec)
```

How to use SQL to join tables

- You can also use other SQL commands on the joined tables.

- List the products that were stocked after 2016-01-15 and their vendors information by their price in ascending.

```
SELECT   P_DESCRIPT, P_PRICE, P_INDATE, V_NAME, V_CONTACT, V_PHONE
FROM     PRODUCT, VENDOR
WHERE    PRODUCT.V_CODE = VENDOR.V_CODE
AND      P_INDATE > '2016-01-15'
ORDER BY P_PRICE;
```


```
mysql> select p_descript, p_price, p_indate, v_name, v_contact, v_phone
    -> from product, vendor
    -> where product.v_code = vendor.v_code
    -> and p_indate > '2016-01-15'
    -> order by p_price;
+-----------------------------------+---------+---------------------+---------
--------+-----------+----------+
| p_descript                        | p_price | p_indate            | v_name
       | v_contact | v_phone  |
+-----------------------------------+---------+---------------------+---------
--------+-----------+----------+
| 1.25-in. metal screw, 25          |    6.99 | 2016-03-01 00:00:00 | Bryson,
Inc.     | Smithson  | 223-3234 |
| 2.5-in. wd. screw, 50             |    8.45 | 2016-02-24 00:00:00 | D&E Supp
ly       | Singh     | 228-3245 |
| Claw hammer                       |    9.95 | 2016-01-20 00:00:00 | Bryson,
Inc.     | Smithson  | 223-3234 |
| B&D cordless drill, 1/2-in.       |   38.95 | 2016-01-20 00:00:00 | Rubicon
Systems  | Orton     | 456-0092 |
| Steel matting, 4'x8'x1/6", .5" mesh |  119.95 | 2016-01-17 00:00:00 | Rubicon
Systems  | Orton     | 456-0092 |
| Hicut chain saw, 16 in.           |  256.99 | 2016-02-07 00:00:00 | ORDVA, I
nc.      | Hakford   | 898-1234 |
+-----------------------------------+---------+---------------------+---------
--------+-----------+----------+
6 rows in set (0.05 sec)
```

Practice 9

- List customers and the date they generated invoices.

TABLE	ATTRIBUTES TO BE SHOWN	LINKING ATTRIBUTE
CUSTOMER	CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_BALANCE	CUS_CODE
INVOICE	INV_NUMBER, INV_DATE	CUS_CODE

- For the customers who live in area ‘615’, find out their information and the date they generated invoices. Show same attributes as the above query. List the records by their balance in descending.

- List customers and the date they generated invoices.

```
mysql> select cus_lname, cus_fname, cus_initial, cus_areacode, cus_balance,inv_d
ate
    -> from customer, invoice
    -> where customer.cus_code = invoice.cus_code;
+-----------+-----------+-------------+---------------+-------------+-----------
---------+
| cus_lname | cus_fname | cus_initial | cus_areacode | cus_balance | inv_date
         |
+-----------+-----------+-------------+---------------+-------------+-----------
---------+
| Orlando   | Myron     | NULL        | 615           |        0.00 | 2016-01-16
00:00:00 |
| Dunne     | Leona     | K           | 713           |        0.00 | 2016-01-16
00:00:00 |
| Smith     | Kathy     | W           | 615           |      345.86 | 2016-01-16
00:00:00 |
| Dunne     | Leona     | K           | 713           |        0.00 | 2016-01-17
00:00:00 |
| Farriss   | Anne      | G           | 713           |      216.55 | 2016-01-17
00:00:00 |
| Orlando   | Myron     | NULL        | 615           |        0.00 | 2016-01-17
00:00:00 |
| O'Brian   | Amy       | B           | 713           |        0.00 | 2016-01-17
00:00:00 |
| Dunne     | Leona     | K           | 713           |        0.00 | 2016-01-17
00:00:00 |
+-----------+-----------+-------------+---------------+-------------+-----------
---------+
8 rows in set (0.06 sec)
```

- For the customers who live in area ‘615’, find out their information and the date they generated invoices. Show same attributes as the above query. List the records by their balance in descending.

```
mysql> select cus_lname, cus_fname, cus_initial, cus_areacode, cus_balance,inv_d
ate
    -> from customer, invoice
    -> where customer.cus_code = invoice.cus_code
    -> and cus_areacode = '615'
    -> order by cus_balance desc;
+-----------+-----------+-------------+--------------+-------------+------------
---------+
| cus_lname | cus_fname | cus_initial | cus_areacode | cus_balance | inv_date
         |
+-----------+-----------+-------------+--------------+-------------+------------
---------+
| Smith     | Kathy     | W           | 615          |      345.86 | 2016-01-16
00:00:00 |
| Orlando   | Myron     | NULL        | 615          |        0.00 | 2016-01-16
00:00:00 |
| Orlando   | Myron     | NULL        | 615          |        0.00 | 2016-01-17
00:00:00 |
+-----------+-----------+-------------+--------------+-------------+------------
---------+
3 rows in set (0.00 sec)
```

Join more tables

- When join three or more tables, you need to specify a join condition for ***each pair*** of the tables.
- The number of join condition will be (# tables for join – 1)
  - 3 tables should give 2 conditions
  - 5 tables should give 4 conditions
- *Example:*

List the CUS_LNAME, INV_NUMBER, INV_DATE, P_DESCRIPT for all invoices of customer 10014

--How many tables are needed for join?

--What are the attributes to link each pair of tables?

TABLE	ATTRIBUTES TO BE SHOWN	LINKING ATTRIBUTE
CUSTOMER	CUS_LNAME	CUS_CODE
INVOICE	INV_NUMBER, INV_DATE	CUS_CODE, INV_NUMBER
LINE		INV_NUMBER, P_CODE
PRODUCT	P_DESCRIPT	P_CODE

```
SELECT   CUS_LNAME, INV_NUMBER, INV_DATE, P_DESCRIPT
FROM     CUSTOMER, INVOICE, LINE, PRODUCT
WHERE    CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
AND      INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND      LINE.P_CODE = PRODUCT.P_CODE
AND      CUS_CODE = '10014'
ORDER BY INV_NUMBER;
```

```
ERROR 1052 (23000): Column 'inv_number' in field list is ambiguous
```

SQL interpreter feels confused of the source of columns INV_NUMBER and CUS_CODE

Join more tables

```
SELECT   CUS_LNAME, INVOICE.INV_NUMBER, INV_DATE, P_DESCRIPT
FROM     CUSTOMER, INVOICE, LINE, PRODUCT
WHERE    CUSTOMER.CUS_CODE = INVOICE.CUS_CODE
AND      INVOICE.INV_NUMBER = LINE.INV_NUMBER
AND      LINE.P_CODE = PRODUCT.P_CODE
AND      CUSTOMER.CUS_CODE = '10014'
ORDER BY INV_NUMBER;
```

```
+-----------+------------+---------------------+--------------------------+
| cus_lname | inv_number | inv_date            | p_descript               |
+-----------+------------+---------------------+--------------------------+
| Orlando   |       1001 | 2016-01-16 00:00:00 | 7.25-in. pwr. saw blade  |
| Orlando   |       1001 | 2016-01-16 00:00:00 | Claw hammer              |
| Orlando   |       1006 | 2016-01-17 00:00:00 | B&D jigsaw, 12-in. blade |
| Orlando   |       1006 | 2016-01-17 00:00:00 | Claw hammer              |
| Orlando   |       1006 | 2016-01-17 00:00:00 | Hicut chain saw, 16 in.  |
| Orlando   |       1006 | 2016-01-17 00:00:00 | 1.25-in. metal screw, 25 |
+-----------+------------+---------------------+--------------------------+
6 rows in set (0.05 sec)
```

Practice 10

- Find the customers who ordered “saw” after ‘2016-01-15’. List the customer last name, customer first name, invoice number, invoice date and product description.
  - How many tables are needed for join?

    CUSTOMER, PRODUCT, INVOICE, LINE
  - What are the attributes to link each pair of tables?

TABLE	ATTRIBUTES TO BE SHOWN	LINKING ATTRIBUTE
CUSTOMER	CUS_LNAME, CUS_FNAME	CUS_CODE
INVOICE	INV_NUMBER, INV_DATE	CUS_CODE, INV_NUMBER
LINE		INV_NUMBER, P_CODE
PRODUCT	P_DESCRIPT	P_CODE

- Find the customers who ordered “saw” after ‘01/15/2016’. List the customer last name, customer first name, invoice number, invoice date and product description.

```
mysql> select cus_lname, cus_fname, invoice.inv_number, inv_date, p_descript
    -> from customer, invoice, line, product
    -> where customer.cus_code = invoice.cus_code
    -> and invoice.inv_number = line.inv_number
    -> and line.p_code = product.p_code
    -> and p_descript like '%saw%'
    -> and inv_date > '2016-01-15'
    -> order by inv_number;
+-----------+-----------+------------+---------------------+--------------------
------+
| cus_lname | cus_fname | inv_number | inv_date            | p_descript
      |
+-----------+-----------+------------+---------------------+--------------------
------+
| Orlando   | Myron     |       1001 | 2016-01-16 00:00:00 | 7.25-in. pwr. saw b
lade  |
| Smith     | Kathy     |       1003 | 2016-01-16 00:00:00 | 7.25-in. pwr. saw b
lade  |
| Orlando   | Myron     |       1006 | 2016-01-17 00:00:00 | Hicut chain saw, 16
 in.  |
| Orlando   | Myron     |       1006 | 2016-01-17 00:00:00 | B&D jigsaw, 12-in.
blade |
| O'Brian   | Amy       |       1007 | 2016-01-17 00:00:00 | 7.25-in. pwr. saw b
lade  |
+-----------+-----------+------------+---------------------+--------------------
------+
5 rows in set (0.00 sec)
```

Thank you & Questions