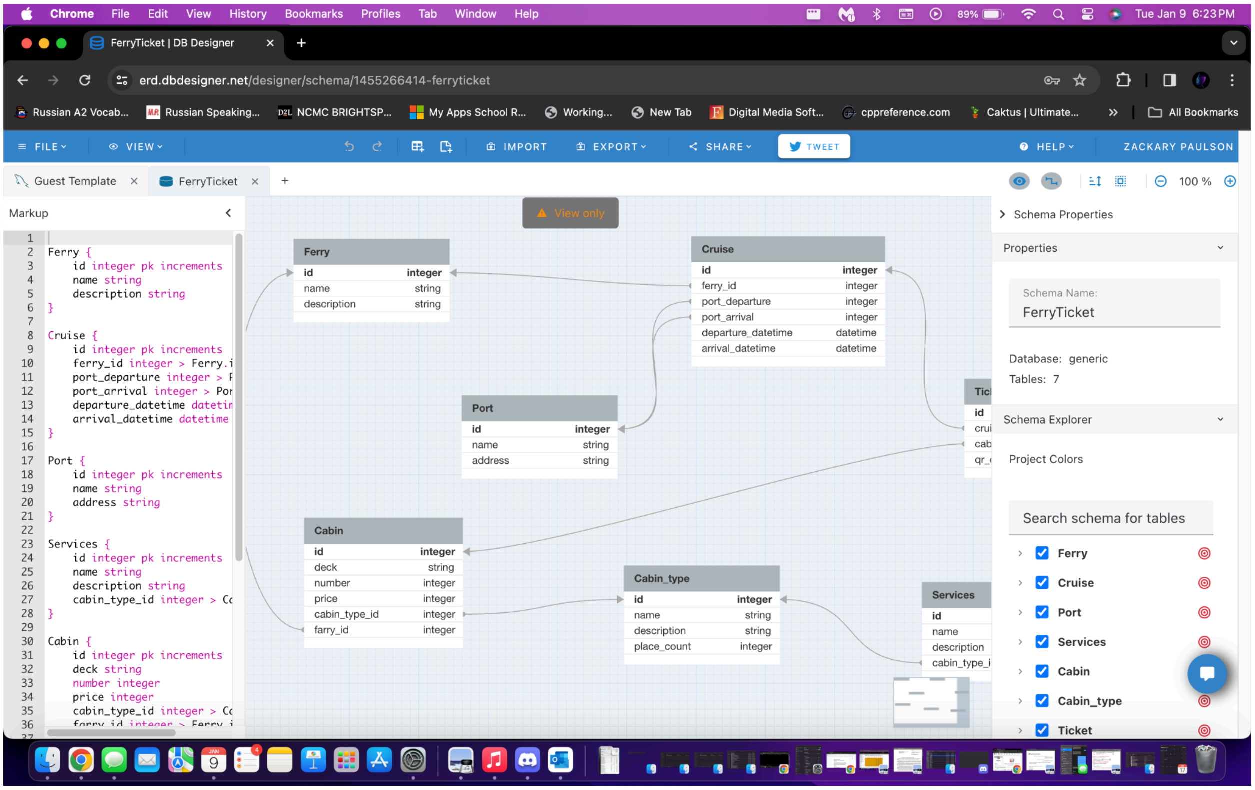
Zackary Paulson

E1 - Databases Refresher

1 - Description of what is happening between the 7 Tables:

Description of the ferry table: It looks like the Ferry table is made up of three different data types one being an int called id that is an integer and is being incremented while the other two data types name and description are strings data types. Moving onto Cruise it looks as though it also shares the id value like in Ferry but then it uses a foreign key to show where the ferry.id came from, this repeats also for port departure and port arrival. This way Cruise and keep track of where that info is coming from. The departure_daytime and the arrival_daytime seem to just be additional data. As for port, it looks like it is just an int that increments and has two different string values in its name and address. In the services table, it looks like it is also incrementing an int as well as having two other data types as type string and another foreign key for cabin type. As for cabin_type, it looks like it also has an int var that is an integer and increments. Then it has a deck which is a string and then I see a number that is type int as well as a price that is type int. Then there is a foreign key for cabin_type_id and lastly, there is a foreign key that farry_id. Next, there is cabin_type which also has an int that increments and there are two other data types of type string name and description, and finally, another integer that holds the place_count. Ticket is the final table and it holds an int called id that is also incrementing and then two other foreign key relationships cruise_id and cabin_id and finally what looks like a QR code binary so it can hold whatever kind of larger file you'd like there.

All in all, it looks like these tables use foreign key relationships to track what is going on. This way pieces of information for example id can be transferred and be used in all of the tables so that the values that are stored in one table could also give another value to another table for example, the cruise could take that information from the ferry like the id and store that data inside of it's ferry_id to identify the ferry that will be used on the cruise. But at the same time, the Port also gives its id value to the cruise that way you know which port you're going into.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
89% Tue Jan 9 6:23 PM
FerryTicket | DB Designer
erd.dbdesigner.net/designer/schema/1455266414-ferryticket
Russian A2 Vocab... Russian Speaking... NCMC BRIGHTSP... My Apps School R... Working... New Tab Digital Media Soft... cppreference.com Caktus | Ultimate... All Bookmarks
FILE VIEW IMPORT EXPORT SHARE TWEET HELP ZACKARY PAULSON
Guest Template FerryTicket
100 %
View only

Markup

```
Ferry {
    id integer pk increments
    name string
    description string
}

Cruise {
    id integer pk increments
    ferry_id integer > Ferry.i
    port_departure integer > F
    port_arrival integer > Por
    departure_datetime datetin
    arrival_datetime datetime
}

Port {
    id integer pk increments
    name string
    address string
}

Services {
    id integer pk increments
    name string
    description string
    cabin_type_id integer > Cc
}

Cabin {
    id integer pk increments
    deck string
    number integer
    price integer
    cabin_type_id integer > Cc
    farry_id integer > Ferry.i
```

Ferry	
id	integer
name	string
description	string

Cruise	
id	integer
ferry_id	integer
port_departure	integer
port_arrival	integer
departure_datetime	datetime
arrival_datetime	datetime

Port	
id	integer
name	string
address	string

Cabin	
id	integer
deck	string
number	integer
price	integer
cabin_type_id	integer
farry_id	integer

Cabin_type	
id	integer
name	string
description	string
place_count	integer

Services
id
name
description
cabin_type_i

Tic
id
crui
cab
qr_

Schema Properties

Properties

Schema Name: FerryTicket

Database: generic

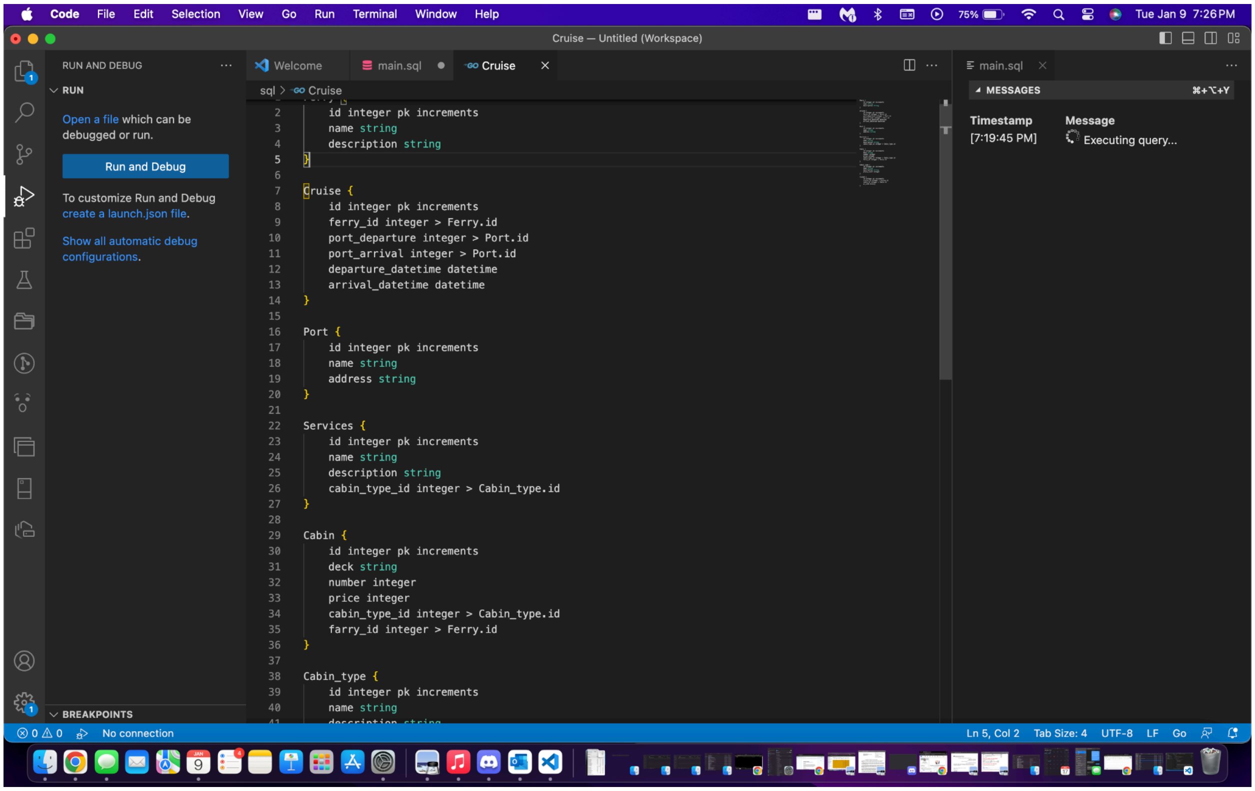
Tables: 7

Schema Explorer

Project Colors

Search schema for tables

☑ Ferry
☑ Cruise
☑ Port
☑ Services
☑ Cabin
☑ Cabin_type
☑ Ticket



Open a file which can be debugged or run.

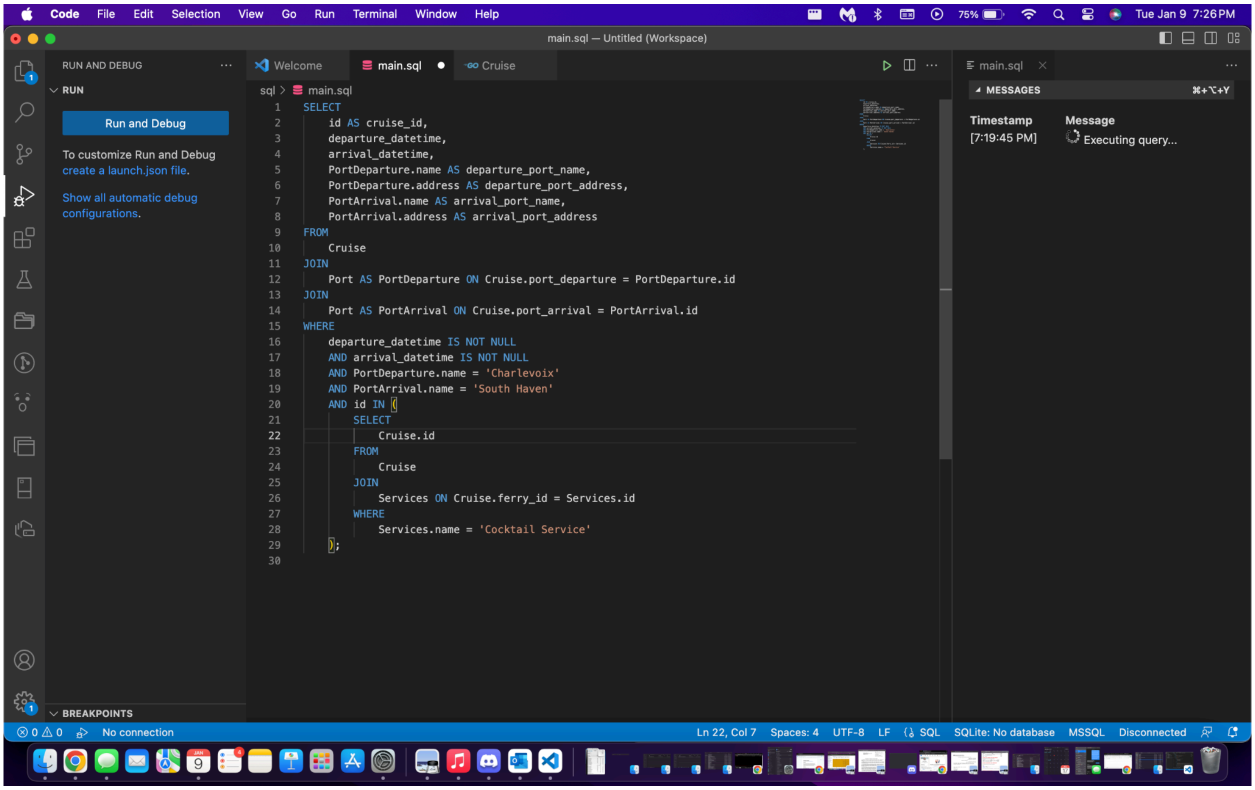
To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

```
    id integer pk increments
    name string
    description string
}

Cruise {
    id integer pk increments
    ferry_id integer > Ferry.id
    port_departure integer > Port.id
    port_arrival integer > Port.id
    departure_datetime datetime
    arrival_datetime datetime
}

Port {
    id integer pk increments
    name string
    address string
}

Services {
    id integer pk increments
    name string
    description string
    cabin_type_id integer > Cabin_type.id
}

Cabin {
    id integer pk increments
    deck string
    number integer
    price integer
    cabin_type_id integer > Cabin_type.id
    farry_id integer > Ferry.id
}

Cabin_type {
    id integer pk increments
    name string
    description string
```

Timestamp	Message
[7:19:45 PM]	Executing query...

⊗ 0 ⚠ 0 No connection Ln 5, Col 2 Tab Size: 4 UTF-8 LF Go



Code File Edit Selection View Go Run Terminal Window Help
Tue Jan 9 7:26 PM
main.sql — Untitled (Workspace)
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
Welcome
main.sql
Cruise
sql > main.sql
MESSAGES
Ln 22, Col 7 Spaces: 4 UTF-8 LF SQL SQLite: No database MSSQL Disconnected
No connection

```sql
SELECT
    id AS cruise_id,
    departure_datetime,
    arrival_datetime,
    PortDeparture.name AS departure_port_name,
    PortDeparture.address AS departure_port_address,
    PortArrival.name AS arrival_port_name,
    PortArrival.address AS arrival_port_address
FROM
    Cruise
JOIN
    Port AS PortDeparture ON Cruise.port_departure = PortDeparture.id
JOIN
    Port AS PortArrival ON Cruise.port_arrival = PortArrival.id
WHERE
    departure_datetime IS NOT NULL
    AND arrival_datetime IS NOT NULL
    AND PortDeparture.name = 'Charlevoix'
    AND PortArrival.name = 'South Haven'
    AND id IN (
        SELECT
            Cruise.id
        FROM
            Cruise
        JOIN
            Services ON Cruise.ferry_id = Services.id
        WHERE
            Services.name = 'Cocktail Service'
    );
```

```
Timestamp      Message
[7:19:45 PM]   Executing query...
```

3. In other tables the port is being called by putting more than one value into the Port.id which is a violation of normalization because of the redundancy of data.