

# MySQL + airportdb

---

Copyright 2025 Jaeyoung Chun | Handong Global University

This guide will walk you through setting up MySQL with the **airportdb** sample database using Docker.

## Create Data Directories

If this is your first time, navigate to your **bmp** workspace and create the following data directories:

```
cd ~/projects/bmp/workspace
mkdir mysql
cd mysql
mkdir data
mkdir scratch
```

The **data** directory will be used to contain the **airportdb** database. The files in the **data** directory persists even if the contain stops.

The **scratch** directory is a temporary directory where **airportdb** will be downloaded onto.

The final directory structure should look like this:

```
workspace
├── mysql
│   ├── data
│   └── scratch
```

## Download **airportdb**

Download the **airportdb** archive:

```
curl -o scratch/airport-db.tar.gz
https://downloads.mysql.com/docs/airport-db.tar.gz
```

Extract the archive:

```
tar xvzf scratch/airport-db.tar.gz
mv airport-db ./scratch/
```

## Download MySQL Docker Image

Pull the MySQL Docker images from Docker Hub:

```
docker pull mysql:8.0.32
```

## Run the MySQL Container

For macOS

Run the MySQL container:

```
docker run -d --name aix_mysql \
  --rm \
  -p 3306:3306 \
  -v $(pwd)/data:/var/lib/mysql \
  -v $(pwd)/scratch:/scratch \
  -e MYSQL_ROOT_PASSWORD=abcd1234 \
  mysql:8.0.32
```

For Windows using PowerShell

It's a very long command, but you should type it all in one line because it's a single command:

```
docker run -d --name aix_mysql --rm -p 3306:3306 -v
${PWD}/data:/var/lib/mysql -v ${PWD}/scratch:/scratch -e
MYSQL_ROOT_PASSWORD=abcd1234 mysql:8.0.32
```

## Load airportdb

Open an interactive Bash shell inside the running `aix_mysql` container:

```
docker exec -it aix_mysql bash
```

Connect to the MySQL Console. Enter the password if asked.

```
mysql -p
```

Run the following statements, then exit from the MySQL console by pressing `control+d`:

```
SHOW GLOBAL VARIABLES LIKE 'local_infile';
SET GLOBAL local_infile = 'ON';
SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

Navigate to the `/scratch` directory:

```
cd /scratch
```

Make sure you see the `airportdb` files:

```
ls -al
```

Connect to the MySQL Shell. Enter the password if asked.

```
mysqlsh
```

Run the following command to load the `airportdb` dump. This will take some time. Once done, exit from the MySQL Shell by pressing `control+d`. This step can take quite long (sometimes hours). Please be patient.

```
util.loadDump("airport-db")
```

In case, it takes too long, you might want to increase the number of threads to speed things up. For example,

```
util.loadDump("airport-db", { threads:20 })
```

While loading, you might see this error: `Lock wait timeout exceeded; try restarting transaction`. You can safely ignore this error for now.

Once the loading process is complete, clean up the `/scratch` directory:

```
rm -rf /scratch/*
```

Exit the container by typing `exit`.

## Connect to MySQL

Use one of the following software to check if you can connect to your MySQL database. We will be using DataGrip in the class.

- DataGrip: <https://www.jetbrains.com/datagrip/>

- MySQL Workbench: <https://www.mysql.com/products/workbench/>
- DBeaver: <https://dbeaver.io/download/>

Connection Information:

- Hostname: `127.0.0.1`
- Port: `3306`
- Username: `root`
- Password `abcd1234`

## Stop the MySQL Container

```
docker stop aix_mysql
```

## Reference

- [MySQL Docker Hub](#)
- [airportdb Documentation](#)