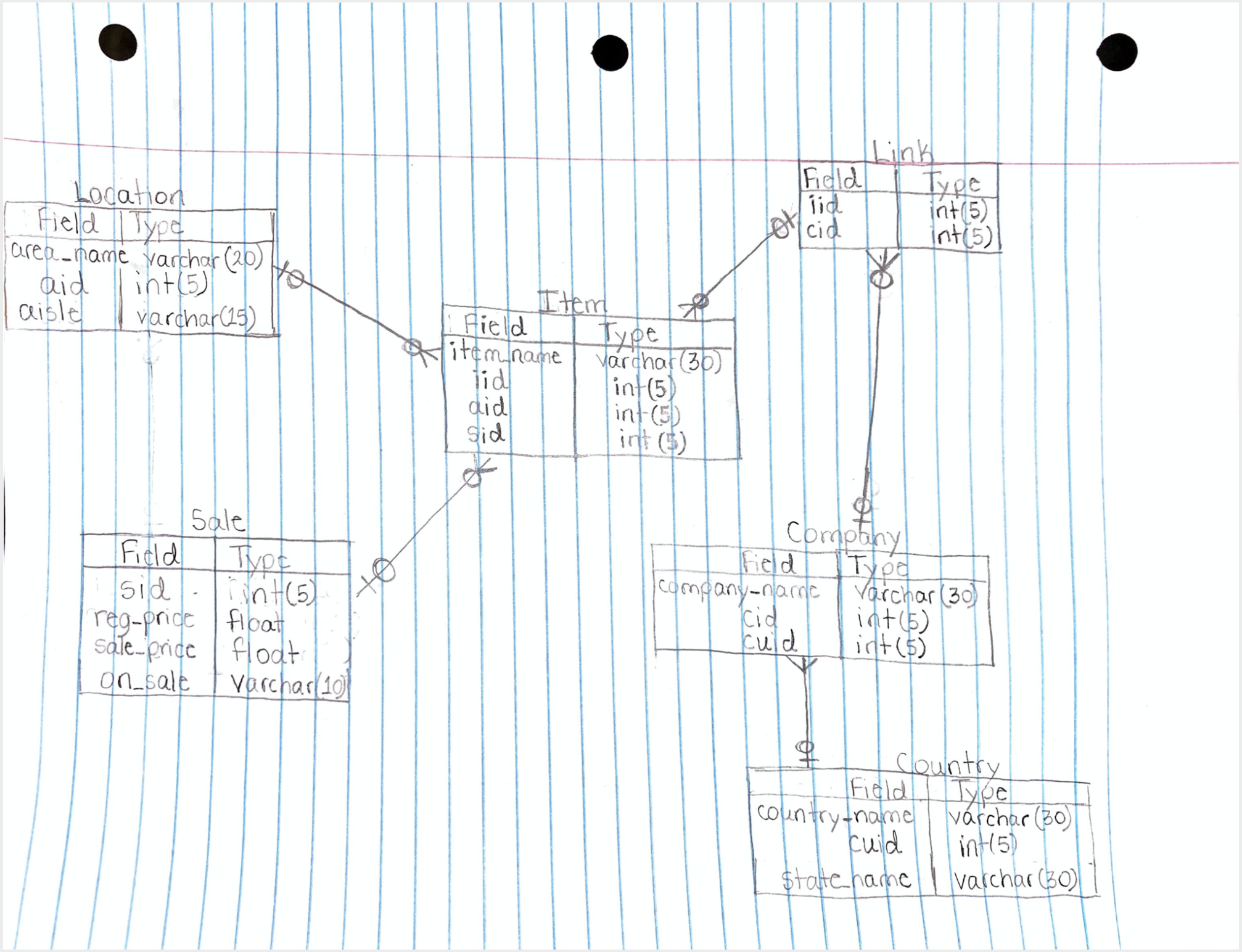
**CMPSC 431W – Week 9 assignment – One point per step**

**STEP 1:** Obtain the CSV data file. Note that the first row of data are the column headers.

**STEP 2:** Analyze the file. Create a data model that captures all of the data shown in the file. **The data model must be in third normal form**. Paste the crow's foot diagram below showing your data model.

[Paste crow's foot diagram here]



**STEP 3:** Load the data from the csv file into the various tables as defined by your data model. For each table, paste a DESCRIBE and a SELECT \* FROM below:

**REMOVE THE FIRST ROW for each and change rows at bottom**

**LOCATION:**

mysql> describe bkp5190\_location;

+--------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| aid | int(6) | NO | | NULL | |

| area\_name | varchar(20) | NO | PRI | NULL | |

| aisle | varchar(20) | NO | | NULL | |

+--------+-------------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> select \* from bkp5190\_area\_name;

+--------+--------------+---------+

| aid | area\_name | aisle |

+--------+--------------+---------+

| 0 | area\_name | Aisle |

| 5000 | canned foods | aisle 5 |

| 5001 | dairy | aisle 2 |

| 5002 | frozen | aisle 4 |

| 5003 | meat & fish | aisle 3 |

| 5004 | produce | aisle 1 |

+--------+--------------+---------+

6 rows in set (0.00 sec)

**ITEM TABLE:**

mysql> describe bkp5190\_item;

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| iid | int(6) | NO | PRI | NULL | |

| aid | int(6) | NO | | NULL | |

| sid | int(6) | NO | | NULL | |

| item | varchar(30) | NO | | NULL | |

+---------+-------------+------+-----+---------+-------+

4 rows in set (0.01 sec)

mysql> select \* from bkp5190\_item;

+--------+--------+---------+----------------+

| iid | aid | sid | item |

+--------+--------+---------+----------------+

| 0 | 0 | 0 | Item |

| 2000 | 5004 | 3000 | corn |

| 2001 | 5003 | 3001 | chicken |

| 2002 | 5003 | 3002 | hamburger |

| 2003 | 5000 | 3003 | corn |

| 2004 | 5003 | 3004 | chicken |

| 2005 | 5001 | 3005 | yogurt |

| 2006 | 5002 | 3006 | tuna |

| 2007 | 5004 | 3007 | eggplant |

| 2008 | 5002 | 3008 | pizza |

| 2009 | 5000 | 3009 | corn |

| 2010 | 5000 | 3010 | baked beans |

| 2011 | 5004 | 3011 | tomato |

| 2013 | 5004 | 3012 | banana |

| 2013 | 5003 | 3013 | salmon |

| 2014 | 5000 | 3014 | spinach |

| 2015 | 5001 | 3015 | dozen eggs |

| 2016 | 5003 | 3016 | tuna |

| 2017 | 5000 | 3017 | spinach |

| 2018 | 5001 | 3018 | whole milk |

| 2019 | 5002 | 3019 | waffles |

| 2020 | 5004 | 3020 | cherry |

| 2021 | 5002 | 3021 | spinach |

| 2022 | 5004 | 3022 | apple |

| 2023 | 5002 | 3023 | ice cream |

| 2024 | 5003 | 3024 | steak |

| 2025 | 5001 | 3025 | skim milk |

| 2026 | 5004 | 3026 | pear |

| 2027 | 5003 | 3027 | chicken breast |

| 2028 | 5004 | 3028 | plum |

| 2029 | 5004 | 3029 | lettuce |

| 2030 | 5001 | 3030 | 2% milk |

+--------+--------+---------+----------------+

32 rows in set (0.00 sec)

**SALES TABLE WITH PRICES:**

mysql> describe bkp5190\_sales;

+--------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+-------------+------+-----+---------+-------+

| sid | int(6) | NO | PRI | NULL | |

| reg\_price | double | NO | | NULL | |

| sale\_price | double | NO | | NULL | |

| OnSale | varchar(20) | NO | | NULL | |

+--------------+-------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> select \* from bkp5190\_sales;

+---------+--------------+------------+---------------+

| sid | reg\_price | sale\_price | OnSale |

+---------+--------------+------------+---------------+

| 0 | 0 | 0 | onOnSale |

| 3000 | 2.9 | 2.54 | regular price |

| 3001 | 0.07 | 0.06 | regular price |

| 3002 | 0.57 | 0.5 | on sale |

| 3003 | 2.91 | 2.55 | regular price |

| 3004 | 2.41 | 2.11 | regular price |

| 3005 | 3.12 | 2.73 | on sale |

| 3006 | 3.22 | 2.82 | regular price |

| 3007 | 4.55 | 3.99 | regular price |

| 3008 | 2.33 | 2.04 | regular price |

| 3009 | 0.44 | 0.39 | regular price |

| 3010 | 2.78 | 2.44 | on sale |

| 3011 | 1.18 | 1.03 | regular price |

| 3012 | 2.2 | 1.93 | on sale |

| 3013 | 2.43 | 2.13 | regular price |

| 3014 | 1.35 | 1.18 | regular price |

| 3015 | 1.73 | 1.52 | regular price |

| 3016 | 0.74 | 0.65 | regular price |

| 3017 | 4.7 | 4.12 | regular price |

| 3018 | 0.15 | 0.13 | regular price |

| 3019 | 2.71 | 2.38 | regular price |

| 3020 | 0.04 | 0.04 | regular price |

| 3021 | 3.01 | 2.64 | regular price |

| 3022 | 4.83 | 4.23 | regular price |

| 3023 | 0.32 | 0.28 | regular price |

| 3024 | 0.92 | 0.81 | regular price |

| 3025 | 3.08 | 2.7 | regular price |

| 3026 | 3.36 | 2.95 | regular price |

| 3027 | 4.82 | 4.22 | regular price |

| 3028 | 0.42 | 0.37 | regular price |

| 3029 | 3.83 | 3.36 | regular price |

| 3030 | 3.94 | 3.45 | regular price |

+---------+--------------+------------+---------------+

32 rows in set (0.00 sec)

**CONNECTION TABLE:**

mysql> describe bkp5190\_Link;

+-----------+--------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------+------+-----+---------+-------+

| cid | int(6) | NO | PRI | NULL | |

| iid | int(6) | NO | PRI | NULL | |

+-----------+--------+------+-----+---------+-------+

2 rows in set (0.00 sec)

mysql> select \* from bkp5190\_Link;

+-----------+--------+

| cid | iid |

+-----------+--------+

| 0 | 0 |

| 1000 | 2000 |

| 1000 | 2001 |

| 1000 | 2021 |

| 1000 | 2022 |

| 1000 | 2029 |

| 1000 | 2030 |

| 1001 | 2000 |

| 1001 | 2002 |

| 1001 | 2005 |

| 1001 | 2011 |

| 1001 | 2012 |

| 1001 | 2030 |

| 1002 | 2003 |

| 1002 | 2008 |

| 1002 | 2012 |

| 1003 | 2004 |

| 1003 | 2007 |

| 1003 | 2009 |

| 1003 | 2014 |

| 1003 | 2016 |

| 1003 | 2021 |

| 1003 | 2023 |

| 1003 | 2024 |

| 1003 | 2026 |

| 1003 | 2029 |

| 1004 | 2001 |

| 1004 | 2005 |

| 1004 | 2006 |

| 1004 | 2019 |

| 1004 | 2027 |

| 1004 | 2029 |

| 1004 | 2030 |

| 1005 | 2002 |

| 1005 | 2006 |

| 1005 | 2017 |

| 1005 | 2023 |

| 1005 | 2027 |

| 1006 | 2007 |

| 1006 | 2013 |

| 1006 | 2025 |

| 1006 | 2028 |

| 1007 | 2008 |

| 1008 | 2009 |

| 1008 | 2010 |

| 1008 | 2013 |

| 1008 | 2014 |

| 1008 | 2015 |

| 1008 | 2025 |

| 1008 | 2026 |

| 1009 | 2015 |

| 1009 | 2017 |

| 1009 | 2022 |

| 1010 | 2000 |

| 1010 | 2004 |

| 1010 | 2016 |

| 1010 | 2018 |

| 1010 | 2020 |

| 1010 | 2026 |

| 1010 | 2028 |

| 1011 | 2020 |

| 1011 | 2027 |

| 1012 | 2003 |

| 1012 | 2007 |

| 1012 | 2020 |

| 1012 | 2022 |

| 1012 | 2024 |

| 1012 | 2028 |

| 1013 | 2003 |

| 1013 | 2004 |

| 1013 | 2005 |

+-----------+--------+

71 rows in set (0.00 sec)

**COMPANY TABLE:**

mysql> describe bkp5190\_company ;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| cid | int(6) | NO | PRI | NULL | |

| cuid | int(6) | NO | | NULL | |

| company\_name | varchar(25) | NO | | NULL | |

+-------------+-------------+------+-----+---------+-------+

3 rows in set (0.01 sec)

mysql> select \* from bkp5190\_company;

+-----------+-----------+--------------+

| cid | cuid | company\_name |

+-----------+-----------+--------------+

| 0 | 0 | company\_name |

| 1000 | 6000 | Lambda Inc. |

| 1001 | 6001 | Epsilon Inc. |

| 1002 | 6002 | Kappa Inc. |

| 1003 | 6003 | Xi Inc. |

| 1004 | 6004 | Gamma Inc. |

| 1005 | 6005 | Nu Inc. |

| 1006 | 6006 | Beta Inc. |

| 1007 | 6007 | Zeta Inc. |

| 1008 | 6008 | Mu Inc. |

| 1009 | 6009 | Iota Inc. |

| 1010 | 6010 | Alpha Inc. |

| 1011 | 6011 | Delta Inc. |

| 1012 | 6012 | Eta Inc. |

| 1013 | 6013 | Theta Inc. |

+-----------+-----------+--------------+

15 rows in set (0.00 sec)

**COUNTRY TABLE:**

mysql> describe bkp5190\_country;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| cuid | int(6) | NO | PRI | NULL | |

| country\_name | varchar(30) | NO | | NULL | |

| state\_name | varchar(5) | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> select \* from bkp5190\_country;

+-----------+-------------+-----------+

| cuid | country\_name | state\_name |

+-----------+-------------+-----------+

| 0 | Country | State |

| 6000 | Mexico | N/A |

| 6001 | USA | DE |

| 6002 | Canada | N/A |

| 6003 | USA | NJ |

| 6004 | USA | NY |

| 6005 | USA | PA |

+-----------+-------------+-----------+

7 rows in set (0.00 sec)

**STEP 4-10:** Paste the answers (SQL and the results) below to answer each of the questions given. Do not create any temporary tables in order to answer these.

**STEP 4:** List every country from which a vendor comes. List each country only once.

mysql> select distinct country\_name from bkp5190\_country;

+-------------+

| country\_name |

+-------------+

| Country |

| Mexico |

| USA |

| Canada |

+-------------+

**4 rows in set (0.00 sec)**

**STEP 5:** List every item and every vendor who supplies that item. Show the product (including the area\_name) and the vendor. Sort by product, area\_name, vendor.

mysql> Select item, area\_name, company\_name from bkp5190\_area\_name A, bkp5190\_item I, bkp5190\_Link T, bkp5190\_company C WHERE A.aid = I.aid AND I.iid = T.iid AND C.cid = T.cid ORDER BY item, area\_name, company\_name;

+----------------+--------------+--------------+

| item | area\_name | company\_name |

+----------------+--------------+--------------+

| 2% milk | dairy | Epsilon Inc. |

| 2% milk | dairy | Gamma Inc. |

| 2% milk | dairy | Lambda Inc. |

| apple | produce | Eta Inc. |

| apple | produce | Iota Inc. |

| apple | produce | Lambda Inc. |

| baked beans | canned foods | Mu Inc. |

| banana | produce | Epsilon Inc. |

| banana | produce | Kappa Inc. |

| cherry | produce | Alpha Inc. |

| cherry | produce | Delta Inc. |

| cherry | produce | Eta Inc. |

| chicken | meat & fish | Alpha Inc. |

| chicken | meat & fish | Gamma Inc. |

| chicken | meat & fish | Lambda Inc. |

| chicken | meat & fish | Theta Inc. |

| chicken | meat & fish | Xi Inc. |

| chicken breast | meat & fish | Delta Inc. |

| chicken breast | meat & fish | Gamma Inc. |

| chicken breast | meat & fish | Nu Inc. |

| corn | canned foods | Eta Inc. |

| corn | canned foods | Kappa Inc. |

| corn | canned foods | Mu Inc. |

| corn | canned foods | Theta Inc. |

| corn | canned foods | Xi Inc. |

| corn | produce | Alpha Inc. |

| corn | produce | Epsilon Inc. |

| corn | produce | Lambda Inc. |

| dozen eggs | dairy | Iota Inc. |

| dozen eggs | dairy | Mu Inc. |

| eggplant | produce | Beta Inc. |

| eggplant | produce | Eta Inc. |

| eggplant | produce | Xi Inc. |

| hamburger | meat & fish | Epsilon Inc. |

| hamburger | meat & fish | Nu Inc. |

| ice cream | frozen | Nu Inc. |

| ice cream | frozen | Xi Inc. |

| lettuce | produce | Gamma Inc. |

| lettuce | produce | Lambda Inc. |

| lettuce | produce | Xi Inc. |

| pear | produce | Alpha Inc. |

| pear | produce | Mu Inc. |

| pear | produce | Xi Inc. |

| pizza | frozen | Kappa Inc. |

| pizza | frozen | Zeta Inc. |

| plum | produce | Alpha Inc. |

| plum | produce | Beta Inc. |

| plum | produce | Eta Inc. |

| salmon | meat & fish | Beta Inc. |

| salmon | meat & fish | Mu Inc. |

| skim milk | dairy | Beta Inc. |

| skim milk | dairy | Mu Inc. |

| spinach | canned foods | Iota Inc. |

| spinach | canned foods | Mu Inc. |

| spinach | canned foods | Nu Inc. |

| spinach | canned foods | Xi Inc. |

| spinach | frozen | Lambda Inc. |

| spinach | frozen | Xi Inc. |

| steak | meat & fish | Eta Inc. |

| steak | meat & fish | Xi Inc. |

| tomato | produce | Epsilon Inc. |

| tuna | meat & fish | Alpha Inc. |

| tuna | meat & fish | Gamma Inc. |

| tuna | meat & fish | Nu Inc. |

| tuna | meat & fish | Xi Inc. |

| waffles | frozen | Gamma Inc. |

| whole milk | dairy | Alpha Inc. |

| yogurt | dairy | Epsilon Inc. |

| yogurt | dairy | Gamma Inc. |

| yogurt | dairy | Theta Inc. |

+----------------+--------------+--------------+

71 rows in set (0.00 sec)

**STEP 6:** If you bought one of every frozen item, how much would you spend? (This must be done with a single select, the result of which is the total.)

mysql> Select SUM(reg\_price) from bkp5190\_sales S, bkp5190\_item I, bkp5190\_area\_name A WHERE S.sid = I.sid and A.aid = I.aid AND A.area\_name = "frozen";

+-------------------+

| SUM(reg\_price) |

+-------------------+

| 8.81 |

+-------------------+

1 row in set (0.00 sec)

**STEP 7:** List every product that is on sale.

mysql> Select item from bkp5190\_item I, bkp5190\_sales S WHERE I.sid = S.sid AND S.OnSale = "on sale";

+-------------+

| item |

+-------------+

| hamburger |

| yogurt |

| baked beans |

| banana |

+-------------+

4 rows in set (0.00 sec)

**STEP 8:** List every product that might have come from New Jersey. Show the product, the area\_name, and the price.

mysql> Select item, area\_name, IF(S.OnSale = "on sale", S.sale\_price, S.reg\_price) as "price" from bkp5190\_item I, bkp5190\_sales S, bkp5190\_area\_name A, bkp5190\_Link T, bkp5190\_company C, bkp5190\_country K WHERE A.aid = I.aid AND S.sid = I.sid AND I.iid = T.iid AND T.cid = C.cid AND C.cuid = K.cuid AND K.state\_name = "NJ";

+----------------+-------------+-------+

| item | area\_name | price |

+----------------+-------------+-------+

| yogurt | dairy | 2.73 |

| 2% milk | dairy | 3.94 |

| waffles | frozen | 2.71 |

| chicken | meat & fish | 0.07 |

| tuna | meat & fish | 3.22 |

| chicken breast | meat & fish | 4.82 |

| lettuce | produce | 3.83 |

+----------------+-------------+-------+

7 rows in set (0.01 sec)

**STEP 9:** Find all of the products that exist in more than one area\_name; show the product and the area\_names, sorted by product, area\_name.*The first few lines should look like this:*

+---------+--------------+

| product | area\_name |

+---------+--------------+

| chicken | canned foods |

| chicken | meat & fish |

mysql> Select item, area\_name, COUNT(area\_name) from bkp5190\_area\_name A, bkp5190\_item I WHERE A.aid = I.aid GROUP BY item, area\_name HAVING COUNT(area\_name) > 1;

+---------+--------------+-------------+

| item | area\_name | COUNT(area\_name) |

+---------+--------------+-------------+

| chicken | meat & fish | 2 |

| corn | canned foods | 2 |

| spinach | canned foods | 2 |

| tuna | meat & fish | 2 |

+---------+--------------+-------------+

4 rows in set (0.00 sec)

**STEP 10:** In one command, determine the single area\_name name which has the most items in it. The answer should be ONE of ["canned foods", "meat & fish", "produce", "dairy", "frozen"].

mysql> SELECT area\_name, COUNT(\*) as count FROM bkp5190\_area A, bkp5190\_item I WHERE A.aid = I.aid GROUP BY area\_name ORDER BY count DESC LIMIT 1;

+---------+------------+

| area | count |

+---------+------------+

| produce | 9 |

+---------+------------+

1 row in set (0.00 sec)