Setting Up the airportdb Database

Table of Contents

This document describes `airportdb` sample database installation, structure, and usage.

The `airportdb` sample database is adapted from the Flughafen DB developed by Stefan Proell, Eva Zangerle, Wolfgang Gassler.

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2024-01-20 (revision: 90)

1 Preface and Legal Notices

This document describes `airportdb` sample database installation, structure, and usage.

Legal Notices

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

2 Introduction

The `airportdb` database is a large data set intended for use with MySQL DB Systems on Oracle Cloud Infrastructure (OCI) and HeatWave. The database is approximately 2GB in size and consists of 14 tables containing a total of 55,983,205 records.

Table 1 airportdb Tables

Table Name	Rows
`booking`	50831531
`flight`	416429
`flight_log`	0
`airport`	9939
`airport_reachable`	0
`airport_geo`	9854
`airline`	113
`flightschedule`	9851
`airplane`	5583
`airplane_type`	342
`employee`	1000
`passenger`	36346
`passengerdetails`	37785
`weatherdata`	4626432

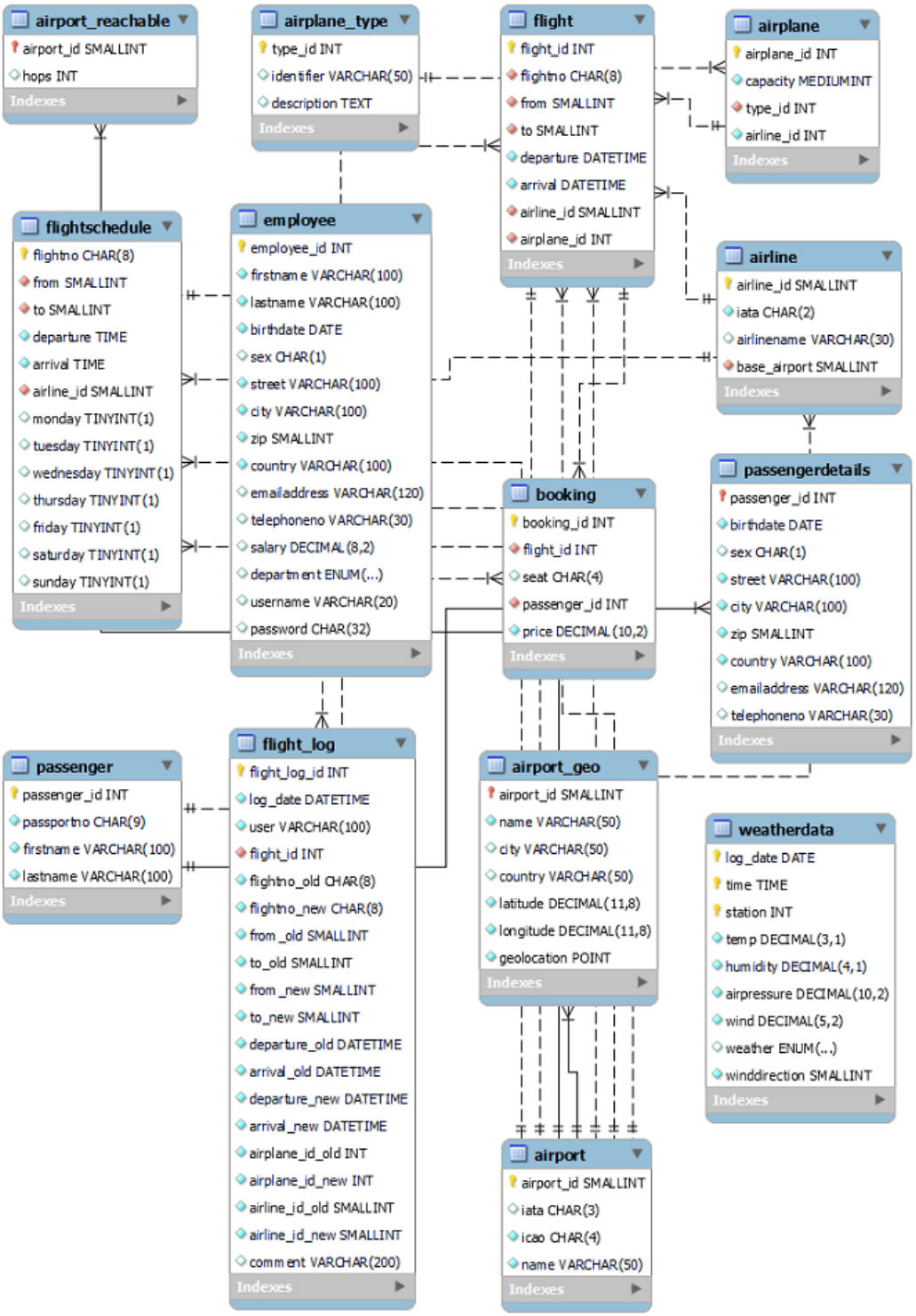
The `airportdb` data files were produced using the MySQL Shell Schema Dump Utility.

Data files produced by the MySQL Shell Schema Dump Utility include DDL files for creating the schema structure, compressed `.tsv` files that contain the data, and `.json` metadata files.

3 airportdb Structure

The following diagram provides an overview of the `airportdb` database structure.

Figure 1 airportdb Schema



4 Installation

The following instructions describe how to install the `airportdb` database on a MySQL DB System on Oracle Cloud Infrastructure (OCI). Two installation methods are provided:

- Installing airportdb from Object Storage
- Installing airportdb from a Compute Instance

The Object Storage method requires uploading the `airportdb` data files to an Object Storage bucket from where they are loaded into the MySQL DB System. The Compute instance method does not involve an Object Storage bucket; data files are downloaded to the Compute instance, from where they are loaded into the MySQL DB System.

> **Warning**
>
> The `airportdb` sample database can be installed on an on-premise MySQL Server instance, but due to the volume of data, it is only advisable to do so if your system has sufficient disk space and is capable of supporting large load operations consisting of millions of rows of data.

Installing airportdb from Object Storage

The installation procedure involves downloading the `airportdb` sample database, uploading the data files to an Object Storage bucket, and importing the data from the Object Storage bucket into the MySQL DB System using the MySQL Shell Dump Loading utility.

Before you begin, ensure that you have the following:

- The MySQL DB System's Endpoint (its private IP address), and a MySQL DB System administration user name and password. For information about retrieving the MySQL DB System Endpoint, see MySQL DB System Details.
- The public IP address of the Compute instance used to connect to the MySQL DB System. For information about the Compute Service, see Overview of the Compute Service.
- MySQL Shell 8.0.22 or higher installed on the Compute instance. For installation instructions, see Connecting to the MySQL DB System with SSH and MySQL Shell.
- A valid OCI CLI configuration file. See SDK and CLI Configuration File. If you have not installed and configured the OCI CLI, you must either install it or create a configuration file manually.
- Access to Object Storage and an existing bucket. For information about Object Storage, see Object Storage Overview.
- The Object Storage bucket name and namespace. For information about namespaces, see Understanding Object Storage Namespaces.

To install the `airportdb` sample database:

1. Download the `airportdb` database to a temporary location such as `/tmp/` or `C:\temp\` and unpack it. The `airportdb` sample database is provided for download as a compressed `tar` or Zip archive. The download is approximately 640 MBs in size.

   ```
   wget https://downloads.mysql.com/docs/airport-db.tar.gz
   tar xvzf airport-db.tar.gz
   ```

   or

   ```
   wget https://downloads.mysql.com/docs/airport-db.zip
   unzip airport-db.zip
   ```

   Unpacking the compressed `tar` or Zip archive results in a single directory named `airport-db`, which contains the data files.

2. Log in to your Oracle Cloud Infrastructure (OCI) account and upload the `airportdb` data files from the `airport-db` directory to an Object Storage bucket. For information about loading data into an Object Storage bucket, see Putting Data into Object Storage.

3. SSH to the Compute instance using the opc user and the public IP address of the compute instance. For additional information about connecting to a Compute instance, see Connecting to an Instance.

```
ssh opc@computeInstancePublicIP
```

4. Start MySQL Shell and connect to the MySQL DB System Endpoint. For additional information about connecting to a DB System, see Connecting to the MySQL DB System with SSH and MySQL Shell.

```
mysqlsh Username@IPAddressOfMySQLDBSystemEndpoint
```

5. Load the `airportdb` data files into the MySQL DB System using the MySQL Shell Dump Loading utility. The bucket name and namespace are required.

```
MySQL>JS> util.loadDump("airportdb", {threads: 16, deferTableIndexes: "all",
          osBucketName: "bucket-name", osNamespace: "name_space",
          ignoreVersion: true})
```

> **Note**
>
> The `deferTableIndexes: "all"` option defers creating secondary indexes until after the table data is loaded, which significantly reduces load times. If you intend to use `airportdb` with HeatWave, which does not use secondary indexes, you can avoid creating secondary indexes by specifying the `loadIndexes: "false"` option instead of `deferTableIndexes: "all"`. For more information about MySQL Dump Load options, see MySQL Shell Dump Loading utility.

After the data is imported into the MySQL DB System, you can load the tables into HeatWave. For instructions, see Section 5, “Loading airportdb into HeatWave”.

Installing airportdb from a Compute Instance

The installation procedure involves downloading the `airportdb` database to a Compute instance and importing the data from the Compute instance into the MySQL DB System using the MySQL Shell Dump Loading utility.

Before you begin, ensure that you have the following:

- The MySQL DB System's Endpoint (its private IP address), and a MySQL DB System administration user name and password. For information about retrieving the MySQL DB System Endpoint, see MySQL DB System Details.
- The public IP address of the Compute instance used to connect to the MySQL DB System. For information about the Compute Service, see Overview of the Compute Service.
- MySQL Shell 8.0.22 or higher installed on the Compute instance. For installation instructions, see Connecting to the MySQL DB System with SSH and MySQL Shell.
- A valid OCI CLI configuration file. See SDK and CLI Configuration File. If you have not installed and configured the OCI CLI, you must either install it or create a configuration file manually.

To install the `airportdb` database:

1. SSH to the Compute instance using the opc user and the public IP address of the compute instance. For additional information about connecting to a Compute instance, see Connecting to an Instance.

```
ssh opc@computeInstancePublicIP
```

2. Download the `airportdb` sample database and unpack it. The `airportdb` sample database is provided for download as a compressed `tar` or Zip archive. The download is approximately 640 MBs in size.

```
wget https://downloads.mysql.com/docs/airport-db.tar.gz
tar xvzf airport-db.tar.gz
```

or

```
wget https://downloads.mysql.com/docs/airport-db.zip
unzip airport-db.zip
```

Unpacking the compressed `tar` or Zip archive results in a single directory named `airport-db`, which contains the data files.

3. Start MySQL Shell and connect to the MySQL DB System Endpoint. For additional information about connecting to a DB System, see Connecting to the MySQL DB System with SSH and MySQL Shell.

```
mysqlsh Username@IPAddressOfMySQLDBSystemEndpoint
```

4. Load the `airportdb` database into the MySQL DB System using the MySQL Shell Dump Loading utility.

```
MySQL>JS> util.loadDump("airport-db", {threads: 16, deferTableIndexes: "all",
          ignoreVersion: true})
```

> **Note**
>
> The `deferTableIndexes: "all"` option defers creating secondary indexes until after the table data is loaded, which significantly reduces load times. If you intend to use `airportdb` with HeatWave, which does not use secondary indexes, you can avoid creating secondary indexes by specifying the `loadIndexes: "false"` option instead of `deferTableIndexes: "all"`. For more information about MySQL Dump Load options, see MySQL Shell Dump Loading utility.

After the data is imported into the MySQL DB System, you can load the tables into HeatWave. For instructions, see Section 5, “Loading airportdb into HeatWave”.

5 Loading airportdb into HeatWave

The following procedure describes how to load the `airportdb` database from a MySQL DB System into HeatWave.

Before you begin:

- The `airportdb` database must be loaded in the MySQL DB System. If you have not done so, see Section 4, “Installation”.
- A HeatWave Cluster must be enabled for use with your MySQL DB System. For information about adding a HeatWave Cluster to a DB System, see Adding a HeatWave Cluster to a DB System.

To load the `airportdb` into HeatWave:

1. SSH to the Compute instance from your local machine using the opc user and the public IP address of the compute instance. For additional information about connecting to a Compute instance, see Connecting to an Instance.

```
ssh opc@computeInstancePublicIP
```

2. Start MySQL Shell and connect to the MySQL DB System's Endpoint (its private IP address). For additional information about connecting to a DB System, see Connecting to the MySQL DB System with SSH and MySQL Shell.

```
mysqlsh Username@IPAddressOfMySQLDBSystemEndpoint
```

> **Note**
>
> Alternatively, you can connect to the DB System using a MySQL Client:

```
mysql --host IPAddressOfMySQLDBSystemEndpoint -u Username -p
```

3. Change the MySQL Shell execution mode to SQL and run the following Auto Parallel Load command to load the `airportdb` tables into HeatWave.

```
MySQL>JS> \sql
MySQL>SQL> CALL sys.heatwave_load(JSON_ARRAY('airportdb'), NULL);
```

Tables can be unloaded from HeatWave using the following statements:

```
USE airportdb;

ALTER TABLE booking SECONDARY_UNLOAD;
ALTER TABLE flight SECONDARY_UNLOAD;
ALTER TABLE flight_log SECONDARY_UNLOAD;
ALTER TABLE airport SECONDARY_UNLOAD;
ALTER TABLE airport_reachable SECONDARY_UNLOAD;
ALTER TABLE airport_geo SECONDARY_UNLOAD;
ALTER TABLE airline SECONDARY_UNLOAD;
ALTER TABLE flightschedule SECONDARY_UNLOAD;
ALTER TABLE airplane SECONDARY_UNLOAD;
ALTER TABLE airplane_type SECONDARY_UNLOAD;
ALTER TABLE employee SECONDARY_UNLOAD;
ALTER TABLE passenger SECONDARY_UNLOAD;
ALTER TABLE passengerdetails SECONDARY_UNLOAD;
ALTER TABLE weatherdata SECONDARY_UNLOAD;
```

6 License for the airportdb Database

FlughafenDB. (2015). Stefan Proell, Eva Zangerle, Wolfgang Gassler. www.flughafendb.cc. https://doi.org/10.5281/zenodo.3968361

Please contact http://www.mysql.com/about/contact/ for more information.