**SQL Queries**

CREATE DATABASE ev\_charging\_db;

USE ev\_charging\_db;

CREATE TABLE ev\_stations (

Station\_Name VARCHAR(255),

Street\_Address VARCHAR(255),

City VARCHAR(100),

State VARCHAR(50),

ZIP VARCHAR(20),

Latitude FLOAT,

Longitude FLOAT,

EV\_Level1\_Num INT,

EV\_Level2\_Num INT,

EV\_DC\_Fast\_Count INT,

EV\_Connector\_Types VARCHAR(255),

EV\_Network VARCHAR(255),

EV\_Pricing VARCHAR(255),

Open\_Date DATE,

Access\_Days\_Time VARCHAR(255),

Groups\_With\_Access\_Code VARCHAR(255),

Status\_Code VARCHAR(50),

Total\_Chargers INT,

Open\_Year INT,

Station\_Age\_Years FLOAT,

Predicted\_Chargers FLOAT,

Charger\_Gap FLOAT

);

SHOW DATABASES;

USE ev\_charging\_db;

SHOW TABLES;

SELECT \* FROM ev\_stations LIMIT 10;

DROP TABLE ev\_stations;

RENAME TABLE `ev final clean` TO ev\_stations;

**1)Total Stations by state**

SELECT State, COUNT(\*) AS Total\_Stations

FROM ev\_stations

GROUP BY State

ORDER BY Total\_Stations DESC;

**Output:**

|  |  |
| --- | --- |
| **State** | **Total\_stations** |
| AK | 18 |
| AL | 175 |
| AR | 93 |
| AZ | 402 |
| CA | 3143 |
| CO | 598 |
| CT | 386 |
| DC | 132 |
| DE | 48 |
| FL | 1215 |
| GA | 633 |
| HI | 254 |
| IA | 137 |
| ID | 84 |
| IL | 580 |
| IN | 224 |
| KS | 224 |
| KY | 113 |
| LA | 107 |
| MA | 509 |
| MD | 654 |
| ME | 172 |
| MI | 401 |
| MN | 351 |
| MO | 348 |
| MS | 70 |
| MT | 49 |
| NC | 607 |
| ND | 26 |
| NE | 81 |
| NH | 111 |
| NJ | 419 |
| NM | 78 |
| NV | 224 |
| NY | 1133 |
| OH | 484 |
| OK | 118 |
| OR | 553 |
| PA | 516 |
| RI | 110 |
| SC | 244 |
| SD | 42 |
| TN | 355 |
| TX | 942 |
| UT | 245 |
| VA | 615 |
| VT | 207 |
| WA | 847 |
| WI | 312 |
| WV | 81 |
| WY | 56 |

**2)Total Stations by EV Network**

SELECT `EV Network`, COUNT(\*) AS Total\_Stations

FROM ev\_stations

GROUP BY `EV Network`

ORDER BY Total\_Stations DESC;

**Output:**

|  |  |
| --- | --- |
| **EV Network** | **Total\_Stations** |
| Non-Networked | 6311 |
| ChargePoint Network | 5137 |
| Tesla Destination | 3560 |
| SemaCharge Network | 1148 |
| Tesla | 778 |
| Blink Network | 763 |
| Volta | 506 |
| Greenlots | 446 |
| EV Connect | 423 |
| Electrify America | 115 |
| FLO | 112 |
| OpConnect | 77 |
| Webasto | 60 |
| eVgo Network | 40 |
| POWERFLEX | 23 |
| EVGATEWAY | 17 |
| FCN | 6 |
| AMPUP | 3 |
| CHARGELAB | 1 |

**3) Growth of stations over years**

SELECT `Open Year`, COUNT(\*) AS Stations\_Opened

FROM ev\_stations

GROUP BY `Open Year`

ORDER BY `Open Year`;

**Output:**

|  |  |
| --- | --- |
| **Open Year** | **Stations\_Opened** |
| 1995 | 1 |
| 1996 | 2 |
| 1997 | 6 |
| 1998 | 16 |
| 1999 | 12 |
| 2000 | 3 |
| 2002 | 18 |
| 2004 | 15 |
| 2005 | 2 |
| 2006 | 3 |
| 2007 | 1 |
| 2008 | 26 |
| 2009 | 18 |
| 2010 | 75 |
| 2011 | 1089 |
| 2012 | 1650 |
| 2013 | 727 |
| 2014 | 1074 |
| 2015 | 1823 |
| 2016 | 2278 |
| 2017 | 2168 |
| 2018 | 3085 |
| 2019 | 4355 |
| 2020 | 1076 |
| 2021 | 3 |

**4)Top 10 cities by charger gap**

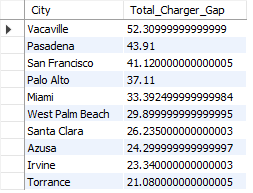
SELECT City, SUM(`Charger Gap`) AS Total\_Charger\_Gap

FROM ev\_stations

GROUP BY City

ORDER BY Total\_Charger\_Gap DESC

LIMIT 10;

**Output:**

**5)Overutilized station Gap**

SELECT COUNT(\*) AS Overutilized\_Stations

FROM ev\_stations

WHERE `Charger Gap` > 2;

**Output:**



**6) Average charger gap across all stations**

SELECT ROUND(AVG(`Charger Gap`), 2) AS Average\_Charger\_Gap

FROM ev\_stations;

**Output:**

****

**7)State with highest average charger gap (find the most problematic states)**

SELECT State, ROUND(AVG(`Charger Gap`), 2) AS Avg\_Charger\_Gap

FROM ev\_stations

GROUP BY State

ORDER BY Avg\_Charger\_Gap DESC

LIMIT 1;

**Output:**



**8) Predicted Vs Actual charges**

SELECT

SUM(`Total Chargers`) AS Total\_Actual\_Chargers,

SUM(`Predicted Chargers`) AS Total\_Predicted\_Chargers,

SUM(`Predicted Chargers`) - SUM(`Total Chargers`) AS Total\_Charger\_Gap

FROM ev\_stations;

**Output:**

****

**9) Cluster Analysis**

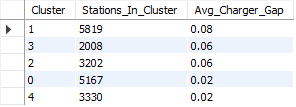
SELECT Cluster, COUNT(\*) AS Stations\_In\_Cluster, ROUND(AVG(`Charger Gap`), 2) AS Avg\_Charger\_Gap

FROM ev\_stations

GROUP BY Cluster

ORDER BY Avg\_Charger\_Gap DESC;

**Output:**

****