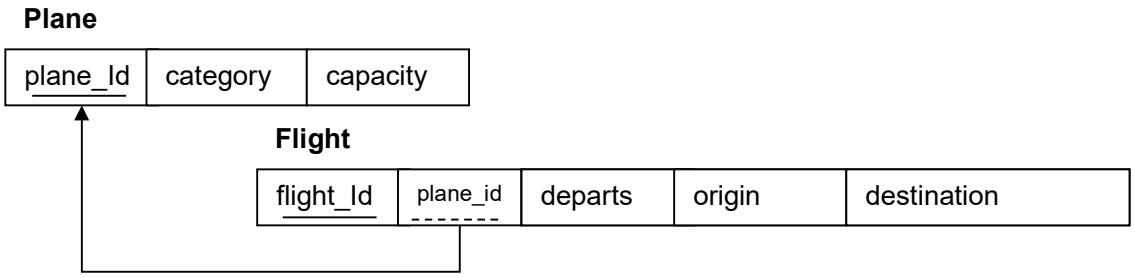
COMP9120 Database Management Systems **Semester 2, 2016**

Tutorial Week 3: The Relational Model

You'll need to logon to Oracle using SQLDeveloper in this lab. If you can't remember how to do this, the week 1 lab notes should provide a brief summary.

Exercise 1. Flight Booking Schema

Here is a relational model diagram for part of a flight booking database schema:



Write DDL statements for these two relations. Execute the statements within your Oracle account to test them out. You are using Oracle 12c syntax so you may wish to refer to the Oracle 12c `CREATE TABLE` documentation at:

http://docs.oracle.com/database/121/SQLRF/statements_7002.htm#i2062833

as well as:

http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/Oracle/user_guide/oracle_guide.html#ct

You should choose appropriate data types for your attributes based upon the following details:

(i) Each plane has a unique alphanumeric ID of up to 8 characters, a category (either 'jet' or 'turboprop'), and capacity (maximum number of passengers);

(ii) A flight has a unique numerical ID, departure date, the plane making the flight, the origin and destination;

Check the documentation on Oracle data types at:

http://docs.oracle.com/database/121/SQLRF/sql_elements001.htm#SQLRF30020

Also make sure you capture all the appropriate key constraints (there is one foreign and two primary keys).

Once the tables are created, add another integrity constraint to capture the extra rule that a plane cannot make more than one flight a day. `ALTER TABLE` syntax is documented at:

http://docs.oracle.com/database/121/SQLRF/statements_3001.htm#i2109868

http://docs.oracle.com/database/121/SQLRF/statements_3001.htm#SQLRF01001

If you'd like to remove one of the tables that you've created, you may do so with `DROP TABLE`, after which you can create the table afresh.

Exercise 2. Populating your database

Add data to your relations using `INSERT` statements (don't forget to `COMMIT` these changes). Then try inserting, updating and deleting data to violate the integrity constraints of the database. See if you can trigger an error for each of the classes of constraints described in Exercise 1 above.

You can inspect the contents of each relation with, e.g.:

```
SELECT * FROM Plane;
SELECT * FROM Flight;
```