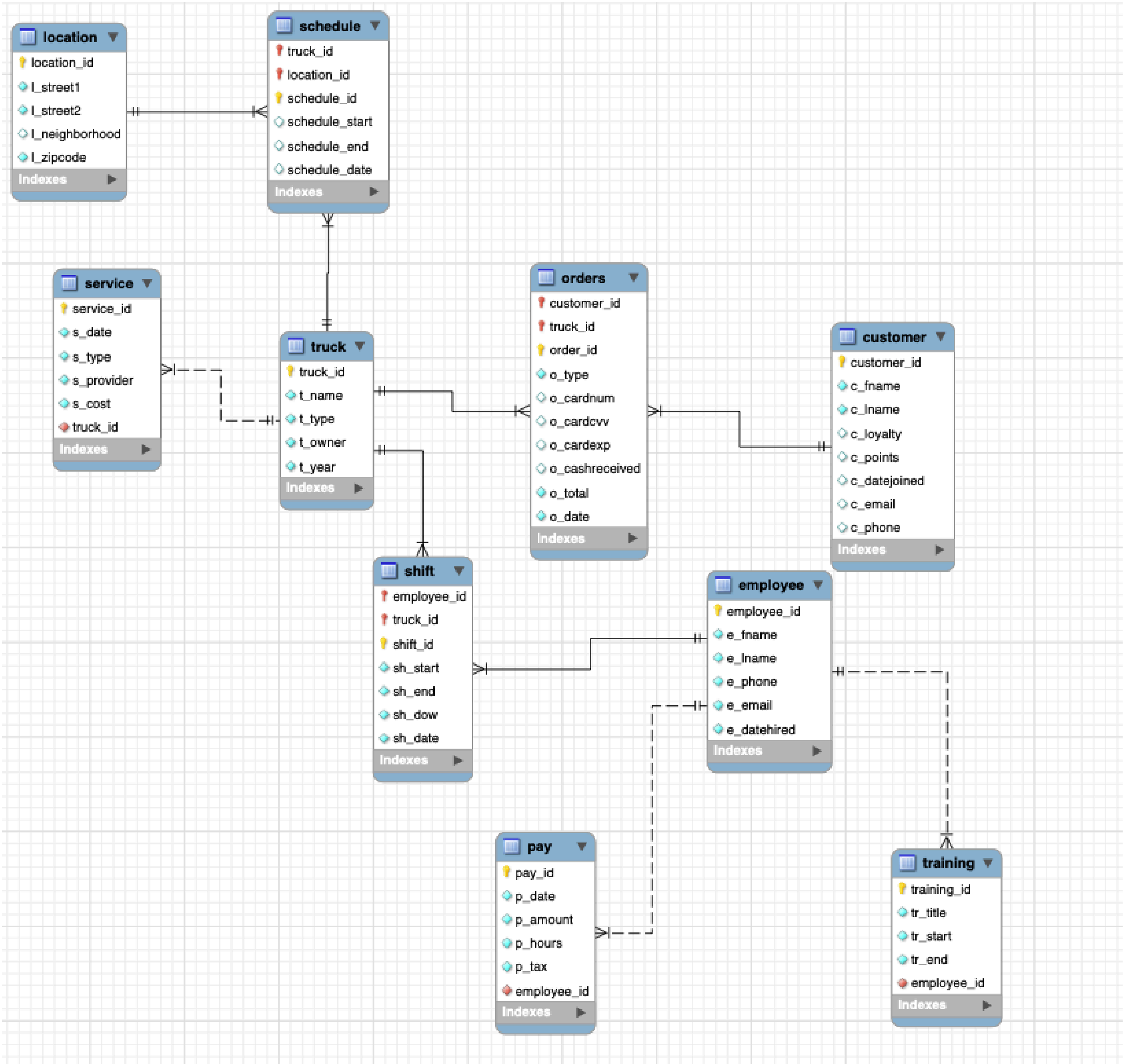
Team Project #2

Database Management (BUS-315)

Prof. Larson

Thomas Haller
Avery Myers
Aidan Lorsson

Data Model



Our final data model has ten tables that primarily focus on the relationships between trucks, employees, and customers. The purpose of our database is for FFC to be able to track where customers are placing orders, how experienced employees are and where they are being scheduled, as well as where trucks are located. In addition to these main goals, we also have a service table to keep track of truck maintenance and a pay table to monitor employee paystubs. We chose to exclude information on menus and inventory for trucks, as our database is more focused on employee and customer interactions with the trucks as well as the functionality of the trucks rather than the specifics of the food being served.

Our model has several many-to-many relationships as well as three one-to-many relationships. Truck and employee constitute a many-to-many relationship with shift as the associative table. Trucks have many employees, and employees can work on different trucks during different shifts. While truck and employee have an identifying relationship with shift, meaning both primary keys of truck and employee are part of shift's primary key, shift also has shift_id as part of its primary key. This is because an employee can work on a certain truck more than once. Truck and customer also constitute a many-to-many relationship with orders as the associative table. Trucks can have many customers, and customers can place orders at all four trucks. Order_id is the third component of the orders table's primary key, because customers can place an order at the same truck more than once. The third many-to-many relationship is between truck and location with an associative table called schedule. Trucks can move around to many locations in the Shenandoah area, and the same location can have different trucks. Schedule_id is part of the primary key in the associative table because it differentiates between daily scheduling. All three many-to-many relationships have a surrogate key in the associative tables along with the primary keys of the two independent tables. This is because all three associative tables require a third attribute for the primary key to be unique.

The employee table has one-to-many relationships with pay and training. The pay table covers biweekly paystubs, and employees get paid multiple times. Employees may also have multiple training certifications. The other one-to-many relationship in our data model is between truck and service. Trucks can have multiple service appointments, costing different amounts and for different maintenance issues.

The orders table tracks both cash and card orders, which is why the cash and card attributes are nullable. Furthermore, the attribute o_type tracks whether or not the order is online or in-person. We chose to consolidate online orders and in-person orders into one table to reduce redundancies in the database. The attributes covering credit card payments and cash payments can be null, assuming that a customer buys food from a truck through either a credit card or cash in an individual payment. However, the same customers can use cash or different cards for different orders.

The customer table tracks basic information like customer names for ordering purposes, but the table also tracks information on the loyalty program. Customers' reward points will be calculated in this table and will be updated when customers gain new reward points. The attribute c_loyalty is meant to capture the customer's loyalty status. Customers can be bronze, gold, or platinum members of the loyalty program, so we included this attribute to track what level an individual customer is. We assume while most customers will be part of the rewards program, some customers may opt out of that option. Therefore, the columns related to the loyalty program are nullable.

Data Dictionaries

Employee

	Field	Data Type	Key Type	Description	Constraints	Allowable Values	Example
1	employee_id	INT	Primary Key	Unique identifier for each employee	Not Null, Unique	Integer, Unique	1
2	e_fname	VARCHAR(45)	None	Employee's first name	Not Null	String, Max Length	Alex

						45 chars	
3	e_lname	VARCHAR(45)	None	Employee's last name	Not Null	String, Max Length 45 chars	Johnson
4	e_phone	VARCHAR(45)	None	Employee's phone number	Not Null	String, Phone number format	333-444-5555
5	e_email	VARCHAR(45)	None	Employee's email address	Not Null, Unique	String, Email format	alexjohnso@email.com
7	e_datehired	DATE	None	The date the employee was hired	Not Null	Date, YYYY-MM-DD	2022-01-01

Orders

Field	Data Type	Key Type	Description	Constraints	Allowable Values	Example
order_id	INT	Primary Key	Unique identifier for each order	Not Null, Unique	Integer, Unique	1001
customer_id	INT	Primary Key	Reference to the customer's ID	Not Null	Integer, Existing customer ID	1
truck_id	INT	Primary Key	Reference to the truck's ID	Not Null	Integer, Existing truck ID	1

o_date	DATE	None	Date when the order was placed	Not Null	Date, YYYY-MM-DD	2022-01-15
o_type	VARCHAR(225)	None	Type of the order (e.g., Pickup, Delivery)	Not Null	String	'Online'
o_cardnum	VARCHAR(16)	None	Credit card number used for the order		String, Card number format	'1234123412341234'
o_cardcvv	VARCHAR(4)	None	Card security code		String, max length 4 characters	'123'
o_cardexp	VARCHAR(45)	None	Card expiration date		String, max length 45 characters	2025-08
o_cashreceived	DECIMAL(10,2)	None	Cash received for the order		Decimal, 2 decimal places	NULL
o_total	DECIMAL(10,2)	None	Total price of the order	Not Null	Decimal, 2 decimal places	45.99

Truck

	Field	Data Type	Key Type	Description	Constraints	Allowable Values	Example
1	t_id	INT	Primary Key	Unique identifier for the truck	Not Null, Unique	Integer, Unique	1

2	t_name	VARCHAR(45)	None	Name or model of the truck	Not Null	String	'Lex Mex”
3	t_type	VARCHAR(45)	None	Type of the truck	Not Null	String, Max Length 45 chars	Chevy P30
4	t_year	VARCHAR(45)	None	Year of the truck model	Not Null	Integer, Year	2021
5	t_owner	VARCHAR(45)	None	Owner of the truck	Not Null	String	'Mac Fowler”

Shift

Field	Data Type	Key Type	Description	Constraints	Allowable Values	Example
shift_id	INT	Primary Key	Unique identifier for each shift	Not Null, Unique	Integer, Unique	1
employee_id	INT	Primary Key	Reference to the employee's ID	Not Null	Integer, Existing employee ID	1
truck_id	INT	Primary Key	Reference to the truck's ID	Not Null	Integer, Existing truck ID	1
sh_start	TIME	None	Time when the shift starts	Not Null	Time, HH:MM:SS	'08:15:30'

sh_end	TIME	None	Time when the shift ends	Not Null	Time, HH:MM:SS	'17:05:30'
sh_dow	VARCHAR(45)	None	Day of the week when the shift takes place	Not Null	String, Day name	'Monday'
sh_date	DATE	None	Date when the shift takes place	Not Null	Date, YYYY-MM-DD	2022-01-15

<u>10 Business Questions, Justifications, and Their Queries</u>

1. **Employee Scheduling:**

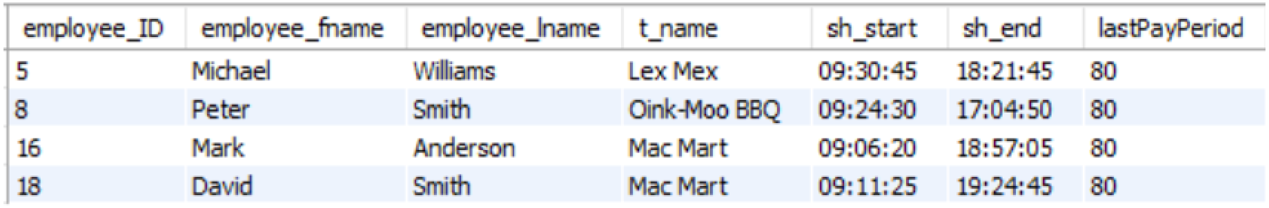
Question: Which employees who worked 80 hours for their last paycheck are scheduled to start their shift between 9 am and 10 am on Monday on which food trucks?

Justification: This query can run for FFC to ensure that food trucks are adequately staffed with full-time workers. FFC can use this query regularly to look at daily data by changing the day of the week and the time frame to see which employees are being staffed on which trucks.

Query:

```
SELECT employee_ID, employee_fname, employee_lname, t_name, sh_start, sh_end, p_hours AS lastPayPeriod
FROM employee
JOIN shift USING (employee_id) JOIN truck USING(truck_id) JOIN pay USING(employee_id)
WHERE sh_dow = 'Monday' AND sh_start BETWEEN '09:00:00' AND '10:00:00' AND p_hours = 80;
```

Result:



employee_ID	employee_fname	employee_lname	t_name	sh_start	sh_end	lastPayPeriod
5	Michael	Williams	Lex Mex	09:30:45	18:21:45	80
8	Peter	Smith	Oink-Moo BBQ	09:24:30	17:04:50	80
16	Mark	Anderson	Mac Mart	09:06:20	18:57:05	80
18	David	Smith	Mac Mart	09:11:25	19:24:45	80

2. Trucks and Locations:

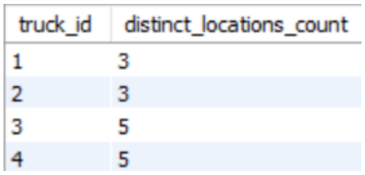
Question: Identify the trucks that have served in at least two different locations during 2023, and show the count of distinct locations for each of those trucks.

Justification: This query can run yearly or quarterly to help FCC identify trucks that are reaching customers in a greater geographical area than others. By identifying which trucks have been to more locations, FFC can contact truck owners to gather data on which locations were more optimal for business.

Query:

```
SELECT truck_id, COUNT(DISTINCT location_id) AS distinct_locations_count
FROM schedule
WHERE YEAR(schedule_date) = 2023
GROUP BY truck_id
HAVING truck_id IN (SELECT truck_id FROM schedule
WHERE YEAR(schedule_date) = 2023
GROUP BY truck_id HAVING COUNT(DISTINCT location_id) >= 2);
```

Result:



truck_id	distinct_locations_count
1	3
2	3
3	5
4	5

3. Scheduling

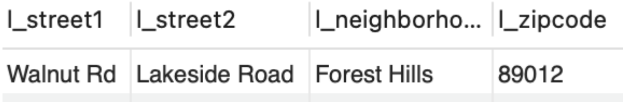
Question: Are there any locations that haven't been on the schedule?

Justification: This query shows FFC what desired locations haven't been scheduled. This query can help FFC reach out to truck owners to prioritize scheduling their trucks there so that the company can collect data on all 10 of the company's desired locations.

Query:

```
SELECT l_street1, l_street2, l_neighborhood, l_zipcode
FROM location
LEFT JOIN schedule USING(location_id)
WHERE truck_id IS NULL;
```

Result:



l_street1	l_street2	l_neighborho...	l_zipcode
Walnut Rd	Lakeside Road	Forest Hills	89012

4. Employee Training:

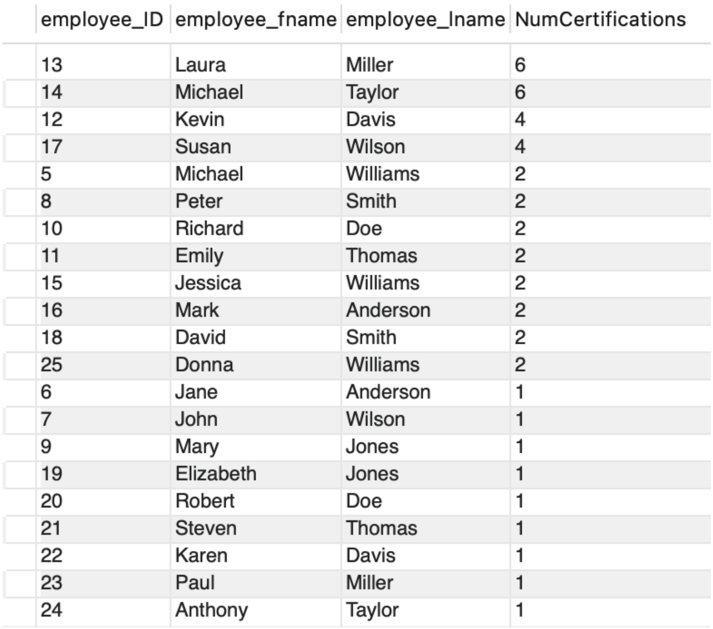
QuestionL: List the number of training certifications held by employees who have an above-average paycheck than other employees at FFC, taking into consideration tax withholdings from pay stubs. Order the results by the highest to lowest number of training certifications.

Justification: A question like this can help FFC determine if there is a correlation between high-paid employees and formal training procedures. Workers who get paid more than an average employee should be high-quality workers who have multiple training qualifications. This business question can ensure FFC that employees who are getting rewarded with high pay have multiple training certifications. Furthermore, the query allows FFC executives to reward highly-trained employees and give warnings to employees with a low number of training certifications.

Query:

```
SELECT employee_ID, employee_fname, employee_lname, COUNT(tr_title) AS NumCertifications
FROM training JOIN employee USING(employee_id) JOIN pay USING(employee_id)
WHERE (p_amount - p_tax) > (SELECT AVG(p_amount - p_tax) FROM pay)
GROUP BY employee_ID
ORDER BY NumCertifications DESC;
```

Result:



employee_ID	employee_fname	employee_lname	NumCertifications
13	Laura	Miller	6
14	Michael	Taylor	6
12	Kevin	Davis	4
17	Susan	Wilson	4
5	Michael	Williams	2
8	Peter	Smith	2
10	Richard	Doe	2
11	Emily	Thomas	2
15	Jessica	Williams	2
16	Mark	Anderson	2
18	David	Smith	2
25	Donna	Williams	2
6	Jane	Anderson	1
7	John	Wilson	1
9	Mary	Jones	1
19	Elizabeth	Jones	1
20	Robert	Doe	1
21	Steven	Thomas	1
22	Karen	Davis	1
23	Paul	Miller	1
24	Anthony	Taylor	1

5. Customer Loyalty Trends:

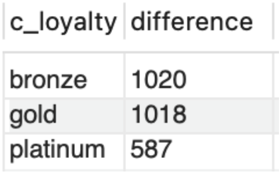
Question: Calculate the difference in days from the first customer to join the loyalty program to the most recent customer to join the program. Group the difference by bronze loyalty members, then gold loyalty members, then platinum loyalty members.

Justification: This query can help FFC evaluate the scope of the customer loyalty program as well as determine how recently customers began to level up in loyalty tiers.

Query:

```
SELECT c_loyalty, DATEDIFF(MAX(c_datejoined),MIN(c_datejoined)) AS difference FROM customer
GROUP BY c_loyalty
ORDER BY c_loyalty ASC;
```

Result:



c_loyalty	difference
bronze	1020
gold	1018
platinum	587

6. **Employee Compensation:**

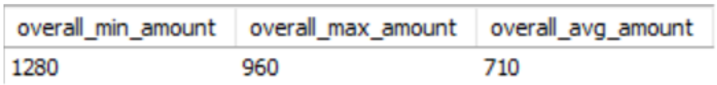
Question: Calculate the minimum, maximum, and average amount an employee has been paid between September and November.

Justification: These calculations will yield a distribution of employee pay over the past quarter, assuming that workers are paid biweekly. FFC can find this information every quarter to see how employee work trends are changing and to make sure employees are being paid enough.

Query:

```
SELECT MIN(p_amount) AS overall_min_amount, MAX(p_amount) AS overall_max_amount,
AVG(p_amount) AS overall_avg_amount
FROM pay
WHERE MONTH(p_date) BETWEEN 9 AND 11;
```

Result:



overall_min_amount	overall_max_amount	overall_avg_amount
1280	960	710

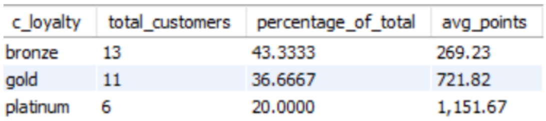
7. **Customer Loyalty:**

Question: What percentage of customers are categorized as bronze, gold, and platinum loyalty members, and what is the average number of points in each category?

Justification: This information can be found every couple of months so that FFC can track if customers are moving up in the loyalty point system as well as see the distribution of customers across the program.

Query:
SELECT c_loyalty, COUNT(customer_id) AS total_customers, COUNT(customer_id) / SUM(COUNT(customer_id)) OVER () * 100 AS percentage, FORMAT(AVG(c_points),2) AS avg_points
FROM customer
GROUP BY c_loyalty
ORDER BY c_loyalty;

Result:



c_loyalty	total_customers	percentage_of_total	avg_points
bronze	13	43.3333	269.23
gold	11	36.6667	721.82
platinum	6	20.0000	1,151.67

8. **Shift Efficiency:**

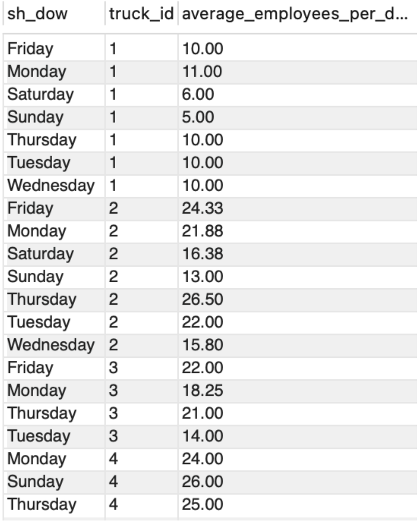
Question: What is the average number of employees working on each day of the week during 2023 for each truck? Order the results by trucks in ascending order, and round the average number of employees to the hundredth.

Justification: This question can run at the end of each fiscal year so that FFC managers can see how employees are being allocated throughout the week. This data can help the company improve shift and service efficiency to lead to faster customer service and reduced wait times.

Query:
SELECT sh_dow, truck_id, FORMAT(AVG(DISTINCT employee_id),2) AS average_employees_per_day

FROM shift
WHERE YEAR(sh_date) = 2023
GROUP BY sh_dow, truck_id
ORDER BY truck_id;

Result:



sh_dow	truck_id	average_employees_per_d...
Friday	1	10.00
Monday	1	11.00
Saturday	1	6.00
Sunday	1	5.00
Thursday	1	10.00
Tuesday	1	10.00
Wednesday	1	10.00
Friday	2	24.33
Monday	2	21.88
Saturday	2	16.38
Sunday	2	13.00
Thursday	2	26.50
Tuesday	2	22.00
Wednesday	2	15.80
Friday	3	22.00
Monday	3	18.25
Thursday	3	21.00
Tuesday	3	14.00
Monday	4	24.00
Sunday	4	26.00
Thursday	4	25.00

9. **Truck Maintenance Costs:**

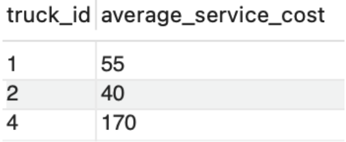
Question: What is the average service cost of propane-related repairs for each truck in 2022?

Justification: This question can be used every year to determine which trucks are costing FFC the most money based on certain maintenance issues. This data can help the company decide when to replace trucks based on how much money is being spent on certain repairs, such as propane refills.

Query:

```
SELECT truck_id, AVG(s_cost) AS average_service_cost
FROM service
WHERE s_type REGEXP 'Propane' and YEAR(s_date) = '2022'
GROUP BY truck_id
ORDER BY truck_id;
```

Result:



truck_id	average_service_cost
1	55
2	40
4	170

10. **Revenue Calculation:**

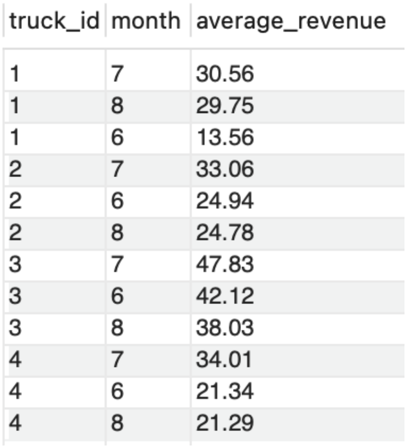
Question: What is the average revenue per month generated by each truck in the summer of 2023, considering both online and in-person orders? Order the results by ascending truck numbers, as well as most to least revenue, and round the average revenue to the hundredth.

Justification: Justification: This query is useful for assessing the financial performance of each food truck in the Summer of 2023. By understanding the average revenue generated by each truck, FFC can identify top-performing trucks and strategize marketing or operational efforts to enhance the profitability of underperforming ones. This query can be executed for any season of the year by changing the months around in the WHERE clause. This information aids in making informed business decisions and allocating resources effectively for improved overall revenue.

Query:

```
SELECT truck_id, MONTH(o_date) AS month, FORMAT(AVG(o_total),2) AS average_revenue
FROM orders
JOIN truck USING(truck_id)
WHERE YEAR(o_date) = 2023 AND MONTH(o_date) BETWEEN '6' AND '8'
GROUP BY MONTH(o_date), truck_id
ORDER BY truck_id, average_revenue DESC;
```

Result:



truck_id	month	average_revenue
1	7	30.56
1	8	29.75
1	6	13.56
2	7	33.06
2	6	24.94
2	8	24.78
3	7	47.83
3	6	42.12
3	8	38.03
4	7	34.01
4	6	21.34
4	8	21.29

Query Matrix:

	Query 1	**Q 2**	**Q 3**	**Q 4**	**Q 5**	**Q 6**	**Q 7**	**Q 8**	**Q 9**	**Q 10**
1. Subquery		x		x						
2. BETWEEN	x									x
3. GROUP BY				x	x		x	x	x	x
4. HAVING		x								

5. ORDER BY				x	x		x	x	x	x
6. IN/NOT IN		x								
7. Aggregate function				x	x	x		x	x	
8. REGEXP									x	
9. Date function(s) - MONTH, YEAR, DATEDIFF					x	x		x	x	x
10. Calculation				x			x			
11. LEFT JOIN		x								
12. DISTINCT		x						x		
13. FORMAT							x	x		x