1.4 Relational Databases: Homework (Basic)

Open the Flight Search UI of the educational application and find some flights that give results (please refer to General Materials of the course for information about the application and your account).

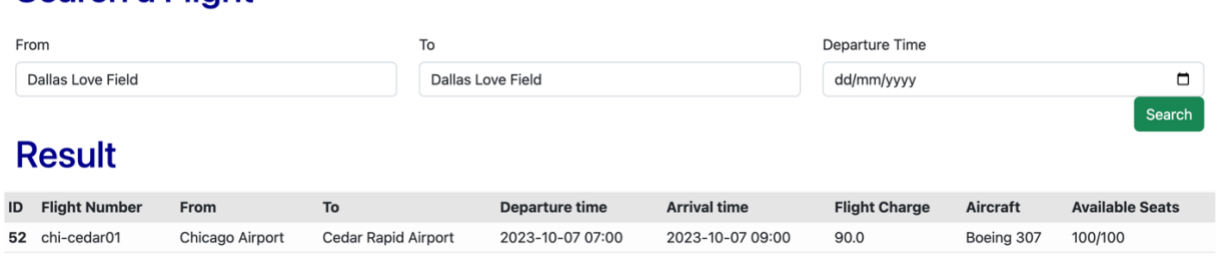
1. Explore how objects are shown on UI, and find them in the DB.
2. Where is it in the DB (use ER-diagram to find)
3. How do the tables relate to each other?
4. What types of relations can you see?
5. What data types are used?

Open the Passenger booking UI of the educational application. Answer the same questions.

****

1. **Explore how objects are shown on UI, and find them in the DB.**

A page: **Search Flight**



We can see on the UI 3 fields in the area "Search a Flight":
From – which relates to table airport in the DB, field **airport_name**. *I think that the realization is through departure_airport_airport_id (table flight) (which is a foreign key from the table airport) to table airport.*
To – which also relates to table airport in the DB, field **airport_name.** *I think that the realization is through destination_airport_airport_id (table flight), (which is a foreign key from the table airport) to table airport.*

Departure Time *(shown as Departure Time on UI* 

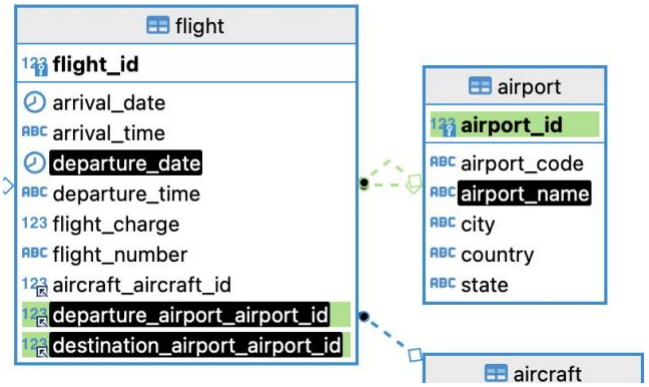
instead of Departure date) – which relates to table flight (**departure_date**) in the DB.

We can see on the UI in the area "Result" fields:
ID - which relates to table flight **flight_id** (Primary key).
Flight Number - which relates to table flight **flight_number.**
From - which relates to table airport **airport_name** through foreign key departure_airport_airport_id from table flight.

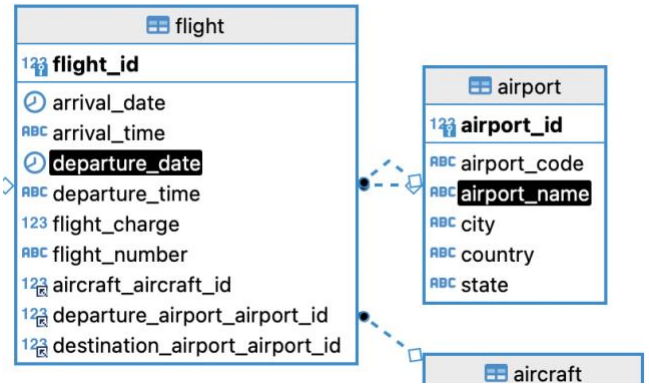
To - which relates to table airport **airport_name** through foreign key destination_airport_airport_id from table flight.
Departure time - which relates to table flight, fields **departure_date** + **departure_time.**
Arrival time - which relates to table flight, fields **arrival_date**+ **arrival_time.**
Flight Charge - which relates to table flight **flight_charge.**
Aircraft - - which relates to table aircraft **model.**
Available Seats - which relates to table aircraft **number_of_seats.** I have a question here – In the DB we have number_of_seats on the aircraft. How can we know a real number of available seats? Are there any calculations in the process?

2. **Where is it in the DB (use ER-diagram to find)**

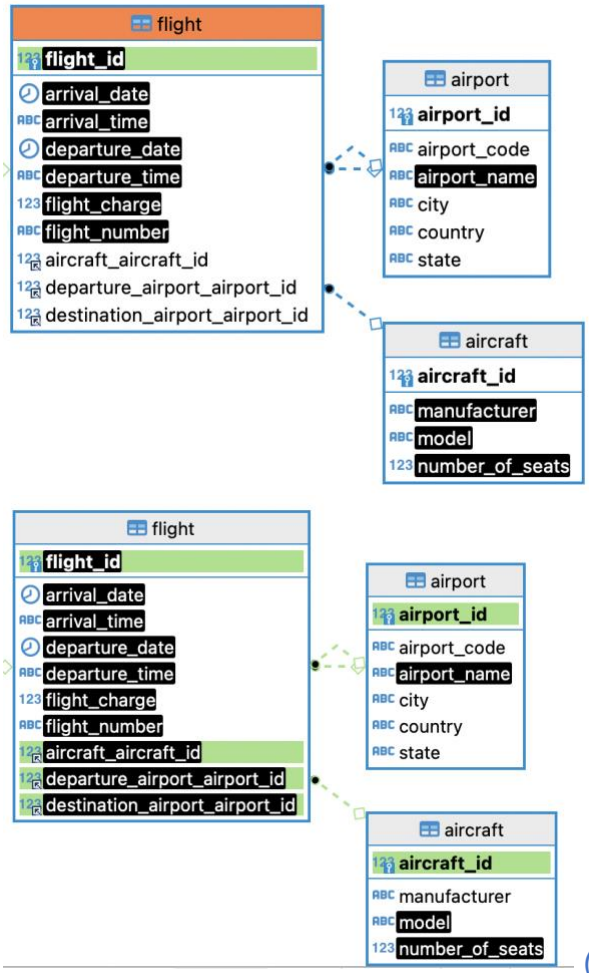
Search a Flight:



(with FK)



Result:

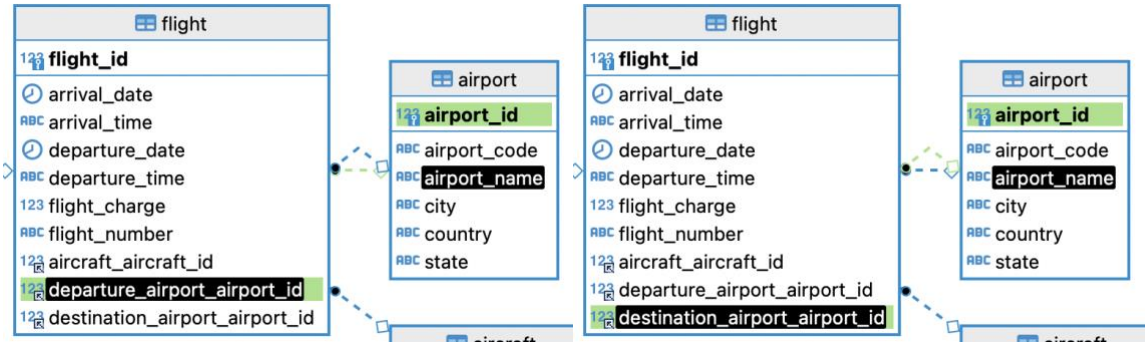


(with FK)

3. **How do the tables relate to each other?**

Search a Flight:

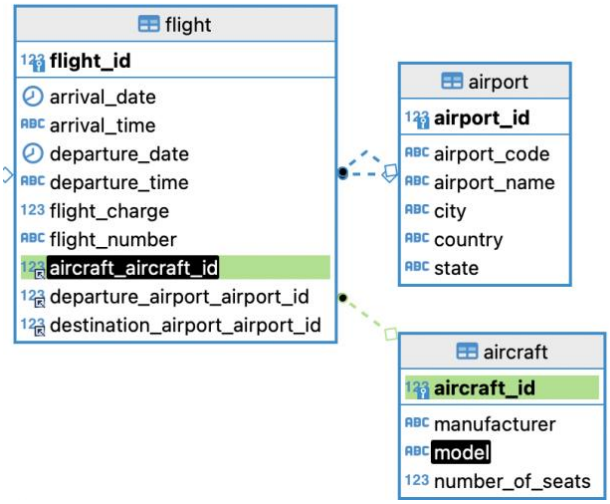
We have 2 tables related: flight and airport. There is a double relationship between the flight and the airport tables and 2 airport IDs in the flight table.



In the first relationship airport_id (Primary Key in the table airport) is linked as **departure_airport_airport_id** to the table flight (foreign key from the table airport). And through this relationship we get departure_airport_airport_id - > airport_name.
In the second relationship airport_id (Primary Key in the table airport) is linked as **destination_airport_airport_id** to the table flight (foreign key from the table airport). Through this relationship we get destination_airport_airport_id -> airport_name.

Result:

We have 3 related tables. The relationship between flight and airport described above. There is also a relationship between flight and aircraft where aircraft_id (Primary Key in the table aircraft) is linked as aircraft_aircraft_id to the table flight (foreign key from the table aircraft). Through this relationship we get aircraft_aircraft_id -> model.



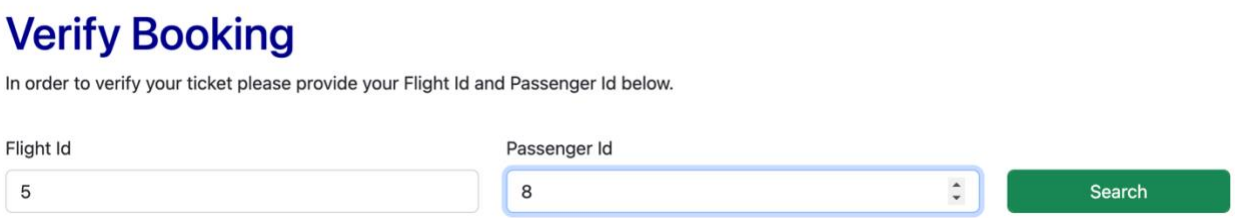
4. **What types of relations can you see?**
   Types of relationships:
   One-to-many (flight - airport). One airport could service several flights, but having several airports in one flight is impossible on this DB.
   One-to-many (flight - aircraft). One aircraft could fly in several flights.

5. **What data types are used?**

Column Name	Data Type	
flight_id	bigint	
arrival_date	date	
arrival_time	varchar(255)	
departure_date	date	
departure_time	varchar(255)	
flight_charge	double	
flight_number	varchar(255)	
aircraft_aircraft_id (FK)	*aircraft_id*	*bigint*
departure_airport_airport_id (FK)	*airport_id*	*int*
destination_airport_airport_id (FK)		
airport_name	varchar(255)	
model	varchar(255)	
number_of_seats	int	

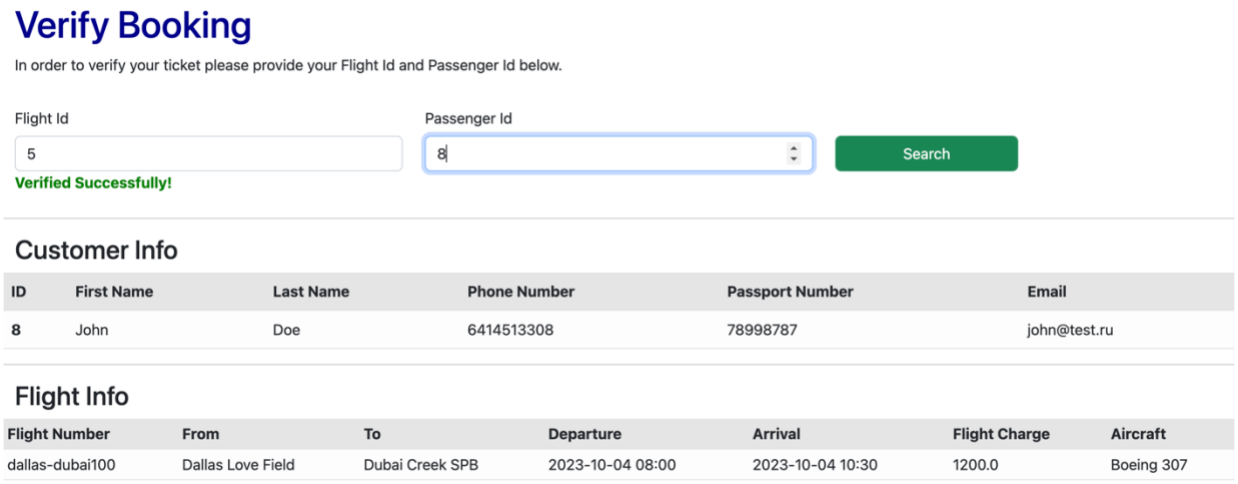
Open the Passenger booking UI of the educational application. Answer the same questions.

1. **Explore how objects are shown on UI, and find them in the DB.**
   A page: **Verify Booking**
   Search:



After pressing on Search button fields Flight Id and Passenger Id are empty. We cannot verify or check our search results. Is it right or not?
How should a user know what is the Passenger Id? I think it is more relevant to search according to passengers passport, for example.

Result:



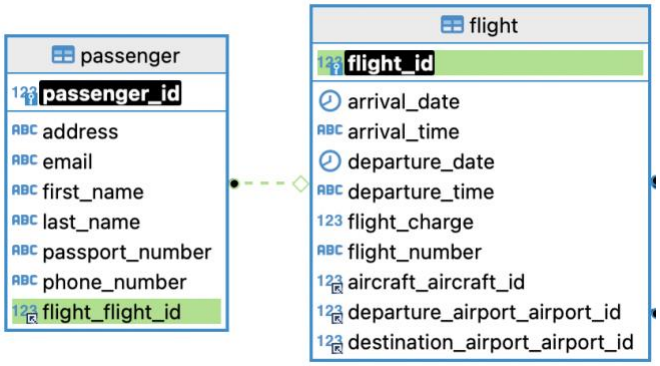
We can see on the UI 2 fields in the "Search area":
Flight Id - which relates to table passenger, field **flight_flight_id** in the DB. *I think that the realization is through flight_flight_id (table passenger) (which is a foreign key from the table passenger) to table flight.*
Passenger Id - which relates to table passenger, field **passenger_id** in the DB.

Customer Info:
ID - which relates to table passenger **passenger_id** (Primary key).
First Name - which relates to table passenger **first_name**
Last Name - which relates to table passenger **last_name**
Phone Number - which relates to table passenger **phone_number**
Passport Number - which relates to table passenger **passport_number**
Email - which relates to table passenger **email**

Flight Info:
Flight Number - which relates to table flight **flight_number** in the DB.
From - which relates to table airport **airport_name** through foreign key departure_airport_airport_id from table flight.
To - which relates to table airport **airport_name** through foreign key destination_airport_airport_id from table flight.
Departure - which relates to table flight, fields **departure_date** + **departure_time.**

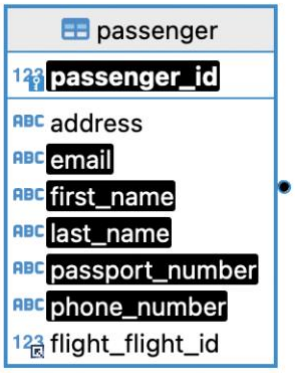
Arrival - which relates to table flight, fields **arrival_date**+ **arrival_time.**
Flight Charge - which relates to table flight **flight_charge.**
Aircraft- which relates to table aircraft **model.**

2. **Where is it in the DB (use ER-diagram to find)**
   Search engine:

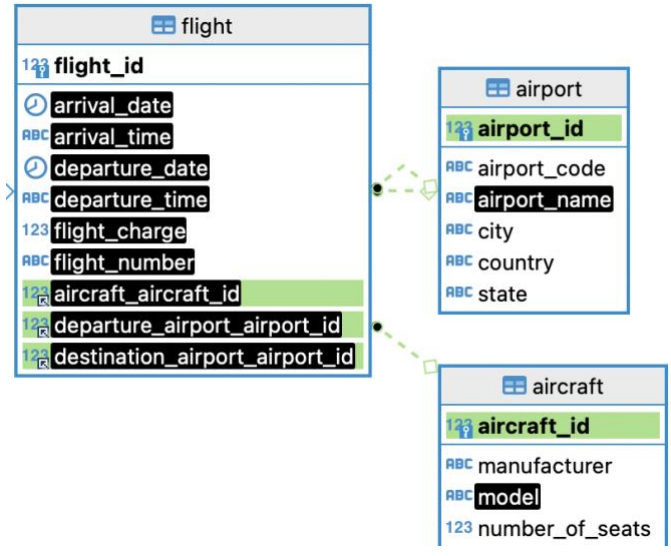


(with FK)

Customer Info:

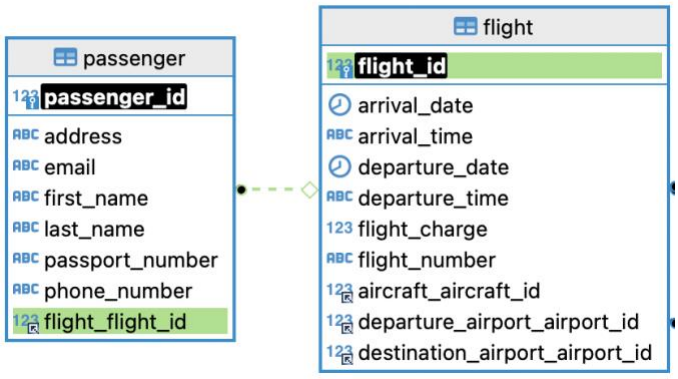


Flight Info:



(with FK)

3. **How do the tables relate to each other?**
   Search engine:

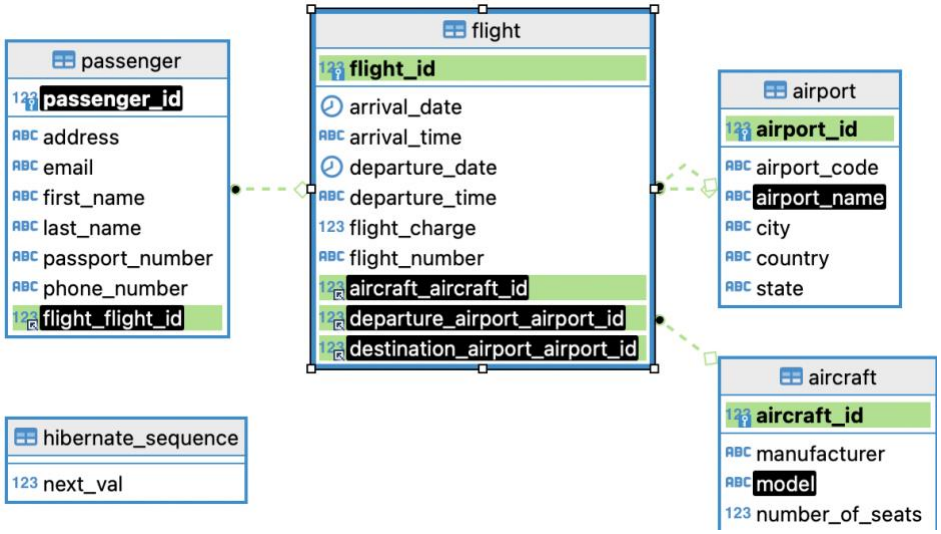


We have 2 tables related: passenger and flight.

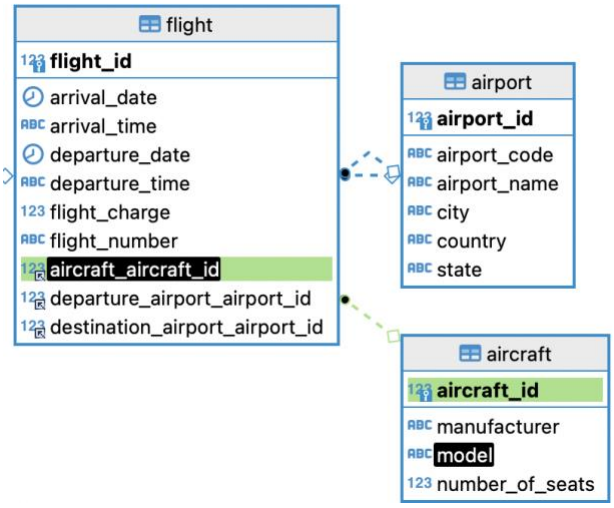
In this relationship flight_id (Primary Key in the table flight) is linked as **flight_flight_id** to the table passenger (foreign key from the table flight). And through this relationship we get flight_flight_id -> passenger_id.

Result:

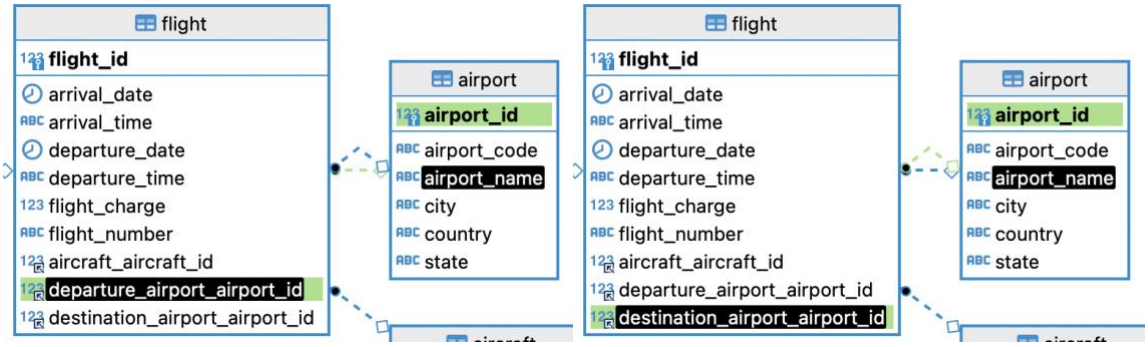
We have 4 related tables. The relationship between flight and passenger described above.



There is also a relationship between flight and aircraft where aircraft_id (Primary Key in the table aircraft) is linked as aircraft_aircraft_id to the table flight (foreign key from the table aircraft). Through this relationship we get aircraft_aircraft_id -> model.



We also have 2 tables related: flight and airport. There is a double relationship between the flight and the airport tables and 2 airport IDs in the flight table.



In the first relationship airport_id (Primary Key in the table airport) is linked as **departure_airport_airport_id** to the table flight (foreign key from the table airport). And through this relationship we get departure_airport_airport_id - > airport_name.
In the second relationship airport_id (Primary Key in the table airport) is linked as **destination_airport_airport_id** to the table flight (foreign key from the table airport). Through this relationship we get destination_airport_airport_id -> airport_name.

4. **What types of relations can you see?**

Types of relationships:
One-to-many (flight - airport). One airport could service several flights, but having several airports in one flight is impossible on this DB.
One-to-many (flight - aircraft). One aircraft could fly in several flights.
One-to-Many (passenger - flight). One flight relates to many passengers, but having several flights to one passenger is impossible on this DB.

5. **What data types are used?**

Column Name	Data Type
flight_id	bigint
passenger_id	bigint
first_name	varchar(255)
last_name	varchar(255)

phone_number	varchar(255)	
passport_number	varchar(255)	
flight_number	varchar(255)	
airport_name	varchar(255)	
departure_date	date	
departure_time	varchar(255)	
arrival_date	date	
arrival_time	varchar(255)	
flight_charge	double	
aircraft_aircraft_id (FK)	*aircraft_id*	*bigint*
departure_airport_airport_id (FK)	*airport_id*	*int*
destination_airport_airport_id (FK)		
flight_flight_id (FK)	*flight_id*	*bigint*
model	varchar(255)	