

## Assignment 2

1. Show the busiest site in Ireland.

```
In [2]: vehicle_counter_DF.registerTempTable("transport")
```

```
In [3]: the busiest site in Ireland.
        = spark.sql("select cosit as busiest_site, count(*) as count from transport group by cosit order by count(cosit) desc limit 1");
        .show()

+-----+-----+
|busiest_site|count|
+-----+-----+
|1508|98292|
+-----+-----+
```

```
In [4]: busiest_site.select("busiest_site", "count")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="busiest_site", keyspace = "assignment2")\
.save(mode="append")
```

```
cqlsh:assignment2> create table busiest_site(busiest_site int, count int, primary key(busiest_site));
cqlsh:assignment2> select * from busiest_site;
```

| busiest_site | count |
|--------------|-------|
| 1508         | 98292 |

```
(1 rows)
```

2. Show the average distance between vehicles on all M50 sites.

```
In [26]: # 2.Show the average distance between vehicles on all M50 sites.

distance_between_vehicles = spark.sql("select AVG(gap) as avg_gap from transport where cosit in(1500,1501,1502,1503,1504,1505,1506,1507,1508,1509,1510,1511,1512,1513,1514,1515,1516,1517,1518,1519,1520,1521,1522,1523,1524,1525,1526,1527,1528,1529,1530,1531,1532,1533,1534,1535,1536,1537,1538,1539,1540,1541,1542,1543,1544,1545,1546,1547,1548,1549,1550,1551,1552,1553,1554,1555,1556,1557,1558,1559,1560,1561,1562,1563,1564,1565,1566,1567,1568,1569,1570,1571,1572,1573,1574,1575,1576,1577,1578,1579,1580,1581,1582,1583,1584,1585,1586,1587,1588,1589,1590,1591,1592,1593,1594,1595,1596,1597,1598,1599,1600,1601,1602,1603,1604,1605,1606,1607,1608,1609,1610,1611,1612,1613,1614,1615,1616,1617,1618,1619,1620,1621,1622,1623,1624,1625,1626,1627,1628,1629,1630,1631,1632,1633,1634,1635,1636,1637,1638,1639,1640,1641,1642,1643,1644,1645,1646,1647,1648,1649,1650,1651,1652,1653,1654,1655,1656,1657,1658,1659,1660,1661,1662,1663,1664,1665,1666,1667,1668,1669,1670,1671,1672,1673,1674,1675,1676,1677,1678,1679,1680,1681,1682,1683,1684,1685,1686,1687,1688,1689,1690,1691,1692,1693,1694,1695,1696,1697,1698,1699,1700,1701,1702,1703,1704,1705,1706,1707,1708,1709,1710,1711,1712,1713,1714,1715,1716,1717,1718,1719,1720,1721,1722,1723,1724,1725,1726,1727,1728,1729,1730,1731,1732,1733,1734,1735,1736,1737,1738,1739,1740,1741,1742,1743,1744,1745,1746,1747,1748,1749,1750,1751,1752,1753,1754,1755,1756,1757,1758,1759,1760,1761,1762,1763,1764,1765,1766,1767,1768,1769,1770,1771,1772,1773,1774,1775,1776,1777,1778,1779,1780,1781,1782,1783,1784,1785,1786,1787,1788,1789,1790,1791,1792,1793,1794,1795,1796,1797,1798,1799,1800,1801,1802,1803,1804,1805,1806,1807,1808,1809,1810,1811,1812,1813,1814,1815,1816,1817,1818,1819,1820,1821,1822,1823,1824,1825,1826,1827,1828,1829,1830,1831,1832,1833,1834,1835,1836,1837,1838,1839,1840,1841,1842,1843,1844,1845,1846,1847,1848,1849,1850,1851,1852,1853,1854,1855,1856,1857,1858,1859,1860,1861,1862,1863,1864,1865,1866,1867,1868,1869,1870,1871,1872,1873,1874,1875,1876,1877,1878,1879,1880,1881,1882,1883,1884,1885,1886,1887,1888,1889,1890,1891,1892,1893,1894,1895,1896,1897,1898,1899,1900,1901,1902,1903,1904,1905,1906,1907,1908,1909,1910,1911,1912,1913,1914,1915,1916,1917,1918,1919,1920,1921,1922,1923,1924,1925,1926,1927,1928,1929,1930,1931,1932,1933,1934,1935,1936,1937,1938,1939,1940,1941,1942,1943,1944,1945,1946,1947,1948,1949,1950,1951,1952,1953,1954,1955,1956,1957,1958,1959,1960,1961,1962,1963,1964,1965,1966,1967,1968,1969,1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,2040,2041,2042,2043,2044,2045,2046,2047,2048,2049,2050,2051,2052,2053,2054,2055,2056,2057,2058,2059,2060,2061,2062,2063,2064,2065,2066,2067,2068,2069,2070,2071,2072,2073,2074,2075,2076,2077,2078,2079,2080,2081,2082,2083,2084,2085,2086,2087,2088,2089,2090,2091,2092,2093,2094,2095,2096,2097,2098,2099,2100,2101,2102,2103,2104,2105,2106,2107,2108,2109,2110,2111,2112,2113,2114,2115,2116,2117,2118,2119,2120,2121,2122,2123,2124,2125,2126,2127,2128,2129,2130,2131,2132,2133,2134,2135,2136,2137,2138,2139,2140,2141,2142,2143,2144,2145,2146,2147,2148,2149,2150,2151,2152,2153,2154,2155,2156,2157,2158,2159,2160,2161,2162,2163,2164,2165,2166,2167,2168,2169,2170,2171,2172,2173,2174,2175,2176,2177,2178,2179,2180,2181,2182,2183,2184,2185,2186,2187,2188,2189,2190,2191,2192,2193,2194,2195,2196,2197,2198,2199,2200,2201,2202,2203,2204,2205,2206,2207,2208,2209,2210,2211,2212,2213,2214,2215,2216,2217,2218,2219,2220,2221,2222,2223,2224,2225,2226,2227,2228,2229,2230,2231,2232,2233,2234,2235,2236,2237,2238,2239,2240,2241,2242,2243,2244,2245,2246,2247,2248,2249,2250,2251,2252,2253,2254,2255,2256,2257,2258,2259,2260,2261,2262,2263,2264,2265,2266,2267,2268,2269,2270,2271,2272,2273,2274,2275,2276,2277,2278,2279,2280,2281,2282,2283,2284,2285,2286,2287,2288,2289,2290,2291,2292,2293,2294,2295,2296,2297,2298,2299,2300,2301,2302,2303,2304,2305,2306,2307,2308,2309,
```

```
In [29]: distance_between_vehicles.select('avg_gap')\
         .write.format("org.apache.spark.sql.cassandra")\
         .options(table="avg_gap", keyspace = "assignment2")\
         .save(mode="append")
```

```
(1 rows)
cqlsh:assignment2> create table avg_gap(avg_gap float, primary key(avg_gap));
cqlsh:assignment2> select * from avg_gap;

avg_gap
-----
4.27486
(1 rows)
```

3. What site has recorded the highest temperature? Show the hour of the day.

```
In [36]: # What site has recorded the highest temperature? Show the hour of the day.
sql("select cosit as site, hour,minute, MAX(temperature) as max_temp from transport group by cosit,hour,minute order by MAX(temper")
```

```
+-----+-----+-----+
|site|hour|minute|max_temp|
+-----+-----+-----+
|1015| 13| 4| 12.0|
|1015| 18| 47| 12.0|
|1015| 18| 48| 12.0|
|1015| 12| 32| 12.0|
|1015| 18| 49| 12.0|
+-----+-----+-----+
```

```
In [52]: Maximum_temperature.select("site", "hour", "minute", "max_temp")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="maximum_temperature", keyspace = "assignment2")\
.save(mode="append")
```

```
cqlsh:assignment2> create table Maximum_temperature(site int, hour int, minute int, max_temp float, primary key(
site));
cqlsh:assignment2> select * from maximum_temperature;

site | hour | max_temp | minute
-----+-----+-----+-----
(0 rows)
cqlsh:assignment2> select * from maximum_temperature;

site | hour | max_temp | minute
-----+-----+-----+-----
1015 | 18 | 12 | 49
(1 rows)
```

4. Show total number of WIM sites available in the dataset?

```
In [53]: # 4. Show total number of WIM sites available in the dataset?
```

```
Total_WIM_sites = spark.sql("select COUNT(DISTINCT cosit) as wim_site from transport group where weight IS NOT NULL");
Total_WIM_sites.show();
```

```
+-----+
|WIM_site|
+-----+
| 325|
+-----+
```

```
In [58]: Total_WIM_sites.select('wim_site')\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="total_wim_sites", keyspace = "assignment2")\
.save(mode="append")
```

```
cqlsh:assignment2> create table total_wim_sites(wim_site int, primary key(wim_site));
cqlsh:assignment2> select * from total_wim_sites;

wim_site
-----
325

(1 rows)
```

## 5. Compute the average speed for each site on M50.

```
In [67]: compute the average speed for each site on M50.
spark.sql("select cosit, avg(speed) as avg_speed from transport where cosit IN (1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510)");
```

| cosit | avg_speed         |
|-------|-------------------|
| 1507  | 95.00087226856925 |
| 1506  | 89.01114291227168 |
| 1504  | 84.95810635538263 |
| 1502  | 84.54840331627523 |
| 1501  | 83.7040682651283  |
| 1505  | 83.56696008249519 |
| 1508  | 81.37383510356896 |
| 1503  | 79.29442543070944 |
| 1012  | 78.86071923672837 |
| 1509  | 78.19974476773864 |
| 1500  | 76.23903247455863 |

```
In [62]: Avg_speed.select("cosit", "avg_speed")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="avg_speed", keyspace = "assignment2")\
.save(mode="append")
```

```
cqlsh:assignment2> create table avg_speed(cosit int, avg_speed float, primary key(cosit));
cqlsh:assignment2> select * from avg_speed;
```

| cosit | avg_speed |
|-------|-----------|
| 1500  | 76.23903  |
| 1503  | 79.29443  |
| 1012  | 78.86072  |
| 1501  | 83.70407  |
| 1509  | 78.19975  |
| 1502  | 84.5484   |
| 1508  | 81.37383  |

(7 rows)

## 6. Show total number of counts by vehicle class. Order results in descending.

```
In [69]: # 6.Show total number of counts by vehicle class. Order results in descending.
```

```
Total_numberof_vehicles = spark.sql("select classname, count(class) as total_vehicle_count from transport group by classname order by total_vehicle_count desc");
Total_numberof_vehicles.show()
```

| classname | total_vehicle_count |
|-----------|---------------------|
| CAR       | 3472965             |
| LGV       | 498505              |
| HGV_ART   | 216978              |
| HGV_RIG   | 135202              |
| BUS       | 29347               |
| CARAVAN   | 21224               |
| MBIKE     | 14682               |
| null      | 396                 |

```
In [74]: Total_numberof_vehicles.select("classname", "total_vehicle_count")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="total_vehicles", keyspace = "assignment2")\
.save(mode="append")
```

```
SyntaxException: line 1:80 no viable alternative at input '(' (... text, total_vehicle_count int, primary key[()...))
cqlsh:assignment2> create table total_vehicles(classname text, total_vehicle_count int, primary key(total_vehicle_count));
cqlsh:assignment2> select * from total_vehicles;
```

| total_vehicle_count | classname |
|---------------------|-----------|
| 29347               | BUS       |
| 14682               | MBIKE     |
| 498505              | LGV       |
| 396                 | null      |
| 3472965             | CAR       |
| 135202              | HGV_RIG   |
| 21224               | CARAVAN   |
| 216978              | HGV_ART   |

(8 rows)

## 7. List the top 3 busiest sites on M50.

```
In [78]: # 7. List the top 3 busiest sites on M50.

Top3_busiest_sites = spark.sql("select cosit, count(*) as vehicle_count from transport where cosit IN (1500, 1501, 1502, 1503, 1509)
Top3_busiest_sites.show()
```

| cosit | vehicle_count |
|-------|---------------|
| 1508  | 98292         |
| 1502  | 89498         |
| 1503  | 86195         |
| 1501  | 83205         |
| 1509  | 78360         |

```
In [79]: Top3_busiest_sites.select("cosit", "vehicle_count")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="busiest_sites", keyspace = "assignment2")\
.save(mode="append")
```

```
(8 rows)
cqlsh:assignment2> create table busiest_sites(cosit int, vehicle_count int, primary key(cosit));
cqlsh:assignment2> select * from busiest_sites;
```

| cosit | vehicle_count |
|-------|---------------|
| 1503  | 86195         |
| 1501  | 83205         |
| 1509  | 78360         |
| 1502  | 89498         |
| 1508  | 98292         |

(5 rows)

## 8. What is the busiest site on M6?

```
In [84]: # 8.What is the busiest site on M6?

Busiest_site_on_M6 = spark.sql("select cosit, count(*) as count from transport where cosit IN (3601,3602, 3603, 3604, 3605, 2006)
Busiest_site_on_M6.show()
```

| cosit | count |
|-------|-------|
| 20062 | 12387 |

```
In [88]: Busiest_site_on_M6.select("cosit", "count")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="m6_site", keyspace = "assignment2")\
.save(mode="append")
```

```
cqlsh:assignment2> create table m6_site(cosit int, count int, primary key(cosit));
cqlsh:assignment2> select * from m6_site;

cosit | count
-----+-----
20062 | 12387

(1 rows)
```

## 9. What site reports the highest number of HGVs?

In [91]: #9. What site reports the highest number of HGVs?

```
Highest_HGV_site_count = spark.sql("select cosit, count(*) as hgv_count from transport where class = 5 or class = 6 group by cosit")
Highest_HGV_site_count.show()
```

```
+-----+-----+
|cosit|hgv_count|
+-----+-----+
| 997| 12031|
+-----+-----+
```

In [92]: Highest\_HGV\_site\_count.select("cosit", "hgv\_count")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="highest\_numberof\_hgv", keyspace = "assignment2")\
.save(mode="append")

```
cqlsh:assignment2> create table highest_numberof_hgv(cosit int, hgv_count int, primary key(cosit));
cqlsh:assignment2> select * from highest_numberof_hgv;

cosit | hgv_count
-----+-----
997 | 12031

(1 rows)
```

## 10. Calculate the total number of vehicles on each site on M7.

In [93]: #10. Calculate the total number of vehicles on each site on M7.

```
Total_numberof_vehicles_on_M7 = spark.sql("select cosit, count(*) as total_vehicles from transport where cosit IN (3703, 3704, 20074, 20075, 200713, 20076, 20078, 20077, 200722, 20079, 200715, 200721, 200719, 200718, 200717, 200716)")
Total_numberof_vehicles_on_M7.show()
```

```
+-----+-----+
|cosit|total_vehicles|
+-----+-----+
| 20074| 26331|
| 20075| 26148|
| 200713| 25864|
| 20076| 22952|
| 20078| 17977|
| 20077| 17712|
| 200722| 17568|
| 20079| 16296|
| 200715| 13944|
| 200721| 12441|
| 200719| 8445|
| 200718| 7694|
| 200717| 7203|
| 200716| 7073|
+-----+-----+
```

In [94]: Total\_numberof\_vehicles\_on\_M7.select("cosit", "total\_vehicles")\
.write.format("org.apache.spark.sql.cassandra")\
.options(table="total\_vehicles\_on\_m7", keyspace = "assignment2")\
.save(mode="append")

```
cqlsh:assignment2> create table total_vehicles_on_m7(cosit int, total_vehicles int, primary key(cosit));
cqlsh:assignment2> select * from total_vehicles_on_m7;

cosit | total_vehicles
-----+-----
200713 | 25864
200722 | 17568
20075 | 26148
200721 | 12441
200719 | 8445
200715 | 13944
20078 | 17977
200716 | 7073
20079 | 16296
20074 | 26331
20076 | 22952
20077 | 17712
200718 | 7694
200717 | 7203

(14 rows)
cqlsh:assignment2> 
```