Merge Queries

Merge Traffic Data with Incident Data

```
SELECT * FROM "incidents"."data_incidents" as incident
INNER JOIN "incidents"."data_traffic" as traffic
on incident.xdsegid = traffic.xd_id AND incident.window_id = traffic.window_id;
```

Explanation: Join by primary key XDSegID to line up traffic data for the segment closest to a particular incident. Join by window_id to ensure 6hr windows are matched up between both datasets.

Note: Window_ID is a string with the format "X.YYY.ZZZZ", where X is the window number (1,2,3,4), Y is the Day number (1-365), and Z is the Year.

Result Table is called **traffic_incidents_merge**

Merge Weather with Traffic-Incident-Merge

```
SELECT * FROM "incidents"."traffic_incidents_merge" as traf_inc
INNER JOIN "incidents"."weather" as weather
on traf_inc.window_id = weather.window_id;
```

Explanation: Join by primary key XDSegID to line up weather data for the segment closest to a particular incident. Join by window_id to ensure 6hr windows are matched up between both datasets.

Result Table is called **intermediate_merge**

Merge Roads with Traffic-Incident-Weather-Merge

```
SELECT * FROM "incidents"."intermediate_merge" as traf_inc
LEFT JOIN "incidents"."roads_id" as roads
ON intermediate_merge.xd_id = roads.segid;
```

Explanation: Join by primary key SegID in order to later classify the cluster of roads (road group) corresponding to the XDSegID of the incident in question.

Result Table is called **showdown_merge**
