Assignment 1 — Bike Shop SQL Practice

In this lab, you'll create a small bike rental database in **phpMyAdmin**, seed it with data, and practice writing SQL statements to explore the data. You'll then display your results in a browser using PHP to run and print your queries.

Project Description

In this example, you're building a small database for a local **bike rental shop**. The shop needs to track:

- **Bikes** — each bike's model, type, and hourly rate.
- **Customers** — first and last name of each customer.
- **Rentals** — which bike is rented by which customer, along with start and end times.

Database Tables

Create your database and tables in **phpMyAdmin**. Use these field names and types. Be sure to mark all required fields as `NOT NULL` and use `AUTO_INCREMENT` for primary keys.

Table	Field	Type	Notes
bikes	bike_id	INT	Primary Key, AUTO_INCREMENT
	model	VARCHAR(100)	Required
	type	VARCHAR(50)	Road, Mountain, Hybrid, etc.
	hourly_rate	DECIMAL(5,2)	Rental cost per hour
customers	customer_id	INT	Primary Key, AUTO_INCREMENT
	first_name	VARCHAR(50)	Required
	last_name	VARCHAR(50)	Required
rentals	rental_id	INT	Primary Key, AUTO_INCREMENT
	bike_id	INT	FK → bikes.bike_id
	customer_id	INT	FK → customers.customer_id
	start_time	TIME	Rental start time

Table	Field	Type	Notes
	end_time	TIME	Nullable (if not returned yet)

SQL Setup & Seed

```
INSERT INTO bikes (model, type, hourly_rate) VALUES
('Trek Marlin 6', 'Mountain', 12.00),
('Giant Escape 3', 'Road', 10.00),
('Co-op DRT 1.1', 'Hybrid', 13.00),
('Specialized Sirrus X', 'Road', 15.00),
('Cannondale Quick 4', 'Hybrid', 11.00);

INSERT INTO customers (first_name, last_name) VALUES
('Alex', 'Smith'),
('Taylor', 'Smith'),
('Casey', 'Nguyen'),
('Jordan', 'Lee'),
('Riley', 'Patel');

INSERT INTO rentals (bike_id, customer_id, start_time, end_time) VALUES
(1, 1, '10:15:00', '11:45:00'),
(2, 3, '13:00:00', '14:10:00'),
(3, 2, '15:05:00', '15:50:00'),
(4, 5, '16:00:00', NULL),
(5, 4, '09:20:00', NULL);
```

Your Task — Practice SQL Queries

Write and test these SQL statements, from easiest to more advanced:

1. Show all customers, sorted by **last_name**, then **first_name**.
2. Show all bikes that are **available**.
3. Show which **bike** is rented by which **customer** (JOIN).
4. Show all rentals with **start_time** before noon.
5. Show the **top 3 most expensive bikes** (hint: `ORDER BY` and `LIMIT`).
6. Show all **open rentals** (where `end_time` IS NULL).

7. Show a multiple JOIN query (bikes → rentals → customers).
8. Update the **end_time** for the first rental using `NOW()`.
9. Update the **available** status of a bike to 0 (rented) by bike_id.

Functions to Create in `functions.php` (with hints)

Each function should **return an SQL string**. You'll call these from your PHP page to run and display results.

- `sqlAllCustomers()` — select all customers; *ORDER BY* `last_name`, `first_name`.
- `sqlAvailableBikes()` — select bikes where `available = 1`; include `bike_id`, `model`, `type`, `hourly_rate`.
- `sqlBikeRentals()` — JOIN `rentals` → `bikes` and `customers`; show *bike model*, *customer name*, *start/end* times.
- `sqlMorningRentals()` — rentals with `start_time < '12:00:00'`; consider ordering by `start_time`.
- `sqlTop3Bikes()` — bikes sorted by `hourly_rate` (DESC) with *LIMIT 3*.
- `sqlOpenRentals()` — rentals where `end_time IS NULL`; include who and which bike.
- `sqlAllJoins()` — multi-join of `rentals`, `customers`, `bikes`; include model, type, rate, names, start/end; optionally *ORDER BY* `start_time`.
- `sqlUpdateEndTime($rentalId)` — UPDATE a single rental; set `end_time` to `NOW()` for the given `$rentalId`.
- `sqlUpdateBikeStatus($bikeId)` — UPDATE a single bike; set `available = 0` (mark rented) for the given `$bikeId`.

Tip: keep function names exactly as listed so you can reuse them in Part 2.

Deliverables

- Use the Export tab in phpMyAdmin to create an **SQL file** that includes *all tables and data*. Save it in your project under `data/bike_shop.sql`.
- Create **bike_shop.php** in your project to display your SQL query results.
- Push your project to a new GitHub repository (do not reuse a class repo).
- Submit your **GitHub repo link** in Canvas.

✅ Expected Results — Use These to Verify Your Output

Compare your results with the tables below to confirm that your SQL queries return the correct data. Your column headings and formatting may differ, but the data should match.

1) All Customers (ORDER BY last, first)

customer_id	first_name	last_name
4	Jordan	Lee
3	Casey	Nguyen
5	Riley	Patel
1	Alex	Smith
2	Taylor	Smith

2) Available Bikes

bike_id	model	type	hourly_rate
1	Trek Marlin 6	Mountain	12.00
3	Co-op DRT 1.1	Hybrid	13.00
5	Cannondale Quick 4	Hybrid	11.00

3) Bikes Rented by Each Customer (JOIN)

bike_id	model	customer	start_time	end_time
1	Trek Marlin 6	Alex Smith	10:15	11:45
2	Giant Escape 3	Casey Nguyen	13:00	14:10
3	Co-op DRT 1.1	Taylor Smith	15:05	15:50

4) Top 3 Most Expensive Bikes

model	hourly_rate
Specialized Sirrus X	15.00
Co-op DRT 1.1	13.00

model	hourly_rate
Trek Marlin 6	12.00

5) Open Rentals (end_time IS NULL)

rental_id	bike	customer	start_time	end_time
4	Specialized Sirrus X	Riley Patel	16:00	NULL
5	Cannondale Quick 4	Jordan Lee	09:20	NULL

Back to Top

6) Rentals + Customers + Bikes (Multiple JOIN)

This query joins `rentals` → `customers` → `bikes` to show all active and completed rentals with full details.

rental_id	customer	bike_model	bike_type	hourly_rate	start_time	end_time
1	Alex Smith	Trek Marlin 6	Mountain	12.00	10:15	11:45
2	Casey Nguyen	Giant Escape 3	Road	10.00	13:00	14:10
3	Taylor Smith	Co-op DRT 1.1	Hybrid	13.00	15:05	15:50
4	Riley Patel	Specialized Sirrus X	Road	15.00	16:00	NULL
5	Jordan Lee	Cannondale Quick 4	Hybrid	11.00	09:20	NULL

Back to Top