```sql
-- vehicle info --
-- requirements addressed: joins multiple tables, uses WHERE clause, aggregation,
uses joins --
CREATE VIEW vehicle_info AS
    SELECT
        vm.model AS vehicle_model,
        vm.model_year,
        vm.CAFV_eligibility,
        vm.vehicle_id,
        vm.range_id,
        er.electric_range,
        em.make,
        et.electric_type
    FROM
        vehicle_models vm
            JOIN
        vehicle_makes em ON vm.make_id = em.make_id
            JOIN
        electric_ranges er ON vm.range_id = er.range_id
            JOIN
        electric_types et ON vm.electric_type_id = et.electric_type_id
    WHERE
        vm.CAFV_eligibility = 'Clean Alternative Fuel Vehicle Eligible';


-- view vehicle counts by make -- !!!!!!!! BROKEN!!!!!!!!!!!!!!!!!!--
-- requirments addressed: join used, filters by CAFV, uses COUNT() --
CREATE VIEW counts_by_make AS
    SELECT
        em.make, COUNT(vm.vehicle_id) AS CAFV_eligible_vehicle_count
    FROM
        vehicle_models vm
            JOIN
        vehicle_makes em ON vm.make_id = em.make_id
    WHERE
        vm.CAFV_eligibility = 'Clean Alternative Fuel Vehicle Eligible'
    GROUP BY em.make;


-- vehicle location info --
-- requirements addressed: used joins, no filtering, uses a linking table --
CREATE VIEW vehicle_location AS
    SELECT
        v.vehicle_id,
        vl.county,
        vl.city,
        vl.state,
        vl.postal_code,
        ld.legislative_district
    FROM
        vehicles v
            JOIN
        vehicle_locations vl ON v.location_id = vl.location_id
            JOIN
        legislative_districts ld ON vl.district_id = ld.district_id;


-- viewing vehicles by range and type --
-- requirements addressed: joins tables, filters based off range --
```

```sql
CREATE VIEW range_and_type AS
    SELECT
        vm.vehicle_id,
        vm.model,
        vm.model_year,
        er.electric_range,
        et.electric_type
    FROM
        vehicle_models vm
            JOIN
        electric_ranges er ON vm.range_id = er.range_id
            JOIN
        electric_types et ON vm.electric_type_id = et.electric_type_id
    WHERE
        er.electric_range > 20;




-- requirements addressed: uses join, subquery, and utilizes subquery --
CREATE VIEW make_model_counts AS
    SELECT
        em.make, vm.model, COUNT(vm.vehicle_id) AS vehicle_count
    FROM
        vehicle_models vm
            JOIN
        vehicle_makes em ON vm.make_id = em.make_id
    WHERE
        vm.range_id IN (SELECT
                range_id
            FROM
                electric_ranges
            WHERE
                electric_range > 20)
    GROUP BY em.make , vm.model;


-- 6. Average Electric Range by Vehicle Type
-- Calculate the average electric range for each EV type.
CREATE VIEW Avg_Electric_Range AS
SELECT
    et.electric_type,
    AVG(er.electric_range) AS average_range
FROM
    Vehicle_models vmo
JOIN
    Electric_types et ON vmo.electric_type_id = et.electric_type_id
JOIN
    Electric_ranges er ON vmo.range_id = er.range_id
WHERE
    er.electric_range IS NOT NULL
GROUP BY
    et.electric_type;


-- 7. Charging Accessibility Insights
-- Calculate the distribution of EV registrations by geographic coordinates
(charging infrastructure analysis).
CREATE VIEW Vehicle_Count_By_Location AS
```

```sql
SELECT
    vl.city,
    vl.state,
    COUNT(DISTINCT v.vehicle_id) AS vehicle_count
FROM
    Vehicles v
JOIN
    Vehicle_Locations vl ON v.location_id = vl.location_id
GROUP BY
    vl.city,
    vl.state
ORDER BY
    vehicle_count DESC;
```