



# *48<sup>th</sup> Street Watersports*

## *Company Database Tutorial Manual*



Dominic Oertel  
Uyen Tang  
Kayla Carter

# TABLE OF CONTENTS

|   |    |
|---|----|
| 1. INTRODUCTION.....                        | 3  |
| 1.1 WHO SHOULD USE THIS APPLICATION.....    | 3  |
| 1.2 WHO SHOULD USE THIS GUIDE.....          | 4  |
| 2. KEY FEATURES.....                        | 5  |
| 3. ADMINISTRATIVE USERS - OVERVIEW.....     | 6  |
| 3.1 GETTING STARTED.....                    | 6  |
| 3.2 LOGGING IN.....                         | 7  |
| 3.3 POSSIBLE COMMANDS.....                  | 8  |
| 4. ADMINISTRATIVE COMMANDS - PRINTING.....  | 9  |
| 4.1 GETTING ALL CONTENTS.....               | 9  |
| 4.2 GETTING CONTENTS BY CATEGORY.....       | 11 |
| 5. ADMINISTRATIVE COMMANDS - SEARCHING..... | 12 |
| 6. ADMINISTRATIVE COMMANDS - ADDING.....    | 13 |
| 7. ADMINISTRATIVE COMMANDS - DELETING.....  | 15 |
| 8. ADMINISTRATIVE COMMANDS - UPDATING.....  | 17 |
| 9. CONCLUSION.....                          | 19 |

Public download link:

<https://github.com/Nachtwind1100011/457GroupProject>

# 1. Introduction

## 1.1 Who Should Use This Application

The subject of this tutorial manual is the online database of 48th Street Watersports, Ocean City. As a rental business, 48th Street Watersports needs to keep constant track of the products it is releasing to customers, including details such as pricing, customer contact information, customers who need to sign the safety waiver, which customer has which equipment, and more.

While the company does not need a virtual database to function on a fundamental level, implementation of this concept would nonetheless offer a range of benefits. From reduction of paperwork and clutter to increases in time efficiency. Modernization of information tracking offers a quick and convenient way for administrators to search for and view high-level information tables.

The company database will provide an interface from the admin perspective. Administrators may use the essential functions provided. Entering and updating data, searching, deleting entries, all this and more will be possible with the 48th Street Watersports database developed by our group.

## **1.2 Who Should Use This Guide**

This tutorial will guide unfamiliar users through the features and functions offered by the database's user interface. It is intended for company administrators who wish to:

1. Enter, update, or remove data relating to employees, customers, rentals, tours, products, and pricing.
2. Search the database for details on specific entries using their primary key.
3. Print the contents of the database by category or as a whole.

Administrators who wish to carry out these tasks but are unsure of how should read on.

## 2. Features

### 2.1 Features for Users

The following are key features for administrators in the company:

1. Log in to the database
2. Print contents of entire database
3. Print contents of specific category
4. Search database by category using primary keys
5. Insert data entries into database
6. Delete entries from database
7. Update entries in database

Additional features will continue to be added as the development cycle goes on.

## 3. Administrative Users - Overview

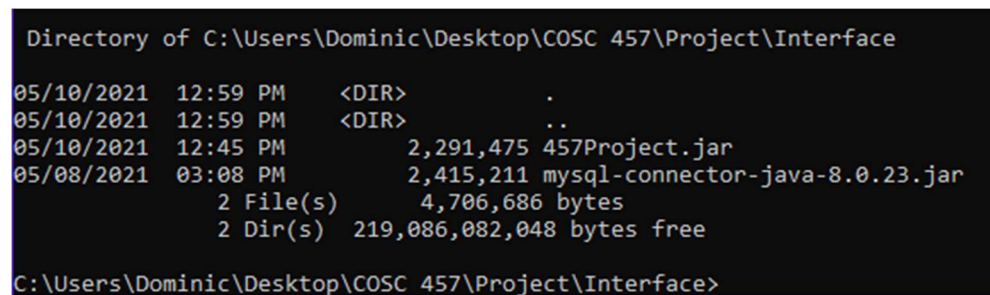
### 3.1 Getting started

First things first, users who wish to access the company database will need a dedicated directory containing two files:

1. 457Project.jar
2. Mysql-connector-java-8.0.23\*

The user's system will also need Java 8 installed to run the interface jar.

To start, enter the command line and navigate to the directory containing the two files.



```
Directory of C:\Users\Dominic\Desktop\COSC 457\Project\Interface
05/10/2021  12:59 PM    <DIR>          .
05/10/2021  12:59 PM    <DIR>          ..
05/10/2021  12:45 PM             2,291,475 457Project.jar
05/08/2021  03:08 PM             2,415,211 mysql-connector-java-8.0.23.jar
                2 File(s)              4,706,686 bytes
                2 Dir(s)  219,086,082,048 bytes free

C:\Users\Dominic\Desktop\COSC 457\Project\Interface>
```

Figure 3.1.1

Enter the following command to launch the interface program:

- `java -jar 457Project.jar`

\*Mysql-connector-java-8.0.23\* is included in the project repository. However, it may also be downloaded from Oracle's website. It is not exclusive to this program or modified by Group 6 in any way.

### 3.2 Logging in

Upon running 457Project.jar, the user should get a login prompt. The login is as follows:

- Username: a57group\_user
- Password: password

```
C:\Users\Dominic\Desktop\COSC 457\Project\Interface>java -jar 457Project.jar
Username:
```

Figure 3.2.1

Once these have been entered, assuming the user's IP address is authorized to connect remotely to the MySQL server, the user should be greeted with the following:

```
C:\Users\Dominic\Desktop\COSC 457\Project\Interface>java -jar 457Project.jar
Username: a57group_user
Password: password
Connected to database

print *category* to show contents
search *category* *primary key value* to search database
insert *category* to add to database
delete *category* *primary key value* to delete entry from database
update *category* *primary key value* to change entry from database
help to print this menu

For more information, see tutorial manual

What operation would you like to carry out?
```

Figure 3.2.2

Note: If the login information was put in incorrectly, or the user's IP address is not authorized, or the MySQL connector jar is not present, the user will be shown an error message like the following.

```
C:\Users\Dominic\Desktop\COSC 457\Project\Interface>java -jar 457Project.jar
Username: user
Password: pass
An error has occurred
java.sql.SQLException: Access denied for user 'user'@'pool' (using password: YES)
    at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
    at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:833)
    at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:453)
    at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:246)
    at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:198)
    at java.sql.DriverManager.getConnection(Unknown Source)
    at java.sql.DriverManager.getConnection(Unknown Source)
    at JDBC.main(JDBC.java:21)
```

Figure 3.2.3

### 3.3 List of possible commands

All commands are case sensitive.

- print
  - all
  - employees
  - customers
  - rentals
  - products
  - rates
- search
  - employees
  - employees.name
  - customers
  - customers.name
  - rentals
  - products
  - products.type
  - rates
- insert
  - employee
  - customer
  - rental
  - product
  - rate
- delete
  - employee
  - customer
  - rental
  - product
  - rate
- update
  - employee
  - customer
  - rental
  - product
  - rate



## 4. Administrative Commands - Printing

### 4.1 Getting contents of entire database

Once successfully logged in, the user has a variety of options to choose from on what they want to do. First and foremost, retrieving database contents.

To have the contents of the database printed to the user's command line, enter the following command(s). It may be entered on one line or two.

- print all

This will prompt the server to retrieve all entries of every table in the database.

```
What operation would you like to carry out?
print
all
Employees:
Name, Employee ID, Position, Phone #, email

    Kayla Carter
    123456 manager
    4435551011 kcarter@watersports.com

    Dominic Oertel
    123123 cashier
    3015551111 doertel@watersports.com

    Uyen Tang
    654321 tourGuide
    2025557076 utang@watersports.com

Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd
```

Figure 4.1.1

Once all data has been retrieved and printed, the user will get the following confirmation message and prompt to enter a new command.

```
Rates for different products:
Product Type, Rate ID, Timeframe, Value

    Kayak 0
    Hourly 9.99

    Kayak 1
    Trihourly 24.99

    Kayak 2
    Daily 99.99

End of records.

What operation would you like to carry out?
```

**Figure 4.1.2**

## 4.2 Getting contents of database by category

If the user wishes to get details of one specific category/table of data stored, they may narrow their search parameters by entering the name of the table they wish to print as the second word in the command.

- `print *category*`

An example is shown here:

```
What operation would you like to carry out?
print
customers
Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd

    Willem Defoe
    7035551012 wd@gmail.com
    10 Downing St

    Stacy Abrams
    3015558844 sa4p@gmail.com
    1600 Penn Ave NW
```

Figure 4.2.1

Note that categories and all possible print commands are listed on page 8 of this manual.

## 5. Administrative Commands - Searching

One of the most essential functions of a database is to enable the user to parse potentially massive amounts of entries for a result matching specific criteria. This interface program provides the user with two types of searching:

1. By primary key
2. By name\*

This is done using the command “search” followed by a valid category. The categories which have search by name enabled may be parsed using:

- search \*category\*.name

It is essential to use the dot operator in these instances, making the command after “search” one word.

When searching by either function, the result(s) matching the criteria given will be printed to the command line. If no result is found in the database, nothing will be printed.

Below is an example of searching the employee table using name.

```
What operation would you like to carry out?
search
employees.name
Enter first name: Uyen
Enter last name: Tang
Employee:
Name, Employee ID, Position, Phone #, email

    Uyen Tang
    654321 tourGuide
    2025557076 utang@watersports.com
What operation would you like to carry out?
```

Figure 5.1

\*only tables employees, customers, and products offer this function

## 6. Administrative Commands - Adding Entries

Inserting new tuples into the database is as simple as 123. First, enter the following command:

- insert \*category\*

Next, enter the table you wish to insert another entry into. In the example shown below, the developer is registering a new customer with the company.

Customers before new addition:

```
What operation would you like to carry out?
print
customers
Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd

    Willem Defoe
    7035551012 wd@gmail.com
    10 Downing St

    Stacy Abrams
    3015558844 sa4p@gmail.com
    1600 Penn Ave NW

    Robin Williams
    7035552024 rwill@hotmail.com
    Heaven

What operation would you like to carry out?
```

Figure 6.1

After selecting which table to add to, the user will be prompted to enter various details. These are added one at a time.

```
Invalid Category
What operation would you like to carry out?
insert customer

First name: Sam
Last name: Jackson
Phone number: 7035559092
Address: 12 Lampkin Lane
Email: jwinfield@gmail.com

Customer successfully added
```

Figure 6.2

Note: Some entries must be a single word/sequence of characters with no white spaces included (phone #, IDs, etc.). **Address must be 3.**

Now, when printing the customers table, a new entry appears:

```
What operation would you like to carry out?
print customers
Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd

    Willem Defoe
    7035551012 wd@gmail.com
    10 Downing St

    Stacy Abrams
    3015558844 sa4p@gmail.com
    1600 Penn Ave NW

    Robin Williams
    7035552024 rwill@hotmail.com
    Heaven

    Sam Jackson
    7035559092 jwinfield@gmail.com
    12 Lampkin Lane

What operation would you like to carry out?
```

Figure 6.3

## 7. Administrative Commands - Deleting Entries

Deleting entries from the database is also made as convenient as possible. Whether an employee has left the company, or a product or rate(s) no longer needed, all the user needs to delete entries is their primary key value.

The user is prompted for this information upon entering the “delete” command as well as the table to be deleted from.

- delete \*category\*

In the example below, the developer will delete an entry from the products table:

```
What operation would you like to carry out?
print products
Products:
Product ID, Type, Availability

    0000 Kayak available
    0001 Kayak available
    0002 Kayak available
    1000 Board available
    1001 Board available
    2000 Sailboat available
    2001 Sailboat available
    2002 Sailboat available
```

Figure 7.1

Take note of the Product ID of the product meant to be deleted. In this case, it is “0002”.

Next, the “delete product” command is entered. A prompt will appear asking for the Product ID of the entry meant to be removed.

```
What operation would you like to carry out?  
delete product  
Enter product ID  
0002  
Product deleted
```

Figure 7.2

Now if the contents of products is printed we see the entry is no longer there.

```
What operation would you like to carry out?  
print products  
Products:  
Product ID, Type, Availability  
  
0000 Kayak available  
0001 Kayak available  
1000 Board available  
1001 Board available  
2000 Sailboat available  
2001 Sailboat available  
2002 Sailboat available
```

Figure 7.3



## 8. Administrative Commands - Updating Existing Entries

Once data entries have been made in the company server, they are still easily modifiable. The user would simply need the primary key value of whatever entry they wish to modify.

Enter the command:

- `update *category*`

The user should then be prompted for a primary key value. Entering a correct value will then result in another series of prompts asking for the tuple's various values. In effect, all attributes may be changed when the update command is used.

The following is an example of the database user updating a customer's email and address:

```
What operation would you like to carry out?
print customers
Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd

    Willem Defoe
    7035551012 wd@gmail.com
    10 Downing St

    Stacy Abrams
    3015558844 sa4p@gmail.com
    1600 Penn Ave NW
```

Figure 8.1

Here we see the initial email addresses, which also serve as the primary key for customers.

Using the email address of the desired customer to be modified, in this case Stacy Abrams, the primary key value is entered and the user is prompted to re-enter the customer's information.

```
What operation would you like to carry out?
update customer
Enter customer email
sa4p@gmail.com

First name: Stacy
Last name: Abrams
Phone number: 3015558844
Address: 1600 Penn Avenue
Email: stabr@gmail.com
Customer with email sa4p@gmail.com updated
```

Figure 8.2

After all the prompts are answered, a confirmation message is printed. The user can also verify the update using the command "print customers".

```
What operation would you like to carry out?
print customers
Customers:
Name, Phone #, email, Address

    Leo DiCaprio
    2025558988 ldc@gmail.com
    7800 York Rd

    Willem Defoe
    7035551012 wd@gmail.com
    10 Downing St

    Stacy Abrams
    3015558844 stabr@gmail.com
    1600 Penn Avenue
```

Figure 8.3

We now see the email and address have been successfully changed.

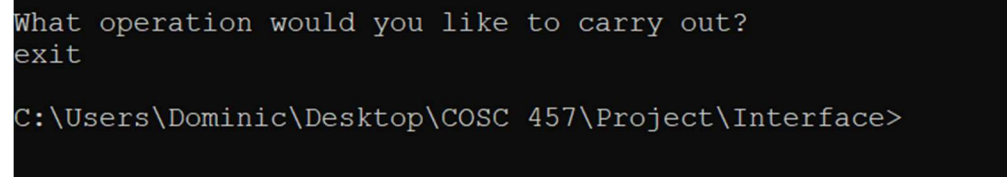
## 9. Conclusion

Once the user is finished with the application, logout is very simple.

Simply entering the command:

- `exit`

will log the user out and end the program.



```
What operation would you like to carry out?  
exit  
  
C:\Users\Dominic\Desktop\COSC 457\Project\Interface>
```

**Figure 9.1**