Isaac Hill
CS 3520-X02
Tavish Burnah

Project 1

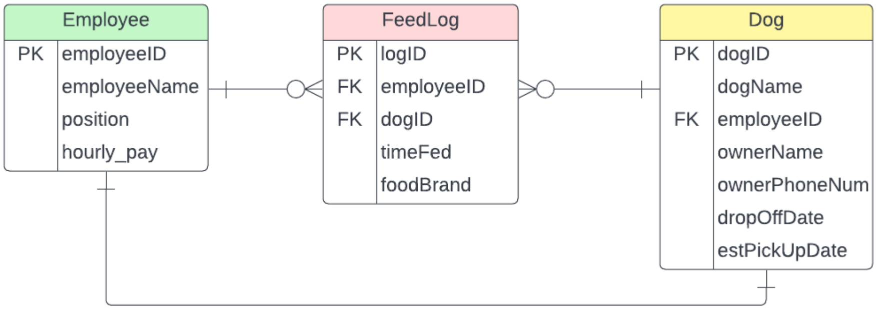
Scenario 1: Dog Boarding Facility Feeding Tracker
A Dog Boarding Facility wants to have a program that is able to track what employees are assigned to certain dogs. This will also track what time the dog was fed and what type of food was given to the dog. The dogs should also have data on their owner's name, phone #, the time they were dropped off, and the estimated time for them to be picked up. Employees must be assigned to at least one dog.

Scenario 2: Bike Shop Commission Tracker
A bike shop wants to have a program that is able to track what bike(s) employees sold to a customer. This should also track the commission rate that the employee has so they can make money off of a sale. The customer order should have an employee associated with it with their employeeID as well as the total amount of the order. An order will only have one bike associated with it but can have varying quantities of said bike.

Scenario 3: Towing Company Car Tracker
A towing company wants to have a program that is able to track what cars have been in their lot. Each car will have a towing log that shows what employee was responsible for towing the car, when the car was towed, location of the tow, and the price to get it out. Employees that tow a car will also receive a commission based on each tow. Employees working for the company don't have to be towing cars.

ERDs:

Scenario 1: Dog Boarding Facility Feeding Tracker

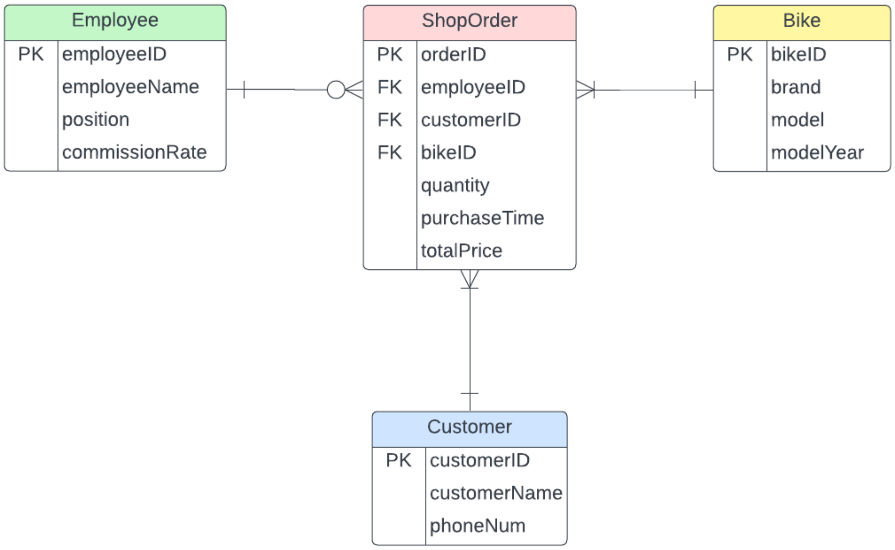


Employee	
PK	employeeID
	employeeName
	position
	hourly_pay

FeedLog	
PK	logID
FK	employeeID
FK	dogID
	timeFed
	foodBrand

Dog	
PK	dogID
	dogName
FK	employeeID
	ownerName
	ownerPhoneNum
	dropOffDate
	estPickUpDate

Scenario 2: Bike Shop Commission Tracker



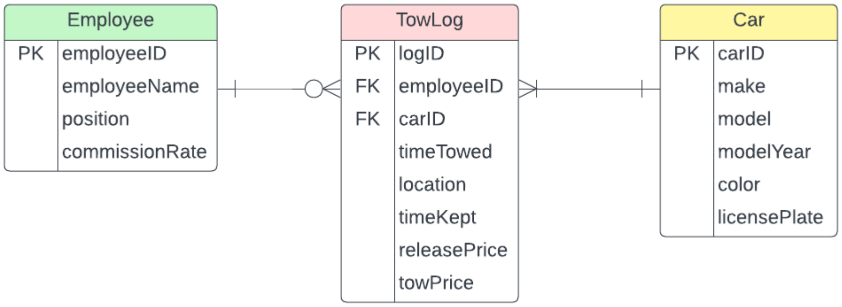
Employee	
PK	employeeID
	employeeName
	position
	commissionRate

ShopOrder	
PK	orderID
FK	employeeID
FK	customerID
FK	bikeID
	quantity
	purchaseTime
	totalPrice

Bike	
PK	bikeID
	brand
	model
	modelYear

Customer	
PK	customerID
	customerName
	phoneNum

Scenario 3: Towing Company Car Tracker



Employee	
PK	employeeID
	employeeName
	position
	commissionRate

TowLog	
PK	logID
FK	employeeID
FK	carID
	timeTowed
	location
	timeKept
	releasePrice
	towPrice

Car	
PK	carID
	make
	model
	modelYear
	color
	licensePlate

Scenario 1

Database and Table Creation SQL:

```
CREATE DATABASE Feeding_Tracker;
CREATE TABLE employee
( employeeID   INT NOT NULL primary key,
  employeeName VARCHAR(30),
  position      VARCHAR(30),
  hourly_pay   DECIMAL(5,2)
);

CREATE TABLE dog
( dogID         INT NOT NULL primary key,
  dogName       VARCHAR(30),
  employeeID    INT NOT NULL,
  foreign key (employeeID) references employee(employeeID),
  ownerName     VARCHAR(30),
  ownerPhoneNum  VARCHAR(15),
  dropOffDate   DATE,
  estPickUpDate  DATE
);

CREATE TABLE feedlog
( logID    INT NOT NULL primary key,
  employeeID INT,
  dogID    INT,
  foreign key (employeeID) references employee(employeeID),
  foreign key (dogID) references dog(dogID),
  timeFed  DATETIME,
  foodBrand  VARCHAR(30)
);
```

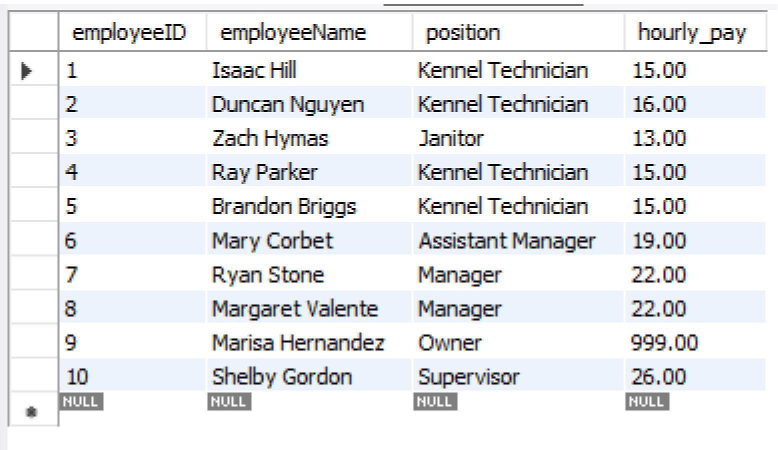
Data Entry:

```
INSERT INTO employee (employeeID, employeeName, position, hourly_pay)
VALUES (0001, "Isaac Hill", "Kennel Technician", 15.00),
      (0002, "Duncan Nguyen", "Kennel Technician", 16.00),
      (0003, "Zach Hymas", "Janitor", 13.00),
      (0004, "Ray Parker", "Kennel Technician", 15.00),
      (0005, "Brandon Briggs", "Kennel Technician", 15.00),
      (0006, "Mary Corbet", "Assistant Manager", 19.00),
      (0007, "Ryan Stone", "Manager", 22.00),
      (0008, "Margaret Valente", "Manager", 22.00),
      (0009, "Marisa Hernandez", "Owner", 999.00),
      (0010, "Shelby Gordon", "Supervisor", 26.00);
```

```
INSERT INTO dog (dogID, dogName, employeeID, ownerName,
ownerPhoneNum, dropOffDate, estPickUpDate)
VALUES (0001, 'Dexter', 0003, 'Mike Tamms', '801-278-7234',
'2023-02-15', '2023-03-05'),
      (0002, 'Eris', 0009, 'Marisa Hernandez', '435-555-2314',
'2023-02-23', '2023-03-15'),
      (0003, 'Hunny', 0005, 'David Tannen', '614-282-2681',
'2023-02-03', '2023-03-17'),
      (0004, 'Elroy', 0001, 'Blake Jenson', '607-288-7494',
'2023-02-01', '2023-03-10'),
      (0005, 'Melly', 0007, 'Logan Bryne', '352-452-6029',
'2023-02-02', '2023-03-09'),
      (0006, 'Harley', 0004, 'Ray Parker', '760-396-1056',
'2023-02-15', '2023-03-08'),
      (0007, 'Maverick', 0006, 'Aiden Foster', '434-888-2083',
'2023-02-20', '2023-03-05'),
      (0008, 'Kayana', 0010, 'Jacob Squire', '910-686-5252',
'2023-02-21', '2023-03-04'),
      (0009, 'Soldier', 0008, 'Robin Wayne', '201-555-7584',
'2023-02-23', '2023-03-01'),
      (0010, 'Minty', 0002, 'Arthur Smith', '952-536-8488',
'2023-02-23', '2023-03-02');

INSERT INTO feedlog (logID, employeeID, dogID, timeFed, foodBrand)
VALUES #(0001, 0003, 0001, '2023-03-01 10:30:23', 'Instinct')
      (0002, 0009, 0002, '2023-03-01 10:25:23', 'Purina'),
      (0003, 0005, 0003, '2023-03-01 10:40:45', 'Blue Buffalo'),
      (0004, 0001, 0004, '2023-03-01 10:45:00', 'Nulo'),
      (0005, 0007, 0005, '2023-03-01 10:33:05', 'Purina'),
      (0006, 0004, 0006, '2023-03-01 10:10:10', 'Purina'),
      (0007, 0006, 0007, '2023-03-01 10:24:27', 'Eagle Pack'),
      (0008, 0010, 0008, '2023-03-01 10:55:37', 'Nulo'),
      (0009, 0008, 0009, '2023-03-01 10:11:11', 'Instinct'),
      (0010, 0002, 0010, '2023-03-01 10:03:55', 'Purina');
```

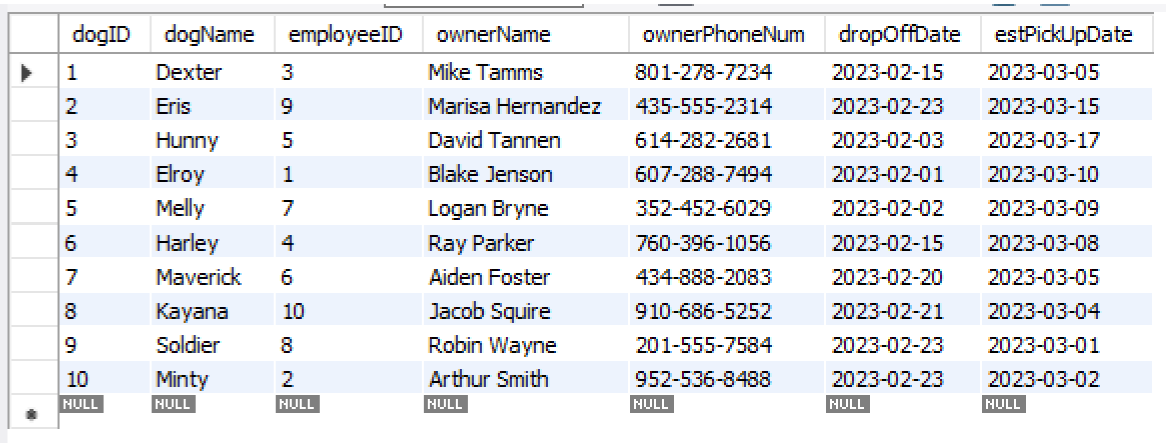
Database Pictures:

Employee Table



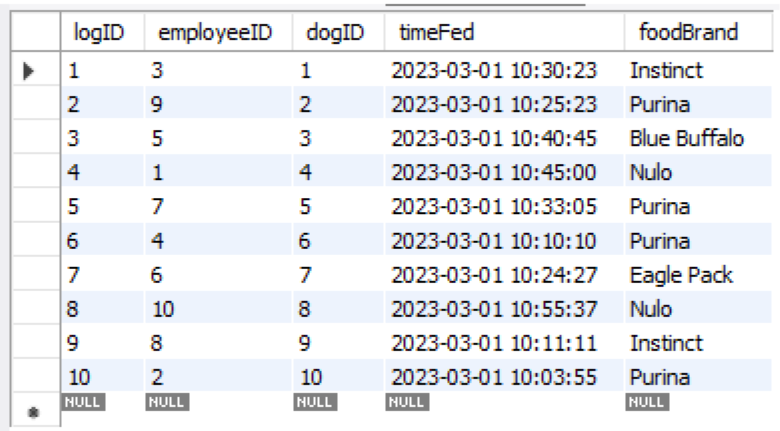
employeeID	employeeName	position	hourly_pay
1	Isaac Hill	Kennel Technician	15.00
2	Duncan Nguyen	Kennel Technician	16.00
3	Zach Hymas	Janitor	13.00
4	Ray Parker	Kennel Technician	15.00
5	Brandon Briggs	Kennel Technician	15.00
6	Mary Corbet	Assistant Manager	19.00
7	Ryan Stone	Manager	22.00
8	Margaret Valente	Manager	22.00
9	Marisa Hernandez	Owner	999.00
10	Shelby Gordon	Supervisor	26.00
NULL	NULL	NULL	NULL

Dog Table



dogID	dogName	employeeID	ownerName	ownerPhoneNum	dropOffDate	estPickUpDate
1	Dexter	3	Mike Tamms	801-278-7234	2023-02-15	2023-03-05
2	Eris	9	Marisa Hernandez	435-555-2314	2023-02-23	2023-03-15
3	Hunny	5	David Tannen	614-282-2681	2023-02-03	2023-03-17
4	Elroy	1	Blake Jenson	607-288-7494	2023-02-01	2023-03-10
5	Melly	7	Logan Bryne	352-452-6029	2023-02-02	2023-03-09
6	Harley	4	Ray Parker	760-396-1056	2023-02-15	2023-03-08
7	Maverick	6	Aiden Foster	434-888-2083	2023-02-20	2023-03-05
8	Kayana	10	Jacob Squire	910-686-5252	2023-02-21	2023-03-04
9	Soldier	8	Robin Wayne	201-555-7584	2023-02-23	2023-03-01
10	Minty	2	Arthur Smith	952-536-8488	2023-02-23	2023-03-02
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Feed Log Table

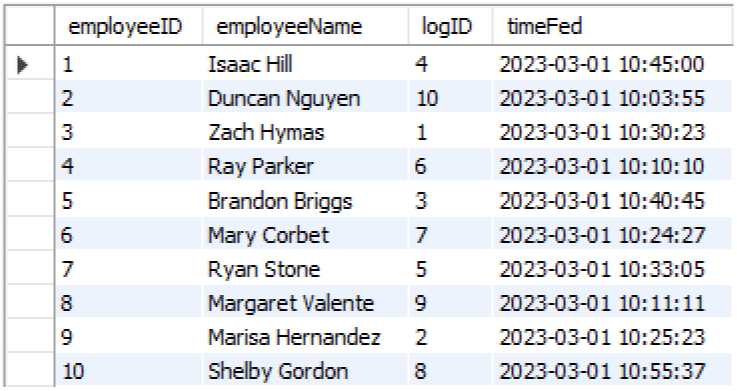


logID	employeeID	dogID	timeFed	foodBrand
1	3	1	2023-03-01 10:30:23	Instinct
2	9	2	2023-03-01 10:25:23	Purina
3	5	3	2023-03-01 10:40:45	Blue Buffalo
4	1	4	2023-03-01 10:45:00	Nulo
5	7	5	2023-03-01 10:33:05	Purina
6	4	6	2023-03-01 10:10:10	Purina
7	6	7	2023-03-01 10:24:27	Eagle Pack
8	10	8	2023-03-01 10:55:37	Nulo
9	8	9	2023-03-01 10:11:11	Instinct
10	2	10	2023-03-01 10:03:55	Purina
NULL	NULL	NULL	NULL	NULL

mySQL Queries:

Query 1:

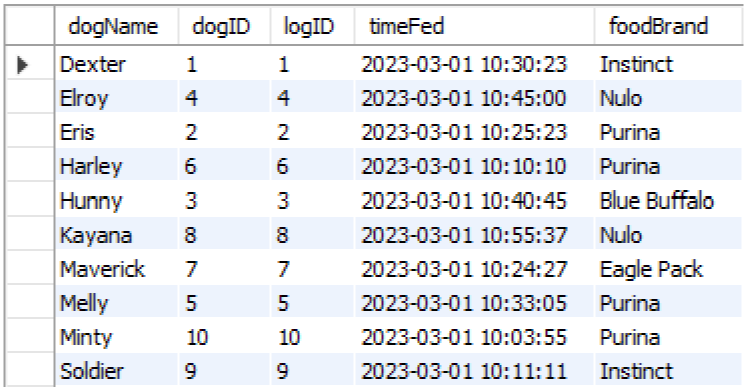
```
#Joins Employee with the feed log to show what time they fed the dog
and is ordered by employee ID

SELECT employee.employeeID, employee.employeeName, feedlog.logID,
feedlog.timeFed
FROM employee
LEFT JOIN feedlog ON employee.employeeID = feedlog.employeeID
ORDER BY employee.employeeID;
```



employeeID	employeeName	logID	timeFed
1	Isaac Hill	4	2023-03-01 10:45:00
2	Duncan Nguyen	10	2023-03-01 10:03:55
3	Zach Hymas	1	2023-03-01 10:30:23
4	Ray Parker	6	2023-03-01 10:10:10
5	Brandon Briggs	3	2023-03-01 10:40:45
6	Mary Corbet	7	2023-03-01 10:24:27
7	Ryan Stone	5	2023-03-01 10:33:05
8	Margaret Valente	9	2023-03-01 10:11:11
9	Marisa Hernandez	2	2023-03-01 10:25:23
10	Shelby Gordon	8	2023-03-01 10:55:37

Query 2:

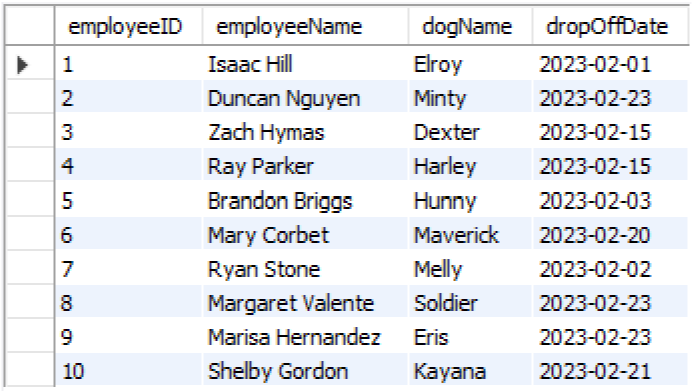
```
#Joins Dog with the feed log to show what time they were fed and what
type of food they ate and is ordered by dog name

SELECT dog.dogName, dog.dogID, feedlog.logID, feedlog.timeFed,
feedlog.foodBrand
FROM dog
LEFT JOIN feedlog ON dog.dogID = feedlog.dogID
ORDER BY dog.dogName;
```



dogName	dogID	logID	timeFed	foodBrand
Dexter	1	1	2023-03-01 10:30:23	Instinct
Elroy	4	4	2023-03-01 10:45:00	Nulo
Eris	2	2	2023-03-01 10:25:23	Purina
Harley	6	6	2023-03-01 10:10:10	Purina
Hunny	3	3	2023-03-01 10:40:45	Blue Buffalo
Kayana	8	8	2023-03-01 10:55:37	Nulo
Maverick	7	7	2023-03-01 10:24:27	Eagle Pack
Melly	5	5	2023-03-01 10:33:05	Purina
Minty	10	10	2023-03-01 10:03:55	Purina
Soldier	9	9	2023-03-01 10:11:11	Instinct

Query 3:

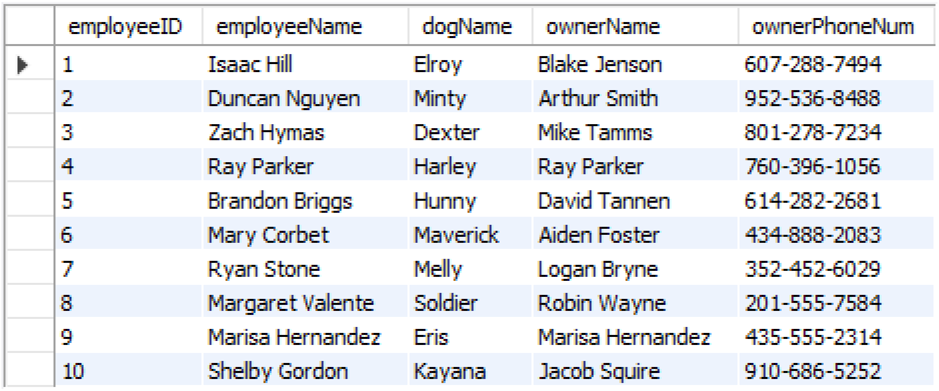
```
#Joins employee with the dog they are responsible for and shows the
time the dog got dropped off and is ordered by employee ID

SELECT employee.employeeID, employee.employeeName, dog.dogName,
dog.dropOffDate
FROM employee
LEFT JOIN dog ON employee.employeeID = dog.employeeID
ORDER BY employee.employeeID;
```



	employeeID	employeeName	dogName	dropOffDate
▶	1	Isaac Hill	Elroy	2023-02-01
	2	Duncan Nguyen	Minty	2023-02-23
	3	Zach Hymas	Dexter	2023-02-15
	4	Ray Parker	Harley	2023-02-15
	5	Brandon Briggs	Hunny	2023-02-03
	6	Mary Corbet	Maverick	2023-02-20
	7	Ryan Stone	Melly	2023-02-02
	8	Margaret Valente	Soldier	2023-02-23
	9	Marisa Hernandez	Eris	2023-02-23
	10	Shelby Gordon	Kayana	2023-02-21

Query 4:

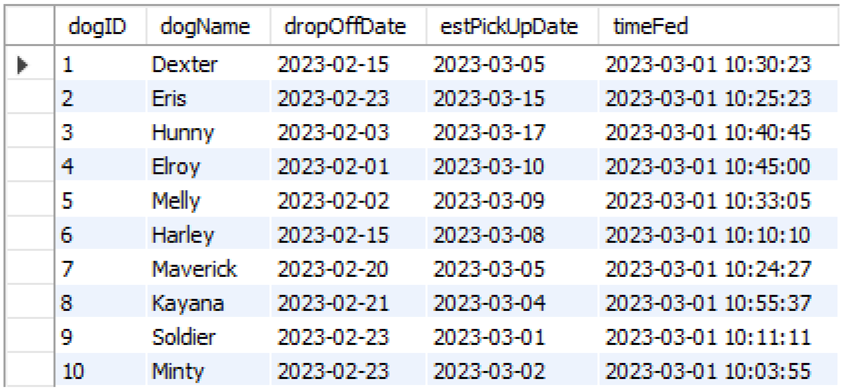
```
#Joins employee with the dog they are responsible for and shows the
owner and owner phone number and is ordered by employee name

SELECT employee.employeeID, employee.employeeName, dog.dogName,
dog.ownerName, dog.ownerPhoneNum
FROM employee
LEFT JOIN dog ON employee.employeeID = dog.employeeID
ORDER BY employee.employeeID;
```



	employeeID	employeeName	dogName	ownerName	ownerPhoneNum
▶	1	Isaac Hill	Elroy	Blake Jenson	607-288-7494
	2	Duncan Nguyen	Minty	Arthur Smith	952-536-8488
	3	Zach Hymas	Dexter	Mike Tamms	801-278-7234
	4	Ray Parker	Harley	Ray Parker	760-396-1056
	5	Brandon Briggs	Hunny	David Tannen	614-282-2681
	6	Mary Corbet	Maverick	Aiden Foster	434-888-2083
	7	Ryan Stone	Melly	Logan Bryne	352-452-6029
	8	Margaret Valente	Soldier	Robin Wayne	201-555-7584
	9	Marisa Hernandez	Eris	Marisa Hernandez	435-555-2314
	10	Shelby Gordon	Kayana	Jacob Squire	910-686-5252

Query 5:

```
#Joins dog with feed log to show the dog dropoff time, pickup time,
and the time they were last fed and is ordered by dog ID

SELECT dog.dogID, dog.dogName, dog.dropOffDate, dog.estPickUpDate,
feedlog.timeFed
FROM dog
LEFT JOIN feedlog ON dog.dogID = feedlog.dogID
ORDER BY dog.dogID;
```



	dogID	dogName	dropOffDate	estPickUpDate	timeFed
▶	1	Dexter	2023-02-15	2023-03-05	2023-03-01 10:30:23
	2	Eris	2023-02-23	2023-03-15	2023-03-01 10:25:23
	3	Hunny	2023-02-03	2023-03-17	2023-03-01 10:40:45
	4	Elroy	2023-02-01	2023-03-10	2023-03-01 10:45:00
	5	Melly	2023-02-02	2023-03-09	2023-03-01 10:33:05
	6	Harley	2023-02-15	2023-03-08	2023-03-01 10:10:10
	7	Maverick	2023-02-20	2023-03-05	2023-03-01 10:24:27
	8	Kayana	2023-02-21	2023-03-04	2023-03-01 10:55:37
	9	Soldier	2023-02-23	2023-03-01	2023-03-01 10:11:11
	10	Minty	2023-02-23	2023-03-02	2023-03-01 10:03:55

Scenario 2

Database and Table Creation SQL:

```
CREATE DATABASE Bike_Shop;
CREATE TABLE employee
( employeeID    INT NOT NULL primary key,
  employeeName  VARCHAR(30) NOT NULL,
  position      VARCHAR(30),
  commissionRate   DECIMAL(2,2)
);

CREATE TABLE customer
( customerID   INT NOT NULL primary key,
  customerName VARCHAR(30) NOT NULL,
  phoneNum      VARCHAR(15)
);

CREATE TABLE bike
( bikeID   INT NOT NULL primary key,
  brand    VARCHAR(30) NOT NULL,
  model    VARCHAR(30) NOT NULL
  modelYear     INT
);

CREATE TABLE shoporder
( orderID       INT NOT NULL primary key,
  employeeID   INT NOT NULL,
  customerID   INT NOT NULL,
  bikeID        INT NOT NULL,
  foreign key (employeeID) references employee(employeeID),
  foreign key (customerID) references customer(customerID),
  foreign key (bikeID) references bike(bikeID),
  quantity      INT,
  purchaseTime DATETIME,
  totalPrice   DECIMAL(7,2)
);
```

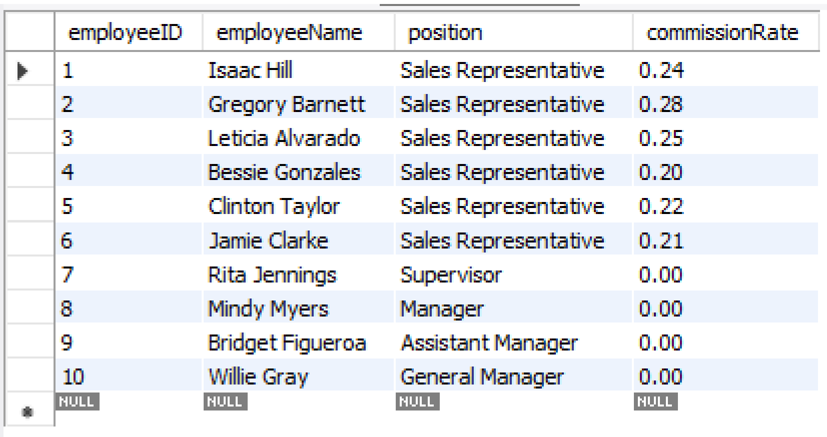
Data Entry:

```
INSERT INTO employee (employeeID, employeeName, position,
commissionRate)
VALUES (0001, 'Isaac Hill', 'Sales Representative', .24),
      (0002, 'Gregory Barnett', 'Sales Representative', .28),
      (0003, 'Leticia Alvarado', 'Sales Representative', .25),
      (0004, 'Bessie Gonzales', 'Sales Representative', .20),
      (0005, 'Clinton Taylor', 'Sales Representative', .22),
      (0006, 'Jamie Clarke', 'Sales Representative', .21),
      (0007, 'Rita Jennings', 'Supervisor', .00),
      (0008, 'Mindy Myers', 'Manager', .00),
      (0009, 'Bridget Figueroa', 'Assistant Manager', .00),
      (0010, 'Willie Gray', 'General Manager', .00);

INSERT INTO customer (customerID, customerName, phoneNum)
VALUES (0001, 'Marshall Klein', '308-489-7223'),
      (0002, 'Deanna George', '850-448-2147'),
      (0003, 'Allison Lawrence', '765-872-7506'),
      (0004, 'Lamar Powers', '575-226-3447'),
      (0005, 'Carolyn Griffith', '479-348-2750'),
      (0006, 'Jared Powell', '202-565-6939'),
      (0007, 'Wilfred Quinn', '718-685-7902'),
      (0008, 'Jenna Walton', '727-257-7777'),
      (0009, 'Terri Barnett', '704-562-4400'),
      (0010, 'Leonard Edwards', '850-338-5819')

INSERT INTO bike (bikeID, brand, model, modelYear)
VALUES (0001, 'Specialized', 'Fuse Comp 29', 2022),
      (0002, 'Specialized', 'Crossroads 1.0', 2022),
      (0003, 'Specialized', 'Jett 24', 2023),
      (0004, 'Trek', 'Checkpoint ALR 5', 2022),
      (0005, 'Trek', 'Checkpoint SL 5', 2023),
      (0006, 'Trek', '820 WSD', 2022),
      (0007, 'Trek', 'Marlin 5 Gen 2', 2023),
      (0008, 'Cannondale', 'Adventure 1', 2022),
      (0009, 'Cannondale', 'CAAD Optimo', 2021),
      (0010, 'Specialized Turbo', 'Como 3.0', 2023);
```

```
INSERT INTO shoporder (orderID, employeeID, customerID, bikeID,
quantity, purchaseTime, totalPrice)
VALUES (0001, 0003, 0004, 0007, 1, '2023-01-28 12:30:13', 632.77),
      (0002, 0005, 0008, 0006, 1, '2023-02-24 08:35:55', 512.80),
      (0003, 0001, 0007, 0001, 1, '2023-01-18 10:12:00', 2932.03),
      (0004, 0002, 0003, 0002, 1, '2023-01-01 13:45:41', 694.59),
      (0005, 0003, 0010, 0003, 1, '2023-03-03 16:39:30', 502.14),
      (0006, 0004, 0009, 00010, 2, '2023-02-11 15:55:29', 6632.41),
      (0007, 0005, 0005, 0005, 1, '2023-02-17 12:00:27', 3332.27),
      (0008, 0005, 0006, 0008, 1, '2023-03-01 14:29:39', 1002.05),
      (0009, 0003, 0002, 0004, 2, '2023-03-01 18:30:11', 2699.99),
      (0010, 0006, 0001, 0009, 2, '2023-03-02 19:00:02', 1132.55),
      (0011, 0001, 0002, 0002, 1, '2023-01-13 08:59:31', 694.59);
```

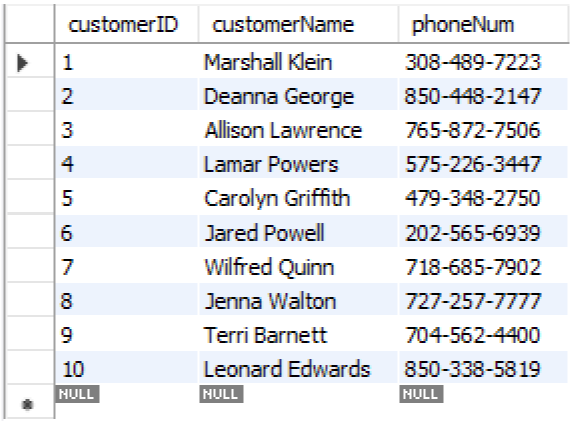
Database Pictures:

Employee Table



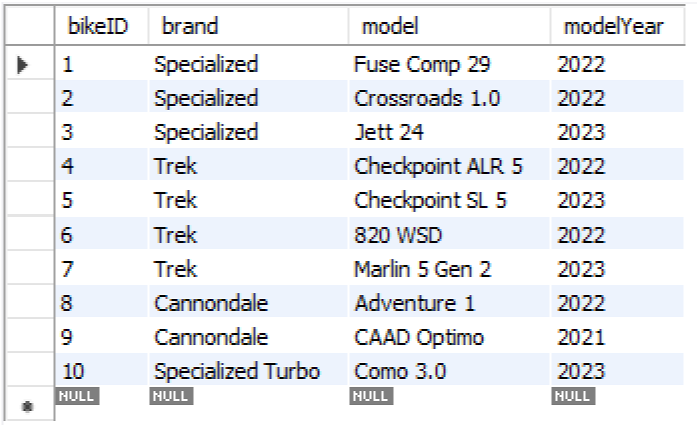
employeeID	employeeName	position	commissionRate
1	Isaac Hill	Sales Representative	0.24
2	Gregory Barnett	Sales Representative	0.28
3	Leticia Alvarado	Sales Representative	0.25
4	Bessie Gonzales	Sales Representative	0.20
5	Clinton Taylor	Sales Representative	0.22
6	Jamie Clarke	Sales Representative	0.21
7	Rita Jennings	Supervisor	0.00
8	Mindy Myers	Manager	0.00
9	Bridget Figueroa	Assistant Manager	0.00
10	Willie Gray	General Manager	0.00
NULL	NULL	NULL	NULL

Customer Table



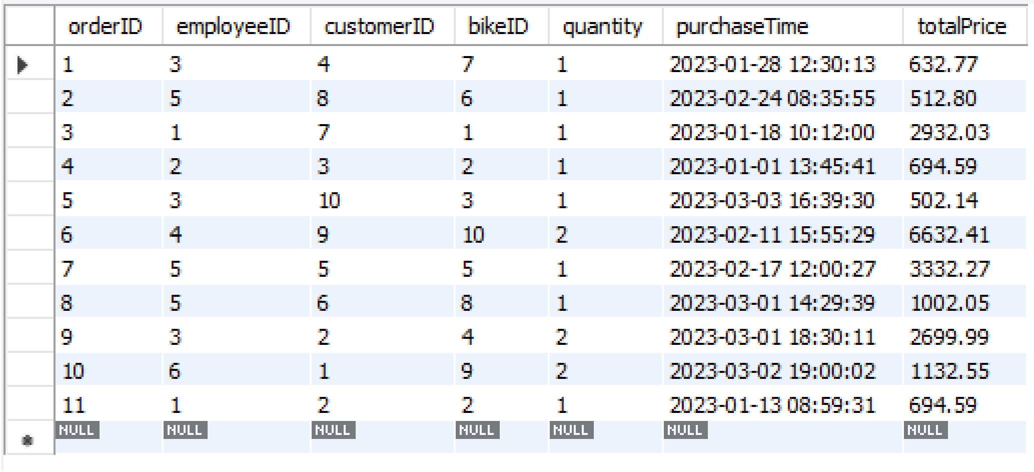
customerID	customerName	phoneNum
1	Marshall Klein	308-489-7223
2	Deanna George	850-448-2147
3	Allison Lawrence	765-872-7506
4	Lamar Powers	575-226-3447
5	Carolyn Griffith	479-348-2750
6	Jared Powell	202-565-6939
7	Wilfred Quinn	718-685-7902
8	Jenna Walton	727-257-7777
9	Terri Barnett	704-562-4400
10	Leonard Edwards	850-338-5819
NULL	NULL	NULL

Bike Table



bikeID	brand	model	modelYear
1	Specialized	Fuse Comp 29	2022
2	Specialized	Crossroads 1.0	2022
3	Specialized	Jett 24	2023
4	Trek	Checkpoint ALR 5	2022
5	Trek	Checkpoint SL 5	2023
6	Trek	820 WSD	2022
7	Trek	Marlin 5 Gen 2	2023
8	Cannondale	Adventure 1	2022
9	Cannondale	CAAD Optimo	2021
10	Specialized Turbo	Como 3.0	2023
NULL	NULL	NULL	NULL

ShopOrder Table

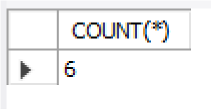


orderID	employeeID	customerID	bikeID	quantity	purchaseTime	totalPrice
1	3	4	7	1	2023-01-28 12:30:13	632.77
2	5	8	6	1	2023-02-24 08:35:55	512.80
3	1	7	1	1	2023-01-18 10:12:00	2932.03
4	2	3	2	1	2023-01-01 13:45:41	694.59
5	3	10	3	1	2023-03-03 16:39:30	502.14
6	4	9	10	2	2023-02-11 15:55:29	6632.41
7	5	5	5	1	2023-02-17 12:00:27	3332.27
8	5	6	8	1	2023-03-01 14:29:39	1002.05
9	3	2	4	2	2023-03-01 18:30:11	2699.99
10	6	1	9	2	2023-03-02 19:00:02	1132.55
11	1	2	2	1	2023-01-13 08:59:31	694.59
NULL	NULL	NULL	NULL	NULL	NULL	NULL

mySQL Queries:

Query 1:

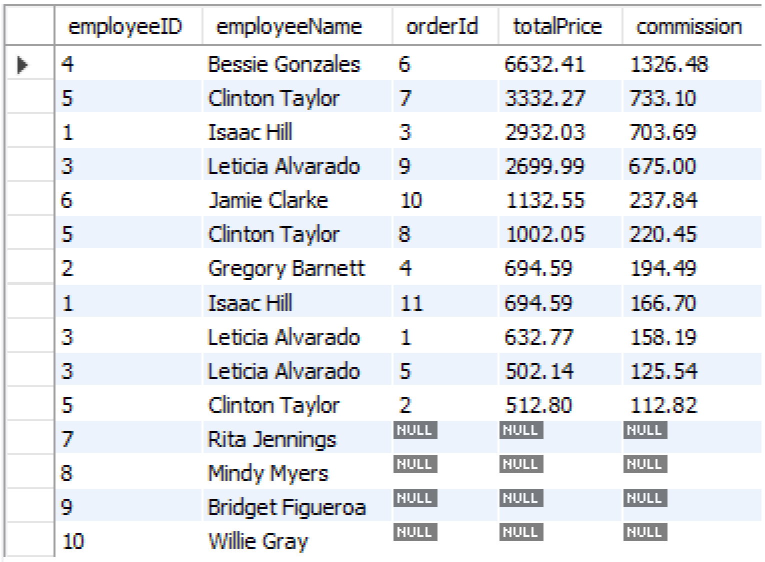
```
#Count the amount of orders that have sold product worth more than
$1000

SELECT COUNT(*)
FROM shoporder
WHERE totalPrice > 1000;
```



COUNT(*)
6

Query 2:

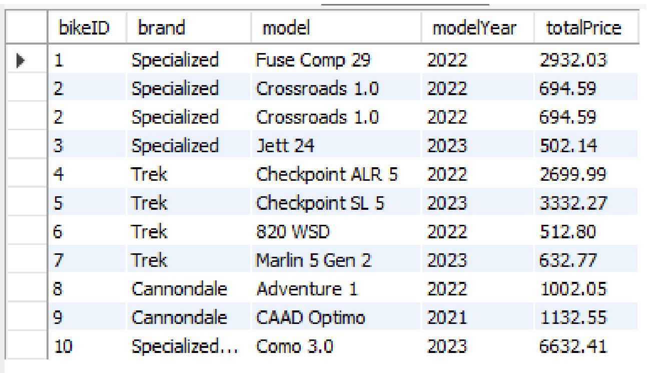
```
#Joins Employees with order to calculate how much money the employee
will make on their commission and is ordered by commission amount

SELECT employee.employeeID, employee.employeeName, shoporder.orderId,
shoporder.totalPrice,
ROUND(employee.commissionRate*shopOrder.totalPrice, 2) AS commission
FROM employee
LEFT JOIN shoporder ON employee.employeeID = shoporder.employeeID
ORDER BY commission DESC;
```



employeeID	employeeName	orderId	totalPrice	commission
4	Bessie Gonzales	6	6632.41	1326.48
5	Clinton Taylor	7	3332.27	733.10
1	Isaac Hill	3	2932.03	703.69
3	Leticia Alvarado	9	2699.99	675.00
6	Jamie Clarke	10	1132.55	237.84
5	Clinton Taylor	8	1002.05	220.45
2	Gregory Barnett	4	694.59	194.49
1	Isaac Hill	11	694.59	166.70
3	Leticia Alvarado	1	632.77	158.19
3	Leticia Alvarado	5	502.14	125.54
5	Clinton Taylor	2	512.80	112.82
7	Rita Jennings	NULL	NULL	NULL
8	Mindy Myers	NULL	NULL	NULL
9	Bridget Figueroa	NULL	NULL	NULL
10	Willie Gray	NULL	NULL	NULL

Query 3:

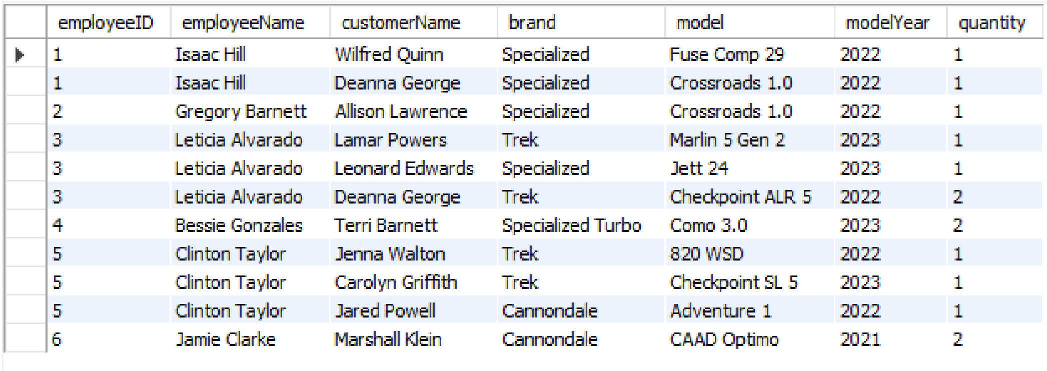
```
#Joins bike with their order to see how much the bike will be after
taxes ordered by bikeID

SELECT bike.bikeID, bike.brand, bike.model, bike.modelYear,
shoporder.totalPrice
FROM bike
LEFT JOIN shoporder ON bike.bikeID = shoporder.bikeID
ORDER BY bike.bikeID;
```



bikeID	brand	model	modelYear	totalPrice
1	Specialized	Fuse Comp 29	2022	2932.03
2	Specialized	Crossroads 1.0	2022	694.59
2	Specialized	Crossroads 1.0	2022	694.59
3	Specialized	Jett 24	2023	502.14
4	Trek	Checkpoint ALR 5	2022	2699.99
5	Trek	Checkpoint SL 5	2023	3332.27
6	Trek	820 WSD	2022	512.80
7	Trek	Marlin 5 Gen 2	2023	632.77
8	Cannondale	Adventure 1	2022	1002.05
9	Cannondale	CAAD Optimo	2021	1132.55
10	Specialized...	Como 3.0	2023	6632.41

Query 4:

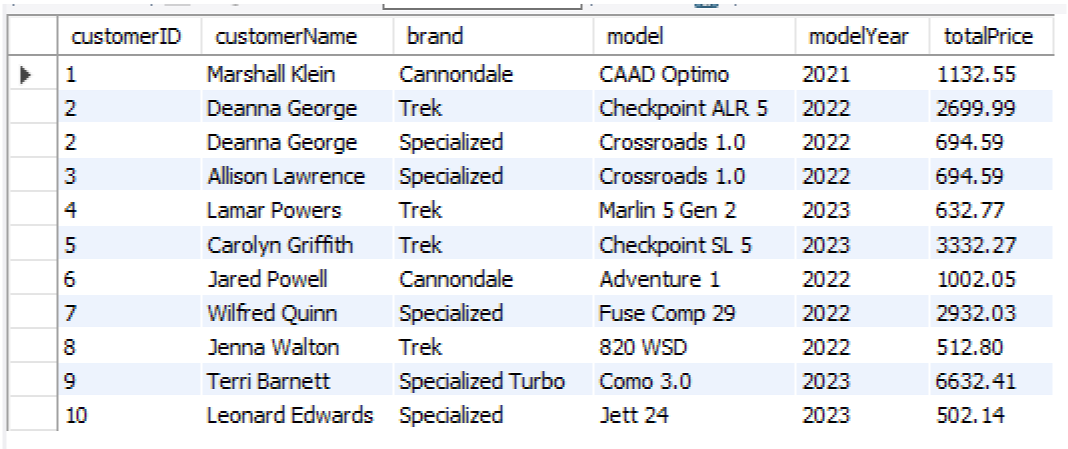
```
#Joins shoporder with employee, customer, and bike to show which
employee sold what type of bike to what customer and how many. This
is ordered by employeeID

SELECT employee.employeeID, employee.employeeName,
customer.customerName, bike.brand, bike.model, bike.modelYear,
shoporder.quantity
FROM shoporder
LEFT JOIN employee ON shoporder.employeeID = employee.employeeID
LEFT JOIN customer ON shoporder.customerID = customer.customerID
LEFT JOIN bike ON shoporder.bikeID = bike.bikeID
ORDER BY employee.employeeID;
```



employeeID	employeeName	customerName	brand	model	modelYear	quantity
1	Isaac Hill	Wilfred Quinn	Specialized	Fuse Comp 29	2022	1
1	Isaac Hill	Deanna George	Specialized	Crossroads 1.0	2022	1
2	Gregory Barnett	Allison Lawrence	Specialized	Crossroads 1.0	2022	1
3	Leticia Alvarado	Lamar Powers	Trek	Marlin 5 Gen 2	2023	1
3	Leticia Alvarado	Leonard Edwards	Specialized	Jett 24	2023	1
3	Leticia Alvarado	Deanna George	Trek	Checkpoint ALR 5	2022	2
4	Bessie Gonzales	Terri Barnett	Specialized Turbo	Como 3.0	2023	2
5	Clinton Taylor	Jenna Walton	Trek	820 WSD	2022	1
5	Clinton Taylor	Carolyn Griffith	Trek	Checkpoint SL 5	2023	1
5	Clinton Taylor	Jared Powell	Cannondale	Adventure 1	2022	1
6	Jamie Clarke	Marshall Klein	Cannondale	CAAD Optimo	2021	2

Query 5:

```
#Joins shoporder with customer, and bike to show how much a customer
spent on buying different bikes and is ordered by customerID

SELECT customer.customerID, customer.customerName, bike.brand,
bike.model, bike.modelYear, shopOrder.totalPrice
FROM shoporder
LEFT JOIN customer ON shoporder.customerID = customer.customerID
LEFT JOIN bike ON shoporder.bikeID = bike.bikeID
ORDER BY customer.customerID;
```



	customerID	customerName	brand	model	modelYear	totalPrice
▶	1	Marshall Klein	Cannondale	CAAD Optimo	2021	1132.55
	2	Deanna George	Trek	Checkpoint ALR 5	2022	2699.99
	2	Deanna George	Specialized	Crossroads 1.0	2022	694.59
	3	Allison Lawrence	Specialized	Crossroads 1.0	2022	694.59
	4	Lamar Powers	Trek	Marlin 5 Gen 2	2023	632.77
	5	Carolyn Griffith	Trek	Checkpoint SL 5	2023	3332.27
	6	Jared Powell	Cannondale	Adventure 1	2022	1002.05
	7	Wilfred Quinn	Specialized	Fuse Comp 29	2022	2932.03
	8	Jenna Walton	Trek	820 WSD	2022	512.80
	9	Terri Barnett	Specialized Turbo	Como 3.0	2023	6632.41
	10	Leonard Edwards	Specialized	Jett 24	2023	502.14

Scenario 3

Database and Table Creation SQL:

```
CREATE DATABASE towing_company;
CREATE TABLE employee
( employeeID    INT NOT NULL primary key,
  employeeName  VARCHAR(30) NOT NULL,
  position      VARCHAR(30),
  commissionRate   DECIMAL(2,2)
);

CREATE TABLE car
( carID         INT NOT NULL primary key,
  make          VARCHAR(30) NOT NULL,
  model         VARCHAR(30) NOT NULL,
  modelYear     INT,
  color         VARCHAR(30),
  licensePlate  VARCHAR(7) NOT NULL
);

CREATE TABLE towlog
( logID         INT NOT NULL primary key,
  employeeID   INT NOT NULL,
  carID         INT NOT NULL,
  foreign key (employeeID) references employee(employeeID),
  foreign key (carID) references car(carID),
  timeTowed     DATETIME,
  location      VARCHAR(30),
  timeKept      VARCHAR(30),
  releasePrice DECIMAL(6,2),
  towPrice      DECIMAL(5,2)
);
```

Data Entry:

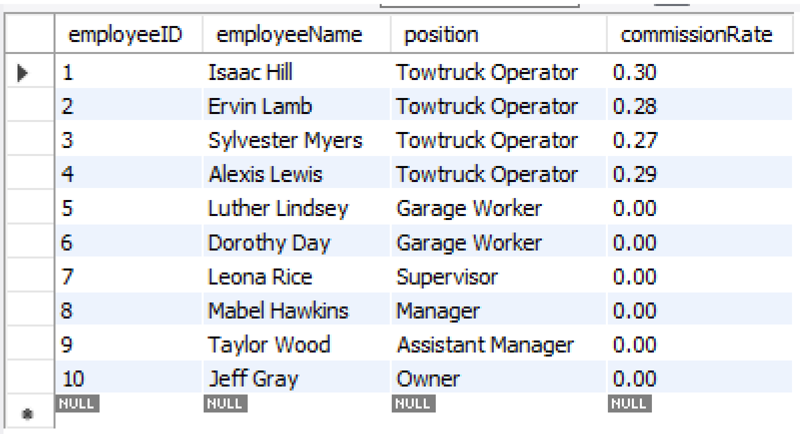
```
INSERT INTO employee (employeeID, employeeName, position,
commissionRate)
VALUES (0001, 'Isaac Hill', 'Towtruck Operator', .30),
      (0002, 'Ervin Lamb', 'Towtruck Operator', .28),
      (0003, 'Sylvester Myers', 'Towtruck Operator', .27),
      (0004, 'Alexis Lewis', 'Towtruck Operator', .29),
      (0005, 'Luther Lindsey', 'Garage Worker', .00),
      (0006, 'Dorothy Day', 'Garage Worker', .00),
      (0007, 'Leona Rice', 'Supervisor', .00),
      (0008, 'Mabel Hawkins', 'Manager', .00),
      (0009, 'Taylor Wood', 'Assistant Manager', .00),
      (0010, 'Jeff Gray', 'Owner', .00);
```

```
INSERT INTO car (carID, make, model, modelYear, color, licensePlate)
VALUES (0001, 'Ford', 'F150', 2020, 'White', 'V26 5KG'),
      (0002, 'Dodge', 'Ram 2500', 2009, 'Black', 'W26 1BP'),
      (0003, 'Toyota', 'Prius', 2012, 'Red', 'U73 4RT'),
      (0004, 'Mercedes-Benz', 'GL500', 2014, 'Gray', 'A20 6AL'),
      (0005, 'Subaru', 'XV', 2017, 'White', 'T30 1YX'),
      (0006, 'Toyota', 'Matrix', 2012, 'Gray', 'T79 1HJ'),
      (0007, 'Volvo', 'V40', 2004, 'White', 'Z82 9WJ'),
      (0008, 'BMW', 'X5', 2023, 'Black', 'Y94 7SX'),
      (0009, 'Maserati', 'Coupe', 2020, 'Blue', '2FAST4U'),
      (0010, 'Ford', 'F250', 2022, 'Gray', 'T57 6MN');
```

```
INSERT INTO towlog (logID, employeeID, carID, timeTowed, location,
timeKept, releasePrice, towPrice)
VALUES (0001, 0004, 0007, '2023-01-03 03:25:05', 'Orem, Utah', '6
Days', 556.75, 100.49),
      (0002, 0001, 0009, '2023-01-08 08:55:46', 'Draper, Utah', '2
Days', 345.08, 126.28),
      (0003, 0003, 0002, '2023-01-15 18:02:38', 'Salt Lake City,
Utah', '1 Day', 209.01, 150.23),
      (0004, 0003, 0003, '2023-01-25 23:39:57', 'Provo, Utah', '2
Days', 298.97, 105.89),
      (0005, 0001, 0008, '2023-02-05 08:00:00', 'Sandy, Utah', '3
Days', 314.38, 135.14),
      (0006, 0002, 0010, '2023-02-12 07:59:22', 'Lehi, Utah', '4
Days', 457.24, 115.99),
      (0007, 0004, 0006, '2023-02-18 12:45:31', 'Saratoga Springs,
Utah', '1 Day', 212.88, 121.18),
      (0008, 0002, 0005, '2023-02-26 13:12:45', 'Orem, Utah', '6
Days', 599.49, 102.42),
      (0009, 0001, 0009, '2023-02-28 15:26:02', 'Draper, Utah', '1
Day', 300.98, 129.73),
      (0010, 0004, 0001, '2023-03-01 02:37:51', 'West Jordan, Utah',
'2 Days', 281.73, 142.03),
      (0011, 0002, 0004, '2023-03-03 05:48:29', 'Bluffdale, Utah', '1
Days', 223.21, 120.48);
```

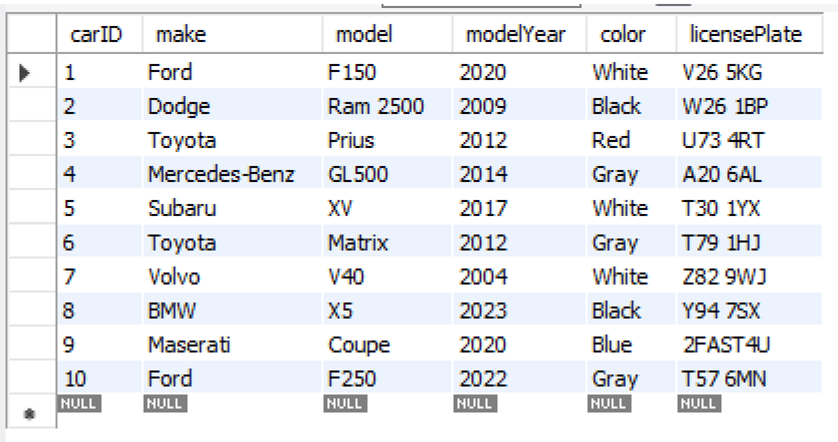
Database Pictures:

Employee Table



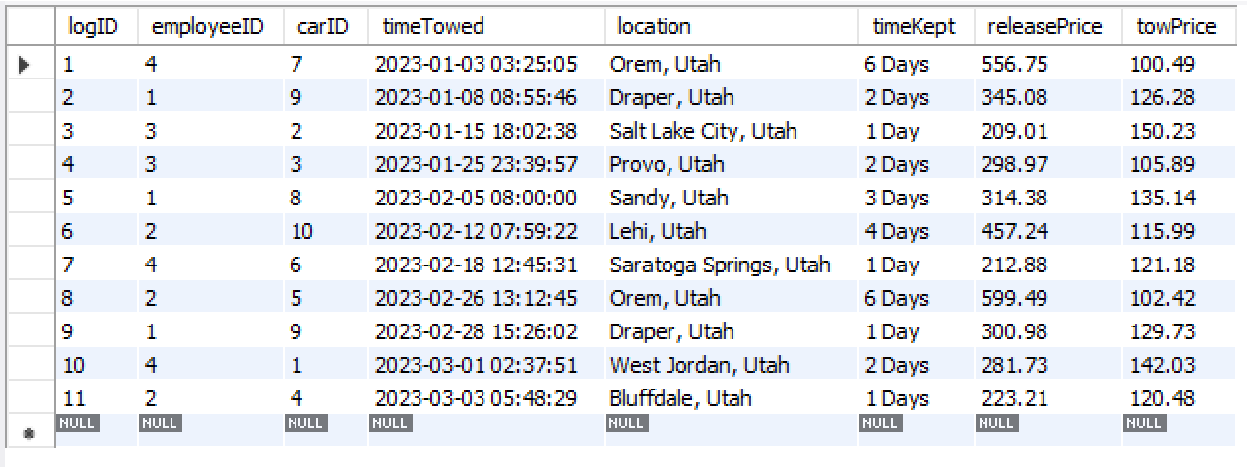
employeeID	employeeName	position	commissionRate
1	Isaac Hill	Towtruck Operator	0.30
2	Ervin Lamb	Towtruck Operator	0.28
3	Sylvester Myers	Towtruck Operator	0.27
4	Alexis Lewis	Towtruck Operator	0.29
5	Luther Lindsey	Garage Worker	0.00
6	Dorothy Day	Garage Worker	0.00
7	Leona Rice	Supervisor	0.00
8	Mabel Hawkins	Manager	0.00
9	Taylor Wood	Assistant Manager	0.00
10	Jeff Gray	Owner	0.00
NULL	NULL	NULL	NULL

Car Table



carID	make	model	modelYear	color	licensePlate
1	Ford	F150	2020	White	V26 5KG
2	Dodge	Ram 2500	2009	Black	W26 1BP
3	Toyota	Prius	2012	Red	U73 4RT
4	Mercedes-Benz	GL500	2014	Gray	A20 6AL
5	Subaru	XV	2017	White	T30 1YX
6	Toyota	Matrix	2012	Gray	T79 1HJ
7	Volvo	V40	2004	White	Z82 9WJ
8	BMW	X5	2023	Black	Y94 7SX
9	Maserati	Coupe	2020	Blue	2FAST4U
10	Ford	F250	2022	Gray	T57 6MN
NULL	NULL	NULL	NULL	NULL	NULL

TowLog Table

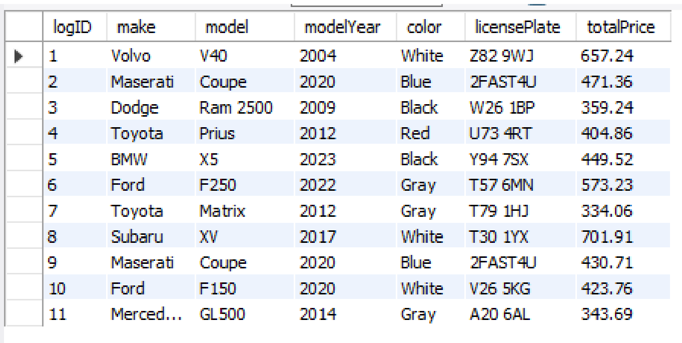


logID	employeeID	carID	timeTowed	location	timeKept	releasePrice	towPrice
1	4	7	2023-01-03 03:25:05	Orem, Utah	6 Days	556.75	100.49
2	1	9	2023-01-08 08:55:46	Draper, Utah	2 Days	345.08	126.28
3	3	2	2023-01-15 18:02:38	Salt Lake City, Utah	1 Day	209.01	150.23
4	3	3	2023-01-25 23:39:57	Provo, Utah	2 Days	298.97	105.89
5	1	8	2023-02-05 08:00:00	Sandy, Utah	3 Days	314.38	135.14
6	2	10	2023-02-12 07:59:22	Lehi, Utah	4 Days	457.24	115.99
7	4	6	2023-02-18 12:45:31	Saratoga Springs, Utah	1 Day	212.88	121.18
8	2	5	2023-02-26 13:12:45	Orem, Utah	6 Days	599.49	102.42
9	1	9	2023-02-28 15:26:02	Draper, Utah	1 Day	300.98	129.73
10	4	1	2023-03-01 02:37:51	West Jordan, Utah	2 Days	281.73	142.03
11	2	4	2023-03-03 05:48:29	Bluffdale, Utah	1 Days	223.21	120.48
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

mySQL Queries:

Query 1:

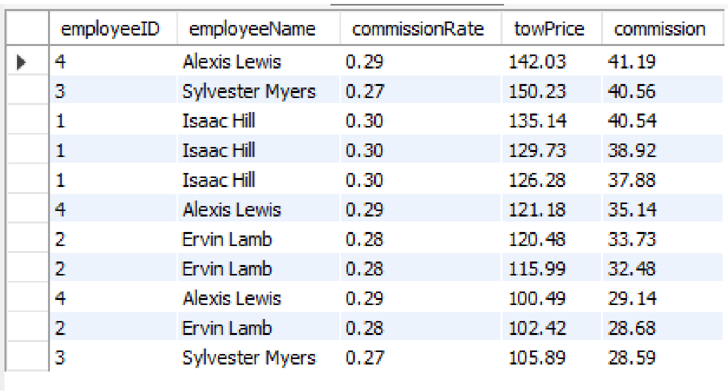
```
#Pairs towlog and car to show how much it took to get the car out of
the lot (releasePrice + towPrice) and this is ordered by logID

SELECT towlog.logID, car.make, car.model, car.modelYear, car.color,
car.licensePlate, (towlog.releasePrice + towlog.towPrice) AS
totalPrice
FROM towlog
INNER JOIN car ON towlog.carID = car.carID
ORDER BY towlog.logID;
```



logID	make	model	modelYear	color	licensePlate	totalPrice
1	Volvo	V40	2004	White	Z82 9WJ	657.24
2	Maserati	Coupe	2020	Blue	2FAST4U	471.36
3	Dodge	Ram 2500	2009	Black	W26 1BP	359.24
4	Toyota	Prius	2012	Red	U73 4RT	404.86
5	BMW	X5	2023	Black	Y94 7SX	449.52
6	Ford	F250	2022	Gray	T57 6MN	573.23
7	Toyota	Matrix	2012	Gray	T79 1HJ	334.06
8	Subaru	XV	2017	White	T30 1YX	701.91
9	Maserati	Coupe	2020	Blue	2FAST4U	430.71
10	Ford	F150	2020	White	V26 5KG	423.76
11	Merced...	GL500	2014	Gray	A20 6AL	343.69

Query 2:

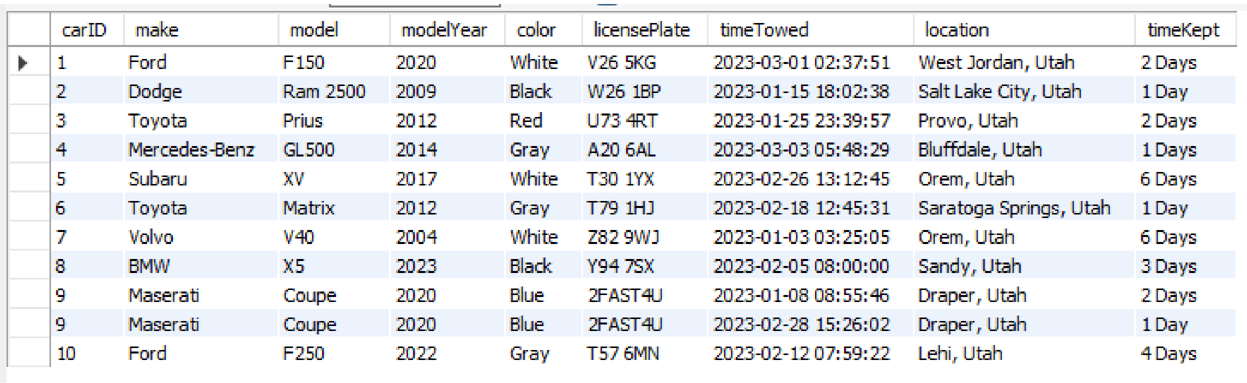
```
#Pairs employee and towlog to show how much the towtruck operator
will make for their commission on the specific tow they did. This is
ordered by the commission

SELECT employee.employeeID, employee.employeeName,
employee.commissionRate, towlog.towPrice, ROUND(towlog.towPrice *
employee.commissionRate, 2) AS commission
FROM employee
INNER JOIN towlog ON employee.employeeID = towlog.employeeID
ORDER BY commission DESC;
```



employeeID	employeeName	commissionRate	towPrice	commission
4	Alexis Lewis	0.29	142.03	41.19
3	Sylvester Myers	0.27	150.23	40.56
1	Isaac Hill	0.30	135.14	40.54
1	Isaac Hill	0.30	129.73	38.92
1	Isaac Hill	0.30	126.28	37.88
4	Alexis Lewis	0.29	121.18	35.14
2	Ervin Lamb	0.28	120.48	33.73
2	Ervin Lamb	0.28	115.99	32.48
4	Alexis Lewis	0.29	100.49	29.14
2	Ervin Lamb	0.28	102.42	28.68
3	Sylvester Myers	0.27	105.89	28.59

Query 3:

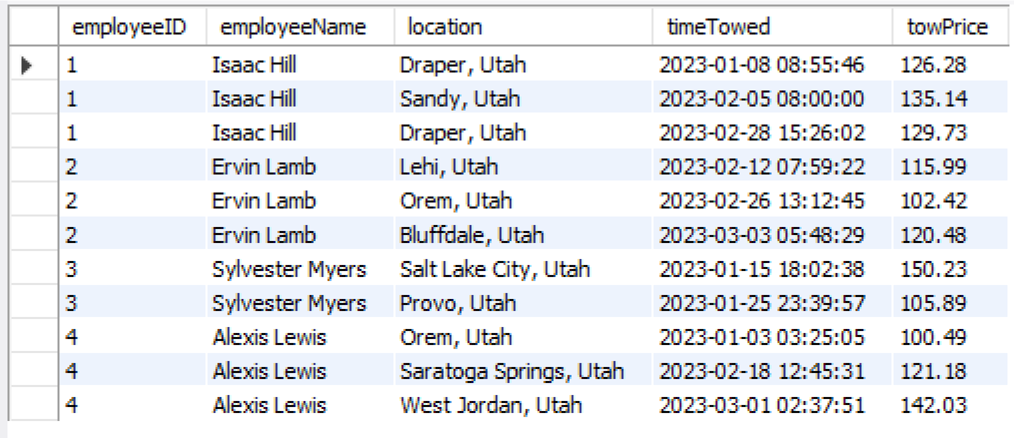
```
#Joins car and towlog to show the information of the car and the
time, location and how long it was kept for. This is ordered by carID

SELECT car.carID, car.make, car.model, car.modelYear, car.color,
car.licensePlate, towlog.timeTowed, towlog.location, towlog.timeKept
FROM car
INNER JOIN towlog ON towlog.carID = car.carID
ORDER BY car.carID;
```



carID	make	model	modelYear	color	licensePlate	timeTowed	location	timeKept
1	Ford	F150	2020	White	V26 5KG	2023-03-01 02:37:51	West Jordan, Utah	2 Days
2	Dodge	Ram 2500	2009	Black	W26 1BP	2023-01-15 18:02:38	Salt Lake City, Utah	1 Day
3	Toyota	Prius	2012	Red	U73 4RT	2023-01-25 23:39:57	Provo, Utah	2 Days
4	Mercedes-Benz	GL500	2014	Gray	A20 6AL	2023-03-03 05:48:29	Bluffdale, Utah	1 Days
5	Subaru	XV	2017	White	T30 1YX	2023-02-26 13:12:45	Orem, Utah	6 Days
6	Toyota	Matrix	2012	Gray	T79 1HJ	2023-02-18 12:45:31	Saratoga Springs, Utah	1 Day
7	Volvo	V40	2004	White	Z82 9WJ	2023-01-03 03:25:05	Orem, Utah	6 Days
8	BMW	X5	2023	Black	Y94 7SX	2023-02-05 08:00:00	Sandy, Utah	3 Days
9	Maserati	Coupe	2020	Blue	2FAST4U	2023-01-08 08:55:46	Draper, Utah	2 Days
9	Maserati	Coupe	2020	Blue	2FAST4U	2023-02-28 15:26:02	Draper, Utah	1 Day
10	Ford	F250	2022	Gray	T57 6MN	2023-02-12 07:59:22	Lehi, Utah	4 Days

Query 4:

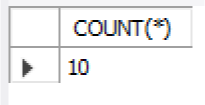
```
#Joins employee and towlog to show where they have towed from, the
time they towed, and the price from the location. This is ordered by
employeeID

SELECT employee.employeeID, employee.employeeName, towlog.location,
towlog.timeTowed, towlog.towPrice
FROM employee
INNER JOIN towlog ON employee.employeeID = towlog.employeeID
ORDER BY employeeID;
```



employeeID	employeeName	location	timeTowed	towPrice
1	Isaac Hill	Draper, Utah	2023-01-08 08:55:46	126.28
1	Isaac Hill	Sandy, Utah	2023-02-05 08:00:00	135.14
1	Isaac Hill	Draper, Utah	2023-02-28 15:26:02	129.73
2	Ervin Lamb	Lehi, Utah	2023-02-12 07:59:22	115.99
2	Ervin Lamb	Orem, Utah	2023-02-26 13:12:45	102.42
2	Ervin Lamb	Bluffdale, Utah	2023-03-03 05:48:29	120.48
3	Sylvester Myers	Salt Lake City, Utah	2023-01-15 18:02:38	150.23
3	Sylvester Myers	Provo, Utah	2023-01-25 23:39:57	105.89
4	Alexis Lewis	Orem, Utah	2023-01-03 03:25:05	100.49
4	Alexis Lewis	Saratoga Springs, Utah	2023-02-18 12:45:31	121.18
4	Alexis Lewis	West Jordan, Utah	2023-03-01 02:37:51	142.03

Query 5:

```
#Counts how many cars have passed through their lot

SELECT COUNT(*)
FROM car
```



COUNT(*)
10

Report

I think this project has really developed my understanding of databases. I've learned a lot about the process of coming up with scenarios and then developing the databases from said scenarios. I think my scenarios overall are pretty good although I had some difficulty coming up with some ideas for some cause I'm not the most creative person. I would suggest maybe having a concrete number of scenarios for students to come up with as I wasn't sure how many we were supposed to do.

When it comes to ERDs, I still struggle with them. I had to edit my ERDs a good amount of times trying to make sure my relationships were correct and everything looked correct. I think my ERDs are pretty solid overall. There are a few relationships that I was a little "iffy" about such as the relationship between ShopOrder and Bike in Scenario 2 and the relationship between TowLog and Car in Scenario 3. I found the Lucid Chart was a very effective way to create ERDs and was very easy to use.

Creating the databases was pretty easy to do. I felt that having a good ERD for a scenario helped in having a roadmap in creating the tables and the values that go along with them. I think the biggest struggle that I had in the entire project came in filling out the tables. The names for employees in each of the scenarios were pretty easy with a random name generator as well as phone number generators for phone numbers. I wanted to use actual bikes and cars for scenario 2 and 3 and I'm not the biggest bike or car guy so I had to do some research finding bikes and cars to use in the tables. Scenario 1 was pretty easy to fill because I work at a dog daycare so I took a lot of inspiration from my job.

Creating the queries for each of the scenarios also took a good amount of time to do. I tried to think of what the needs of the business would be when making the queries. I learned also while creating queries in scenario 2 that you could use multiple joins in one query. I feel, as I went through each scenario, that the queries got better as I went through each scenario. Overall, I think that my queries I have would be useful for a business to use if they were to use this database.

I think there are definitely improvements that could be done for each scenario. For scenario 1, I would want to create an Owner table that would hold the owner information rather than having it on the Dog Table. For scenario 2, I would want to have the ShopOrder Table be able to have multiple types of bikes rather than just one and force the customer to create a new ShopOrder entry to buy another type of bike. For scenario 3, I would want to have a Lot Table that would show what cars are currently in the towing lot rather than not having it and just showing what cars have come through in the past. I've learned a lot about databases in this project and the process behind them. It has really improved my knowledge about how to implement them. I think that having this project be centered around the student coming up with scenarios is a great idea because I think that it makes them a lot more motivated due to them being their own ideas.