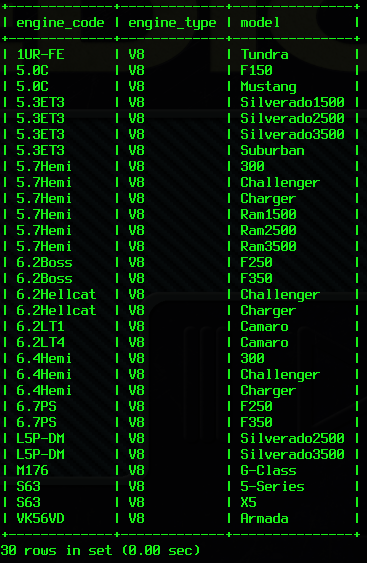
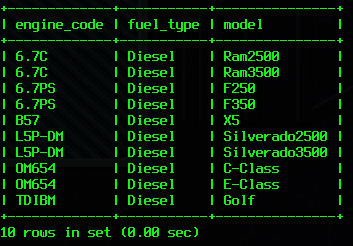
**1) This query pulls all cars with V8's**

SELECT e.engine\_code, e.engine\_type, ue.model FROM engine e INNER JOIN usesengine ue ON e.engine\_code = ue.engine\_code WHERE e.engine\_type = 'V8';



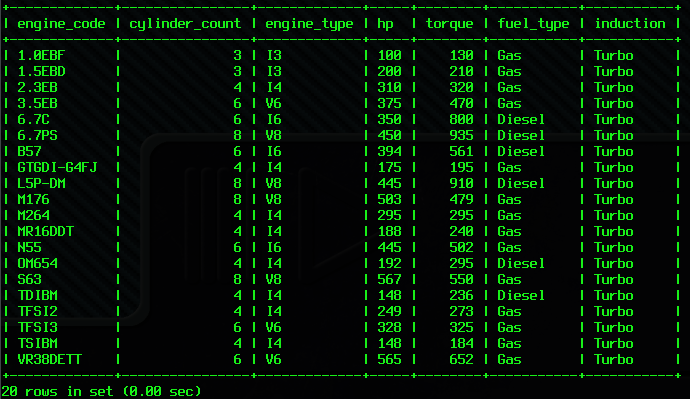
**2) This query pulls all cars with Diesel engines**

SELECT e.engine\_code, e.fuel\_type, ue.model FROM engine e INNER JOIN usesengine ue ON e.engine\_code = ue.engine\_code WHERE e.fuel\_type = 'Diesel';



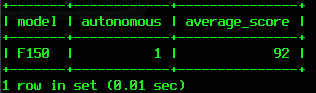
**3) This query pulls all Turbocharged engines from the engine table**

SELECT \* FROM engine WHERE induction='Turbo';



**4) This query pulls all cars with autonomous features and average reviews (> 90%)**

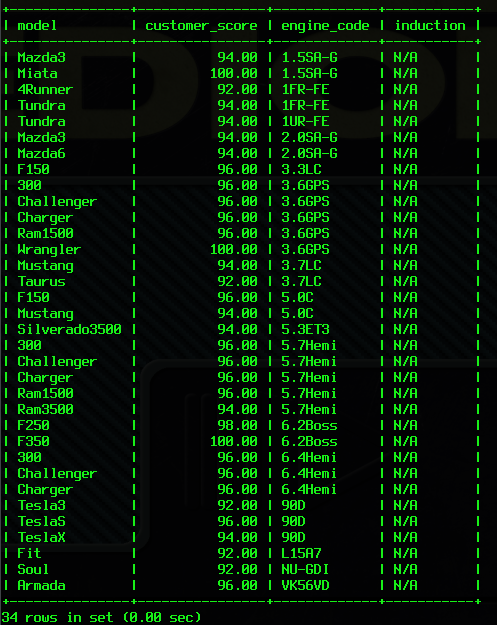
SELECT t.model, t.autonomous, r.average\_score FROM tech t INNER JOIN review r ON t.model = r.model WHERE t.autonomous = '1' AND r.average\_score > 90.00;



This one surprised Hiram and I

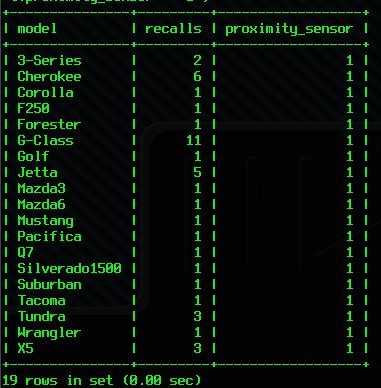
**5) This query pulls all cars with customer reviews ( > 90%) and non N/A engines**

SELECT r.model, r.customer\_score, ue.engine\_code, e.induction FROM review r INNER JOIN usesengine ue ON r.model = ue.model INNER JOIN engine e ON e.engine\_code = ue.engine\_code WHERE r.customer\_score > 90 AND e.induction = 'N/A';



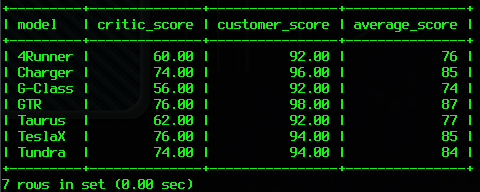
**6) This query pulls are cars with at least 1 recall and proximity sensors**

SELECT c.model, cow.recalls, t.proximity\_sensor FROM car c INNER JOIN tech t ON c.model = t.model INNER JOIN costofownership cow ON c.model = cow.model WHERE cow.recalls > 0 AND t.proximity\_sensor = '1';



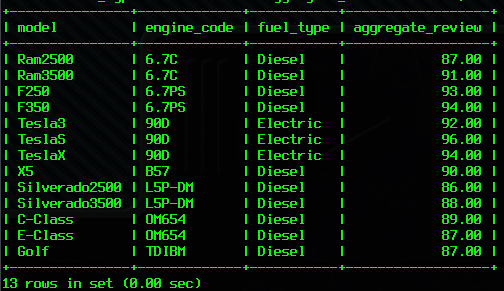
**7) This query pulls cars with >90 customer reviews and <80 critical reviews**

SELECT \* FROM review WHERE customer\_score > 90 AND critic\_score < 80;



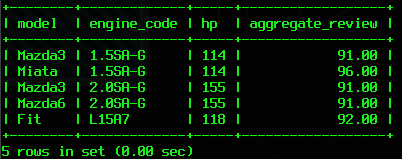
**8) This query pulls all cars that don't use regular gas AND have average reviews >85**

SELECT ue.model, ue.engine\_code, e.fuel\_type, c.aggregate\_review FROM usesengine ue INNER JOIN engine e ON ue.engine\_code = e.engine\_code INNER JOIN car c ON ue.model = c.model WHERE e.fuel\_type != 'Gas' AND c.aggregate\_review > 85.00;



**9) This query pulls cars that have <200 horsepower and average reviews >90**

SELECT ue.model, ue.engine\_code, e.hp, c.aggregate\_review FROM usesengine ue INNER JOIN engine e on ue.engine\_code = e.engine\_code INNER JOIN car c ON ue.model = c.model WHERE e.hp < 200 AND c.aggregate\_review > 90;



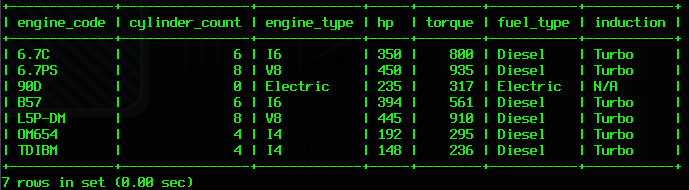
**10) This query pulls cars that have high average reviews and minimal tech**

SELECT c.model, r.average\_score, t.proximity\_sensor, t.auto\_braking, t.adc,t.autonomous FROM car c INNER JOIN review r ON c.model = r.model INNER JOIN tech t ON c.model = t.model WHERE r.average\_score > 90 AND t.proximity\_sensor = '0' AND t.auto\_braking='0' AND t.adc='0' AND t.autonomous='0';



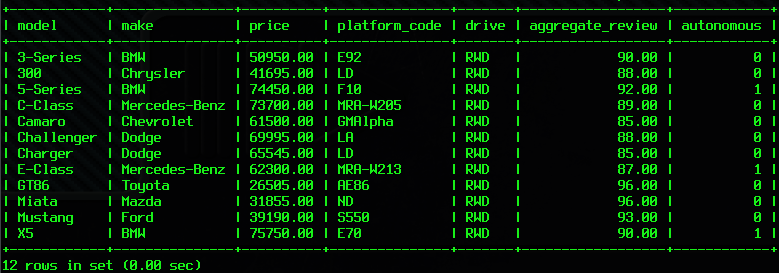
**11) This query pulls all non Gas engines**

SELECT \* FROM engine where fuel\_type != 'Gas';



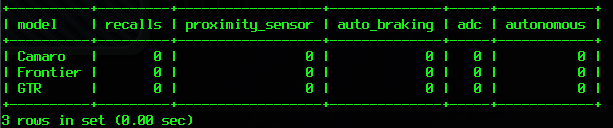
**12) This query pulls all cars that are RWD and cost less than $30000**

SELECT \* FROM car WHERE drive = 'RWD' OR drive ='4/RWD' AND price <= 30000.00;



**13) This query pulls all cars that have no recalls and minimal tech**

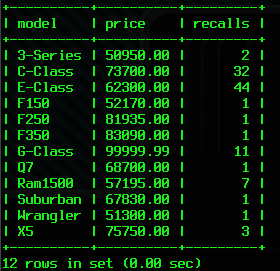
SELECT c.model, cow.recalls, t.proximity\_sensor, t.auto\_braking, t.adc,t.autonomous FROM car c INNER JOIN costofownership cow ON c.model = cow.model INNER JOIN tech t ON c.model = t.model WHERE cow.recalls=0 AND t.proximity\_sensor = '0' AND t.auto\_braking='0' AND t.adc='0' AND t.autonomous='0';



**14) This query pulls cars that have at least 1 recall and cost more than**

**$50,000**

SELECT c.model, c.price, cow.recalls FROM car c INNER JOIN costofownership cow ON c.model = cow.model WHERE c.price >=50000.00 AND cow.recalls >=1;



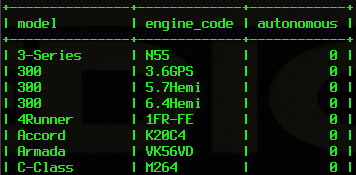
**15) This query pulls engines used in Ford/Chevy/Dodges only**

SELECT c.make, ue.model, ue.engine\_code FROM usesengine ue INNER JOIN car c ON ue.model = c.model WHERE c.make ='Ford' OR c.make = 'Dodge' OR c.make = 'Chevrolet';



**16) This query pulls engines NOT used in autonomous cars**

SELECT ue.model, ue.engine\_code, c.autonomous FROM usesengine ue INNER JOIN car c ON ue.model = c.model WHERE c.autonomous = '0';

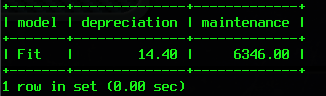




I used a subset, because I don’t think you care about all 80 rows…

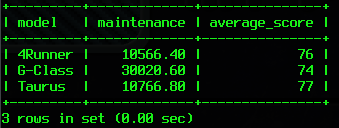
**17) This query pulls all cars with depreciation less than 15% and a cost of ownership < $7,500**

SELECT c.model, cow.depreciation, cow.maintenance FROM car c INNER JOIN costofownership cow ON c.model = cow.model WHERE cow.depreciation < 15.00 AND cow.maintenance < 7500.00;



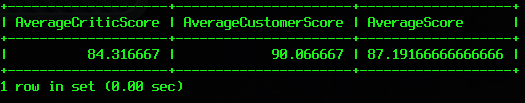
**18) This query pulls all cars with a cost of ownership over $10,000 and average reviews < 80**

SELECT c.model, cow.maintenance, r.average\_score FROM car c INNER JOIN costofownership cow ON c.model = cow.model LEFT JOIN review r ON c.model = r.model WHERE cow.maintenance > 10000.00 AND r.average\_score < 80;



**19) This query pulls up the average scores across the entire review table**

SELECT AVG(review.critic\_score) AS AverageCriticScore, AVG(review.customer\_score) AS AverageCustomerScore, AVG(review.average\_score) AS AverageScore FROM review;



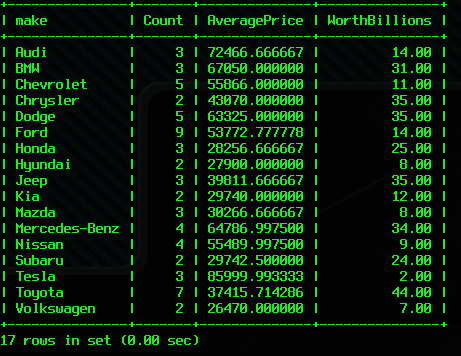
**20) This query pulls up the average review score of each brand**

SELECT car.make, AVG(car.aggregate\_review) AS AverageReview FROM car GROUP BY make;



**21) This query pulls up the brands, how many of their vehicles are in the database, the average price of their vehicles and the brand worth (in billions).**

SELECT car.make, COUNT(car.make) AS Count, AVG(car.price) as AveragePrice, brand.brand\_worth AS WorthBillions FROM car INNER JOIN brand ON car.make = brand.make GROUP BY brand.make;



For some reason, my database made the Worth into that small column. Very frustrating...