

1. Sum the total number of customers who live on a “Street”

```
mysql> select count(Address) Total_Street
-> from Customer
-> where Address like '%St%' or Address like '%Street%' or
-> Address like '%st%' or Address like '%street%';
```

```
+-----+
| Total_Street |
+-----+
|          4   |
+-----+
1 row in set (0.00 sec)
```

2. Find (group by) the total number of produce by type

```
mysql> select sum(Produce.Inventory) Total_Produce, Produce.Type
-> from Produce
-> group by Produce.Type;
```

```
+-----+-----+
| Total_Produce | Type      |
+-----+-----+
|          134 | dairy    |
|          200 | fruit    |
|           79 | meat     |
|          166 | vegetable|
+-----+-----+
4 rows in set (0.00 sec)
```

3. Select farmworkers who are assigned to work with “Pollinating Bees”

```
mysql> select Employee_ID, Position, RFID_Assignment, Salary,
        Experience
-> from BeeHive inner join FarmWorker
-> on FarmWorker.RFID_Assignment = BeeHive.RFID
-> where Bee like '%Honey Bees%';
```

```
+-----+-----+-----+-----+-----+
| Employee_ID | Position | RFID_Assignment | Salary   | Experience |
+-----+-----+-----+-----+-----+
|      78901  | Laborer  |          119    | 52000.00 |          2 |
|      89012  | Manager  |          120    | 32000.75 |          3 |
|      90123  | Manager  |          121    | 90750.25 |         20 |
|      45678  | Laborer  |          125    | 42000.00 |          3 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

4. Select customers ordered by harvest date of purchased produce.

```
mysql> select Harvest,Social_Security_Number, Receipt, Name, Sex, Birthday, Address,
        Telephone
        -> from Customer inner join Produce
        -> on Customer.Social_Security_Number = Produce.Buyer
        -> order by Harvest;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| Harvest | Social_Security_Number | Receipt | Name | Sex | Birthday | Address |
| Telephone |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 2021-02-25 14:12:00 | 66446648 | 1240 | Meka | f | 2003-10-18 | 1 TheFarm Rd, 39 Cheeny Pl |
1225541 |
| 2021-02-25 17:00:00 | 12345678 | 3951 | Mary | f | 1978-08-15 | 12 HighWater St, 1 Ham St |
2508881 |
| 2021-02-25 18:39:00 | 66446648 | 1240 | Meka | f | 2003-10-18 | 1 TheFarm Rd, 39 Cheeny Pl |
1225541 |
| 2021-03-20 01:42:00 | 23456789 | NULL | Suzie | f | 1995-11-16 | 15 Butterfly Rd |
7442112 |
| 2021-03-20 13:00:00 | 55555555 | 5519 | Amanda | f | 2002-04-02 | 12 Miami Ln |
1231231 |
| 2021-03-20 14:30:00 | 23456789 | NULL | Suzie | f | 1995-11-16 | 15 Butterfly Rd |
7442112 |
| 2021-03-21 05:00:00 | 55555555 | 5519 | Amanda | f | 2002-04-02 | 12 Miami Ln |
1231231 |
| 2021-03-21 07:00:00 | 23456789 | NULL | Suzie | f | 1995-11-16 | 15 Butterfly Rd |
7442112 |
| 2021-03-22 09:00:00 | 22222222 | NULL | Phil | NULL | 1964-04-25 | 412 Chaplin Circle, 1118 5 St APT 12 | 8338605
4917505, 1214441 |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

5. Count all farmworkers who are assigned to work with "Poultry"

```
mysql> select count(Employee_ID) Total_Chicken_Farmer
        -> from FarmWorker inner join LiveStock
        -> on FarmWorker.Employee_ID = LiveStock.Caretaker_ID
        -> where Type like "%poultry%" or Type like "%Poultry%";
```

```
+-----+
| Total_Chicken_Farmer |
+-----+
| 6 |
+-----+
1 row in set (0.00 sec)
```

6. Find the average experience of farmworkers who work with “Poultry”

```
mysql> select avg(Experience)
Average_Chicken_Farmer_Experience
-> from FarmWorker inner join LiveStock
-> on FarmWorker.Employee_ID = LiveStock.Caretaker_ID
-> where Type like "%poultry%" or Type like "%Poultry%";
```

```
+-----+
| Average_Chicken_Farmer_Experience |
+-----+
|                                1.5000 |
+-----+
1 row in set (0.01 sec)
```

7. List the employee ID of farmworkers who work with “seed fed” livestock, ordered by the price of produce they harvested.

```
mysql> select Employee_ID, Produce.Price
-> from FarmWorker inner join LiveStock inner join Produce
-> on FarmWorker.Employee_ID = LiveStock.Caretaker_ID and
   FarmWorker.Employee_ID = Produce.HarvestedBy
-> where Feed like "%Seed%" or Feed like "%seed%"
-> order by Produce.Price;
```

```
+-----+-----+
| Employee_ID | Price |
+-----+-----+
|      78901 | 1.75  |
|      78901 | 1.75  |
|      78901 | 1.75  |
|      34567 | 2.99  |
|      78901 | 4.99  |
|      78901 | 4.99  |
|      78901 | 4.99  |
+-----+-----+
7 rows in set (0.00 sec)
```

8. Find the total number of customers who purchased produce valued more than 2\$ harvested by farmworkers that are “managers”

```
mysql> select count(Customer.Social_Security_Number)
        Total_Customers_2_or_More
-> from Customer inner join Produce inner join FarmWorker
-> on Customer.Social_Security_Number = Produce.Buyer and
        Produce.HarvestedBy = FarmWorker.Employee_ID
-> where FarmWorker.Position like "%Manager%" or
        FarmWorker.Position like "%manager%";
```

```
+-----+
| Total_Customers_2_or_More |
+-----+
|                          3 |
+-----+
1 row in set (0.00 sec)
```

9. Finds the average price of produce, harvested by paid farmworkers who also care to “poultry” type livestock.

```
mysql> select avg(Produce.Price) Average_Price
-> from FarmWorker inner join LiveStock inner join Produce
-> on FarmWorker.Employee_ID = LiveStock.Caretaker_ID and
        FarmWorker.Employee_ID = Produce.HarvestedBy
-> where LiveStock.Type like "%poultry%" or LiveStock.Type
        like "%Poultry%" and FarmWorker.Salary > 0;
```

```
+-----+
| Average_Price |
+-----+
|      2.920000 |
+-----+
1 row in set (0.00 sec)
```

- 10. Obtain the customers ssn as well as the employees ID (concat) such that the purchased produce was handled by both parties**

```
mysql> select concat(concat(Customer.Social_Security_Number,
    "-"), FarmWorker.Employee_ID) Customer_Employee_ID
    -> from Customer inner join FarmWorker inner join Produce
    -> on Customer.Social_Security_Number = Produce.Buyer and
    FarmWorker.Employee_ID = Produce.HarvestedBy
    -> where Customer.Receipt >= 0;
```

```
+-----+
| Customer_Employee_ID |
+-----+
| 12345678-78901      |
| 33333333-89012      |
| 55555555-34567      |
| 55555555-0          |
| 66446648-89012      |
| 66446648-78901      |
+-----+
6 rows in set (0.01 sec)
```

- 11. Obtain the livestock ID as well as the produce's barcode (concat) such that the farmworker who worked with both has more than 2 years of experience**

```
mysql> select concat(concat(LiveStock.ID_Number, "-"),
    Produce.Bar_Code) LiveStock_Produce_ID
    -> from FarmWorker inner join Produce inner join LiveStock
    -> on FarmWorker.Employee_ID = LiveStock.Caretaker_ID and
    FarmWorker.Employee_ID = Produce.HarvestedBy
    -> where FarmWorker.Experience > 2;
```

```
+-----+
| LiveStock_Produce_ID |
+-----+
| 222-4117              |
+-----+
1 row in set (0.00 sec)
```

12. Select customers who are female, and purchased produce harvested by a farm worker who works with both “Honey Bees” and “Poultry”, and has at least 1 year of experience.

```
mysql> select Social_Security_Number, Receipt, Name, Sex,
        Birthday, Address, Telephone
-> from Customer inner join FarmWorker inner join BeeHive
        inner join LiveStock inner join Produce
-> on Produce.HarvestedBy = FarmWorker.Employee_ID and
        LiveStock.Caretaker_ID = FarmWorker.Employee_ID and
        Customer.Social_Security_Number = Produce.Buyer and
        FarmWorker.RFID_Assignment = BeeHive.RFID
-> where Customer.Sex IS NOT NULL and Customer.Sex like
        "%f%" and FarmWorker.Experience >= 1 and BeeHive.Bee
        like '%Honey Bees%' and LiveStock.Type like "%poultry%"
        or LiveStock.Type like "%Poultry%";
```

Social_Security_Number	Receipt	Name	Sex	Birthday	Address	Telephone
55555555	5519	Amanda	f	2002-04-02	12 Miami Ln	1231231
12345678	3951	Mary	f	1978-08-15	12 HighWater St, 1 Ham St	2508881
66446648	1240	Meka	f	2003-10-18	1 TheFarm Rd, 39 Cheeny Pl	1225541
12345678	3951	Mary	f	1978-08-15	12 HighWater St, 1 Ham St	2508881
66446648	1240	Meka	f	2003-10-18	1 TheFarm Rd, 39 Cheeny Pl	1225541
23456789	NULL	Suzie	f	1995-11-16	15 Butterfly Rd	7442112
22222222	NULL	Phil	NULL	1964-04-25	412 Chaplin Circle, 1118 5 St APT 12	8338605
23456789	NULL	Suzie	f	1995-11-16	15 Butterfly Rd	7442112
12345678	3951	Mary	f	1978-08-15	12 HighWater St, 1 Ham St	2508881
66446648	1240	Meka	f	2003-10-18	1 TheFarm Rd, 39 Cheeny Pl	1225541

10 rows in set (0.00 sec)

- 13. Obtain Livestock IDs and Customer SSNs (concat) for produce harvested from a farmworker who gets paid more than \$15,000 and works with healthy “Honey Bees.”**

```
mysql> select concat(concat(LiveStock.ID_Number, "-"),
    Customer.Social_Security_Number) LiveStock_Customer_ID
-> from LiveStock inner join Customer inner join
    FarmWorker inner join BeeHive inner join Produce
-> on Produce.HarvestedBy = FarmWorker.Employee_ID and
    LiveStock.Caretaker_ID = FarmWorker.Employee_ID and
    Customer.Social_Security_Number = Produce.Buyer and
    FarmWorker.RFID_Assignment = BeeHive.RFID
-> where FarmWorker.Salary > 15000 and
    BeeHive.Indisposition = 0;
```

```
+-----+
| LiveStock_Customer_ID |
+-----+
| 223-12345678          |
| 224-12345678          |
| 240-12345678          |
| 417-12345678          |
| 223-66446648          |
| 224-66446648          |
| 240-66446648          |
| 417-66446648          |
| 222-55555555          |
+-----+
9 rows in set (0.00 sec)
```

14. Find the count of beehives, whose farmworkers harvested produce valued at more than \$2, and were purchased by a customer who lives on a “Street.”

```
mysql> select count(BeeHive.RFID)
-> from FarmWorker inner join BeeHive inner join Customer
inner join Produce
-> on Produce.HarvestedBy = FarmWorker.Employee_ID and
Customer.Social_Security_Number = Produce.Buyer and
FarmWorker.RFID_Assignment = BeeHive.RFID
-> where Produce.Price > 2 and Customer.Address like
'%St%' or Customer.Address like '%Street%' or
Customer.Address like '%st%' or Customer.Address like
'%street%';
```

```
+-----+
| count(BeeHive.RFID) |
+-----+
|                      2 |
+-----+
1 row in set (0.00 sec)
```

15. Sum the total number of produce that is worth more than \$1.99.

```
mysql> select sum(Produce.Inventory)
      Inventory_Greater_Than_ABuck99
-> from Produce
-> where Produce.Price > 1.99;
```

```
+-----+  
| Inventory_Greater_Than_ABuck99 |  
+-----+  
|                               326 |  
+-----+  
1 row in set (0.00 sec)
```