1. Calculate the usage of Irish road network in terms of percentage grouped by vehicle category.

Code -

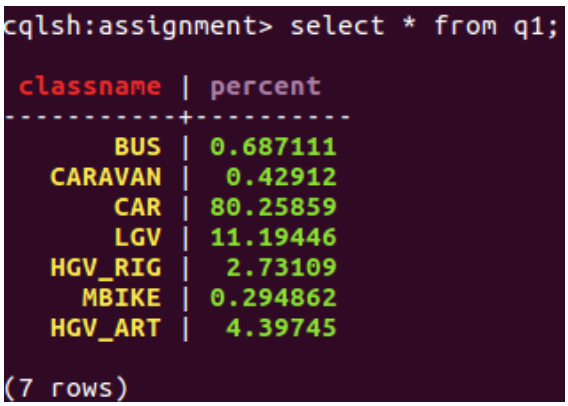
```
In [9]: total_vehicles = df.count()

q1DF = (df
        .groupBy('classname')
        .count()
        .withColumn('percent', (col('count') * 100) / total_vehicles)
        .select('classname', 'percent')
        .orderBy(desc('percent')))
|
q1DF.show()

saveToCassandra(q1DF.filter(col("classname").isNotNull()), "q1")
```

```
+---------+--------------------+
|classname|             percent|
+---------+--------------------+
|      CAR|   80.25858594040197|
|      LGV|  11.194464465420944|
|  HGV_ART|   4.397450167807071|
|  HGV_RIG|  2.7310861887745705|
|      BUS|  0.6871114761643508|
|  CARAVAN| 0.42912036442325563|
|    MBIKE| 0.29486205142905475|
|     null|0.007319345578788326|
+---------+--------------------+
```


```
In [9]: total_vehicles = df.count()

q1DF = (df
        .groupBy('classname')
        .count()
        .withColumn('percent', (col('count') * 100) / total_vehicles)
        .select('classname', 'percent')
        .orderBy(desc('percent')))
|
q1DF.show()

saveToCassandra(q1DF.filter(col("classname").isNotNull()), "q1")
```

```
+---------+--------------------+
|classname|             percent|
+---------+--------------------+
|      CAR|   80.25858594040197|
|      LGV|  11.194464465420944|
|  HGV_ART|   4.397450167807071|
|  HGV_RIG|  2.7310861887745705|
|      BUS|  0.6871114761643508|
|  CARAVAN| 0.42912036442325563|
|    MBIKE| 0.29486205142905475|
|     null|0.007319345578788326|
+---------+--------------------+
```


Output -



```
cqlsh:assignment> select * from q1;

 classname | percent
-----------+----------
       BUS | 0.687111
   CARAVAN |  0.42912
       CAR | 80.25859
       LGV | 11.19446
   HGV_RIG |  2.73109
     MBIKE | 0.294862
   HGV_ART |  4.39745

(7 rows)
```

2. Calculate the highest and lowest hourly flows on M50 - show the hours and total number of vehicle counts.

Code -

```
In [11]: hourDF = df.groupBy('hour').count()

maxDF = hourDF.orderBy(desc('count')).limit(1)
minDF = hourDF.orderBy('count').limit(1)

maxDF.show()
minDF.show()

saveToCassandra(maxDF, "q2_max")
saveToCassandra(minDF, "q2_min")
```

```
+----+------+
|hour| count|
+----+------+
|  16|385850|
+----+------+

+----+-----+
|hour|count|
+----+-----+
|   2|13682|
+----+-----+
```

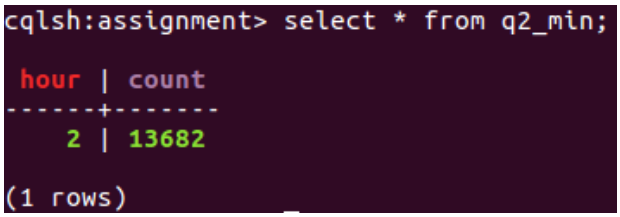

```
In [11]: hourDF = df.groupBy('hour').count()

maxDF = hourDF.orderBy(desc('count')).limit(1)
minDF = hourDF.orderBy('count').limit(1)

maxDF.show()
minDF.show()

saveToCassandra(maxDF, "q2_max")
saveToCassandra(minDF, "q2_min")
```

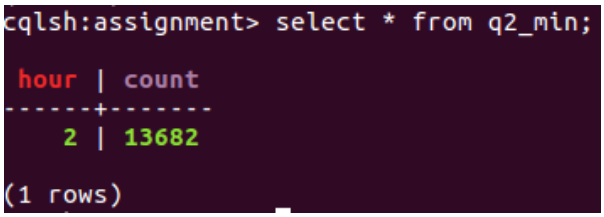
```
+----+------+
|hour| count|
+----+------+
|  16|385850|
+----+------+

+----+-----+
|hour|count|
+----+-----+
|   2|13682|
+----+-----+
```


Output -

```
cqlsh:assignment> select * from q2_max;

 hour | count
------+--------
   16 | 385850

(1 rows)
```


```
cqlsh:assignment> select * from q2_max;

 hour | count
------+--------
   16 | 385850

(1 rows)
```





3. Calculate the evening and morning rush hours on M50 - show the hours and the total counts.

Code -

```
In [13]: hourDF = df.groupBy('hour').count().orderBy('hour')

         morningDF = hourDF.filter((col('hour') < 12) & (col('hour') >= 4))
         eveningDF = hourDF.filter((col('hour') < 20) & (col('hour') >= 16))

         morningDF.show()
         eveningDF.show()

         saveToCassandra(morningDF, "q3_morning")
         saveToCassandra(eveningDF, "q3_evening")
```

```
+----+------+
|hour| count|
+----+------+
|   4| 27187|
|   5| 61937|
|   6|198369|
|   7|299784|
|   8|352862|
|   9|277509|
|  10|256183|
|  11|246847|
+----+------+

+----+------+
|hour| count|
+----+------+
|  16|385850|
|  17|367269|
|  18|314085|
|  19|232409|
```


```
In [13]: hourDF = df.groupBy('hour').count().orderBy('hour')

         morningDF = hourDF.filter((col('hour') < 12) & (col('hour') >= 4))
         eveningDF = hourDF.filter((col('hour') < 20) & (col('hour') >= 16))

         morningDF.show()
         eveningDF.show()

         saveToCassandra(morningDF, "q3_morning")
         saveToCassandra(eveningDF, "q3_evening")
```

```
+----+------+
|hour| count|
+----+------+
|   4| 27187|
|   5| 61937|
|   6|198369|
|   7|299784|
|   8|352862|
|   9|277509|
|  10|256183|
|  11|246847|
+----+------+

+----+------+
|hour| count|
+----+------+
|  16|385850|
|  17|367269|
|  18|314085|
|  19|232409|
```

Output -

```
cqlsh:assignment> select * from q3_morning ;

 hour | count
------+--------
    5 |  61937
   10 | 256183
   11 | 246847
    8 | 352862
    4 |  27187
    7 | 299784
    6 | 198369
    9 | 277509

(8 rows)
```


```
cqlsh:assignment> select * from q3_morning ;

 hour | count
------+--------
    5 |  61937
   10 | 256183
   11 | 246847
    8 | 352862
    4 |  27187
    7 | 299784
    6 | 198369
    9 | 277509

(8 rows)
```


```
cqlsh:assignment> select * from q3_evening  ;

 hour | count
------+--------
   16 | 385850
   19 | 232409
   18 | 314085
   17 | 367269

(4 rows)
```


```
cqlsh:assignment> select * from q3_evening  ;

 hour | count
------+--------
   16 | 385850
   19 | 232409
   18 | 314085
   17 | 367269

(4 rows)
```


4. Calculate average speed between each junction on M50 (e.g., junction 1 - junction 2, junction 2 - junction 3, etc.).

Code -

In [15]:
```
q4DF = (df
        .groupBy('lanename')
        .agg(mean('speed').alias("avg_speed"))
        .orderBy(desc('avg_speed')))

q4DF.show()

saveToCassandra(q4DF, "q4")
```

```
+--------------------+------------------+
|            lanename|         avg_speed|
+--------------------+------------------+
| Southbound 1 (slow)| 135.4469130170314|
|       Northbound 2 |122.31002458344715|
|        Eastbound  2|114.68716172331673|
|  Eastbound 2 (fast)|113.59000942507083|
| Southbound 2 (fast)|111.72458022387893|
|  Westbound 2 (fast)|111.34068965517257|
| Northbound 2 (fast)|110.21109738884894|
|          southbound| 104.4090909090909|
|         Nortbound 1|103.95987028779895|
| Northbound 1 (slow)|103.63843987902779|
|  Westbound 3 (fast)|103.47554310278697|
|  Eastbound 3 (fast)|100.55896097639352|
|        Southbound 2| 97.79121728990314|
|        Northbound 2| 97.76209841746629|
|  Westbound 2 (slow)| 95.40281196241926|
|Southbound Mainli...| 95.25522388059701|
|         Westbound 2| 93.36724880445983|
|         Eastbound 2| 92.79648071706569|
|   Eastbound on slip|  92.7741935483871|
```


In [15]:
```
q4DF = (df
        .groupBy('lanename')
        .agg(mean('speed').alias("avg_speed"))
        .orderBy(desc('avg_speed')))

q4DF.show()

saveToCassandra(q4DF, "q4")
```

```
+--------------------+------------------+
|            lanename|         avg_speed|
+--------------------+------------------+
| Southbound 1 (slow)| 135.4469130170314|
|       Northbound 2 |122.31002458344715|
|        Eastbound  2|114.68716172331673|
|  Eastbound 2 (fast)|113.59000942507083|
| Southbound 2 (fast)|111.72458022387893|
|  Westbound 2 (fast)|111.34068965517257|
| Northbound 2 (fast)|110.21109738884894|
|          southbound| 104.4090909090909|
|         Nortbound 1|103.95987028779895|
| Northbound 1 (slow)|103.63843987902779|
|  Westbound 3 (fast)|103.47554310278697|
|  Eastbound 3 (fast)|100.55896097639352|
|        Southbound 2| 97.79121728990314|
|        Northbound 2| 97.76209841746629|
|  Westbound 2 (slow)| 95.40281196241926|
|Southbound Mainli...| 95.25522388059701|
|         Westbound 2| 93.36724880445983|
|         Eastbound 2| 92.79648071706569|
|   Eastbound on slip|  92.7741935483871|
```


Output -

```
cqlsh:assignment> select * from q4 limit 10;

 lanename              | avg_speed
-----------------------+-----------
  Southbound Off Slip |   75.51857
    Eastbound On Slip |   83.44571
   Eastbound 2 (fast) |  113.59001
      Eastbound 1 BUS |         53
 Southbound Off Right |   46.24775
   Northbound On Left |   45.20036
         Westbound HS |   60.72727
          Southbound  |   62.18173
 Southbound Left Turn |   40.61792
   On Slip Northbound |   55.37979

(10 rows)
```

```
cqlsh:assignment> select * from q4 limit 10;

 lanename              | avg_speed
-----------------------+-----------
  Southbound Off Slip |  75.51857
    Eastbound On Slip |  83.44571
   Eastbound 2 (fast) | 113.59001
      Eastbound 1 BUS |        53
 Southbound Off Right |  46.24775
   Northbound On Left |  45.20036
         Westbound HS |  60.72727
          Southbound  |  62.18173
 Southbound Left Turn |  40.61792
   On Slip Northbound |  55.37979

(10 rows)
```

5. Calculate the top 10 locations with highest number of counts of HGVs (class). Map the COSITs with their names given on the map.

Code -

```
In [17]: q5DF = (df
                 .filter(col('classname').contains('HGV'))
                 .groupBy('lanename')
                 .agg(mean('cosit').alias('avg_cosit'), count('lanename').alias('count'))
                 .orderBy(desc('count')))

         q5DF.show(10)

         saveToCassandra(q5DF, "q5")
```

```
+------------+------------------+-----+
|    lanename|         avg_cosit|count|
+------------+------------------+-----+
|Northbound 1|20693.790824685962|47606|
|Southbound 1| 20370.47938361651|47438|
| Westbound 1| 47984.26970280579|26481|
| Eastbound 1|50842.564949674364|25335|
|  Northbound|11535.248807024242|20956|
|  Southbound| 15721.02364244845|18526|
|Northbound 2| 6737.779673675744|17406|
|Southbound 2| 5250.837445297139|15767|
|   Eastbound|11569.320689406504|13867|
|   Westbound|11157.380619527628|13591|
+------------+------------------+-----+
only showing top 10 rows
```

```
In [17]: q5DF = (df
                 .filter(col('classname').contains('HGV'))
                 .groupBy('lanename')
                 .agg(mean('cosit').alias('avg_cosit'), count('lanename').alias('count'))
                 .orderBy(desc('count')))

         q5DF.show(10)

         saveToCassandra(q5DF, "q5")
```

```
+------------+------------------+-----+
|    lanename|         avg_cosit|count|
+------------+------------------+-----+
|Northbound 1|20693.790824685962|47606|
|Southbound 1| 20370.47938361651|47438|
| Westbound 1| 47984.26970280579|26481|
| Eastbound 1|50842.564949674364|25335|
|  Northbound|11535.248807024242|20956|
|  Southbound| 15721.02364244845|18526|
|Northbound 2| 6737.779673675744|17406|
|Southbound 2| 5250.837445297139|15767|
|   Eastbound|11569.320689406504|13867|
|   Westbound|11157.380619527628|13591|
+------------+------------------+-----+
only showing top 10 rows
```

Output -

lanename	avg_cosit	count
Southbound Off Slip	20117.11392	158
Eastbound On Slip	20078	44
Eastbound 2 (fast)	20042	412
Eastbound 1 BUS	1221	3
Southbound Off Right	1283	83
Northbound On Left	1283	24
Westbound HS	20511	5
Southbound	1036	134
Southbound Left Turn	20802	20
On Slip Northbound	3806	29

(10 rows)

lanename	avg_cosit	count
Southbound Off Slip	20117.11392	158
Eastbound On Slip	20078	44
Eastbound 2 (fast)	20042	412
Eastbound 1 BUS	1221	3
Southbound Off Right	1283	83
Northbound On Left	1283	24
Westbound HS	20511	5
Southbound	1036	134
Southbound Left Turn	20802	20
On Slip Northbound	3806	29

(10 rows)