Multiple Tables with REBU

Let's practice what we learned about joins by combining rows from different tables. Suppose you are a data analyst at REBU, a ridesharing platform. For a project, you were given three tables:

- `trips` - trips information
- `riders` - users data
- `cars` - autonomous cars

Have fun!

1. Let's examine the three tables.

```
SELECT * FROM trips;

SELECT * FROM riders;

SELECT * FROM cars;
```

2. Try out a simple cross join between riders and cars. Is the result useful?

```
SELECT riders.first,
   riders.last,
   cars.model
FROM riders, cars;
```

3. Suppose we want to create a Trip Log with the trips and its users.

```
SELECT *
FROM trips
LEFT JOIN riders
ON trips.rider_id = riders.id;
```

```
SELECT trips.date,
   trips.pickup,
   trips.dropoff,
   trips.type,
   trips.cost,
   riders.first,
   riders.last,
   riders.username
FROM trips
LEFT JOIN riders
ON trips.rider_id = riders.id;
```

4. Suppose we want to create a link between the trips and the cars used during those trips. Find the columns to join on and combine the trips and cars table using an INNER JOIN.

```
SELECT *
FROM trips
JOIN cars
ON trips.car_id = cars.id;
```

5. The new riders data are in! There are three new users this month. Stack the riders table on top of the new table named riders2.

```
SELECT *
FROM riders
UNION
SELECT *
FROM riders2;
```

6. What is the average cost for a trip?

```
SELECT AVG(cost)
FROM trips;
```

7. REBU is looking to do an email campaign for all the irregular users. Find all the riders who have used REBU less than 500 times!

```
SELECT *
FROM riders
WHERE total_trips < 500
UNION
SELECT *
FROM riders2
WHERE total_trips < 500;
```

8. Calculate the number of cars that are active.

```
SELECT *
FROM cars
WHERE status = 'active';
```

9. It's safety recall time for cars that have been on the road for a while. Write a query that finds the two cars that have the highest trips_completed.

```
SELECT *
FROM cars
ORDER BY trips_completed DESC
LIMIT 2;
```