**Correlated Subqueries: In-class exercise 1**

Assume we are using the database shown here



The current contents of the LINE table are shown here:

**INV\_NUMBER LINE\_NUMBER P\_CODE LINE\_UNITS LINE\_PRICE**

**---------- ----------- ---------- ---------- ----------**

**1001 1 13-Q2/P2 1 14.99**

**1001 2 23109-HB 1 9.95**

**1002 1 54778-2T 2 4.99**

**1003 1 2238/QPD 1 38.95**

**1003 2 1546-QQ2 1 39.95**

**1003 3 13-Q2/P2 5 14.99**

**1004 1 54778-2T 3 4.99**

**1004 2 23109-HB 2 9.95**

**1005 1 PVC23DRT 12 5.87**

**1006 1 SM-18277 3 6.99**

**1006 2 2232/QTY 1 109.92**

**1006 3 23109-HB 1 9.95**

**1006 4 89-WRE-Q 1 256.99**

**1007 1 13-Q2/P2 2 14.99**

**1007 2 54778-2T 1 4.99**

**1008 1 PVC23DRT 5 5.87**

**1008 2 WR3/TT3 3 119.95**

**1008 3 23109-HB 1 9.95**

Suppose we want to find the invoice number, product code, and number of units ordered for each entry in LINE – but only for those entries in LINE where the number of units ordered (LINE\_UNITS) is greater than the average number of units ordered *for the specified product code.* To accomplish this, do the following:

1. For each row, look at the product code and then find the average units ordered for that product code.
2. For the current row, is the number ordered larger than the average you calculated in step 1?
3. If so, you include the current row in your result set; otherwise it is omitted.

The steps you just went through are the steps required for the following correlated subquery. Note that for each row in LINE we first executed the inner query, then used that result to determine if the current line should be included in the result set.

**select inv\_number, p\_code, line\_units**

**from line ls**

**where ls.line\_units >**

**(select avg(line\_units)**

**from line la**

**where la.p\_code = ls.p\_code);**