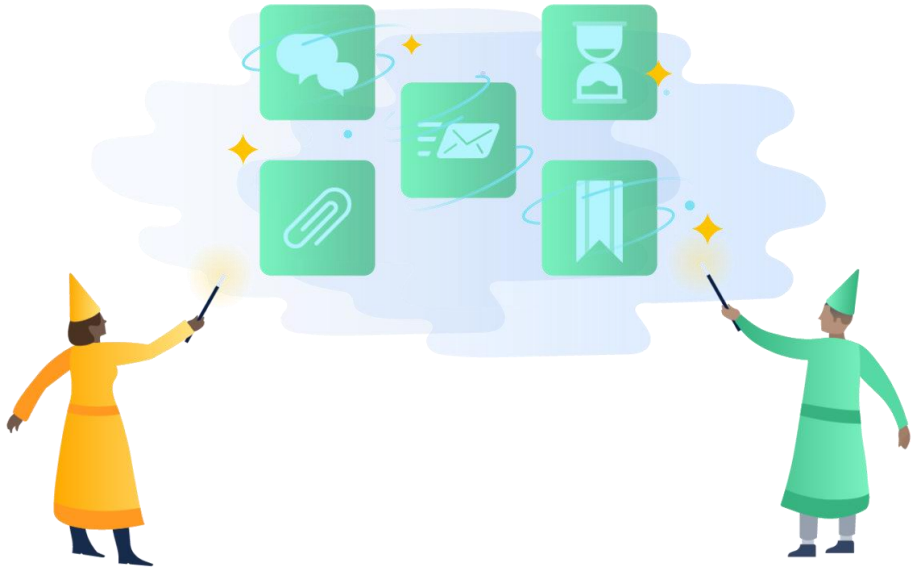
Tracked

Structured Data Modelling Assignment #2

Quantium Airlines

Unpacking the *magic* of Data Modelling



Scenario Introduction

Estimated Time: 1.5 – 3 Hours

You've recently joined **Quantium Airlines**, one of the largest Airlines in Norway, as an entry level Data Analyst in the data and platforms team. Recently Quantium Airlines has been looking at upgrading the basic data model that supports its award-winning Flight Booking System.

You have been asked to work with the platforms team to create a basic data model which captures the following customer information listed in the tables below.

Customer Table

- Customer_ID
- First_Name
- Last_Name

Flight Table

- Flight_ID
- Customer_ID

Schedule Table

- Flight_ID
- Arrival_Time
- Departure_Time
- Departure_Gate
- Departure_Location
- Aircraft_ID
- Arrival_Location

Booking Table

- Booking_Code
- Flight_ID
- Cabin_Class_ID

Price Table

- Price
- Cabin_Class_ID
- Destination

Aircraft Table

- Aircraft_ID
- Aircraft_Operator

Currently all this data resides in separate data tables. It's up to YOU as the data analyst to create an data model for Quantium Airlines that shows how the data is all connected into a singular data model using the dbdiagram.io platform.

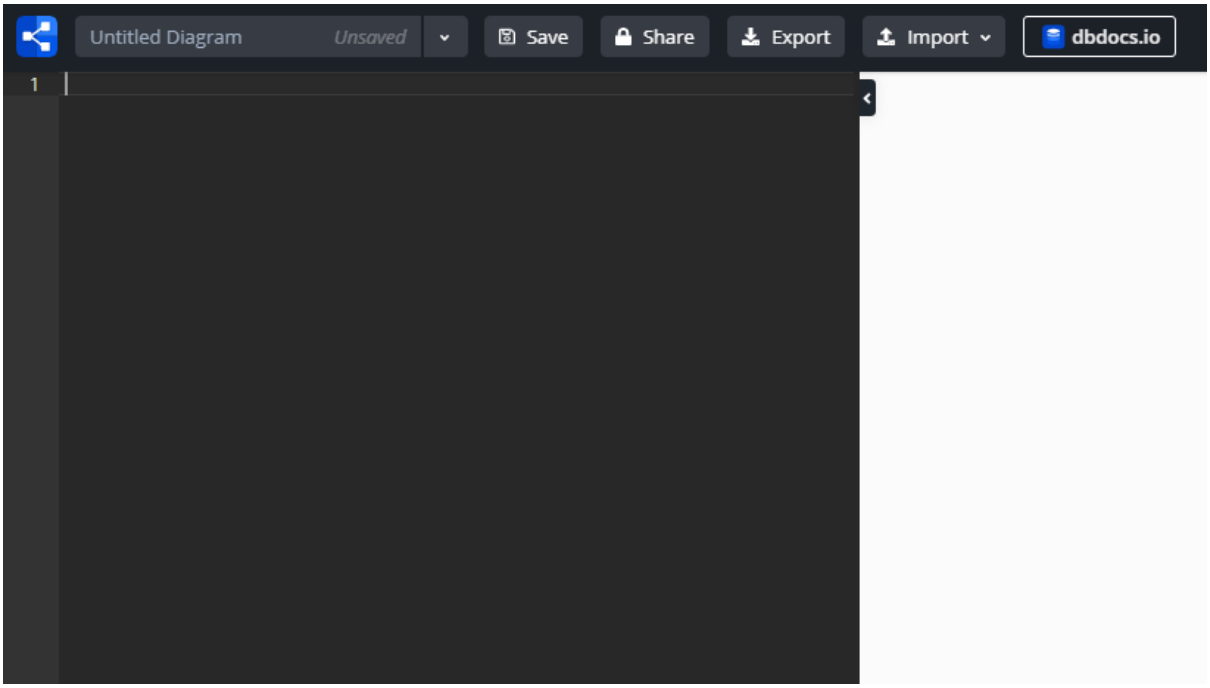
Let's get started.

Instructions

You'll need to reference the Tabular Data File that has been provided **here**. This file includes all the meta-data associated with the data model you need to create, namely the data type, whether the column is a primary key or not and the column name. You will need this information as you create the columns in the dbdiagram.io platform.

1. Using the dbdiagram.io platform, create the tables that contain Customer, Flight, Booking, Schedule, Booking, Price and Aircraft information using the Tabular Data File provided. (*Refer back to Data Modelling Hints Resource if you're not sure how to create these tables)*

   Note: When opening up the dbdiagram.io platform, you will need to **delete all the code that is preloaded in the terminal window** until the terminal is blank before creating any tables.



After discussion with the Flight Booking Manager, she indicates that would like to see not just the table(s) by themselves, but with clearly listed **relationships (i.e. cardinality).**

2. Using the table relationship information provided by the Flight Booking Manager below, modify the data model you've created clearly showing how the tables are linked together.

   Hint: Think carefully regarding the data fields in each table: A relational table has to have relations with other tables – you'll need to look for a 'key' which can be used to link tables together.

The relationship information has been listed below:

- For each **Customer_ID** in the Customer Table**,** this should be reflected in a **One-to-Many** relationship with the **Flight Table**

- For each **Flight_ID** in the Flight Table**,** a customer should be able to have **multiple flights in the Schedule Table reflective of a One-to-Many relationship**

- For each **Flight_ID** in the **Schedule Table**, the booking table should reflect a **One-to-One Relationship** with the **Booking Table**

- For each **Aircraft_ID** in the **Aircraft Table**, a **One-to-One** relationship should be reflected with the **Schedule Table**

- For each **Cabin_Class_ID** entry in the **Booking Table,** this should be reflected and matched with **One-to-One Relationship** in the **Price Table**

3. Save and export your model from the dbdiagram.io platform and send this to your track guide for feedback and review.

Great job! You've successfully built your second data model using the dbdiagram.io platform! Woohoo!