SQL Project - AirBnB Database

Erin Quinn, Hayden Parr, Ty Tinker, Kyle Miller, and Zane Jackson

TABLE OF CONTENTS

Project Scenario

Database Description

Our database is a mock Airbnb database that contains information about properties, their renters, their owners, and the employees that work there from 2018-2020. The tables and data resemble similar characteristics that a company like Airbnb would utilize. We decided to create this ‘simulated’ business as it provides some real-world insight into what information businesses can derive from this data.

We felt that Airbnb would have an interesting ER diagram as there is the company, renters, and customers. Due to there being a lot of data associated with each of the three key components stated previously, we wanted to challenge ourselves with structuring a database that allowed all parts to seamlessly be queried. The business would allow for us to work more closely with Relational Database Management. To make the database useful, we would have to ensure that the primary and foreign keys operated properly.

Another reason for choosing this business was to explore some of the insights and analytics that Airbnb would derive. For example, it would be interesting to explore which regions have the most bookings and the price differences between them. Even though we are unable to collect the large amounts of data that Airbnb stores, we will be able to understand what the role of a business analyst at the company would look like.

Table Themes and Business Logic

Our database would be best used for managers to be able to get insights on their employees, their property owners, their renters, and the details of their bookings. Our database can be broken down into 3 different groups that are all brought together by the bookings table. Those groups are employees, renters, and owners. These groups all meet and are connected to a singular booking. However, renters, owners, and employees may all have more than one booking, but a booking can only have one employee, renter, and owner. Thus, we have many to one relationships.

Within each group there are additional tables that can be joined together to give insights about each group. The renter review tables are connected using the renterid where every renter has a single review and date associated with it. The owners table also gives information about the owner and what property they own. This can be joined with the properties table to see where the property is located. Each property can only have 1 owner, but owners can have multiple properties. This is similar to the renters and bookings tables where a booking can only have one renter, however a renter can make multiple bookings.

The countries table is connected to multiple tables where it can be joined to get the full name of the country in addition to its 2 letter code. This is helpful as sometimes you do not need to see the entire country name, and when a new country is added to the country table there are no referential integrity issues when it comes to inputting those countries into either the renter or property tables.

The employees group has the tables departments, services, and service requests connected to it. The service table has information pertaining to service requests that an employee has completed. Each service has a ticket ID that then corresponds to the service requests table which holds the date of the request. This can be used for managers to see the type of service requests their employees are fulfilling and on what days. The departments table connects to the employees to show the department they work in.

In order to get more information about bookings, you will need to join tables or use aggregate functions. For example, if you wanted to get the email address of the owner of a property that a specific renter has rented, you will need to join the bookings and owners table. Additionally, if you want to see the revenue that a booking is bringing in, you will need to multiply the price per night column by the number of nights.

Important Fields and Data Transformations

With our goal to create a database which efficiently connects renters, owners, and employees, we used a highly centralized, multi-file relational database which supported multiple many to one relationships through the use of one primary central table, “bookings”, and three secondary tables, “owners”, “renters”, and “employees”. Each booking is assigned a key which uniquely identifies it. As each and every booking involves a renter, an employee, and an owner, “bookingID” is the single most important column in our database, as it identifies the unique bookings that each owner, renter, and employee are assigned to. Each key from the three secondary tables has a foreign key pair column in the bookings table, which gives the database join ability. The 3 keys identifying the central tables are significant to keeping both structural integrity and allowing for in-depth analysis.

Beyond the main components of the centralized structure, there are two tables which have an interesting role in analysis; “properties” and “countries”. The properties table would, without its relationship to countries, only serve as a reference for the property each owner uses. However, as it is connected to countries, it can actually be used as a way to bypass the bookings table in queries and quickly provide geographical feedback about owners *and* renters. Rental companies like Airbnb rely heavily on location data for all kinds of business operations, including advertising, mapping, and tracking. The country table absolutely could have been implemented as a column in both the properties and renters tables, but instead it is used as an important connection between the two.

We built our database from the ground up, meaning that we crafted and input our own test data, and didn’t rely on an outside source to populate our fields. Thus, our data did not require any transformations in SQL. However, the process of data input required that the fields are populated correctly, with structural integrity. This meant that when we created key data, we had to fill the foreign key pairs with the same data, multiple times. When it came to creating date data, we had to verify that it was readable to MySQL, following a format of yyyy-mm-dd. We followed standard formats for emails i.e. ‘local-part@domain’, and for extra verifiable integrity, we made each ID column from each table unique.

Normalization

Our normalization technique for producing a set of suitable relations that support our data requirements is close to that of a star schema where it is very normalized. Our database is in a form where all attributes are directly dependent on the primary key, thus it is in third normal form (3NF). Our fact table is the bookings table with the dimension tables being the others surrounding it (employees, owners, renters, etc.). There is little to no data redundancy that allows for less maintenance and the ability to to keep the integrity of the tables. We wanted to ensure that our tables would update automatically by using foreign keys to connect tables, but not duplicating any columns.

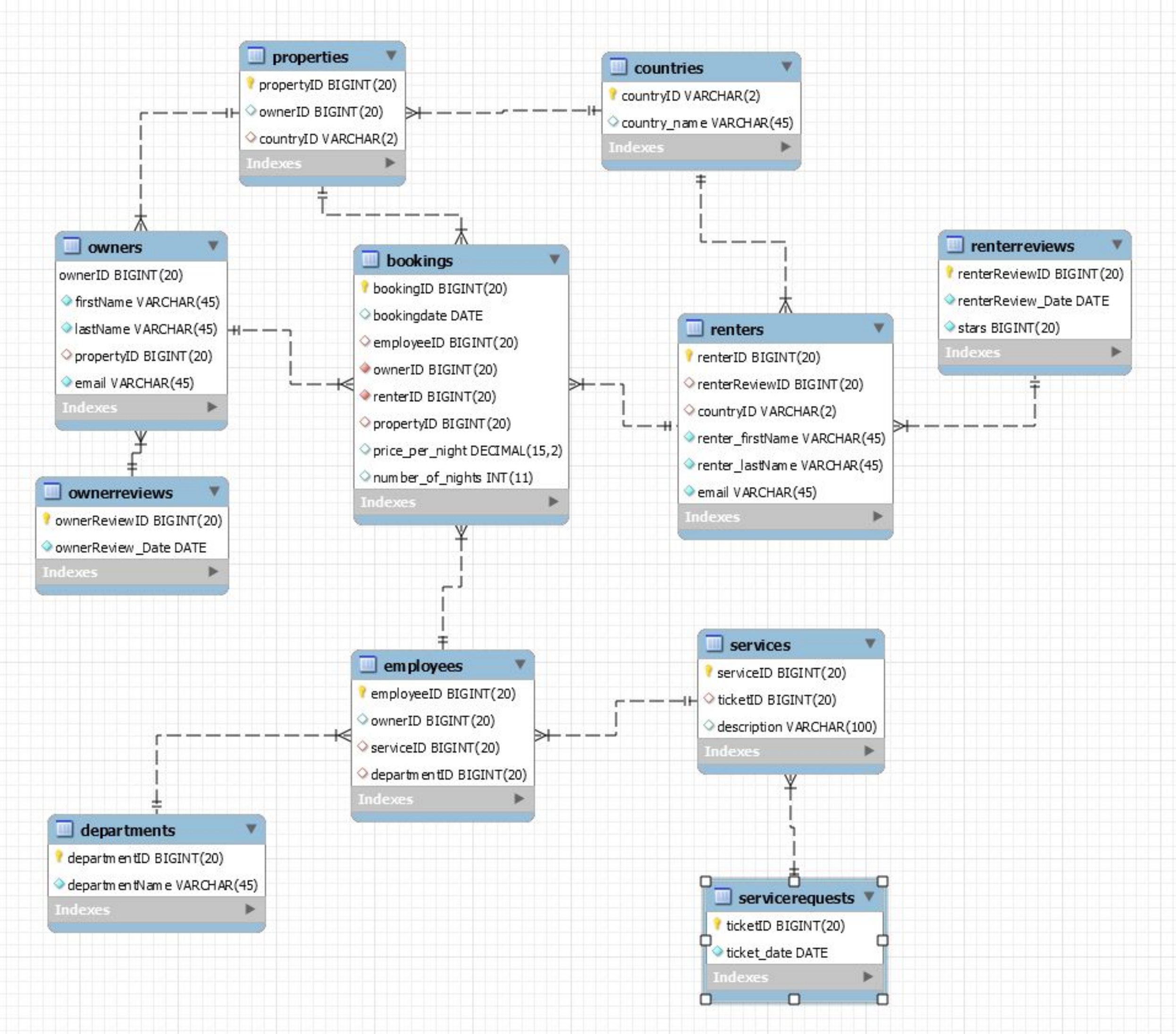
The first step for our group was to create the tables, and organize them in a way that would make the normalization process as easy as possible. We came up with 11 tables, and made sure foreign keys for each table made sense and would run properly. We then used a random data generator called Mockaroo to insert values into our tables. The most important data we needed as ID's. This was because every single table has at least one column with an ID of some sort. For these ID's, we were able to assign each value a number or group of numbers (such as a street address, randomly generated DUNS number, row number, or 10 digit NHS number), which made it easy to avoid duplicates. For each foreign key, we made sure we copied the same values and put them in the correct table to ensure the 'JOIN' function would run properly.

Each table has some kind of id with additional information relating to that id. For example, the renters table has renter id and then information including email, phone number, name, and country. This then connects to the renter reviews and bookings table where they can be joined to include more information about the renter. We did not want to include redundant data in the bookings table, thus we included many tables with foreign keys in the bookings table such that they can be joined when more data is needed about a booking.

We normalized the properties and renters table by creating a separate countries table that can be referenced to find the full name of the country relating to the country code associated with the renter or property id. This creates less data redundancy and the ability to update country information in a single location. We also normalized the renters and owners tables by creating separate reviews tables for each. This allows for multiple reviews of each and to keep less data in the renters and owners tables. While we could have gone to 4NF or 5NF we felt this was the right amount of normalization for our database in order to keep efficiency but also data integrity.

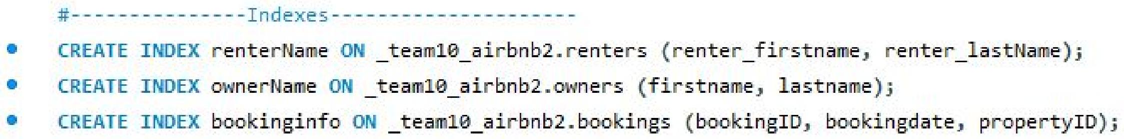
ERD Model

See SQL script for Create Statements



Populate Tables

- See attached file for data export containing insert statements.
- No data transformations were made.
- See attached sql file for indexes.

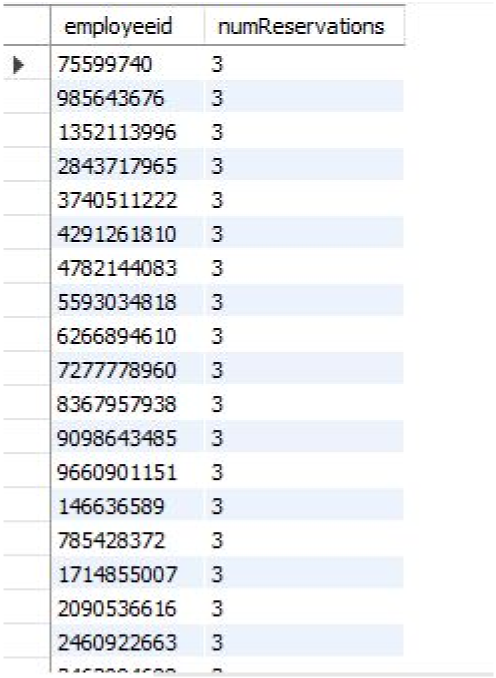


```
#---------------Indexes---------------------
CREATE INDEX renterName ON _team10_airbnb2.renters (renter_firstname, renter_lastName);
CREATE INDEX ownerName ON _team10_airbnb2.owners (firstname, lastname);
CREATE INDEX bookinginfo ON _team10_airbnb2.bookings (bookingID, bookingdate, propertyID);
```

Business Questions

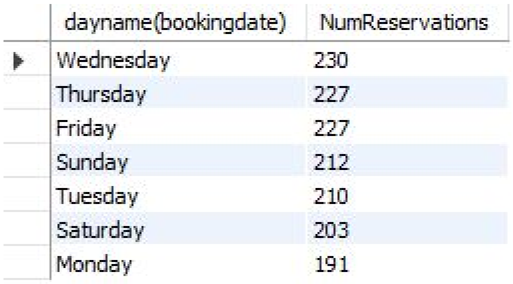
See attached sql file for code.

1. **How many bookings does each employee have?**
    - ○ This allows us to see how employees are performing. As a business we can use this to check if employees are meeting their bookings requirements. In addition we can see who is excelling and who may be struggling.



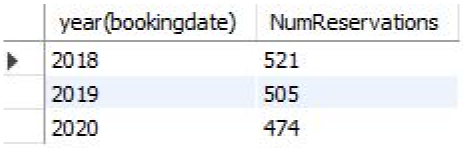
employeeid	numReservations
75599740	3
985643676	3
1352113996	3
2843717965	3
3740511222	3
4291261810	3
4782144083	3
5593034818	3
6266894610	3
7277778960	3
8367957938	3
9098643485	3
9660901151	3
146636589	3
785428372	3
1714855007	3
2090536616	3
2460922663	3

2. **What day of the week has the most bookings?**
    - ○ This allows the business to see what days people are booking their rentals. This allows us to see what days are best to push promotions and other marketing tactics.



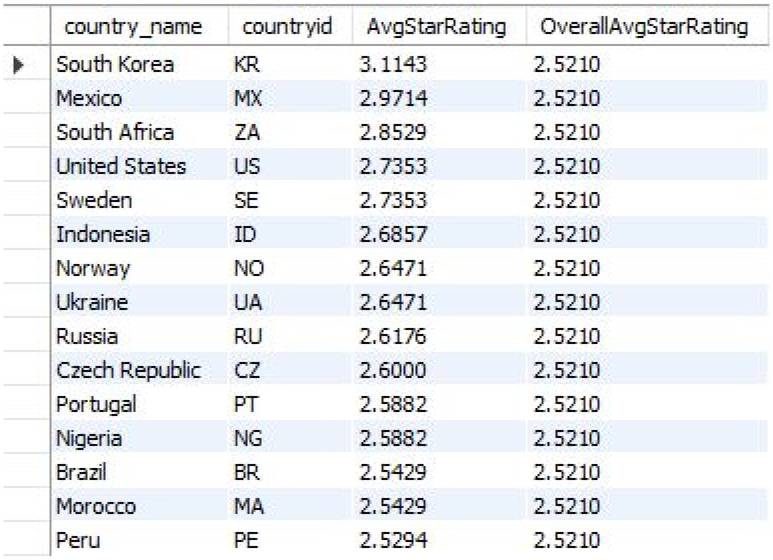
dayname(bookingdate)	NumReservations
Wednesday	230
Thursday	227
Friday	227
Sunday	212
Tuesday	210
Saturday	203
Monday	191

3. **What year has the most bookings?**
    - ○ This allows the business to see year over year trends. You can see that the number of bookings has decreased over the past 2 years.



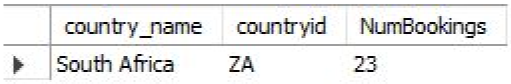
year(bookingdate)	NumReservations
2018	521
2019	505
2020	474

4. **What countries have an average renter star rating higher than the overall average?**
    - ○ This allows the business to see what countries customers like to visit more than the average. We can then look to making further business decisions in these countries such as looking for more properties. It also allows us to see the average rating of properties in these countries.



country_name	countryid	AvgStarRating	OverallAvgStarRating
South Korea	KR	3.1143	2.5210
Mexico	MX	2.9714	2.5210
South Africa	ZA	2.8529	2.5210
United States	US	2.7353	2.5210
Sweden	SE	2.7353	2.5210
Indonesia	ID	2.6857	2.5210
Norway	NO	2.6471	2.5210
Ukraine	UA	2.6471	2.5210
Russia	RU	2.6176	2.5210
Czech Republic	CZ	2.6000	2.5210
Portugal	PT	2.5882	2.5210
Nigeria	NG	2.5882	2.5210
Brazil	BR	2.5429	2.5210
Morocco	MA	2.5429	2.5210
Peru	PE	2.5294	2.5210

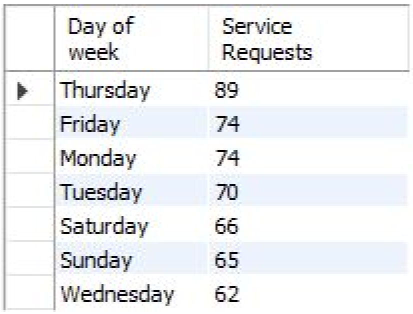
5. **What country is the most commonly booked in 2019?**
    - ○ This allows the business to see the most popular country visited by customers.



country_name	countryid	NumBookings
South Africa	ZA	23

6. Which day of the week had the most service requests(calculated)?

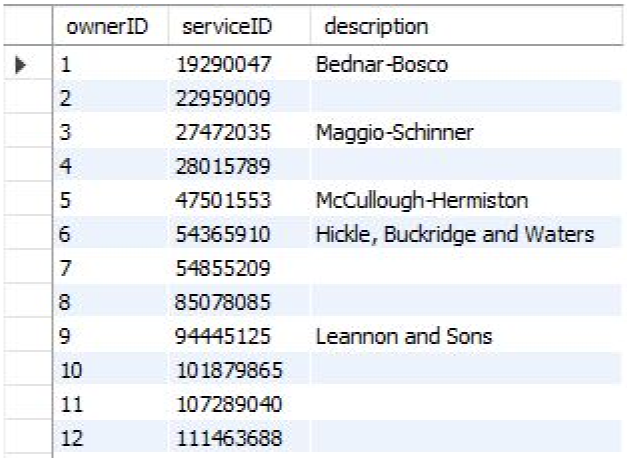
- This query allows us to see the most popular days that service requests were submitted. If there was a trend, we could better prepare for incoming requests and ensure we had the capacity to fill them.



	Day of week	Service Requests
▶	Thursday	89
	Friday	74
	Monday	74
	Tuesday	70
	Saturday	66
	Sunday	65
	Wednesday	62

7. What owners had a serviceID and list the description and date(join)?

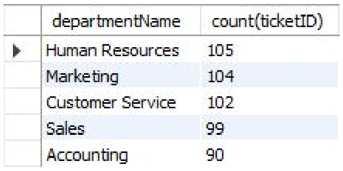
- This allows us to check service requests and which company is hired to fix the issue. This would be important when customer service receives an issue from an owner about the company that is fixing their property. The date is also listed to ensure the request is being completed within a reasonable amount of time.



	ownerID	serviceID	description
▶	1	19290047	Bednar-Bosco
	2	22959009	
	3	27472035	Maggio-Schinner
	4	28015789	
	5	47501553	McCullough-Hermiston
	6	54365910	Hickle, Buckridge and Waters
	7	54855209	
	8	85078085	
	9	94445125	Leannon and Sons
	10	101879865	
	11	107289040	
	12	111463688	

8. How many service requests did each department have(join)?

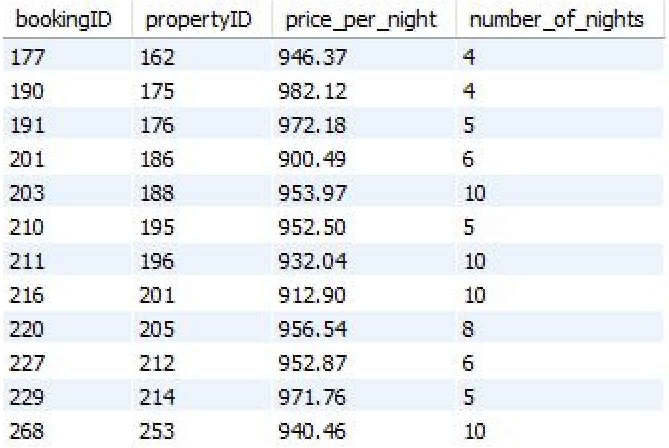
- This query allows us to see if a certain department needs for resources to complete service requests. Also, if one department had a lot of requests, our business may need to find a solution to fix this.



	departmentName	count(ticketID)
▶	Human Resources	105
	Marketing	104
	Customer Service	102
	Sales	99
	Accounting	90

9. How many bookings had a price greater than $900 and rented for more than 3 nights (where)?

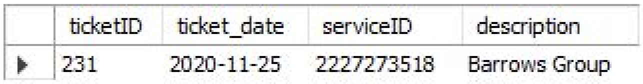
- These properties are considered to be high-end for our business. We could provide this list of properties to our customer service team to make sure they are always in good condition and our guests have a comfortable stay



bookingID	propertyID	price_per_night	number_of_nights
177	162	946.37	4
190	175	982.12	4
191	176	972.18	5
201	186	900.49	6
203	188	953.97	10
210	195	952.50	5
211	196	932.04	10
216	201	912.90	10
220	205	956.54	8
227	212	952.87	6
229	214	971.76	5
268	253	940.46	10

10. What is the most recent service request?

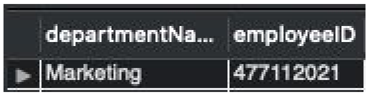
- This allows us to see the information on the most recent service request. This can be used to make sure we are up to date on service requests and can fullfill them promptly.



	ticketID	ticket_date	serviceID	description
▶	231	2020-11-25	2227273518	Barrows Group

11. Which department does the employee with ID 477112021 work for?

○ This highly specific query is useful when seeking out specific employees' department, salary, or other things associated with that employee's background.

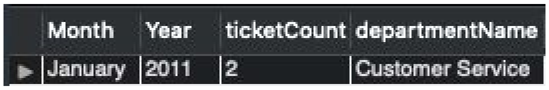


	departmentNa...	employeeID
▶	Marketing	477112021

12. How many support tickets were submitted in the month of January, 2011 for the Customer Service department?

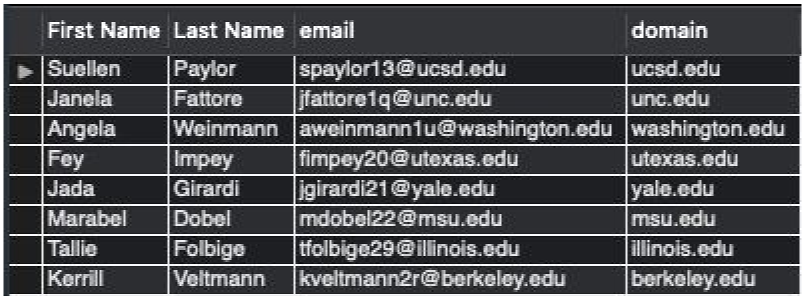
- Support tickets are representative of issues coming from customers and the Customer Service department needs to be able to keep track of a monthly count. Pulling tickets from specific years/months is key to knowing a business.

○



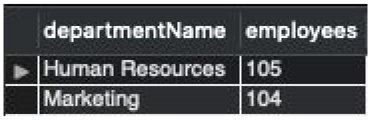
	Month	Year	ticketCount	departmentName
▶	January	2011	2	Customer Service

13. Which renters(s) have email addresses associated with universities (.edu)?

- This interesting query can help the marketing department advertise promotions to students/universities. Additionally, it can help all customer focused departments learn the location/affiliations of their customers through the universities they are employed or study at.
  - 

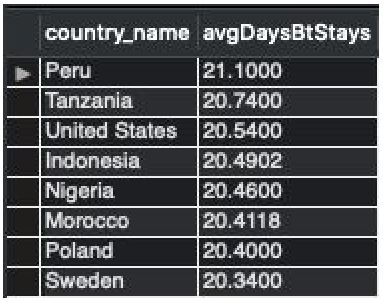
First Name	Last Name	email	domain
Suellen	Paylor	spaylor13@ucsd.edu	ucsd.edu
Janela	Fattore	jfattore1q@unc.edu	unc.edu
Angela	Weinmann	aweinmann1u@washington.edu	washington.edu
Fey	Impey	fimpey20@utexas.edu	utexas.edu
Jada	Girardi	jgirardi21@yale.edu	yale.edu
Marabel	Dobel	mdobel22@msu.edu	msu.edu
Tallie	Folbige	tfolbige29@illinois.edu	illinois.edu
Kerrill	Veltmann	kveltmann2r@berkeley.edu	berkeley.edu

14. Which departments have the most employees? Rank each department by number of employees. Only include the top two.

- This query provides insight into the company, and can help to quickly procure things like payroll. It's important to know the size of the company.
  - 

departmentName	employees
Human Resources	105
Marketing	104

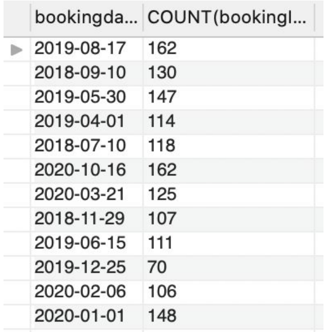
15. What is the average length time between stays in each country?

- This query provides useful statistics to analysts who are measuring the performance of the market of each country. Increased time between stays can represent that certain regions are having trouble finding clients. It also provides insight into which countries are more popular, and can be used in geographic mapping, etc.
  - 

country_name	avgDaysBtStays
Peru	21.1000
Tanzania	20.7400
United States	20.5400
Indonesia	20.4902
Nigeria	20.4600
Morocco	20.4118
Poland	20.4000
Sweden	20.3400

16. Which Month is the most popular for bookings (group by)?

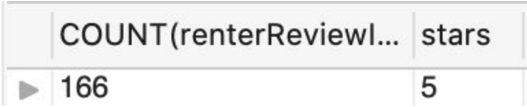
- This allows airbnb to see which months they see the most business. This can be important for seasonal bookings, and traffic issues.



bookingda...	COUNT(bookingI...
2019-08-17	162
2018-09-10	130
2019-05-30	147
2019-04-01	114
2018-07-10	118
2020-10-16	162
2020-03-21	125
2018-11-29	107
2019-06-15	111
2019-12-25	70
2020-02-06	106
2020-01-01	148

17. How many 5 star ratings are there in the database (where)?

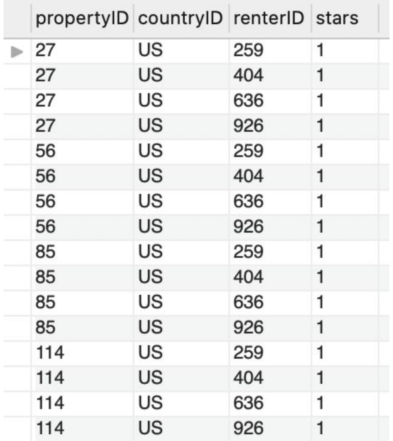
- This can allow the company to assess their best properties. When comparing themselves to other competitors they can use this information to boast about high valued properties that they have to offer.



COUNT(renterReviewI...	stars
166	5

18. Which properties in the US have 1 star ratings (join)?

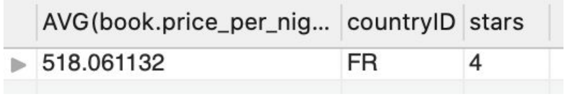
- This query lets airbnb see properties in their database in the US that could use improvements. Bad properties could affect brand image. These properties could be dropped also.



propertyID	countryID	renterID	stars
27	US	259	1
27	US	404	1
27	US	636	1
27	US	926	1
56	US	259	1
56	US	404	1
56	US	636	1
56	US	926	1
85	US	259	1
85	US	404	1
85	US	636	1
85	US	926	1
114	US	259	1
114	US	404	1
114	US	636	1
114	US	926	1

19. What is the average price per night of a property with 4 stars in France (join)?

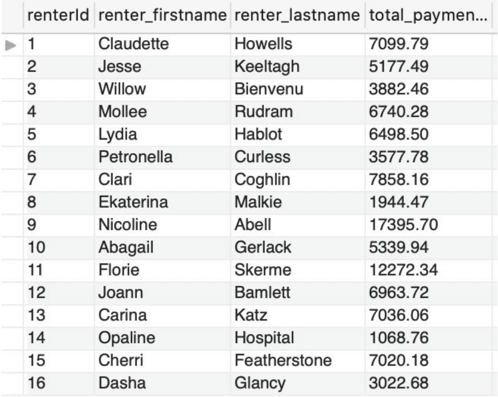
- This specific query can help answer a simple question. Customers wanting to travel to France want to know what they are going to be spending on a nice but not the nicest airbnb.



AVG(book.price_per_nig...	countryID	stars
518.061132	FR	4

20. How much has each renter spent (subquery)?

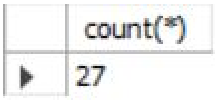
- This allows airbnb to see their top spenders, and perhaps offer them promotions.



	renterId	renter_firstname	renter_lastname	total_paymen...
▶	1	Claudette	Howells	7099.79
	2	Jesse	Keeltagh	5177.49
	3	Willow	Bienvenu	3882.46
	4	Mollee	Rudram	6740.28
	5	Lydia	Hablot	6498.50
	6	Petronella	Curless	3577.78
	7	Clari	Coghlin	7858.16
	8	Ekaterina	Malkie	1944.47
	9	Nicoline	Abell	17395.70
	10	Abagail	Gerlack	5339.94
	11	Florie	Skerme	12272.34
	12	Joann	Bamlett	6963.72
	13	Carina	Katz	7036.06
	14	Opaline	Hospital	1068.76
	15	Cherri	Featherstone	7020.18
	16	Dasha	Glancy	3022.68

21. How many properties are there in Brazil? (Where countryID is BR).

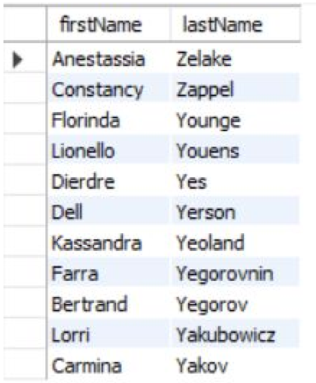
- ○ This query can be used to find the total number of properties in a given region. It can also be used to group or sort regions based on the number of properties found using this query.



	count(*)
▶	27

22. Who are the owners with a last name starting with Z or Y?

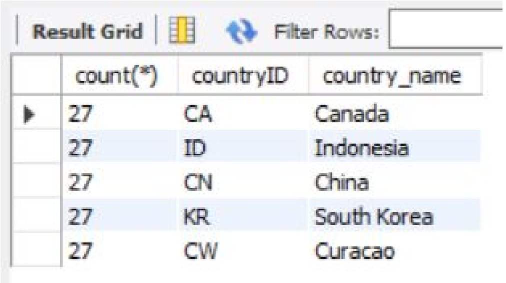
- This query would probably not be used that frequently due to its randomness, but could be useful in cases where there is a massive dataset, and you only want to find customers with a specific start to their last or first name.



	firstName	lastName
▶	Anestassia	Zelake
	Constancy	Zappel
	Florinda	Younge
	Lionello	Youens
	Dierdre	Yes
	Dell	Yerson
	Kassandra	Yeoland
	Farra	Yegorovnin
	Bertrand	Yegorov
	Lorri	Yakubowicz
	Carmina	Yakov

23. What are the top 5 countries (countryID and country name) with the most properties?

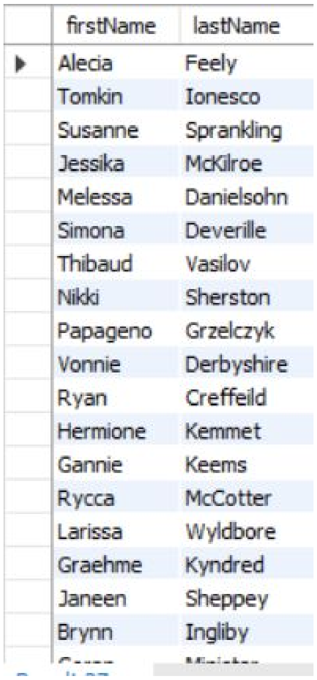
- This query provides a list of the top 5 countries that have the most amount of bookings. This query is very useful when organizing/analyzing data to get statistics on the min/max number of bookings per country.



count(*)	countryID	country_name
27	CA	Canada
27	ID	Indonesia
27	CN	China
27	KR	South Korea
27	CW	Curacao

24. What are the owners' names whose country name is Czech Republic? (join properties, owners

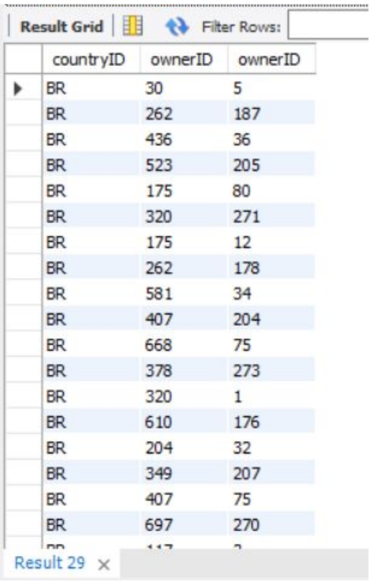
- This query takes into account the joining of three tables together. This is useful if you did not know the countryID for the Czech Republic and wanted to filter based on countryName.



firstName	lastName
Alecia	Feely
Tomkin	Ionesco
Susanne	Sprankling
Jessika	McKilroe
Melessa	Danielsohn
Simona	Deverille
Thibaud	Vasilov
Nikki	Sherston
Papageno	Grzelczyk
Vonnie	Derbyshire
Ryan	Creffeild
Hermione	Kemmet
Gannie	Keems
Rycca	McCotter
Larissa	Wyldbore
Graehme	Kyndred
Janeen	Sheppey
Brynn	Ingliby
[illegible]	[illegible]

25. Find the owners (ownerID) who are located in the same country (countryID).

- This self join compares/groups different owners who have properties in the same country. It is useful in organizing the data and comparing the owners' locations.



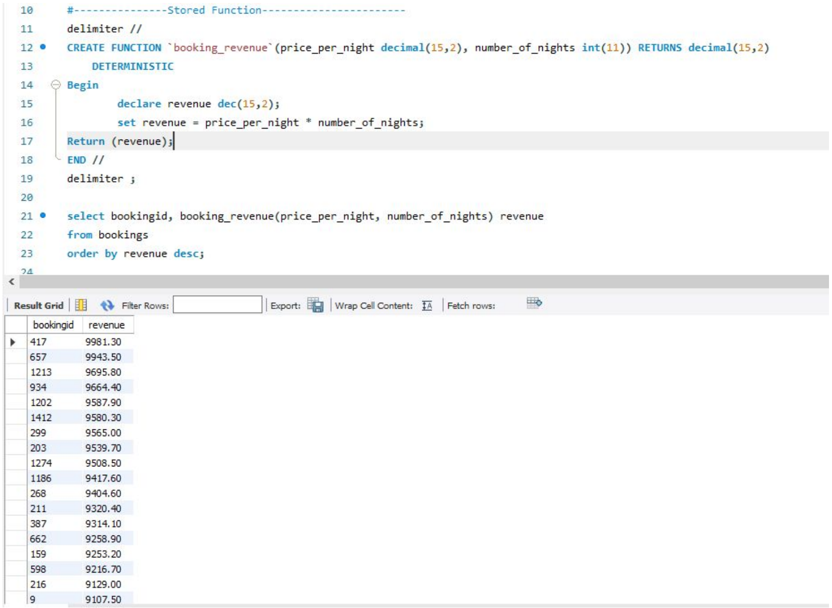
countryID	ownerID	ownerID
BR	30	5
BR	262	187
BR	436	36
BR	523	205
BR	175	80
BR	320	271
BR	175	12
BR	262	178
BR	581	34
BR	407	204
BR	668	75
BR	378	273
BR	320	1
BR	610	176
BR	204	32
BR	349	207
BR	407	75
BR	697	270

Stored Programs

See attached file for code

- 2 Stored Functions

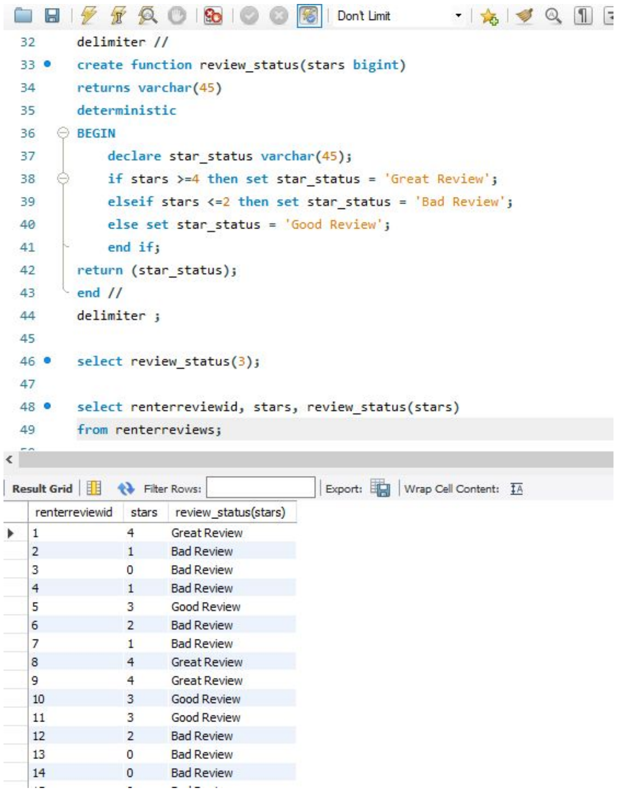
This function calculates the revenue generated by a booking.



```
#--------------Stored Function----------------------
delimiter //
CREATE FUNCTION `booking_revenue`(price_per_night decimal(15,2), number_of_nights int(11)) RETURNS decimal(15,2)
    DETERMINISTIC
Begin
        declare revenue dec(15,2);
        set revenue = price_per_night * number_of_nights;
Return (revenue);
END //
delimiter ;

select bookingid, booking_revenue(price_per_night, number_of_nights) revenue
from bookings
order by revenue desc;
```

bookingid	revenue
417	9981.30
657	9943.50
1213	9695.80
934	9664.40
1202	9587.90
1412	9580.30
299	9565.00
203	9539.70
1274	9508.50
1186	9417.60
268	9404.60
211	9320.40
387	9314.10
662	9258.90
159	9253.20
598	9216.70
216	9129.00
9	9107.50

This function assesses the star review for renters. A star review of 0-2 is “Bad”, 3 is “Good” and Greater than 3 is “Great”.

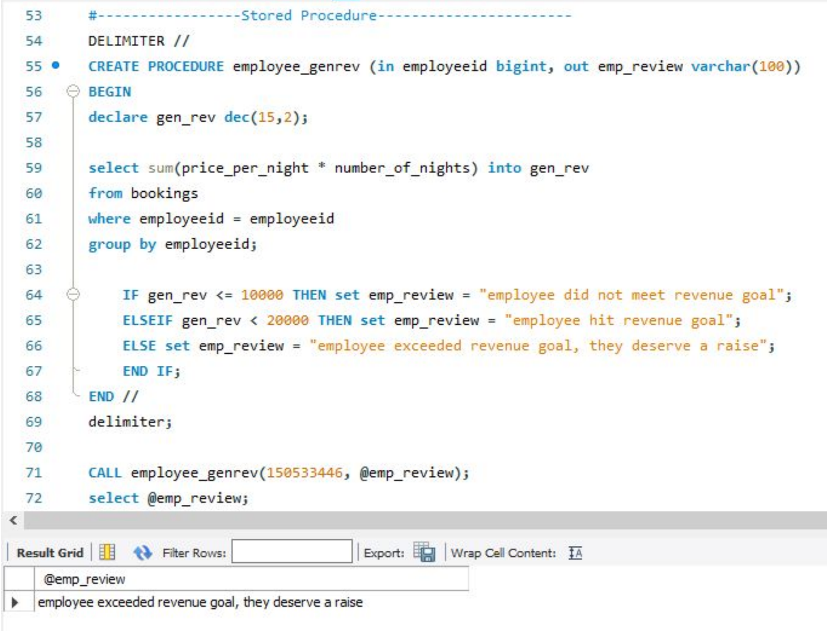


```
delimiter //
create function review_status(stars bigint)
returns varchar(45)
deterministic
BEGIN
    declare star_status varchar(45);
    if stars >=4 then set star_status = 'Great Review';
    elseif stars <=2 then set star_status = 'Bad Review';
    else set star_status = 'Good Review';
    end if;
return (star_status);
end //
delimiter ;

select review_status(3);

select renterreviewid, stars, review_status(stars)
from renterreviews;
```

renterreviewid	stars	review_status(stars)
1	4	Great Review
2	1	Bad Review
3	0	Bad Review
4	1	Bad Review
5	3	Good Review
6	2	Bad Review
7	1	Bad Review
8	4	Great Review
9	4	Great Review
10	3	Good Review
11	3	Good Review
12	2	Bad Review
13	0	Bad Review
14	0	Bad Review

- 1 Stored Procedure

This stored procedure tells you if an employee met their revenue goal.

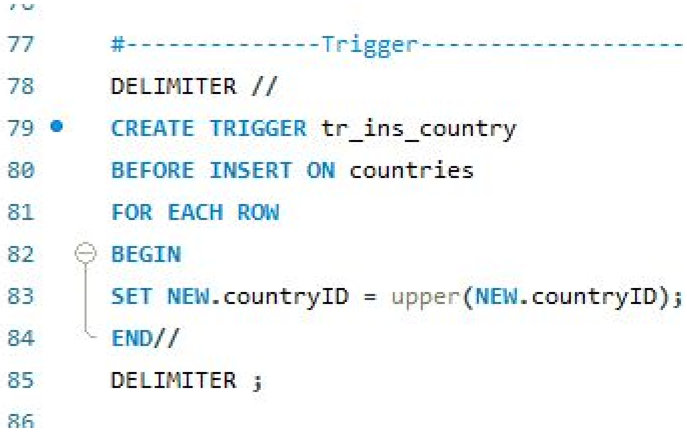


```
#----------------Stored Procedure---------------------
DELIMITER //
CREATE PROCEDURE employee_genrev (in employeeid bigint, out emp_review varchar(100))
BEGIN
declare gen_rev dec(15,2);

select sum(price_per_night * number_of_nights) into gen_rev
from bookings
where employeeid = employeeid
group by employeeid;

    IF gen_rev <= 10000 THEN set emp_review = "employee did not meet revenue goal";
    ELSEIF gen_rev < 20000 THEN set emp_review = "employee hit revenue goal";
    ELSE set emp_review = "employee exceeded revenue goal, they deserve a raise";
    END IF;
END //
delimiter;

CALL employee_genrev(150533446, @emp_review);
select @emp_review;
```

```
@emp_review
employee exceeded revenue goal, they deserve a raise
```

- 1 Trigger
  - See sql script for code.

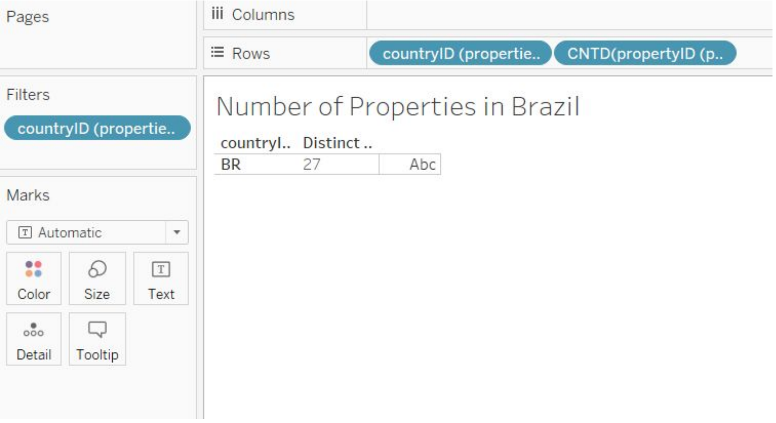
This Trigger makes sure a new country code added to the country table is in uppercase.



```
#-------------Trigger------------------
DELIMITER //
CREATE TRIGGER tr_ins_country
BEFORE INSERT ON countries
FOR EACH ROW
BEGIN
SET NEW.countryID = upper(NEW.countryID);
END//
DELIMITER ;
```

Visualizations

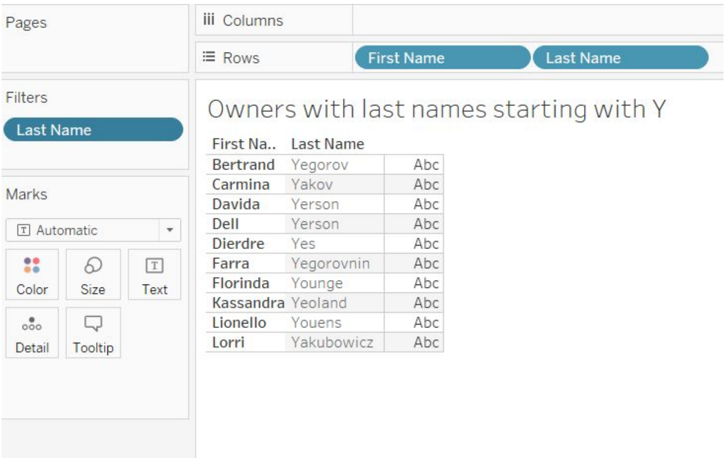
1. **Find the number of properties in Brazil.**
    - This was done by grabbing countryID, and a discrete count of propertyIDs, filtering on countryID that equaled ‘BR’ (Brazil). No significance in doing it in Tableau vs SQL as both provide a total number.



Number of Properties in Brazil

countryI..	Distinct ..	
BR	27	Abc

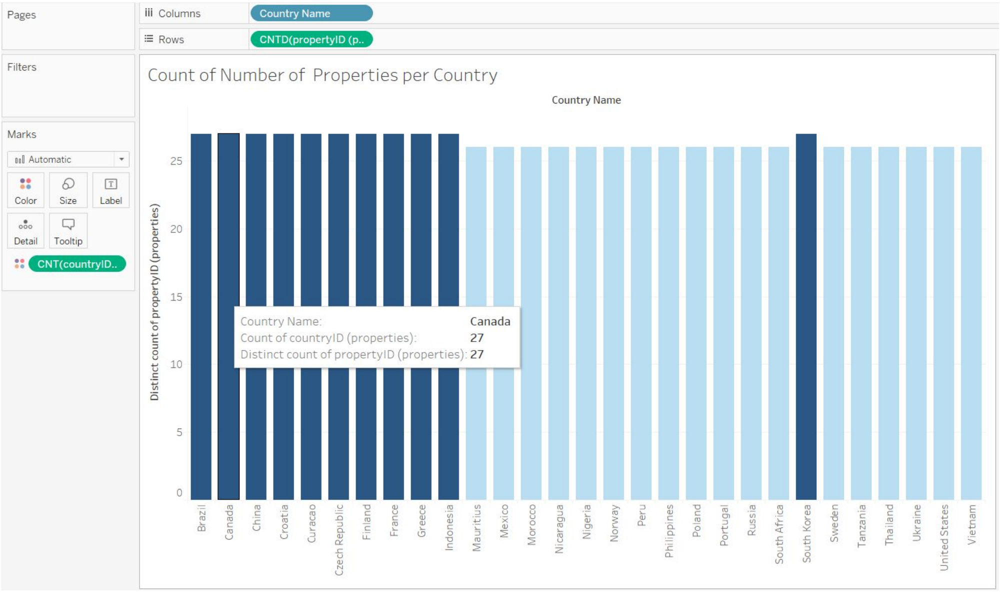
2. **Who are the owners with a last name starting with Y?**
    - This was done by grabbing the first and last names of owners, and then filtering on the last name. To do a special filter, I clicked on the right arrow on last name under filters, clicking edit filter, then choosing the wildcard starts with value ‘y’.



Owners with last names starting with Y

First Na..	Last Name	
Bertrand	Yegorov	Abc
Carmina	Yakov	Abc
Davida	Yerson	Abc
Dell	Yerson	Abc
Dierdre	Yes	Abc
Farra	Yegorovnin	Abc
Florinda	Younge	Abc
Kassandra	Yeoland	Abc
Lionello	Youens	Abc
Lorri	Yakubowicz	Abc

3. **Graph the countries based on the number of properties in each country.**
    - ○ To do this, I did the count of propertyIDs, and grouped by country name. I then clicked on the show me function at the top right, and it provided me with a graph to better visualize the number of properties per country. Done in Tableau to better visualize the data, although for this one in particular they all have either 26 or 27 properties so it does not look the best.



Columns: Country Name

Rows: CNTD(propertyID (p..

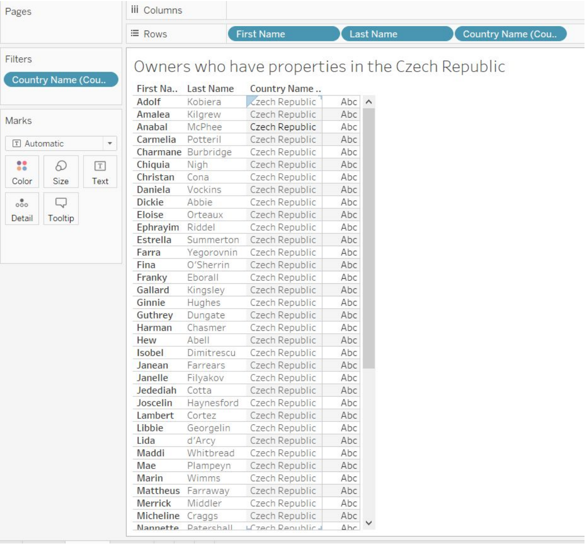
Marks: Automatic

CNT(countryID..

Count of Number of Properties per Country

Country Name: Canada
Count of countryID (properties): 27
Distinct count of propertyID (properties): 27

4. **What are the owners' names whose country name is Czech Republic? (join properties, owners?**
   - This was completed by grabbing first name, last name, and country name while filtering on country name to include only those that are ‘Czech Republic’ by clicking on the arrow on country name under filters and checking the box that says Czech Republic. No significance in doing this on Tableau rather than SQL in terms of appearance.



Columns

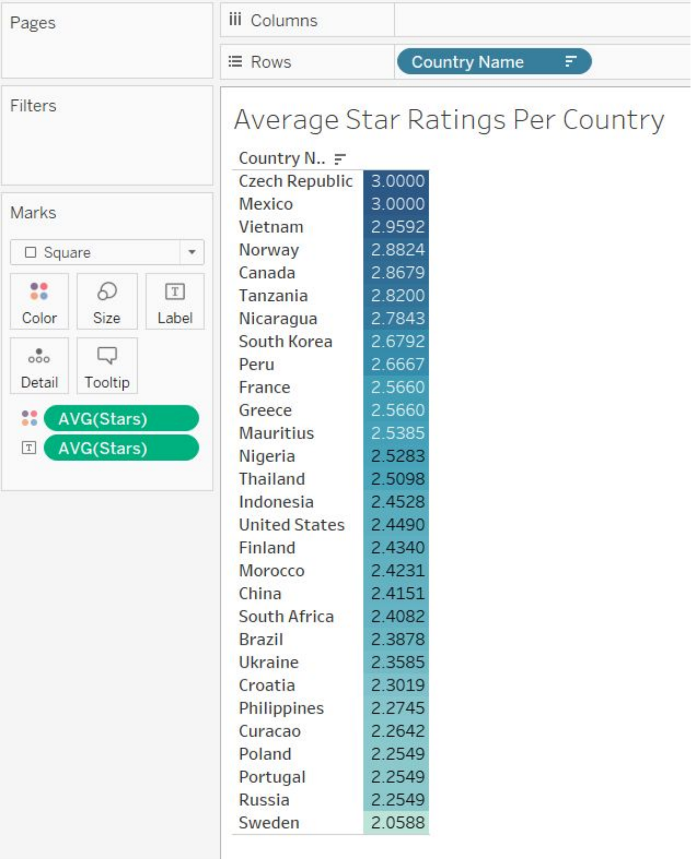
Rows: First Name | Last Name | Country Name (Cou..

Filters: Country Name (Cou..

Owners who have properties in the Czech Republic

First Na..	Last Name	Country Name ..	
Adolf	Kobiera	Czech Republic	Abc
Amalea	Kilgrew	Czech Republic	Abc
Anabal	McPhee	Czech Republic	Abc
Carmelia	Potteril	Czech Republic	Abc
Charmane	Burbridge	Czech Republic	Abc
Chiquia	Nigh	Czech Republic	Abc
Christan	Cona	Czech Republic	Abc
Daniela	Vockins	Czech Republic	Abc
Dickie	Abbie	Czech Republic	Abc
Eloise	Orteaux	Czech Republic	Abc
Ephrayim	Riddel	Czech Republic	Abc
Estrella	Summerton	Czech Republic	Abc
Farra	Yegorovnin	Czech Republic	Abc
Fina	O’Sherrin	Czech Republic	Abc
Franky	Eborall	Czech Republic	Abc
Gallard	Kingsley	Czech Republic	Abc
Ginnie	Hughes	Czech Republic	Abc
Guthrey	Dungate	Czech Republic	Abc
Harman	Chasmer	Czech Republic	Abc
Hew	Abell	Czech Republic	Abc
Isobel	Dimitrescu	Czech Republic	Abc
Janean	Farrears	Czech Republic	Abc
Janelle	Filyakov	Czech Republic	Abc
Jedediah	Cotta	Czech Republic	Abc
Joscelin	Haynesford	Czech Republic	Abc
Lambert	Cortez	Czech Republic	Abc
Libbie	Georgelin	Czech Republic	Abc
Lida	d’Arcy	Czech Republic	Abc
Maddi	Whitbread	Czech Republic	Abc
Mae	Plampeyn	Czech Republic	Abc
Marin	Wimms	Czech Republic	Abc
Mattheus	Farraway	Czech Republic	Abc
Merrick	Middler	Czech Republic	Abc
Micheline	Craggs	Czech Republic	Abc
Nannette	Patershall	Czech Republic	Abc

5. **Show the average number of stars based on rating per country.**
   - This is very easy to read/understand and is extremely helpful in understanding the data. To do this, I grouped by country name, and then grabbed the stars (making sure to filter and choose average) from the renter reviews table.



Columns

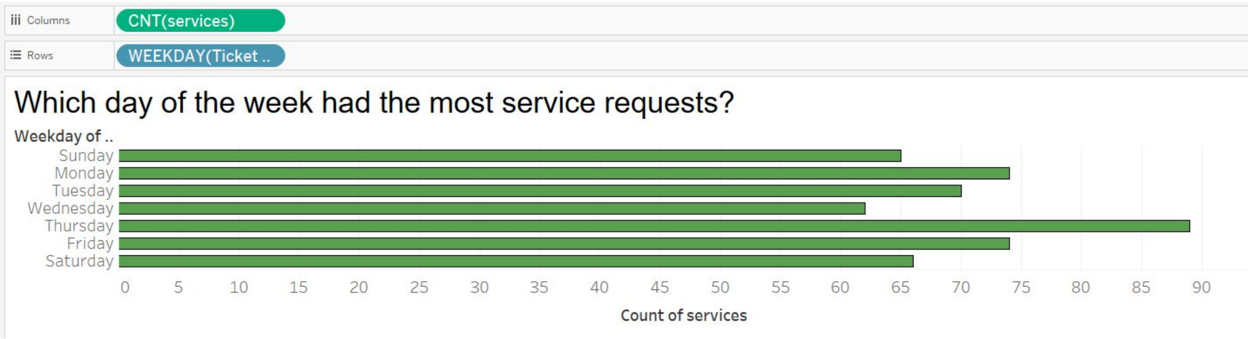
Rows: Country Name

Marks: AVG(Stars) | AVG(Stars)

Average Star Ratings Per Country

Country N..	
Czech Republic	3.0000
Mexico	3.0000
Vietnam	2.9592
Norway	2.8824
Canada	2.8679
Tanzania	2.8200
Nicaragua	2.7843
South Korea	2.6792
Peru	2.6667
France	2.5660
Greece	2.5660
Mauritius	2.5385
Nigeria	2.5283
Thailand	2.5098
Indonesia	2.4528
United States	2.4490
Finland	2.4340
Morocco	2.4231
China	2.4151
South Africa	2.4082
Brazil	2.3878
Ukraine	2.3585
Croatia	2.3019
Philippines	2.2745
Curacao	2.2642
Poland	2.2549
Portugal	2.2549
Russia	2.2549
Sweden	2.0588

6. **Which day of the week had the most service requests?**
   - This bar chart shows the breakdown of how many service requests there were per weekday. This is useful to see if there is a certain day that had an abnormal amount of servicing needs. To do this, I filtered the ticketDate as a weekday and joined it with a count of service ids.

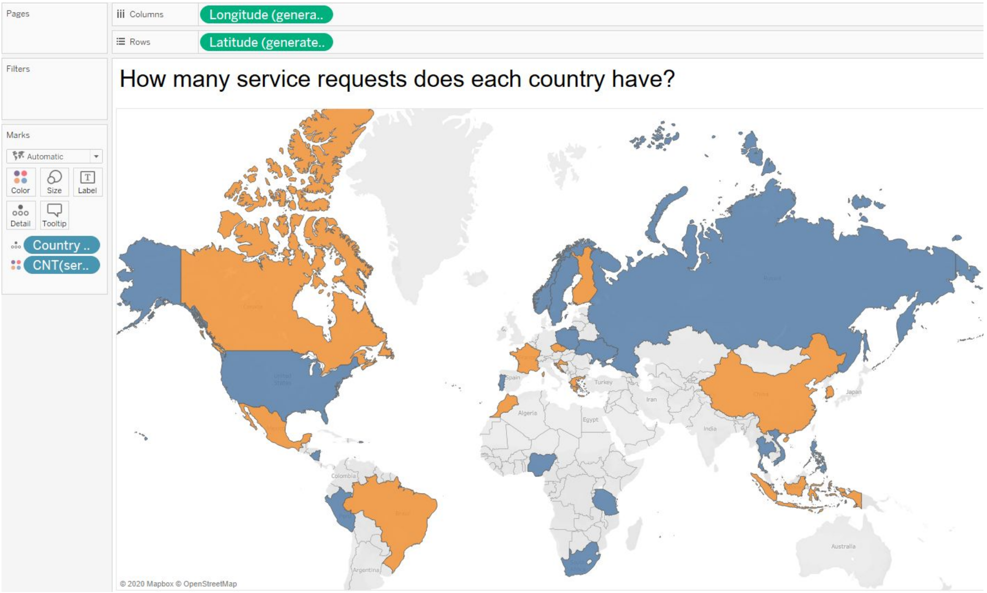


Columns CNT(services)

Rows WEEKDAY(Ticket ..

7. How many service requests does each country have?

- This map shows the number of service requests per country to see if a certain country is having more issues than others. I did this by joining the bookings and service requests table and filtering by each country. The number of requests is portrayed by using the count function.



Columns Longitude (genera..

Rows Latitude (generate..

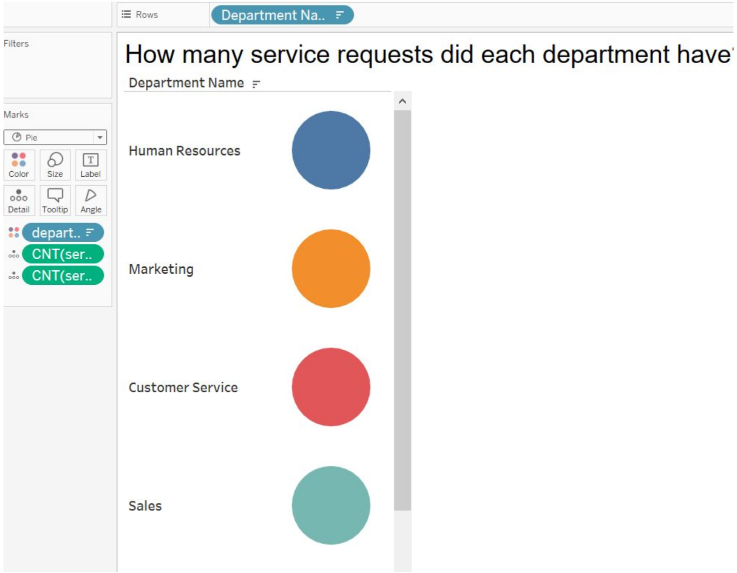
Marks Automatic

Country ..

CNT(ser..

8. How many service requests did each department have?

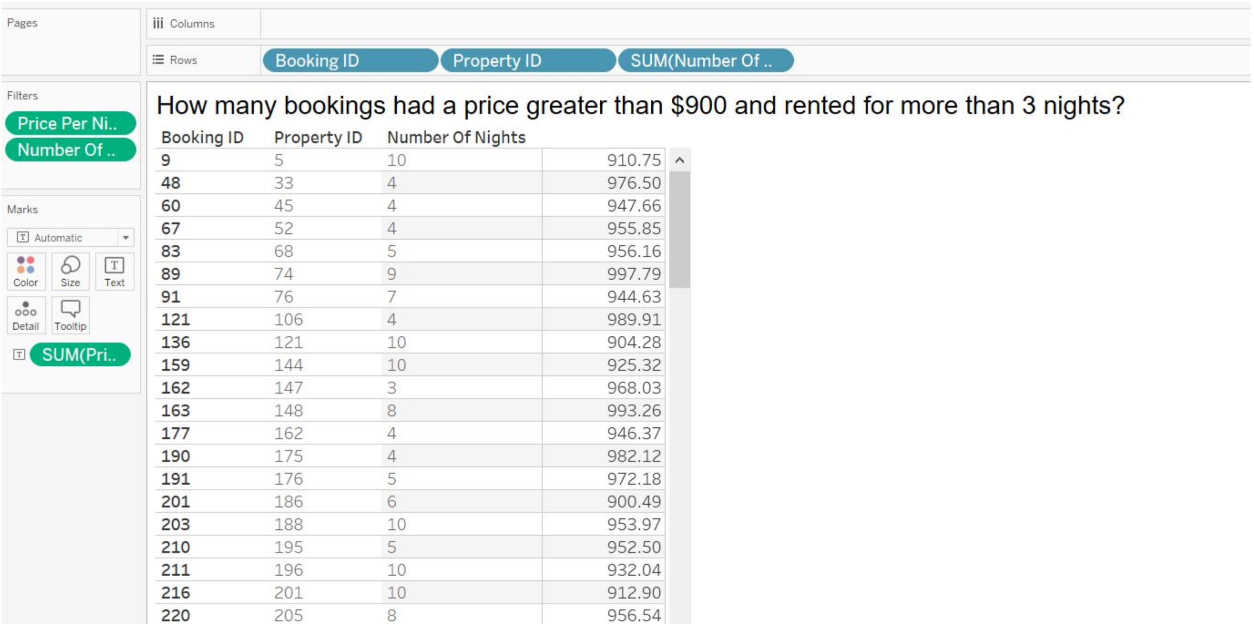
- This pie chart sums the count of service requests by each department using a join between employees and service request tables. This is useful because we can easily see which departments are receiving the most requests.



How many service requests did each department have'

9. Which bookings had a price greater than $900 and a duration of stay greater than 3 nights?

- We can use this table to show our higher-end bookings to make sure we focus our efforts to excellent customer service for these properties. This was created by simply filtering the price and number of nights by $900 and 3 nights.

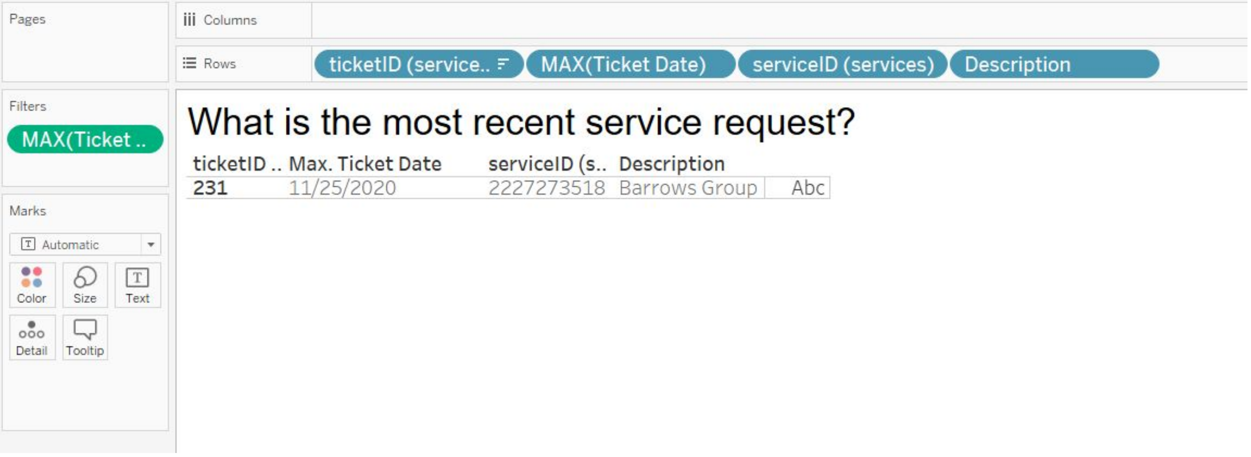


How many bookings had a price greater than $900 and rented for more than 3 nights?

Booking ID	Property ID	Number Of Nights	
9	5	10	910.75
48	33	4	976.50
60	45	4	947.66
67	52	4	955.85
83	68	5	956.16
89	74	9	997.79
91	76	7	944.63
121	106	4	989.91
136	121	10	904.28
159	144	10	925.32
162	147	3	968.03
163	148	8	993.26
177	162	4	946.37
190	175	4	982.12
191	176	5	972.18
201	186	6	900.49
203	188	10	953.97
210	195	5	952.50
211	196	10	932.04
216	201	10	912.90
220	205	8	956.54

10. What is the most recent service request?

- This table can be used to ensure we are taking care of incoming service requests in a timely manner. This was created by joining services and service requests and then filtering by the max date.



Columns

Rows: ticketID (service.. | MAX(Ticket Date) | serviceID (services) | Description

Filters: MAX(Ticket ..

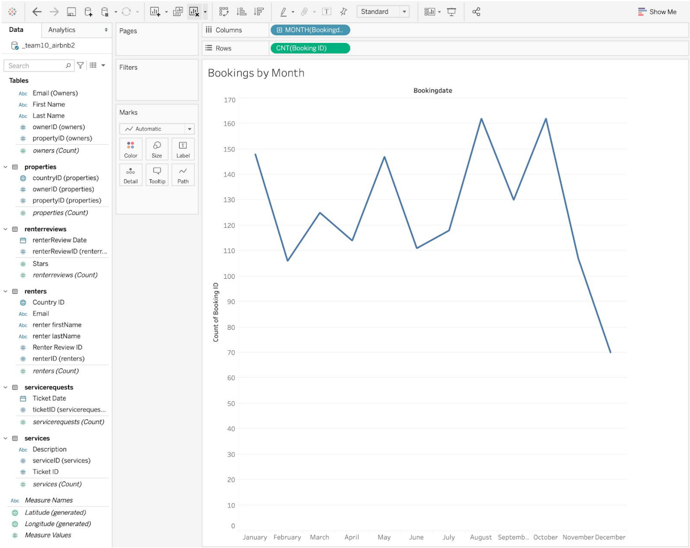
Marks: Automatic

What is the most recent service request?

ticketID ..	Max. Ticket Date	serviceID (s..	Description	
231	11/25/2020	2227273518	Barrows Group	Abc

11. What months are most popular for bookings?

- ○ This graph shows us a clear visual of bookings per month. This can show us where we are seeing the most bookings, and at what time of year. This could be beneficial for seasonal owners/renters. This was created by counting the number of bookings and specifying month on booking date.



Columns: MONTH(Bookingd..

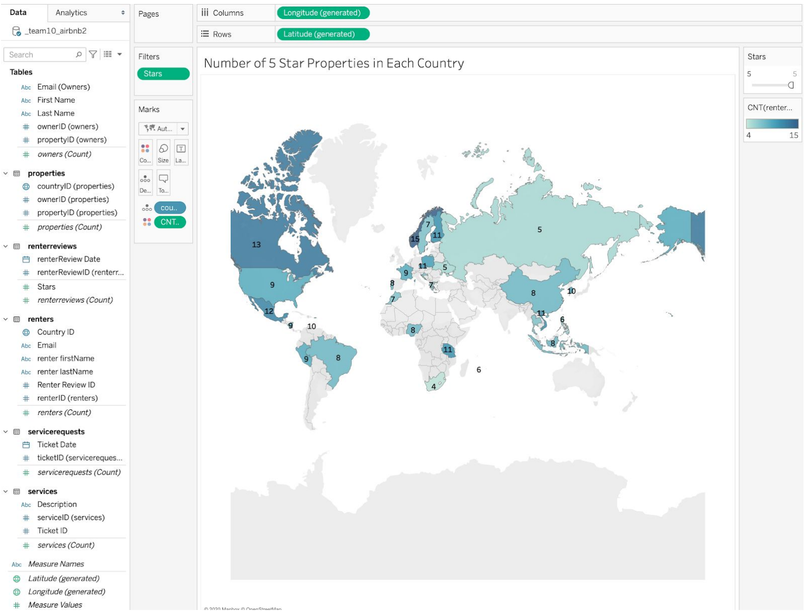
Rows: CNT(Booking ID)

Marks: Automatic

Data source: _team10_airbnb2

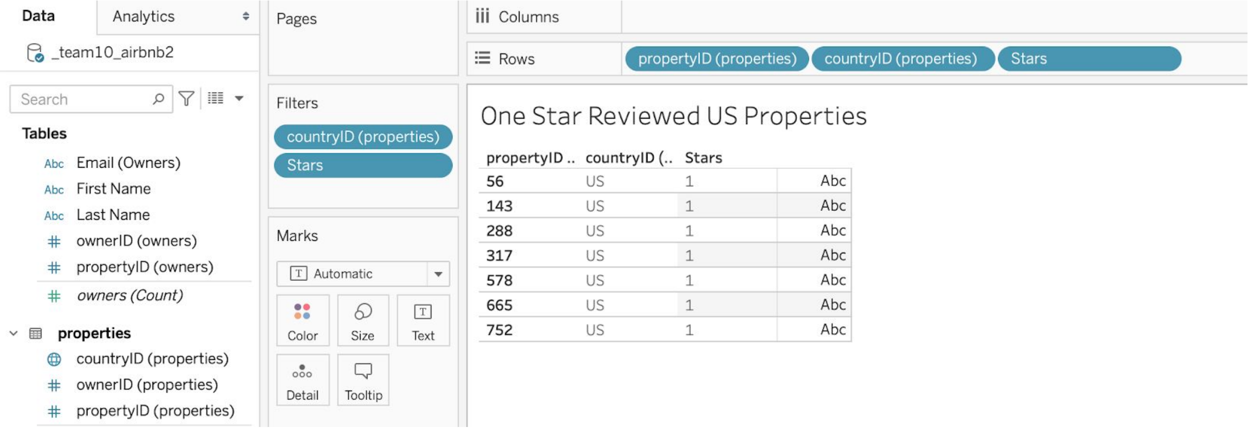
12. How many 5 star properties are there in each country?

- This symbol map shows us how many top notch properties there are across the globe. This was created by filtering stars to be at least 5.


Number of 5 Star Properties in Each Country

13. How many 1 star properties are there in the US?

- This gives us a list of properties in the US with one star reviews. This can guide us to maybe encourage the owners of these properties to spruce up their place… or simply help cut dead weight. Created by filtering US, and by a max rating of 1 star.

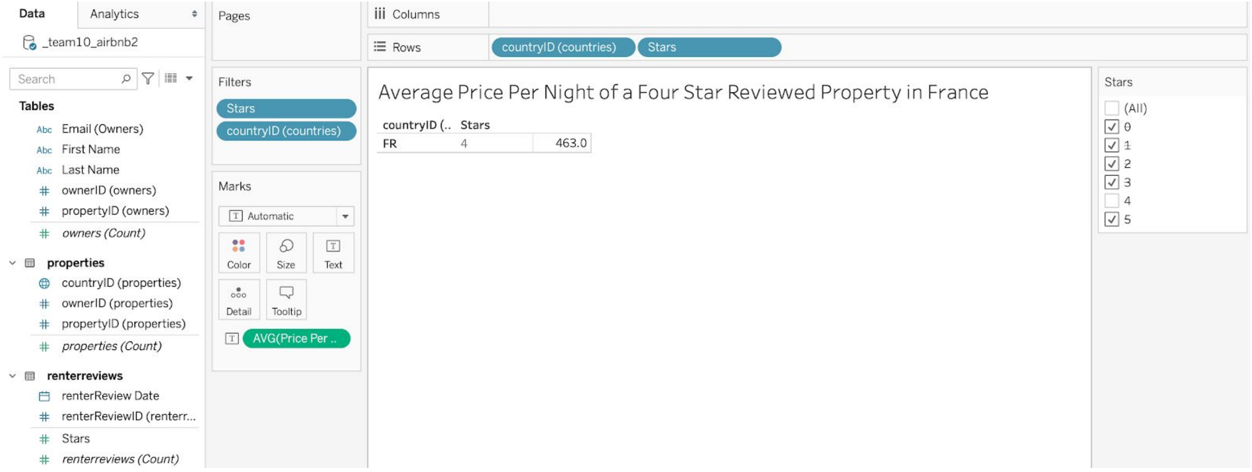


One Star Reviewed US Properties

propertyID ..	countryID (..	Stars	
56	US	1	Abc
143	US	1	Abc
288	US	1	Abc
317	US	1	Abc
578	US	1	Abc
665	US	1	Abc
752	US	1	Abc

14. What is the average price of a 4 star reviewed place in France?

- This is a very specific visualization. This shows us what the usual going rate for a nice place, not the best place in France. Created by filtering stars to only show 4 star reviews, and by using the average measure for price per night.



Columns

Rows: countryID (countries), Stars

Filters: Stars, countryID (countries)

Marks: AVG(Price Per ..

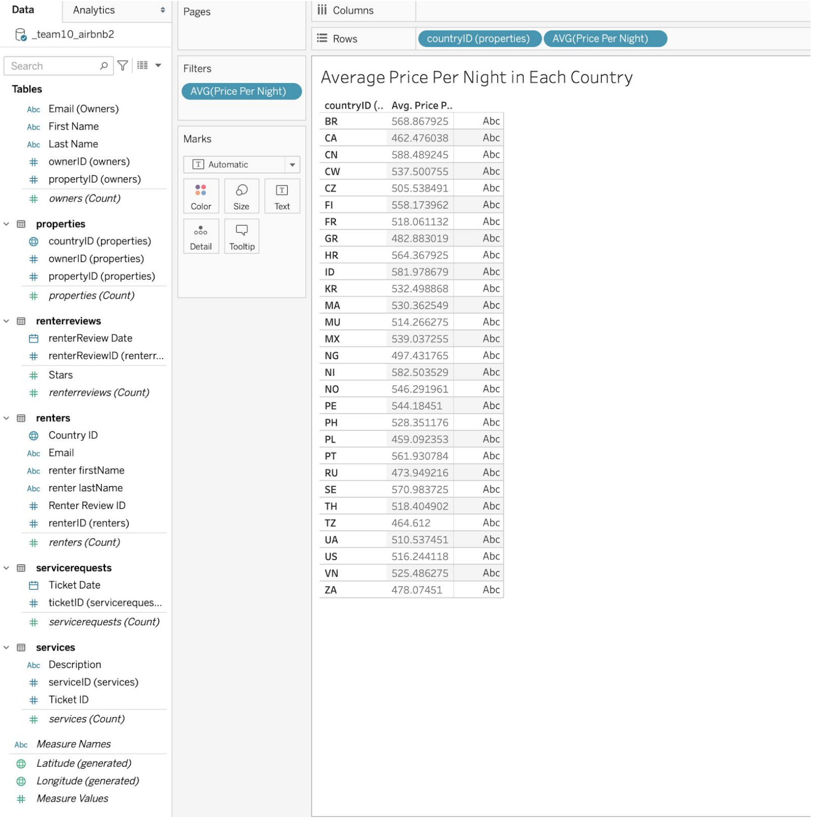
Average Price Per Night of a Four Star Reviewed Property in France

countryID (..	Stars	
FR	4	463.0

Stars
□ (All)
☑ 0
☑ 1
☑ 2
☑ 3
□ 4
☑ 5

15. What is the Average price per night per country?

- This visualization shows us how much it costs to stay around the world! This is nice for us to clearly see the whole world in one table. We can quickly see cheaper and more expensive places to stay. Created by using the average measure for price per night.



Columns

Rows: countryID (properties), AVG(Price Per Night)

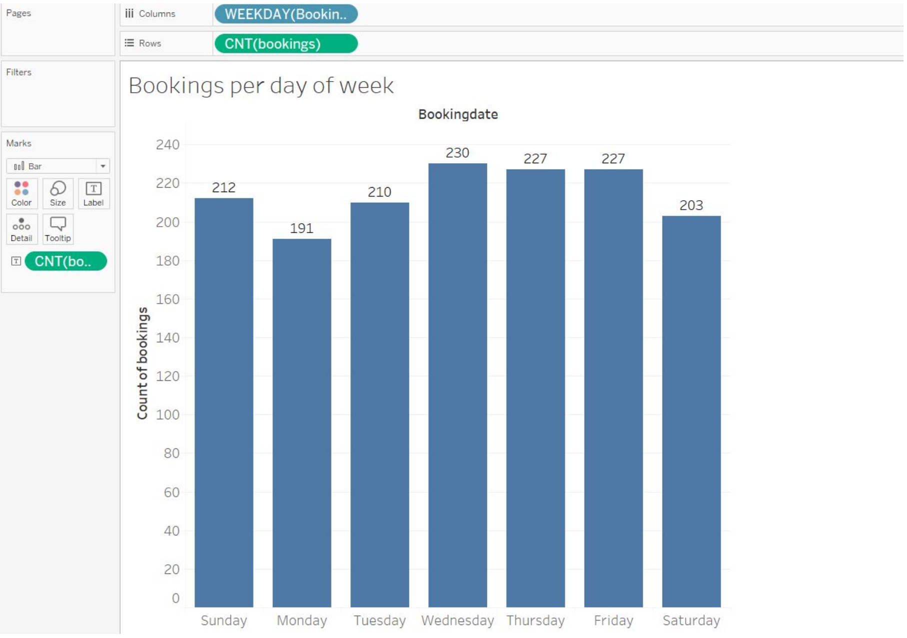
Filters: AVG(Price Per Night)

Average Price Per Night in Each Country

countryID (..	Avg. Price P..	
BR	568.867925	Abc
CA	462.476038	Abc
CN	588.489245	Abc
CW	537.500755	Abc
CZ	505.538491	Abc
FI	558.173962	Abc
FR	518.061132	Abc
GR	482.883019	Abc
HR	564.367925	Abc
ID	581.978679	Abc
KR	532.498868	Abc
MA	530.362549	Abc
MU	514.266275	Abc
MX	539.037255	Abc
NG	497.431765	Abc
NI	582.503529	Abc
NO	546.291961	Abc
PE	544.18451	Abc
PH	528.351176	Abc
PL	459.092353	Abc
PT	561.930784	Abc
RU	473.949216	Abc
SE	570.983725	Abc
TH	518.404902	Abc
TZ	464.612	Abc
UA	510.537451	Abc
US	516.244118	Abc
VN	525.486275	Abc
ZA	478.07451	Abc

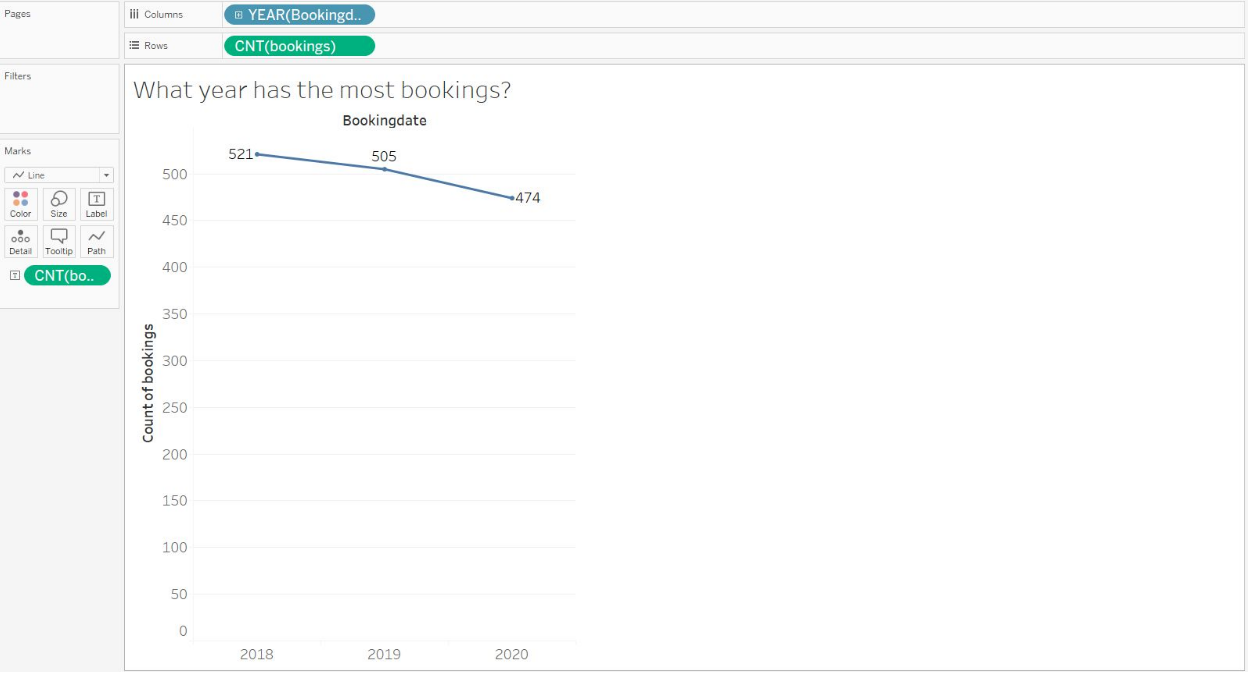
16. Count of bookings per day of week

- This allows us to see the number of bookings per day of week. Using the WEEKDAY function on booking date we can group by the day of the week and count the number of bookings. From this we see that Wednesday has the most bookings with Thursday and Friday tied for second.



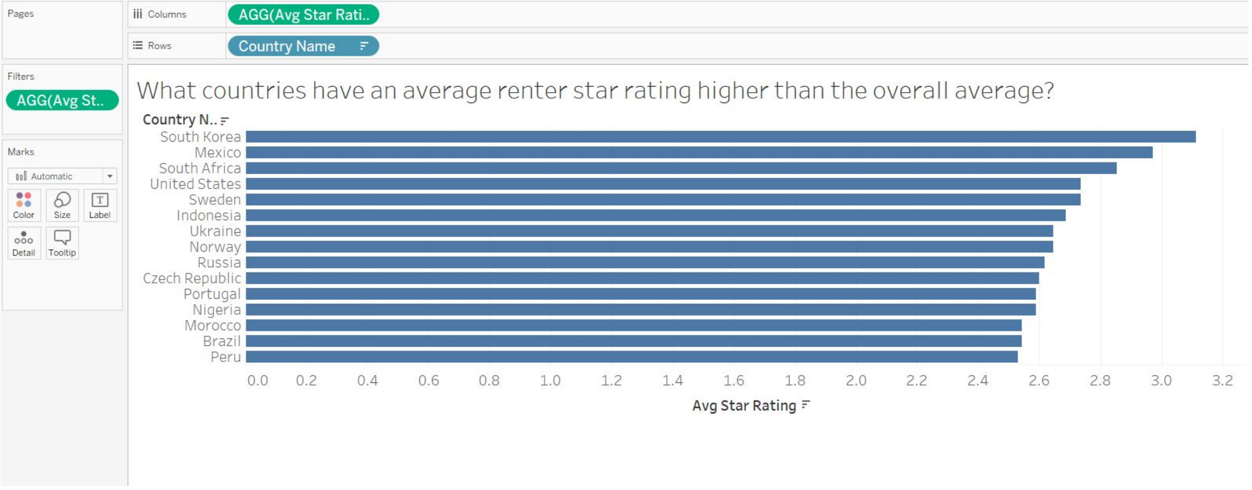
17. Count of bookings per year

- In this graph we are looking at total bookings per year. Using the YEAR function of booking date we can group by year and sum the count of bookings. Over the past 3 years the number of bookings has decreased from 521 to 474.



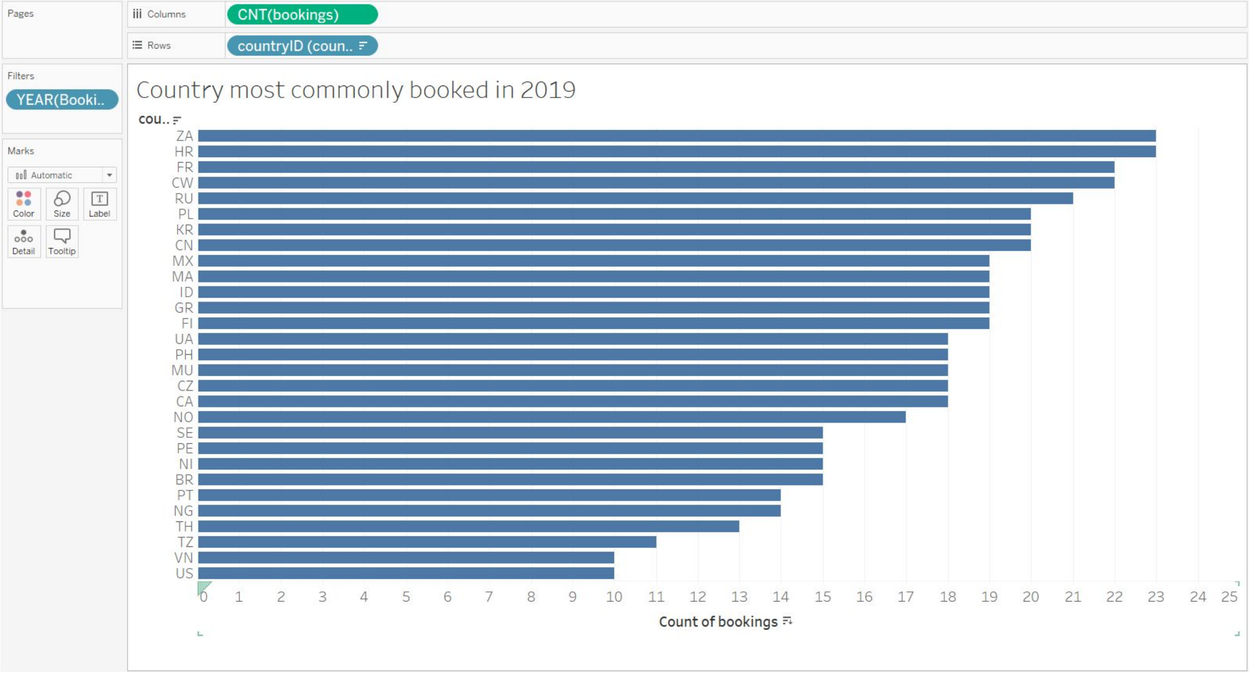
18. Countries with an average renter star rating greater then the overall average

- This chart shows us the countries that have an average star rating higher than the overall average star rating across countries. We grouped by country name in the Rows tab and the AVG star rating in the Columns tab. We then created a calculated function to find the overall average and used that in the Filters tab to filter where the country average was higher. We cans see South Korea has the highest average rating.



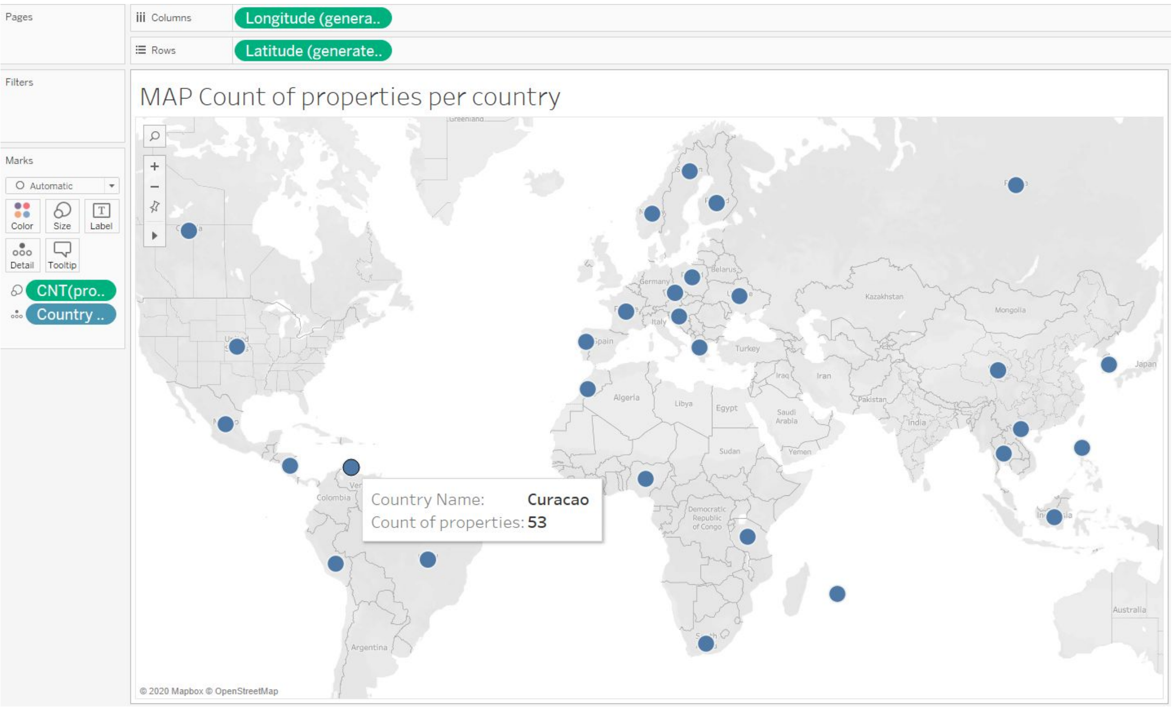
19. What country was booked the most in 2019

- Here we grouped by Country ID in the Rows and aggregated the count of bookings for each country. We then Filtered by the booking date to only count bookings in 2019. We can see ZA or South Africa had the most bookings in 2019.



20. Map of count of properties per country

- This heat map shows us the count of bookings per country. When you hover over a circle you can see the country name and the count of bookings. Overall, they have a pretty even distribution count of bookings as the circles are the same size.



MAP Count of properties per country

Columns: Longitude (genera..

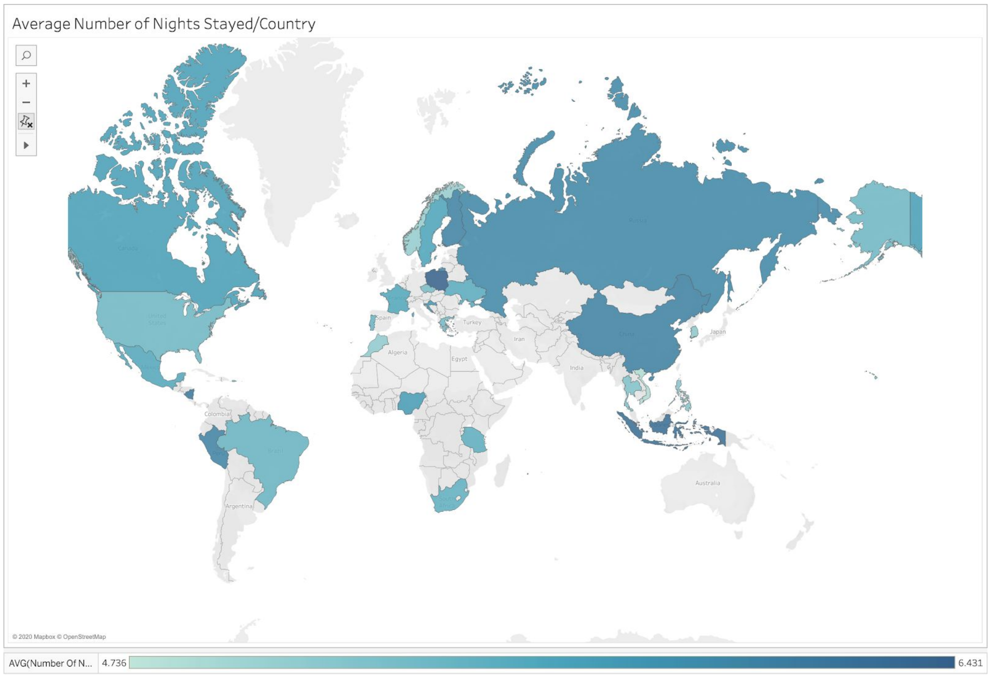
Rows: Latitude (generate..

Country Name: Curacao

Count of properties: 53

21. Average number of Nights stayed per country.

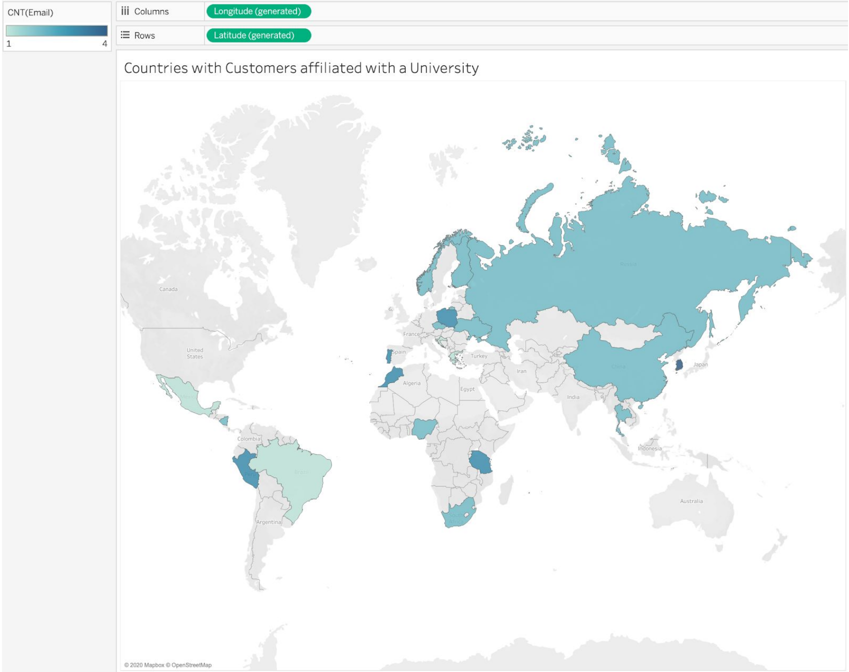
- This graph was created using a join between the bookings, renters, and countries tables, and uses the aggregate function AVG to consolidate the data per country.



Average Number of Nights Stayed/Country

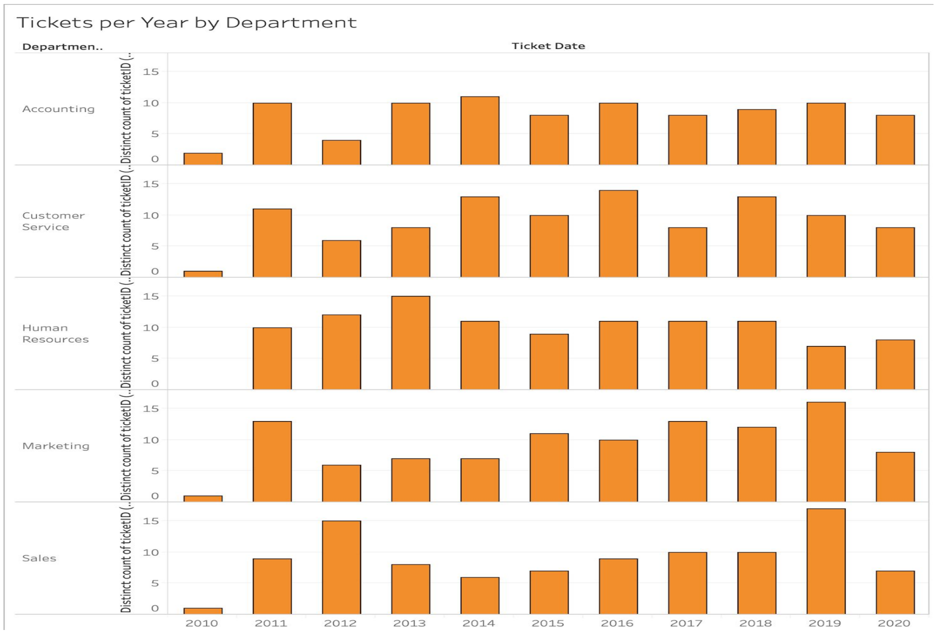
22. Countries with Customers affiliated with a University:

- This map shows the total count of customers with emails that have a domain associated with a university. It only required one join of renters and countries, and one filter which searched for the string ".edu" within the email column in the renters table.



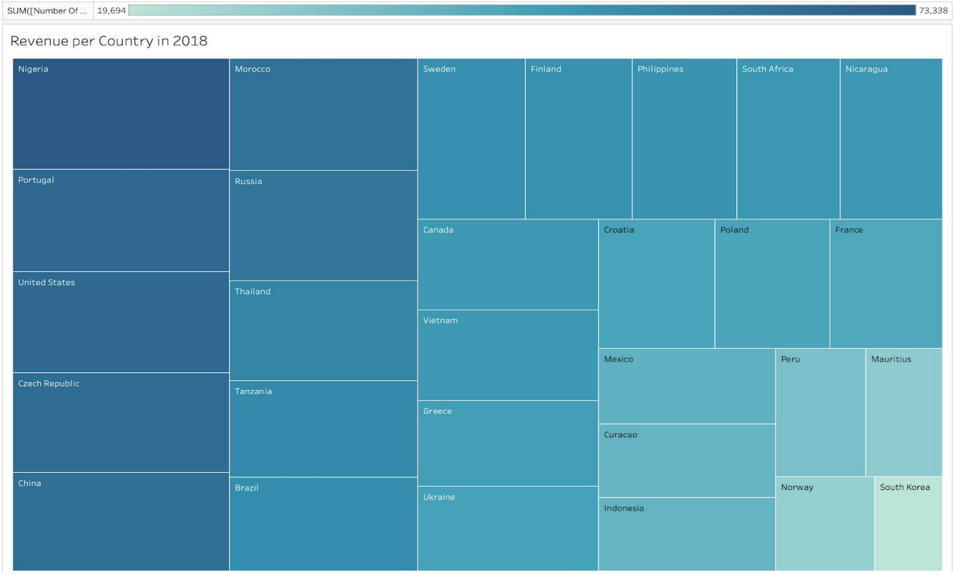
23. Tickets per Year by Department

- This chart represents the number of tickets per year and month, grouped by department. It required a join between the servicerequests, services, employees, and departments tables. It adds extra insight by using year as a filter, so that one could hide particular years or months.



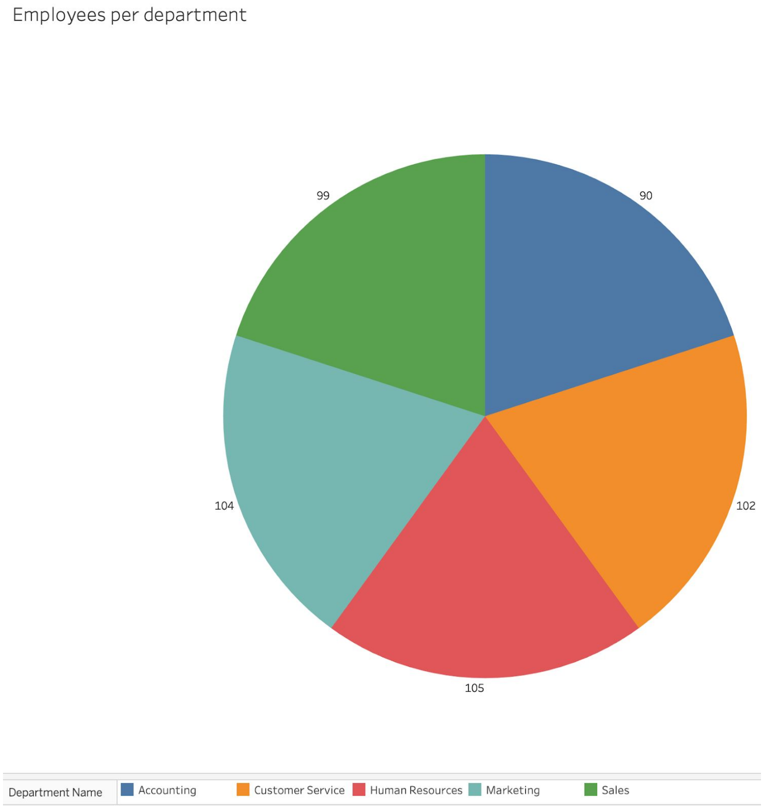
24. Revenue per Country in 2018

- This tree map provides a useful way to quickly find which countries have the highest sum of price per night*number of nights stayed, or “revenue”. It required a join between the bookings, renters, and countries tables, and does not show all the countries, only those with revenue higher than $19,964.



25. Employees per Department

- This simple visualization shows the count of employees per department. No filters were used, just the count aggregation, and three tables were joined: departments, employees, and services.


Employees per department
Department Name
Accounting
Customer Service
Human Resources
Marketing
Sales