

Java Persistence Query Language (JPQL)

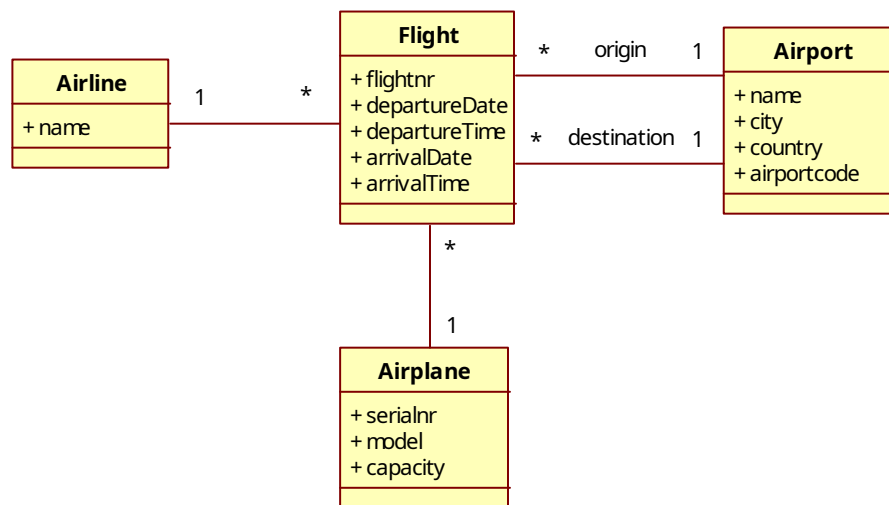
Exercise JPQL.1

The Setup:

In this exercise you will write several JPQL queries against a given data set. Start this exercise by downloading the provided **W2D2-JPQL** project from Sakai, and adding the Hibernate, MySQL and log4j dependencies to it.

The Application:

The provided application uses the following flight scheduling problem domain.



Currently the application simply retrieves and prints all the flight info stored in the database. Running the application should produce output that contains something like:

Flight:	Departs:	Arrives:
NW 36	Detroit 8/6/09 7:10 PM	Amsterdam 8/7/09 9:00 AM
NW 96	Tokyo 8/6/09 3:05 PM	Detroit 8/7/09 1:45 PM
QF 12	Los Angeles 8/5/09 10:30 PM	Sydney 8/7/09 6:15 AM
QF 21	Sydney 8/6/09 9:55 PM	Tokyo 8/7/09 6:55 AM
UA 944	Chicago 8/6/09 2:30 PM	Frankfurt 8/7/09 5:45 AM
UA 934	Los Angeles 8/6/09 12:59 PM	London 8/7/09 7:30 AM
NW 8445	Amsterdam 8/7/09 7:15 AM	London 8/7/09 7:40 AM
NW 1689	Detroit 8/7/09 12:05 PM	Chicago 8/7/09 12:21 PM
QF 3101	Los Angeles 8/7/09 3:00 PM	New York 8/7/09 11:39 PM
QF 4022	Tokyo 8/7/09 11:05 AM	Singapore 8/7/09 5:15 PM
UA 941	Frankfurt 8/7/09 12:45 PM	Chicago 8/7/09 2:53 PM
UA 4842	London 8/7/09 8:10 AM	Amsterdam 8/7/09 10:38 AM

Note, the data for this exercise is loaded from `import.sql` inside the resources directory. If hibernate detects an `import.sql` file at that location it will always run it on startup.

The Exercise:

Look through the application code, and once you feel comfortable with what it does update the JPQL queries in **Application.java** to retrieve the following data:

- a) All flights leaving the USA with a capacity > 500
- b) All airlines that use A380 (model) airplanes. Note: while this query can be done using only implicit joins it's important to also practice using explicit joins.
- c) All fights using 747 planes that don't belong to 'Star Alliance'
- d) All flights leaving before 12pm on 08/07/2009

Hint: you cannot use an **implicit join** to traverse a One-To-Many relationship, and when you use an explicit join you should also use select in your query.

Hint: just like most SQL dialects JPQL expects dates to be formatted according to the ISO 8601 standard which is 'yyyy-mm-dd', not the US mm/dd/yyyy format. Times should be formatted in the 24 hour 'hh:mm:ss' format.