

# COS 30020 – Lab 07

## Task 1: Creating a table and entering data

-- Using your existing database 's103488117\_db', create a new table cars for a used car dealership.

-- Include the following fields in the cars table:

-- car\_id (AUTO\_INCREMENT PRIMARY KEY),

-- make,

-- model,

-- price, and

-- yom (year of manufacture).

```
CREATE TABLE `s103488117_db`.`cars` (  
  `car_id` INT(10) NOT NULL AUTO_INCREMENT,  
  `make` VARCHAR(255) NOT NULL,  
  `model` VARCHAR(255) NOT NULL,  
  `price` DECIMAL(10, 2) NOT NULL,  
  `yom` INT(10) NOT NULL,  
  PRIMARY KEY (`car_id`)  
) ENGINE = InnoDB;
```

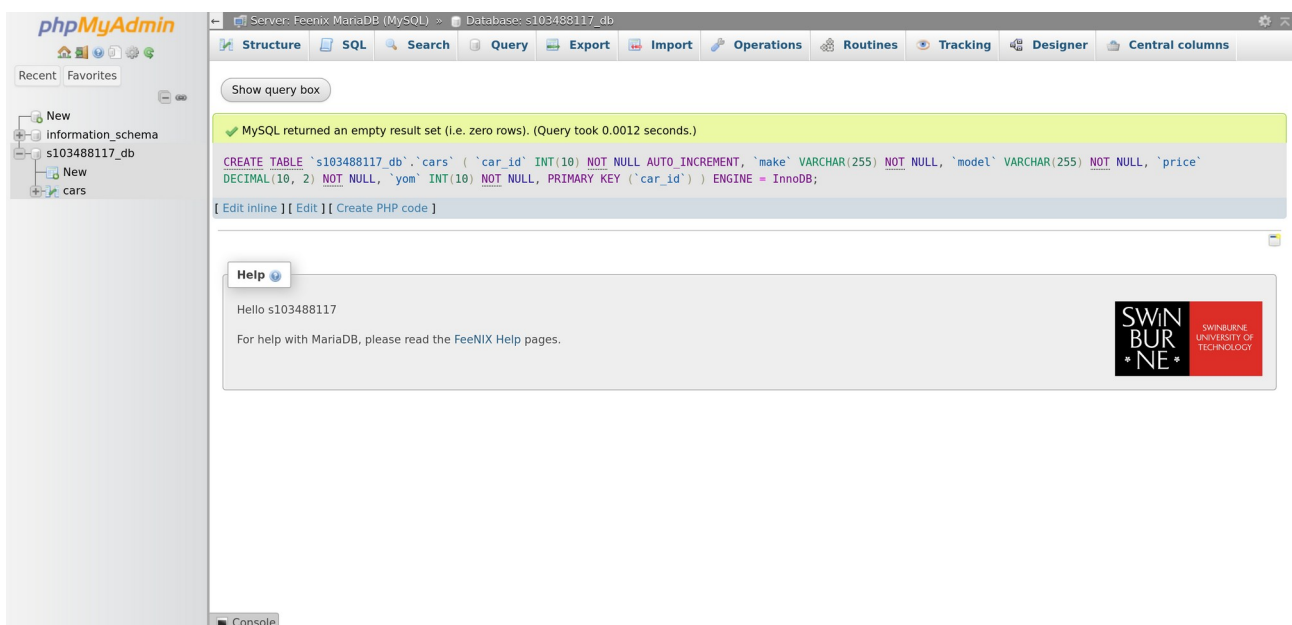


Figure 1: Create table

-- Enter at least 10 records into the table.

```
INSERT INTO cars (make, model, price, yom)
VALUES ('Holden', 'Astra', 14000.00, 2005),
       ('BMW', 'X3', 35000.00, 2004),
       ('Ford', 'Falcon', 39000.00, 2011),
       ('Toyota', 'Corolla', 20000.00, 2012),
       ('Holden', 'Commodore', 13500.00, 2005),
       ('Holden', 'Astra', 8000.00, 2001),
       ('Holden', 'Commodore', 28000.00, 2009),
       ('Ford', 'Falcon', 14000.00, 2007),
       ('Ford', 'Falcon', 7000.00, 2003),
       ('Ford', 'Laser', 10000.00, 2010),
       ('Mazda', 'RX-7', 26000.00, 2000),
       ('Toyota', 'Corolla', 12000.00, 2001),
       ('Mazda', '3', 14500.00, 2009);
```

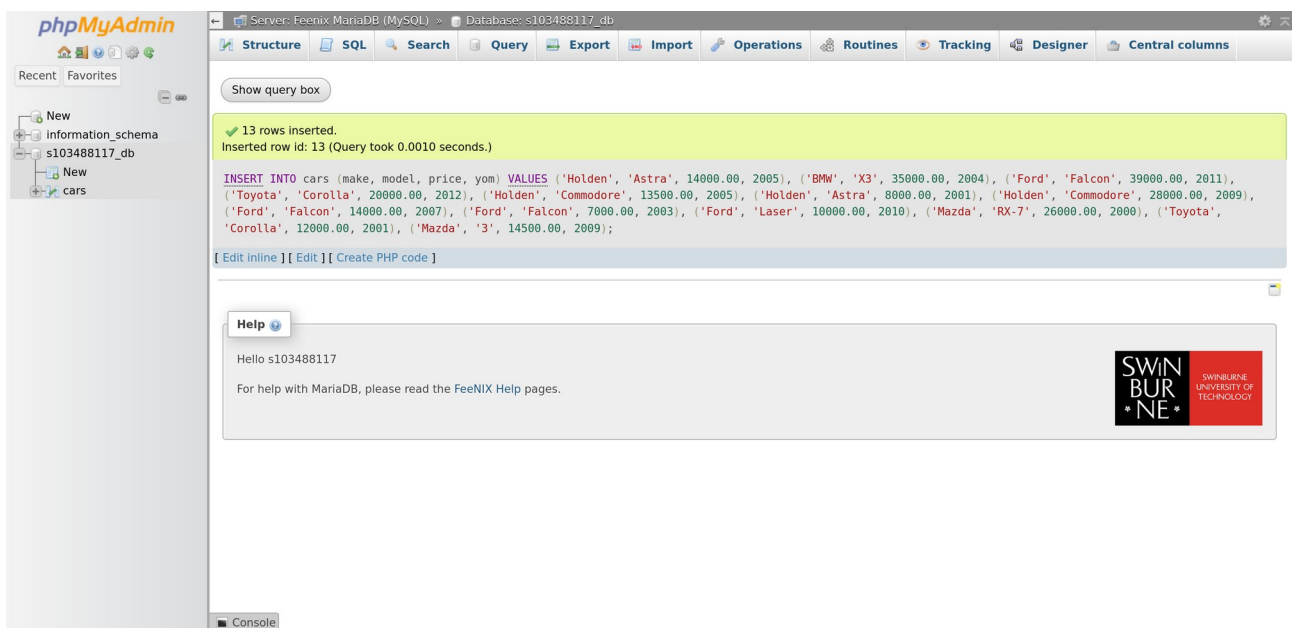


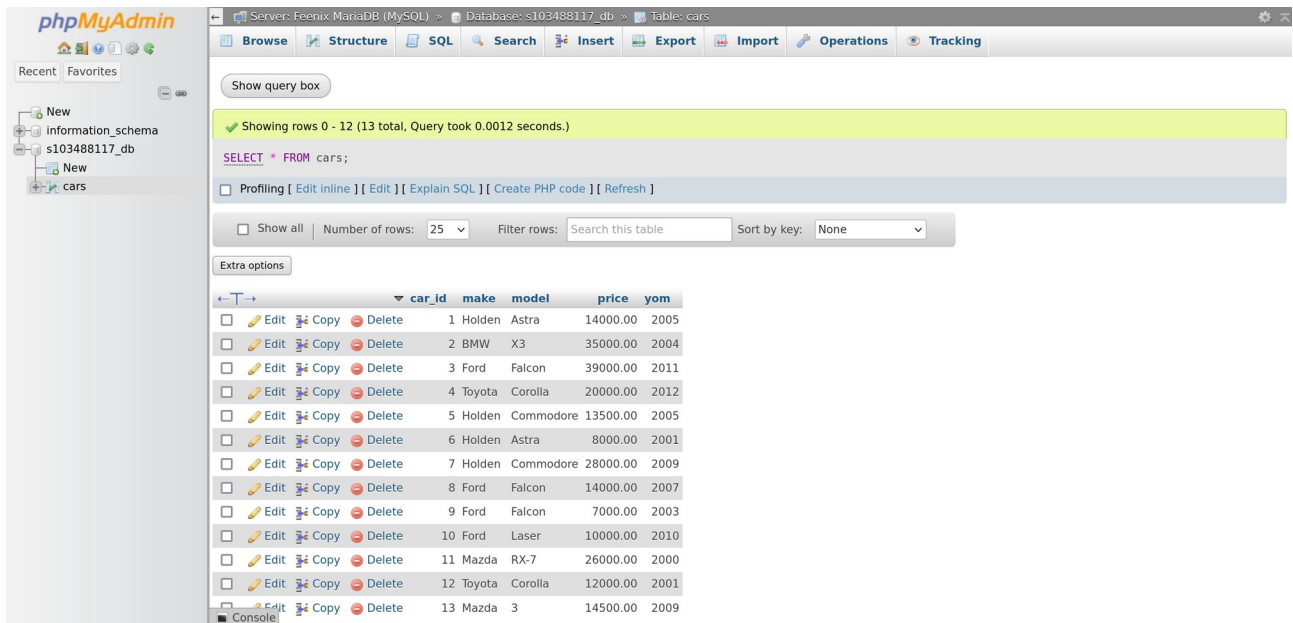
Figure 2: Insert records

## Task 2: Querying the table

-- Write queries that return the following:

-- 1. All records

**SELECT \* FROM cars;**



Showing rows 0 - 12 (13 total, Query took 0.0012 seconds.)

```
SELECT * FROM cars;
```

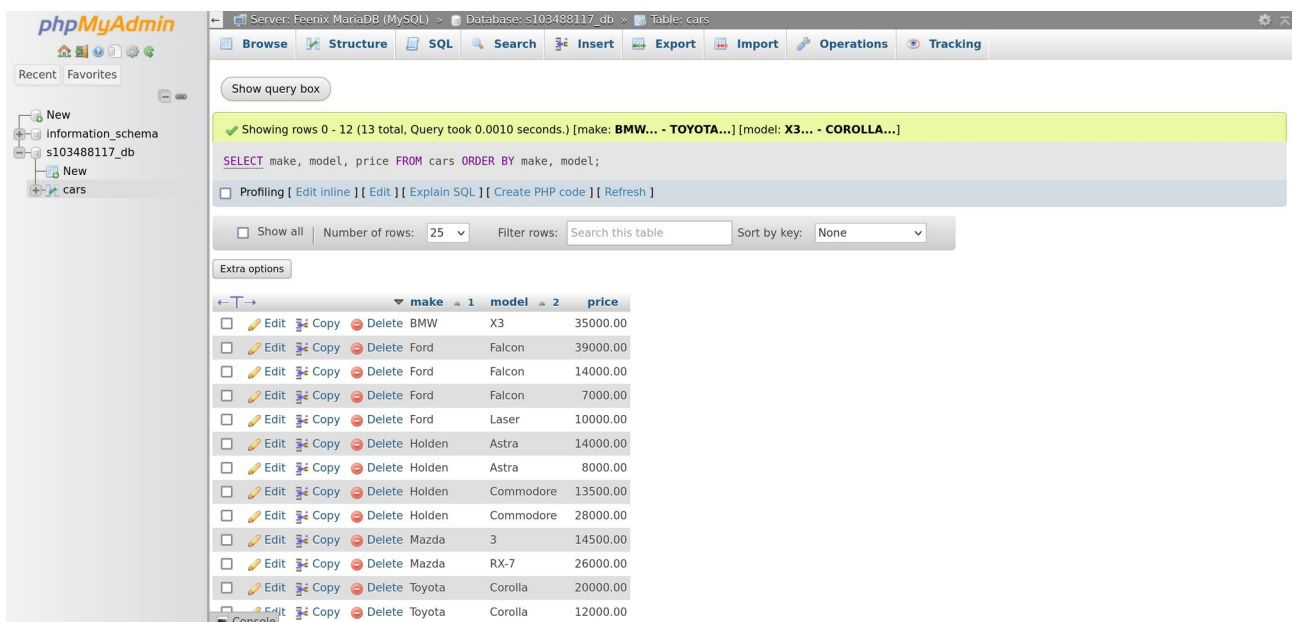
Number of rows: 25 Filter rows: Search this table Sort by key: None

car_id	make	model	price	yom
1	Holden	Astra	14000.00	2005
2	BMW	X3	35000.00	2004
3	Ford	Falcon	39000.00	2011
4	Toyota	Corolla	20000.00	2012
5	Holden	Commodore	13500.00	2005
6	Holden	Astra	8000.00	2001
7	Holden	Commodore	28000.00	2009
8	Ford	Falcon	14000.00	2007
9	Ford	Falcon	7000.00	2003
10	Ford	Laser	10000.00	2010
11	Mazda	RX-7	26000.00	2000
12	Toyota	Corolla	12000.00	2001
13	Mazda	3	14500.00	2009

Figure 3: Query results for all records

-- 2. Make, model, and price, sorted by make and model

**SELECT make, model, price FROM cars ORDER BY make, model;**



Showing rows 0 - 12 (13 total, Query took 0.0010 seconds.) [make: BMW... - TOYOTA...] [model: X3... - COROLLA...]

```
SELECT make, model, price FROM cars ORDER BY make, model;
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

make	model	price
BMW	X3	35000.00
Ford	Falcon	39000.00
Ford	Falcon	14000.00
Ford	Falcon	7000.00
Ford	Laser	10000.00
Holden	Astra	14000.00
Holden	Astra	8000.00
Holden	Commodore	13500.00
Holden	Commodore	28000.00
Mazda	3	14500.00
Mazda	RX-7	26000.00
Toyota	Corolla	20000.00
Toyota	Corolla	12000.00

Figure 4: Query results for make, model, and price sorted by make and model

-- 3. The make and model of the cars which cost \$20,000.00 or more.

**SELECT make, model FROM cars WHERE price >= 20000.00;**

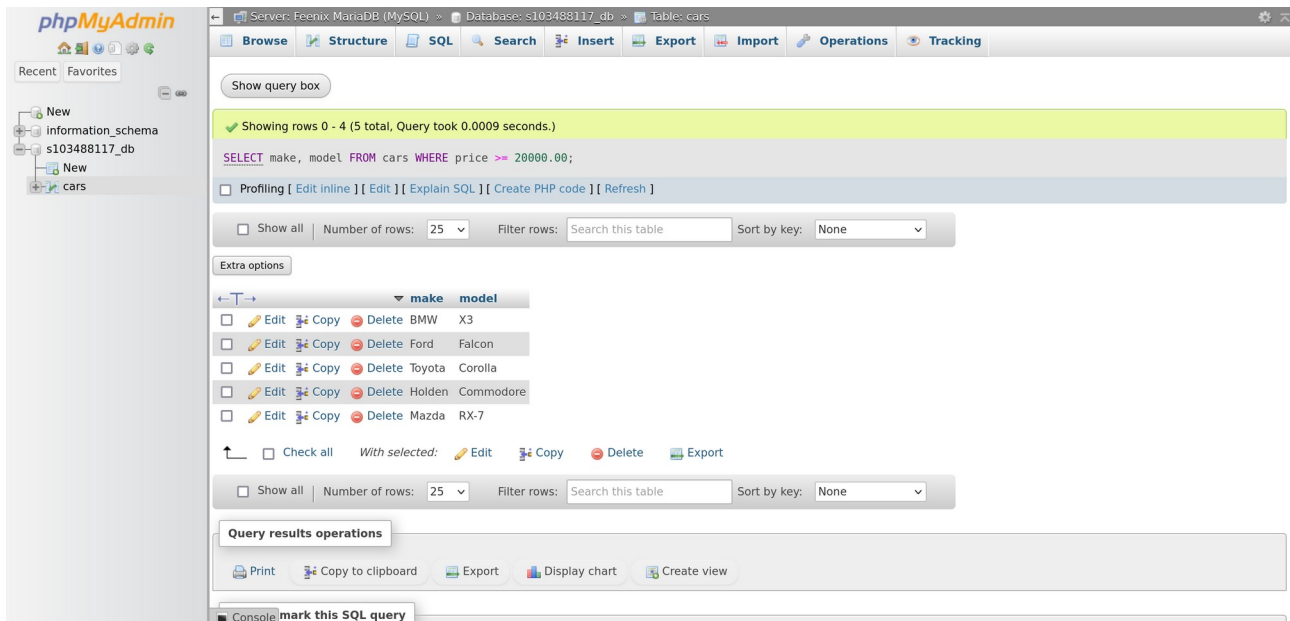


Figure 5: Query results for make and model of the cars which cost \$20,000.00 or more

-- 4. The make and model of the cars which cost below \$15,000.00.

**SELECT make, model FROM cars WHERE price < 15000.00;**

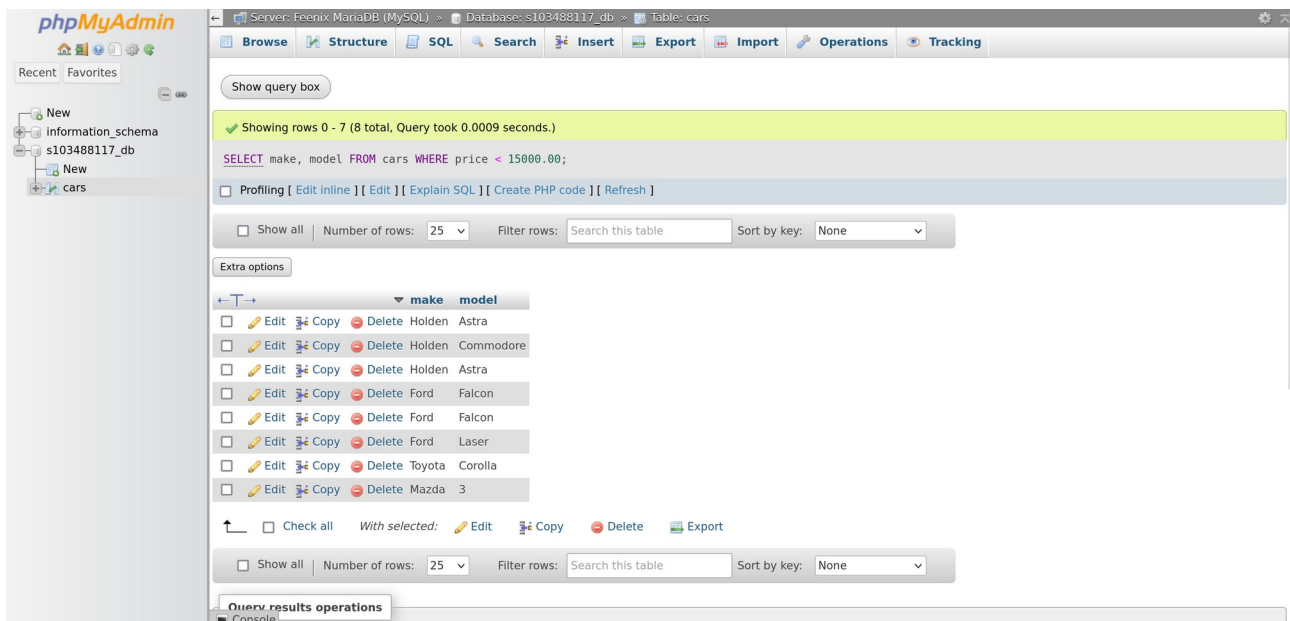
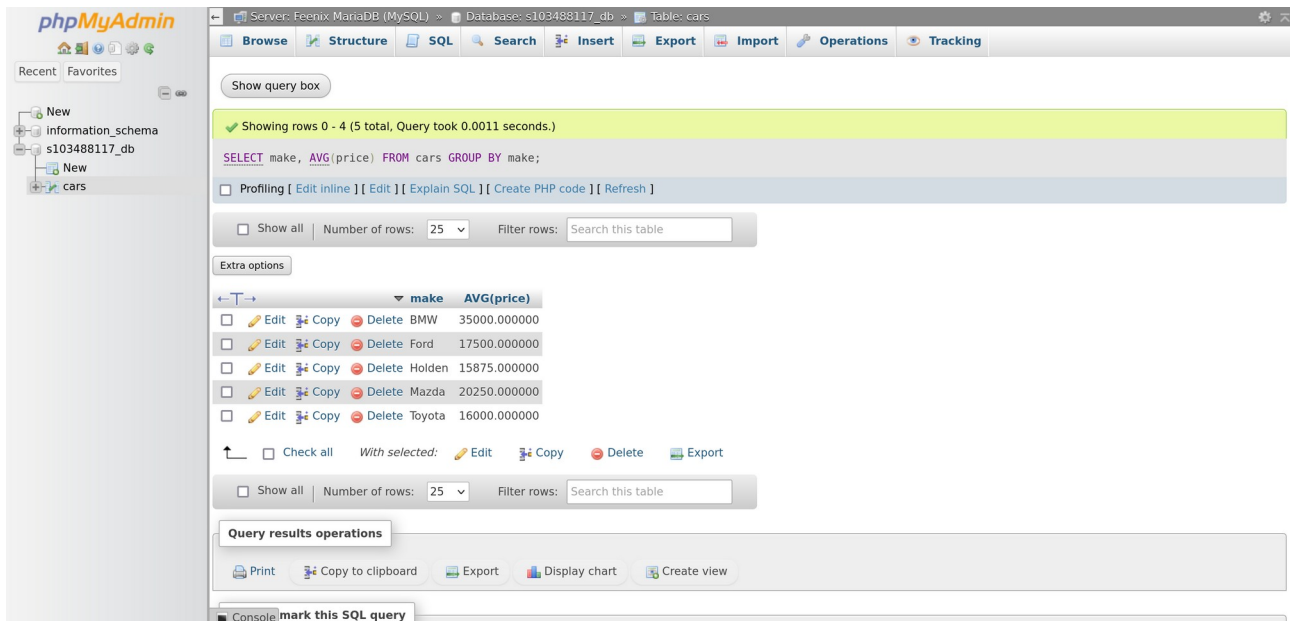


Figure 6: Query results for make and model of the cars which cost below \$15,000.00

-- 5. The average price of cars for similar make.

**SELECT make, AVG(price) FROM cars GROUP BY make;**



The screenshot shows the phpMyAdmin interface with the 'cars' table selected. The query 'SELECT make, AVG(price) FROM cars GROUP BY make;' has been executed, resulting in 5 rows of data. The interface includes a sidebar with the database structure, a top navigation bar, and a main content area with various toolbars and a console.

make	AVG(price)
BMW	35000.000000
Ford	17500.000000
Holden	15875.000000
Mazda	20250.000000
Toyota	16000.000000

Figure 7: Query results for average price of cars for similar make