

Team 3 Milestone 2

Torben Bernhard, Lei Meng, Uchenna Nwoke

Conceptual Model**2****Queries****3****Vendor Queries****4**

Vendor Query 1

4

Vendor Query 2

5

Vendor Query 3

6

Vendor Query 4

7

Vendor Query 5

8

Vendor Query 6

9

Vendor Query 7

10

Customer Queries**12**

Customer Query 1

12

Customer Query 2

13

Customer Query 3

14

Stored procedures**15**

Stored Procedure 1: Get new listing by region

15

Stored Procedure 2: Get most viewed open listings

16

Stored Procedure 3: Get average price and mileage in \$10,000 brackets by vehicle make

18

Frequently Used Views**19**

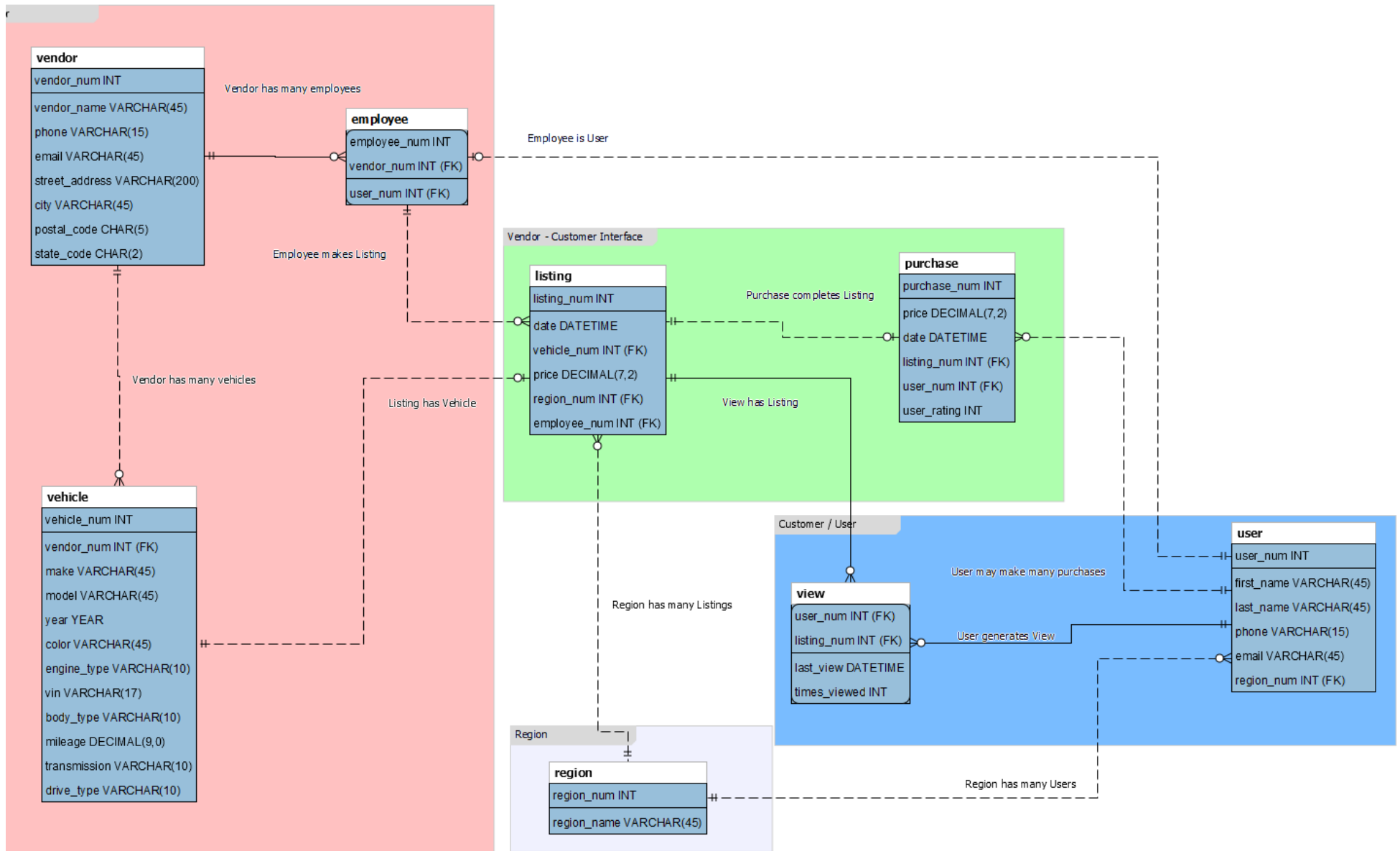
All Open Listings

19

All Closed Listings

20

Conceptual Model



Queries

Vendor Queries

Vendor Queries are queries which give a user (either an employee or vendor organization owner), information about trends in recent listings, purchases, and customer interest. Vendors can use this information to make decisions about how to list their own vehicles and to evaluate employee performance.

1. Fastest selling listings purchased in the last 30 days
2. All open listings with list price below avg purchase price for that make
3. Potential customers with most views but haven't made a purchase yet
4. Average, maximum and minimum listed days of sold vehicles by body types
5. Top 10 salespeople who have created the most revenue.
6. Top 10 selling brands along with average selling price.
7. Most viewed open listings

Customer Queries

Customer queries are customer facing queries which give users searching for a vehicle information they can use when making a purchase.

1. Search for open listings by min vendor rating, make, max price, and max mileage
2. See trends in average price for listings that meet customer specifications.
3. Top 10 purchased models and popularity by region.

Vendor Queries

Vendor Query 1

The following query returns the fastest selling listings purchased in the last 30 days, as well as the difference between the list price and final purchase price.

Vendors can use this information to find out which listings are selling quickly and the difference between purchase and list price.

```

4      -- 1
5      /* This query returns the fastest selling listings purchased in the last 30 days as well as the difference between
6         list and purchase price. Vendors can use this information to determine which listings sell most quickly and
7         how that relates to the purchase price of that listing.
8      */
9      • select
10         timestampdiff(day, AllClosedListings.date, Purchase.date) as Days_Listed,
11         cast(Purchase.date as date) as Purchase_Date,
12         concat('$',format (AllClosedListings.price, 'C2')) as List_Price,
13         concat('$',format (Purchase.price,'C2')) as Purchase_Price,
14         concat('$',format (Purchase.price - AllClosedListings.price, 'C2')) as Price_Differential,
15         Vehicle.make, Vehicle.model, Vehicle.year, Vehicle.mileage, Vehicle.engine_type, Vehicle.body_type
16     from AllClosedListings
17     join Purchase on AllClosedListings.listing_num = Purchase.listing_num
18     join Vehicle on AllClosedListings.vehicle_num = Vehicle.vehicle_num
19     where
20         Purchase.date > date_sub(curdate(), interval 30 day) and
21         Purchase.date < curdate()
22     order by Days_Listed;

```

Days_Listed	Purchase_Date	List_Price	Purchase_Price	Price_Differential	make	model	year	mileage	engine_type	body_type
31	2022-05-10	\$74,629	\$71,445	\$-3,184	Mercedes-Benz	CLK-Class	2009	72	Electric	Sedan
32	2022-05-12	\$58,211	\$58,869	\$658	Ford	E-Series	2007	25910	Hybrid	Wagon
33	2022-04-29	\$80,341	\$79,720	\$-621	Suzuki	Esteem	2001	64521	Hybrid	SUV
35	2022-05-09	\$17,865	\$14,473	\$-3,392	Toyota	Camry	2005	11	Electric	SUV
36	2022-05-07	\$28,833	\$25,462	\$-3,371	Acura	RL	2006	95	Gas	SUV
39	2022-05-16	\$45,534	\$40,612	\$-4,922	Lexus	RX	2003	72	Hybrid	Sedan
44	2022-04-30	\$69,850	\$66,888	\$-2,962	BMW	525	2005	49809	Gas	Sedan
46	2022-04-23	\$63,332	\$60,174	\$-3,158	Dodge	Ramcharger	1993	65	Electric	Sedan
46	2022-05-14	\$90,894	\$90,294	\$-600	BMW	Z4 M	2008	95	Electric	SUV
49	2022-05-01	\$78,118	\$73,123	\$-4,995	Mercury	Grand Marquis	1987	50	Electric	Truck
57	2022-05-08	\$17,118	\$14,993	\$-2,125	Chevrolet	Metro	2001	17	Hybrid	Sedan
57	2022-04-18	\$80,118	\$80,197	\$79	Volkswagen	Golf	1998	98	Electric	Wagon
59	2022-05-15	\$76,677	\$74,242	\$-2,435	Dodge	Ram 2500	2007	13	Electric	Truck

Vendor Query 2

This query returns all open listings where the list price is below the average purchase price.

This query filters for only that vehicle of the same make in the same \$10,000 price bracket and has lower mileage than the average mileage for that make.

Vendors and even customers can use this information to find underpriced listings.

```

31 • select
32     concat('$', format(t0.avgPurchasePrice - AllOpenListings.price, 'C2')) as 'Price Differential',
33     concat('$', format(AllOpenListings.price, 'C2')) as 'List Price',
34     concat('$', format(t0.avgPurchasePrice, 'C2')) as 'Make Average Purchase Price',
35     Vendor.vendor_name as 'Vendor',
36     Vehicle.make, Vehicle.model, Vehicle.year, Vehicle.mileage
37 from AllOpenListings join Vehicle on AllOpenListings.vehicle_num = Vehicle.vehicle_num
38 join (
39     select
40         Vehicle.make,
41         avg(Purchase.price) as avgPurchasePrice
42     from AllClosedListings join Vehicle on AllClosedListings.vehicle_num = Vehicle.vehicle_num
43     join Purchase on AllClosedListings.listing_num = Purchase.listing_num group by Vehicle.make
44 ) as t0 on Vehicle.make = t0.make
45 join (
46     select Vehicle.make,
47         avg(Vehicle.mileage) as avgMileage
48     from Vehicle group by Vehicle.make
49 ) as t1 on Vehicle.make = t1.make
50 join Vendor on Vehicle.vendor_num = Vendor.vendor_num
51 where AllOpenListings.price < t0.avgPurchasePrice
52 and abs(AllOpenListings.price - t0.avgPurchasePrice) < 5000
53 and Vehicle.mileage <= t1.avgMileage
54 order by t0.avgPurchasePrice - AllOpenListings.price desc;

```

	Price Differential	List Price	Make Average Purchase Price	Vendor	make	model	year	mileage
▶	\$4,501	\$53,642	\$58,143	Jennifer's Rad Go-Mobiles	BMW	M3	2010	82
	\$3,982	\$39,617	\$43,600	Queen Victoria's Very Good Cars	Toyota	Previa	1992	86
	\$3,676	\$54,478	\$58,154	Queen Victoria's Very Good Cars	Volkswagen	Eos	2007	2
	\$3,425	\$23,121	\$26,546	Jennifer's Rad Cars	GMC	Yukon XL 1500	2003	37
	\$3,381	\$30,134	\$33,515	Jennifer's Best Vehicles	Mazda	Miata MX-5	2009	53
	\$3,372	\$45,824	\$49,196	Jimbo's Very Good Go-Mobiles	Suzuki	XL-7	2006	11944
	\$3,245	\$56,672	\$59,916	Jimothy's Super Cars	Hyundai	Elantra	1997	41
	\$3,188	\$46,008	\$49,196	Gorben's Super Go-Mobiles	Suzuki	Grand Vitara	2005	30
	\$2,995	\$58,871	\$61,866	Gorben's Very Good Go-Mobiles	Chevrolet	Silverado 2500	2001	68
	\$2,822	\$49,946	\$52,768	Jimbo's Pretty Great Four Wheeled Fun	Nissan	300ZX	1993	70
	\$2,699	\$55,454	\$58,154	Jennifer's Very Good Cars	Volkswagen	GTI	1998	9
	\$2,205	\$24,214	\$26,419	Jennifer's Super Four Wheeled Fun	Pontiac	Montana	2006	6
	\$1,923	\$97,603	\$99,526	Queen Victoria's Super Duper Four Wh...	Bentley	Continental Fl...	2008	84
	\$1,770	\$56,373	\$58,143	Queen Victoria's Super Duper Cars	BMW	8 Series	1996	70
	\$1,384	\$43,431	\$44,815	Jimothy's Super Duper Four Wheeled ...	Honda	Passport	2002	20
	\$1,290	\$58,319	\$59,609	Jimothy's Best Four Wheeled Fun	Ford	E150	1984	54
	\$1,270	\$56,884	\$58,154	Queen Victoria's Rad Vehicles	Volkswagen	GTI	1988	17
	\$1,103	\$55,680	\$56,783	Queen Victoria's Super Duper Four Wh...	Dodge	Intrepid	1999	69
	\$1,098	\$50,069	\$51,166	Jimothy's Best Go-Mobiles	Mercedes-...	300D	1992	11550
	\$570	\$52,198	\$52,768	Jimbo's Super Duper Cars	Nissan	240SX	1998	22155
	\$274	\$50,216	\$50,490	Gorben's Very Good Vehicles	Infiniti	J	1994	1

Vendor Query 3

This query classifies the customers who have viewed vehicle listings on our website many times but didn't make any purchases.

This data will help vendors/admins reach out to potential customers with buying wills.

```

59 • SELECT DISTINCT
60     CONCAT(u.first_name, ' ', u.last_name) AS 'Customer name',
61     u.phone AS 'Phone',
62     u.email AS 'Email',
63     region_name AS Region,
64     SUM(v.times_viewed) AS 'Total views',
65     COUNT(v.listing_num) AS 'Vehicles viewed',
66     MAX(v.last_view) AS 'Last view time'
67 FROM
68     User u
69     INNER JOIN
70     View v ON u.user_num = v.user_num
71     INNER JOIN
72     Region r ON r.region_num = u.region_num
73     LEFT JOIN
74     Purchase p ON v.listing_num = p.listing_num
75 WHERE
76     p.listing_num IS NULL
77 GROUP BY u.user_num
78 ORDER BY SUM(v.times_viewed) DESC
79 LIMIT 30;

```

	Customer name	Phone	Email	Region	Total views	Vehicles viewed	Last view time
►	Christy Verheyden	107-696-2013	cverheyden8m@whitehouse.gov	Crown Hhill	632	34	2022-05-13 03:56:00
	Janie McBrearty	593-836-2660	jmcbreartybe@va.gov	Cedar Park	617	35	2022-05-11 20:07:00
	Edsel Treslove	649-845-3379	etreslovebg@facebook.com	Atlantic	609	32	2022-05-19 21:22:00
	Allison Tackes	475-242-3676	atackes7a@surveymonkey.com	Broadway	600	34	2022-05-19 21:22:00
	Biddy Kuhlen	583-717-6631	bkuhlen8g@google.ru	Pinehurst	596	34	2022-05-24 02:09:00
	Ninnetta Lansdale	151-671-3651	nlansdale5w@photobucket.com	Broadmore	596	34	2022-05-16 02:09:00
	Bron Stephen	702-157-6014	bstephen9k@opensource.org	North Beach	593	35	2022-05-20 10:09:00
	Shanta Fehners	172-782-2349	sfehners5a@nationalgeographic.com	Olympic Hills	591	33	2022-05-23 21:22:00
	Esta Kasting	309-690-4278	ekasting6t@sciencedirect.com	Broadmore	587	32	2022-05-18 20:33:00
	Barthel Oley	516-798-9257	boley6v@cnet.com	Sunset Hill	583	34	2022-05-26 10:09:00
	Shelbi Parsonage	795-850-1736	sparsonage6d@51.la	Maple Leaf	574	34	2022-04-26 13:18:00
	Annamarie Quixley	976-841-3788	aquixley8r@eventbrite.com	Cedar Park	572	33	2022-04-24 09:44:00
	Saunder Oliphand	731-820-1534	soliphandai@ow.ly	Ravenna	571	33	2022-05-08 11:51:00
	Phelia Jandac	704-644-8861	pjandac7s@123-reg.co.uk	Greenwood	568	33	2022-05-06 02:09:00
	Mart Walliker	761-525-9715	mwalliker7g@seesaa.net	North Beach	567	32	2022-05-09 01:36:00
	Rossie Howie	883-131-1849	rhowie9s@google.it	Olympic Hills	567	31	2022-05-20 02:09:00
	Nelia Huglin	189-611-5442	nhuglinbi@feedburner.com	Broadmore	566	33	2022-05-08 00:00:00
	Pattin Spowage	716-703-8658	pspowage5e@gravatar.com	North Beach	565	30	2022-05-07 09:41:00
	Kristine Ruthven	754-294-4123	kruthven6z@arstechnica.com	Madrona	564	34	2022-05-13 11:51:00

Vendor Query 4

This query identifies the average, maximum and minimum days for vehicles sold from listing to purchasing, by vehicle body types.

This data will help vendors know which types of vehicles are selling fast, which may lead to inventory and price optimization strategies.

```

85 • SELECT
86     v.body_type AS 'Vehicle body type',
87     ROUND(AVG(Selling_days), 0) AS 'Average selling days',
88     MAX(Selling_days) AS 'Maximum selling days',
89     MIN(Selling_days) AS 'Minimum selling days'
90 FROM
91     Vehicle v
92     INNER JOIN
93     Listing l ON v.vehicle_num = l.vehicle_num
94     INNER JOIN
95     (SELECT
96         TIMESTAMPDIFF(DAY, l.date, p.date) AS 'Selling_days',
97         l.listing_num
98     FROM
99         Vehicle v
100        INNER JOIN Listing l ON v.vehicle_num = l.vehicle_num
101        INNER JOIN Purchase p ON l.listing_num = p.listing_num) sub ON sub.listing_num = l.listing_num
102 GROUP BY v.body_type
103 ORDER BY AVG(Selling_days);

```

	Vehicle body type	Average selling days	Maximum selling days	Minimum selling days
►	Wagon	44	59	30
	Sedan	45	60	30
	SUV	45	60	30
	Truck	46	59	31

Vendor Query 5

This query lists the top 10 salespeople who have created the most revenue.

This list also includes information regarding which vendor the salesperson belongs to, their total listings, and listing-selling converting rate.

```

109 • SELECT
110     CONCAT(u.first_name, ' ', u.last_name) AS 'Sales person',
111     ven.vendor_name AS 'Vendor',
112     concat('$',format(SUM(p.price),'C2')) AS 'Total sales',
113     (COUNT(l.listing_num)) AS 'Total listings',
114     ROUND((COUNT(p.purchase_num)) / (COUNT(l.listing_num)) * 100,
115           2) AS 'List-sell converting rate %'
116 FROM
117     ((Vehicle v
118     INNER JOIN Listing l ON ((v.vehicle_num = l.vehicle_num)))
119     INNER JOIN Vendor ven ON ven.vendor_num = v.vendor_num
120     INNER JOIN Employee e ON e.employee_num = l.employee_num
121     INNER JOIN User u ON u.user_num = e.user_num
122     LEFT JOIN Purchase p ON ((l.listing_num = p.listing_num)))
123 GROUP BY e.user_num , ven.vendor_name
124 ORDER BY SUM(p.price) DESC
125 LIMIT 10;

```

	Sales person	Vendor	Total sales	Total listings	List-sell converting rate %
►	Red Pallent	Queen Victoria's Super Duper Four Wheeled Fun	\$292,032	13	30.77
	L;urette Astill	Jimbo's Very Good Vehides	\$266,287	12	33.33
	Olivier Abrahamsen	Jennifer's Super Duper Transport Devices	\$262,978	11	36.36
	Priscella Lorain	Jimothy's Best Transport Devices	\$254,765	13	30.77
	Nikolia Fontell	Queen Victoria's Best Vehides	\$252,884	13	30.77
	Ainsley Reinbech	Jimbo's Rad Go-Mobiles	\$242,901	11	36.36
	Lucy Gherarducci	Gorben's Very Good Go-Mobiles	\$239,104	13	30.77
	Farrah Soame	Gorben's Very Good Vehides	\$235,370	13	30.77
	Olia Ware	Jennifer's Best Go-Mobiles	\$222,392	12	33.33
	Zitella Arp	Gorben's Rad Transport Devices	\$211,740	13	23.08

Vendor Query 6

This query lists the top 10 selling brands along with average selling price. This will help vendors and customers know the popular vehicle makes and reference to their business behaviors.

```

131 • SELECT
132     v.make AS `Brand`,
133     COUNT(p.purchase_num) AS `Total sales`,
134     concat('$',FORMAT(AVG(p.price), 2)) AS `Average price`
135 FROM
136     Vehicle v
137     INNER JOIN
138     Listing l ON v.vehicle_num = l.vehicle_num
139     INNER JOIN
140     Purchase p ON l.listing_num = p.listing_num
141 GROUP BY v.make
142 ORDER BY COUNT(p.purchase_num) DESC
143 LIMIT 10;

```

	Brand	Total sales	Average price
►	Ford	16	\$54,385.31
	Chevrolet	15	\$54,146.87
	Dodge	13	\$52,764.46
	Nissan	13	\$46,553.08
	Suzuki	12	\$56,931.08
	Toyota	11	\$54,510.18
	Pontiac	10	\$44,588.10
	Mitsubishi	10	\$52,472.20
	GMC	9	\$38,948.78
	BMW	9	\$60,852.56

Vendor Query 7

This query returns the most viewed open listings whose last view was in the last 30 days.

Vendors and system admin can use this information to gauge recent customer interest.

```

149 • select
150     agg.Listing_Total_Views as Total_Views,
151     cast(t0.mostRecentView as date) as Most_Recent_View,
152     cast(AllOpenListings.date as date) as List_Date,
153     timestampdiff(day, AllOpenListings.date, curdate()) as Days_Listed,
154     concat('$',format(AllOpenListings.price,'C2')) as List_Price,
155     Region.region_name as Region_Name,
156     Vendor.vendor_name as Vendor_Name,
157     Vehicle.make as Make, Vehicle.model as Model, Vehicle.year as Year, Vehicle.mileage as Mileage
158 from
159     AllOpenListings
160 join (
161     select
162         AllOpenListings.listing_num,
163         sum(View.times_viewed) as Listing_Total_Views
164     from AllOpenListings
165         join View on AllOpenListings.listing_num = View.listing_num
166     where
167         View.last_view > date_sub(curdate(), interval 30 day) and
168         View.last_view < curdate()
169     group by AllOpenListings.listing_num
170     order by Listing_Total_Views desc
171 ) as agg on AllOpenListings.listing_num = agg.listing_num
172 join Vehicle
173     on AllOpenListings.vehicle_num = Vehicle.vehicle_num
174 join Vendor
175     on Vehicle.vendor_num = Vendor.vendor_num
176 join Region
177     on AllOpenListings.region_num = Region.region_num
178 join (
179     select max(View.last_view) as mostRecentView, AllOpenListings.listing_num
180     from View join AllOpenListings on View.listing_num = AllOpenListings.listing_num
181     group by AllOpenListings.listing_num) as t0
182 on t0.listing_num = AllOpenListings.listing_num;

```

	Total_Views	Most_Recent_View	List_Date	Days_Listed	List_Price	Region_Name	Vendor_Name	Make	Model	Year	Mileage
▶	296	2022-05-11	2022-04-13	33	\$49,575	Maple Leaf	Jimothy's Best Transport Devices	Lamborghini	Diablo	1994	70
	295	2022-05-07	2022-04-08	38	\$68,315	Atlantic	Queen Victoria's Super Transport Devices	Ford	F-Series	1991	46
	275	2022-05-12	2022-04-15	31	\$77,697	Broadview	Jimothy's Super Duper Go-Mobiles	Mazda	B-Series Plus	1993	33
	268	2022-05-12	2022-04-14	32	\$90,218	Broadmore	Jimbo's Very Good Go-Mobiles	Mazda	Tribute	2001	37
	261	2022-05-11	2022-04-12	34	\$17,480	Broadview	Jennifer's Very Good Four Wheeled Fun	Jaguar	XJ Series	2003	60
	257	2022-05-23	2022-04-24	22	\$30,753	Broadview	Jennifer's Super Vehides	Suzuki	Grand Vitara	2002	32
	253	2022-05-05	2022-04-07	39	\$43,580	Greenwood	Jennifer's Super Vehides	Ford	F-Series	1997	25
	240	2022-05-11	2022-04-13	33	\$82,196	Olympic Hills	Queen Victoria's Super Duper Cars	Subaru	Legacy	1991	47
	236	2022-05-10	2022-04-12	34	\$72,416	View Ridge	Gorben's Super Duper Go-Mobiles	Mitsubishi	Raider	2007	32
	236	2022-06-01	2022-05-04	12	\$65,618	Madison Valley	Jennifer's Very Good Cars	Ford	Aerostar	1992	36
	225	2022-05-10	2022-04-14	32	\$27,229	North Beach	Jennifer's Pretty Great Cars	Suzuki	XL-7	2006	32
	220	2022-05-23	2022-04-25	21	\$88,184	Pinehurst	Jennifer's Super Duper Transport Devices	BMW	8 Series	1997	50
	217	2022-05-11	2022-04-11	35	\$40,926	North Beach	Queen Victoria's Very Good Cars	Mercedes-...	M-Class	2007	57
	215	2022-05-13	2022-04-14	32	\$43,491	Olympic Hills	Jimothy's Best Transport Devices	Nissan	Sentra	1999	21
	206	2022-05-10	2022-04-16	30	\$68,697	Eastlake	Queen Victoria's Very Good Transport D...	Mazda	Tribute	2001	53477
	203	2022-06-02	2022-05-03	13	\$77,364	Bitter Lake	Queen Victoria's Super Four Wheeled Fun	Ferrari	FF	2012	90
	200	2022-05-10	2022-04-14	32	\$51,063	Maple Leaf	Queen Victoria's Pretty Great Vehides	Volvo	S80	2000	47990
	195	2022-05-21	2022-04-21	25	\$61,285	Atlantic	Jimbo's Super Go-Mobiles	Plymouth	Laser	1991	77
	193	2022-05-03	2022-04-03	43	\$78,619	Broadway	Gorben's Very Good Vehides	Chevrolet	Corvette	1962	34
	192	2022-05-04	2022-04-05	41	\$65,420	Bitter Lake	Jimbo's Super Go-Mobiles	Chrysler	Town & Cou...	2011	40
	189	2022-05-22	2022-04-23	23	\$95,083	Ravenna	Jimbo's Best Vehides	Mitsubishi	Truck	1987	60
	189	2022-05-22	2022-04-24	22	\$69,059	Atlantic	Queen Victoria's Rad Cars	Ford	Taurus	1986	27
	188	2022-05-25	2022-04-25	21	\$85,000	Bitter Lake	Jimbo's Pretty Great Four Wheeled Fun	Subaru	Legacy	2006	77

Customer Queries

Customer Query 1

This query allows a user / potential customer to search for open listings from vendors with a minimum vendor rating, by specific make, below max price and below max mileage.

This is a Stored Procedure with the following parameters:

- minimum vendor rating
- vehicle make
- max price
- max mileage.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `CustomerSearch_Rating_Make_Price_Mileage`
(in minVendorRating int, in vehicleMake varchar(20), in maxPrice int, in maxMileage int)
BEGIN
select
    t0.vendor_name,
    t0.AvgRating as 'Vendor Rating',
    AllOpenListings.price,
    Vehicle.make, Vehicle.model, Vehicle.body_type, Vehicle.year, Vehicle.color, Vehicle.engine_type, Vehicle.mileage
from AllOpenListings join Vehicle on AllOpenListings.vehicle_num = Vehicle.vehicle_num
join (
    select Vendor.vendor_num, Vendor.vendor_name, avg(Purchase.user_rating) as AvgRating
    from Purchase join Listing on Purchase.listing_num = Listing.listing_num
    join Vehicle on Vehicle.vehicle_num = Listing.vehicle_num
    join Vendor on Vendor.vendor_num = Vehicle.vendor_num
group by Vendor.vendor_num) as t0 on Vehicle.vendor_num = t0.vendor_num
where
    t0.avgRating >= minVendorRating and
    Vehicle.make = vehicleMake and
    Vehicle.mileage <= maxMileage and
    AllOpenListings.price <= maxPrice;
END
```

	vendor_name	Vendor Rating	price	make	model	body_type	year	color	engine_type	mileage
▶	Jennifer's Very Good Cars	5.0000	65617.69	Ford	Aerostar	Seden	1992	Khaki	Hybrid	36
	Jennifer's Very Good Cars	5.0000	36150.13	Ford	F250	Seden	2009	Orange	Electric	46
	Jennifer's Best Transport Devices	2.0000	59865.40	Ford	Mustang	SUV	1970	Violet	Electric	3
	Jennifer's Super Vehides	3.2500	77725.96	Ford	Taurus	Seden	1997	Aquamarine	Electric	7
	Jennifer's Super Vehides	3.2500	43579.96	Ford	F-Series	Seden	1997	Orange	Hybrid	25
	Jimothy's Super Cars	3.3333	18616.05	Ford	E350	SUV	2008	Mauv	Electric	61
	Jimothy's Rad Cars	2.5000	47448.78	Ford	E-Series	Truck	1991	Red	Electric	92
	Queen Victoria's Super Transport Devices	3.5000	68314.61	Ford	F-Series	SUV	1991	Aquamarine	Electric	46
	Jennifer's Super Duper Transport Devices	2.2500	22365.18	Ford	Expedition	Hachback	2001	Teal	Hybrid	68
	Jennifer's Super Duper Transport Devices	2.2500	19753.86	Ford	E-Series	Minivan	1986	Violet	Electric	57
	Queen Victoria's Super Go-Mobiles	2.0000	29727.13	Ford	Explorer	SUV	1996	Fuscia	Hybrid	39
	Gorben's Pretty Great Cars	5.0000	50780.57	Ford	Explorer	Seden	1992	Orange	Electric	52
	Jennifer's Pretty Great Cars	3.6667	62292.19	Ford	Focus	Coupe	2007	Fuscia	Electric	66
	Jennifer's Super Transport Devices	3.0000	30410.26	Ford	Escape	SUV	2006	Fuscia	Electric	99
	Gorben's Best Vehides	5.0000	19224.24	Ford	Contour	Coupe	1995	Mauv	Hybrid	65
	Queen Victoria's Super Cars	3.5000	43392.53	Ford	Escort	SUV	1996	Goldenrod	Hybrid	79

Customer Query 2

This query returns a month by month breakdown of the average monthly price of listings that meet customer specifications of:

- max price
- max mileage
- min year
- vehicle body type.

Customers can use queries like this to track trends in listing prices to determine the best time to buy the car they're looking for.

	Month	Monthly Average List Price
►	1	52542.498750
	2	57977.917500
	3	46214.436667
	4	52025.740000
	5	28992.430000
	6	45575.087692
	7	52393.131538
	8	62164.303333
	9	50576.771667
	10	51288.902500
	11	67415.830000
	12	57642.705000

```

• CREATE DEFINER=`root`@`localhost` PROCEDURE `GetPriceTrend_LastYear`
  (in priceMax int, in mileageMax int, in minYear int, in bodyType varchar(45))
BEGIN
  select t0.Month as 'Month', t0.AvgListPrice as 'Monthly Average List Price' from (
    select
      month(Listing.date) as 'Month',
      avg(Listing.price) as AvgListPrice
    from
      Listing join Vehicle on Listing.vehicle_num = Vehicle.vehicle_num
    where
      Listing.date >= date_sub(curdate(), interval 1 year) and
      Listing.date <= curdate() and
      Listing.price <= priceMax and
      Vehicle.mileage <= mileageMax and
      Vehicle.year >= minYear and
      Vehicle.body_type = bodyType
    group by month(Listing.date)
    order by month(Listing.date) asc) as t0;
END

```

Customer Query 3

This query returns the regions with the top 10 purchased models, giving first time customers information about popular vehicles per region.

```

208 • select
209     v.make as 'Brand',
210     v.model as 'Model',
211     r.region_name as 'Region',
212     count(p.purchase_num) as 'Total Sales',
213     format(avg(p.price), 2) as 'Average Price'
214 from Region r left join Listing l on r.region_num = l.region_num
215 join Vehicle v on l.vehicle_num = v.vehicle_num
216 join Purchase p on l.listing_num = p.listing_num
217 group by v.make, v.model, r.region_num
218 order by count(p.purchase_num) desc
219 limit 10;

```

	Brand	Model	Region	Total Sales	Average Price
►	Volvo	XC90	Maple Leaf	2	48,590.00
	Pontiac	Grand Prix	Greenwood	2	66,100.50
	Saturn	Ion	Greenwood	1	63,215.00
	Dodge	Charger	Eastlake	1	81,653.00
	Chevrolet	S10	Broadview	1	65,491.00
	Suzuki	Esteem	Broadmore	1	79,720.00
	Ford	Probe	View Ridge	1	36,968.00
	Mercury	Cougar	Sunset Hill	1	84,108.00
	GMC	Envoy	Broadmore	1	30,278.00
	Chevrolet	Camaro	Maple Leaf	1	93,666.00

Stored procedures

Stored Procedure 1: Get new listing by region

This stored procedure retrieves the newly listed (listed within 30 days) vehicles by region. This helps the customers search for new listing vehicles conveniently.

```
1 • CREATE DEFINER=`mm_cpssc502101team03`@`%` PROCEDURE `GetNewListingByRegion`(in date date, in
   region_number int)
2 BEGIN
3     select v.vehicle_num, v.vin, v.make, v.model, v.year, v.color, v.engine_type, v.body_type, v.
   mileage, v.transmission, v.drive_type, l.price, l.listing_num, l.date as 'Listing date', r.
   region_name
4     from Listing l inner join Region r on l.region_num = r.region_num
5     inner join Vehicle v on l.vehicle_num = v.vehicle_num
6     where TIMESTAMPDIFF(DAY, l.date, date) between 0 and 30 and region_number = r.region_num
7     order by 'Listing date';
8 END
```

```
call GetNewListingByRegion('2022-01-01', 1)
```

	vehicle_num	vin	make	model	year	color	engine_type	body_type	mileage	transmission	drive_type	price	listing_num	Listing date	region_name
▶	938	WAUVFAFH3AN426192	Jeep	Wrangler	2008	Indigo	Hybrid	SUV	50322	Automatic	4WD	69379.99	28	2021-12-02 07:47:00	Broadview
	531	1G6DV1EPXB0463087	Audi	90	1995	Yellow	Hybrid	SUV	21	Automatic	FWD	30693.98	168	2021-12-25 02:50:00	Broadview
	52	JHMGE8G38BC214815	Pontiac	Montana	2006	Blue	Hybrid	SUV	6	Automatic	FWD	24213.69	508	2021-12-27 11:25:00	Broadview
	5	2G4WC582861672918	Chevrolet	G-Series 2500	1996	Purple	Gas	SUV	16	Automatic	FWD	38881.37	548	2021-12-24 07:18:00	Broadview
	227	WBAYF8C55FD249187	Ford	E150	1984	Goldenrod	Electric	Sedan	45188	Automatic	FWD	70188.67	580	2021-12-03 07:47:00	Broadview
	562	1G6DC67A480065605	GMC	1500 Club Coupe	1992	Red	Electric	Sedan	72055	Automatic	4WD	31773.99	720	2021-12-26 02:50:00	Broadview

Stored Procedure 2: Get most viewed open listings





This stored procedure returns the most viewed open listings by region in the last (interval) days.

```

1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `GetMostViewedOpenListings`(in days_interval int, in region_num int)
2 BEGIN
3     select
4     agg.Listing_Total_Views as Total_Views,
5     AllOpenListings.listing_num,
6     AllOpenListings.date,
7     AllOpenListings.price,
8     Region.region_name,
9     Vendor.vendor_name,
10    Vehicle.make, Vehicle.model, Vehicle.year, Vehicle.color, Vehicle.engine_type,
11    Vehicle.vin, Vehicle.body_type, Vehicle.mileage, Vehicle.transmission, Vehicle.drive_type
12 from
13     AllOpenListings
14 join (
15     select
16         AllOpenListings.listing_num,
17         sum(View.times_viewed) as Listing_Total_Views
18     from AllOpenListings
19     join View on AllOpenListings.listing_num = View.listing_num
20     where
21         View.last_view > date_sub(curdate(), interval days_interval day) and
22         View.last_view < curdate()
23     group by AllOpenListings.listing_num
24     order by Listing_Total_Views desc
25 ) as agg on AllOpenListings.listing_num = agg.listing_num
26 join Vehicle
27     on AllOpenListings.vehicle_num = Vehicle.vehicle_num
28 join Vendor
29     on Vehicle.vendor_num = Vendor.vendor_num
30 join Region
31     on AllOpenListings.region_num = Region.region_num
32 where AllOpenListings.region_num = region_num
33 order by Total_Views desc;
34 END

```


1 • `call GetMostViewedOpenListings(30, 1);`

Result Grid		Filter Rows:	<input type="text"/>	Export:		Wrap Cell Content:									
	Total_Views	listing_num	date	price	region_name	vendor_name	make	model	year	color	engine_type	vin	body_type	mile	
▶	275	288	2022-04-15 03:33:00	77696.71	Broadview	Jimothy's Super Duper Go-Mobiles	Mazda	B-Series Plus	1993	Pink	Hybrid	1HGCP2F30CA734868	SUV	33	
	261	388	2022-04-12 20:09:00	17479.59	Broadview	Jennifer's Very Good Four Wheeled Fun	Jaguar	XJ Series	2003	Maroon	Electric	1GYS3DEFXER203656	SUV	60	
	257	488	2022-04-24 16:18:00	30753.41	Broadview	Jennifer's Super Vehicles	Suzuki	Grand Vitara	2002	Pink	Gas	2T1BU4EE6CC287637	SUV	32	
	163	840	2022-04-16 03:33:00	80305.03	Broadview	Jimbo's Pretty Great Cars	Mercury	Cougar	1970	Blue	Electric	WAULC58E64A952158	Seden	2831	
	159	620	2022-04-15 11:51:00	95001.87	Broadview	Jimothy's Pretty Great Transport Devices	Lamborghini	Gallardo	2004	Indigo	Gas	1G6KH5E64BU925262	Seden	9243	
	133	308	2022-05-01 09:41:00	29950.04	Broadview	Jennifer's Super Transport Devices	Nissan	Pathfinder	1996	Orange	Electric	JH4NA12641T887448	SUV	52	
	117	940	2022-04-13 20:09:00	18783.00	Broadview	Jimothy's Super Go-Mobiles	Mercedes-Benz	CLS-Class	2007	Pink	Electric	2HNYD18273H625377	Seden	1190	
	83	700	2022-04-05 23:21:00	19625.51	Broadview	Gorben's Pretty Great Cars	Geo	Tracker	1992	Green	Gas	WBAAEAC59AC625713	Seden	8489	
	59	740	2022-04-02 15:18:00	31807.19	Broadview	Gorben's Rad Vehides	Buick	Century	1992	Goldenrod	Electric	1G4PR5SK5C4205869	Seden	6590	
	51	860	2022-05-02 09:41:00	32356.00	Broadview	Queen Victoria's Best Vehides	Volvo	V70	2006	Aquamarine	Hybrid	WAURFAFR6DA461198	Seden	1612	
	7	760	2022-03-30 16:22:00	90941.02	Broadview	Queen Victoria's Best Transport Devices	Saab	5-Sep	2011	Pink	Hybrid	SCFFDAAM5EG448496	Seden	1738	

Stored Procedure 3: Get average price and mileage in \$10,000 brackets by vehicle make

This procedure returns the average mileage and purchase price per make, separated into \$10,000 brackets

```

1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `GetAveragePriceMileageByMake`(in vehicle_make varchar(20))
2 BEGIN
3     select
4         Concat(floor(((AllClosedListings.price ) / 10000) * 10000, ' - ',
5             floor(((AllClosedListings.price ) / 10000) + 1) * 10000) as price_bracket,
6         avg(AllClosedListings.price) as AvgListPrice,
7         avg(Purchase.price) as AvgPurchasePrice,
8         (avg(AllClosedListings.price) - avg(Purchase.price)) AvgPriceDifference,
9         avg(Vehicle.mileage)
10    from
11        AllClosedListings join Purchase on AllClosedListings.listing_num = Purchase.listing_num
12        join Vehicle on AllClosedListings.vehicle_num = Vehicle.vehicle_num
13    where Vehicle.make = vehicle_make
14    group by price_bracket order by price_bracket;
15 END

```

```
1 • call GetAveragePriceMileageByMake('Toyota');
```

Result Grid					
Filter Rows: <input type="text"/>					
Export: Wrap Cell Content:					
	price_bracket	AvgListPrice	AvgPurchasePrice	AvgPriceDifference	avg(Vehicle.mileage)
▶	10000 - 20000	17864.890000	14473.000000	3391.890000	11.0000
	30000 - 40000	31451.270000	27243.000000	4208.270000	94990.0000
	60000 - 70000	65545.890000	61751.000000	3794.890000	88584.0000
	70000 - 80000	74863.980000	70932.000000	3931.980000	66.0000

Frequently Used Views

These views are frequently used in the queries and stored procedures above.

All Open Listings

This view compares listings against purchases to return only the listings that have not been completed by a purchase.

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `mm_cpssc502101team03`.`allopenlistings` AS
6     SELECT
7         `mm_cpssc502101team03`.`listing`.`listing_num` AS `listing_num`,
8         `mm_cpssc502101team03`.`listing`.`date` AS `date`,
9         `mm_cpssc502101team03`.`listing`.`vehicle_num` AS `vehicle_num`,
10        `mm_cpssc502101team03`.`listing`.`price` AS `price`,
11        `mm_cpssc502101team03`.`listing`.`region_num` AS `region_num`,
12        `mm_cpssc502101team03`.`listing`.`employee_num` AS `employee_num`
13    FROM
14        (`mm_cpssc502101team03`.`listing`
15     LEFT JOIN `mm_cpssc502101team03`.`purchase` ON ((`mm_cpssc502101team03`.`listing`.`listing_num` = `mm_cpssc502101team03`.`purchase`.`purchase_num`)))
16    WHERE
17        (`mm_cpssc502101team03`.`purchase`.`purchase_num` IS NULL)

```

All Closed Listings

This view compares listings against purchases to return only the listings that have been completed by a purchase.

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `mm_cpssc502101team03`.`allclosedlistings` AS
6     SELECT
7         `mm_cpssc502101team03`.`listing`.`listing_num` AS `listing_num`,
8         `mm_cpssc502101team03`.`listing`.`date` AS `date`,
9         `mm_cpssc502101team03`.`listing`.`vehicle_num` AS `vehicle_num`,
10        `mm_cpssc502101team03`.`listing`.`price` AS `price`,
11        `mm_cpssc502101team03`.`listing`.`region_num` AS `region_num`,
12        `mm_cpssc502101team03`.`listing`.`employee_num` AS `employee_num`
13    FROM
14        (`mm_cpssc502101team03`.`listing`
15     LEFT JOIN `mm_cpssc502101team03`.`purchase` ON ((`mm_cpssc502101team03`.`listing`.`listing_num` = `mm_cpssc502101team03`.`purchase`.`purchase_num`)))
16    WHERE
17        (`mm_cpssc502101team03`.`purchase`.`purchase_num` IS NOT NULL)
```