A-w1 exercises in SQL

Give SQL expressions for each of the following queries specified in English (assuming the three tables, with all rows, of the earlier RA handout).

1. Write a query that returns the Name and Phone of all Persons owning VRN=123.

Style I like most (consistent use of row variables, such as O and P)

```
SELECT P.Name, P.Phone
FROM Own O, Person P
WHERE O.VRN = 123 AND O.SSN = P.SSN;
```

but other variants possible (recognizing where there is no confusion between attributes across tables in many cases), such as

```
SELECT Name, Phone
FROM Own, Person
WHERE VRN = 123 AND Own.SSN = Person.SSN;
```

WHERE conditions can be listed in any order, though often I will put conditions involving only one table first.

2. Write a query to show the VRN and Mo of each owned Vehicle and the Name and Addr of the Person who owns it.

```
SELECT V.VRN, V.Mo, P.Name, P.Addr    // or O.VRN
FROM Vehicle V, Own O, Person P
WHERE V.VRN = O.VRN AND O.SSN = P.SSN;
```

3. Write a query to show the VRN and Mo of each owned Vehicle BY SOMEONE ON 'Birch' and the Name of the Person who owns it. Write versions of the query that DO and do NOT use a subquery.

//Not using subquery:

```
SELECT V.VRN, V.Mo, P.Name
FROM Vehicle V, Own O, Person P
WHERE V.VRN = O.VRN AND O.SSN = P.SSN AND P.Addr = 'Birch';
```

--Using subqueries (but the above query is better – much simpler):

```
SELECT V.VRN, V.Mo, Temp.Name
FROM
    Vehicle V, Own O,
    (SELECT P.Name, P.SSN FROM Person P WHERE P.Addr = 'Birch')
        AS Temp
WHERE V.VRN = O.VRN AND O.SSN = Temp.SSN;
```

Others possible. You can use sub-queries to help break a problem up, solving a simplified problem, and then building on the solution of that simplified problem

4. Write a query that returns the VRN and Mo of any vehicle that isn't owned.

```
SELECT V.VRN, V.Mo
FROM Vehicle V
WHERE V.VRN NOT IN (SELECT O.VRN FROM Own O);
```

You could also use the EXCEPT operator

```
SELECT V.VRN, V.Mo
FROM Vehicle V
EXCEPT
SELECT V.VRN, V.Mo
FROM Vehicle V, Own O
WHERE V.VRN = O.VRN;
```

```
SELECT V.VRN, V.Mo
FROM Vehicle V
EXCEPT
SELECT V.VRN, V.Mo
FROM Vehicle V
WHERE V.VRN IN (SELECT O.VRN FROM Own O);
```

5. Write a query that returns pairs of SSNs for **(different)** Persons that live at the same Addr.

```
SELECT P1.SSN, P2.SSN
FROM Person P1, Person P2
WHERE P1.Addr = P2.Addr AND
        P1.SSN <> P2.SSN; //example of non-equality join condition
```

This query will list abc, efg AND efg, abc (for example)
If SSN of appropriate type, then < (or >) will list only one of these (e.g., abc,efg):

6. Write a query that returns pairs of Names for (different) Persons that live at the same Addr.

```
SELECT P1.Name, P2.Name     //names could be the same
FROM Person P1, Person P2
WHERE P1.Addr = P2.Addr AND P1.SSN < P2.SSN;
```

7. Write a query that lists pairs of Vehicles by Ma and Mo that are owned by the same Person.

```
SELECT V1.Ma, V1.Mo, V2.Ma V2.Mo
FROM Vehicle V1, Vehicle V2, Own O1, Own O2, Person P
WHERE V1.VRN = O1.VRN AND O1.SSN = P.SSN AND
        P.SSN = O2.SSN AND O2.VRN = V2.VRN AND
        O1.VRN < O2.VRN // need this or other variation

SELECT V1.Ma, V1.Mo, V2.Ma V2.Mo
FROM Vehicle V1, Vehicle V2, Own O1, Own O2
WHERE V1.VRN = O1.VRN AND O1.SSN = O2.SSN AND
        O2.VRN = V2.VRN AND O1.VRN < O2.VRN //could also
                compare V1.VRN and V2.VRN
```

8. Write a query that lists pairs of Persons by SSN and Name that own a Vehicle of the same Ma, Mo and Color

```
SELECT P1.SSN, P1.Name, P2.SSN, P2.Name
FROM Person P1, Person P2, Own O1, Own O2,
        Vehicle V1, Vehicle V2
WHERE P1.SSN < P2.SSN // need this
    AND P1.SSN = O1.SSN AND O1.VRN = V1.VRN
    AND V1.VRN <> V2.VRN  //need this. Why not '<'?
    AND V1.Ma = V2.Ma AND V1.Mo = V2.Mo AND V1.Color = V2.Color
```

9. Suppose that there was a Price attribute on Vehicle. Write a query in which each car is listed (paired) with each other car that costs less than it does.

```
SELECT V1.VRN, V2.VRN
FROM Vehicle V1, Vehicle V2
WHERE V1.Price < V2.Price
```

Other
Write a query that returns the Name and Phone of all Persons owning a Ford.

```
SELECT P.Name, P.Phone
FROM Vehicle V, Own O, Person P
WHERE V.Ma = 'Ford' AND V.VRN = O.VRN AND O.SSN = P.SSN;
```

Write a query that returns the Models and Colors of all vehicles owned by someone on Birch.

```
SELECT V.Mo, V.Color
FROM Vehicle V, Own O, Person P
WHERE P.Addr = 'Birch' AND P.SSN = O.SSN AND V.VRN = O.VRN;
```

Write a query that returns the Name, Addr and Phone of any Person that doesn't own any Vehicle.

```
SELECT P.Name, P.Addr, P.Phone
FROM Person P
WHERE P.SSN NOT IN (SELECT O.SSN FROM Own O);
```

```
SELECT P1.SSN, P2.SSN
FROM Person P1, Person P2
WHERE P1.Addr = P2.Addr AND P1.SSN < P2.SSN;  // '<'
```

Even if you didn't catch on with previous problems, this problem was to simply highlight that not all joins are equality joins