AIRBNB – SQL DATA CLEANING

PROJECT PURPOSE: *To clean and prepare the NYC Airbnb dataset so that it can be ready for analysis.*

Objectives:

- Making Column Names uniform
- Fixing Spelling Errors
- Checking for duplicates
- Handling NULL values
- Data Manipulation (Adding Columns for Smoking, Pets, Wi-Fi)
- Dropping unnecessary columns – Final clean-up

**Dataset pre-processing – The instant_bookable column (BOOLEAN – TRUE/FALSE) contained blank values preventing import into SQL Server. These blank fields were filled in Excel with FALSE to get dataset uploaded.*

```
-------------------------------------------------------------------------------------------------------------
------------------------------ AIRBNB - SQL DATA CLEANING #2 ----------------------------------------
-------------------------------------------------------------------------------------------------------------

USE airbnb;


SELECT * FROM airbnbdata;

-- CREATE COPY OF DATASET --

SELECT * INTO airbnbdatacopy
FROM
	(SELECT * FROM airbnbdata) AS airbnbdatacopy;
```

Making the Column Names uniform

```
/* MAKING COLUMN NAMES UNIFORM */
```

Checking the current column names: Notice how the column names (shown below) in the dataset are not structured the same. The task here will be to make all column names lowercase and ensure underscores divide each word in the column name.

```
-- CHECKING THE CURRENT COLUMN NAMES --

SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'airbnbdatacopy';
```

COLUMN_NAME
id
NAME
host id
host_identity_verified
host name
neighbourhood group
neighbourhood
lat
long
country
country code
instant_bookable
cancellation_policy
room type
Construction year
price
service fee
minimum nights
number of reviews
last review
reviews per month
review rate number
calculated host listings count
availability 365
house_rules
license

Renaming the Columns (Note: not all of the columns needed changing):

```
-- RENAME COLUMNS IN airbnb TABLE --

EXEC sp_RENAME 'airbnbdatacopy.id','airbnb_id','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.NAME','name','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.host id','airbnb_host_id','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.host name','host_name','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.neighbourhood group','neighborhood_group','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.neighbourhood','neighborhood','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.country code','country_code','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.room type','room_type','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.Construction year','construction_year','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.service fee','service_fee','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.minimum nights','minimum_nights','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.number of reviews','number_of_reviews','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.last review','last_review','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.reviews per month','reviews_per_month','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.review rate number','review_rate_number','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.calculated host listings count','calculated_host_listings_cnt','COLUMN';
EXEC sp_RENAME 'airbnbdatacopy.availability 365','availability_365','COLUMN';
```

Checking the new column names:

```
-- CHECKING THE NEW COLUMN NAMES --

SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'airbnbdatacopy';
```

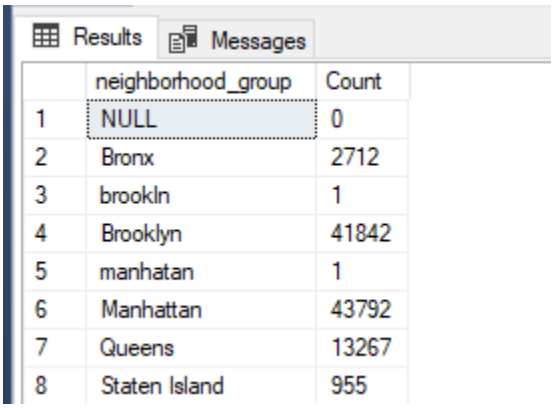
COLUMN_NAME
airbnb_id
name
airbnb_host_id
host_identity_verified
host_name
neighborhood_group
neighborhood
lat
long
country
country_code
instant_bookable
cancellation_policy
room_type
construction_year
price
service_fee
minimum_nights
number_of_reviews
last_review
reviews_per_month
review_rate_number

calculated_host_listings_cnt
availability_365
house_rules
license

Fixing Spelling Errors

Checking the different values showing in the neighborhood_group column

```
/* FIXING SPELLING ERRORS */

-- CHECKING THE DIFFERENT VALUES SHOWING IN THE neighborhood_group COLUMN --

SELECT neighborhood_group,
       COUNT(neighborhood_group) AS 'Count'
FROM airbnbdatacopy
GROUP BY neighborhood_group
ORDER BY neighborhood_group;
```



Results | Messages

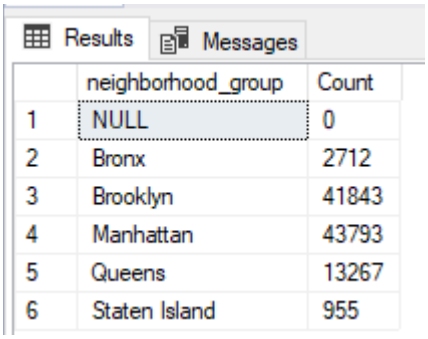
	neighborhood_group	Count
1	NULL	0
2	Bronx	2712
3	brookln	1
4	Brooklyn	41842
5	manhatan	1
6	Manhattan	43792
7	Queens	13267
8	Staten Island	955

Here it can be seen that Brooklyn and Manhattan are showing twice with 2 misspellings (‘brookln’ and ‘manhatan’). All instances of these values will be set to ‘Brooklyn’ and ‘Manhattan’, respectively.

```
-- CORRECTING THE VALUE 'brookln' TO 'Brooklyn' --

UPDATE airbnbdatacopy
SET neighborhood_group = CASE
                            WHEN neighborhood_group = 'brookln' THEN 'Brooklyn'
                            END;

-- CORRECTING THE VALUE 'manhatan' TO 'Manhattan' --

UPDATE airbnbdatacopy
SET neighborhood_group = CASE
                            WHEN neighborhood_group = 'manhatan' THEN 'Manhattan'
                            END;
```

Checking updates

```
-- CHECKING UPDATES --

SELECT neighborhood_group,
       COUNT(neighborhood_group) AS 'Count'
FROM airbnbdatacopy
GROUP BY neighborhood_group
ORDER BY neighborhood_group;
```



Results | Messages

	neighborhood_group	Count
1	NULL	0
2	Bronx	2712
3	Brooklyn	41843
4	Manhattan	43793
5	Queens	13267
6	Staten Island	955

Duplicates

```
/* DUPLICATES */
```

Checking for duplicates:

```
WITH duplicates AS
              (SELECT airbnb_id,
                     ROW_NUMBER() OVER (PARTITION BY airbnb_id, airbnb_host_id ORDER BY airbnb_id) AS
                     ROW_NUM
              FROM airbnbdatacopy)
SELECT * FROM duplicates WHERE ROW_NUM > 1;
```

```
(541 rows affected)
```

Checking the first 3 rows returned:

airbnb_id	ROW_NUM
6026161	2
6026714	2
6027266	2

```
SELECT *
FROM airbnbdatacopy
WHERE airbnb_id IN (6026161, 6026714, 6027266)
ORDER BY airbnb_id;
```

airbnb_id	name	airbnb_host_id	host_identity_verified	host_name	neighborhood_group	neighborhood	lat	long
6026161	Upper East Side 2 bedroom- close to Hospitals-	6.52E+10	verified	Juliana	Manhattan	Upper East Side	40.76222	-73.9603
6026161	Upper East Side 2 bedroom- close to Hospitals-	6.52E+10	verified	Juliana	Manhattan	Upper East Side	40.76222	-73.9603
6026714	Close to East Side Hospitals- Modern 2 Bedroom Apt	3.11E+10	verified	Juliana	Manhattan	Upper East Side	40.76249	-73.9622
6026714	Close to East Side Hospitals- Modern 2 Bedroom Apt	3.11E+10	verified	Juliana	Manhattan	Upper East Side	40.76249	-73.9622
6027266	ACADIA Spacious 2 Bedroom Apt - Close to Hospitals	9.59E+10	verified	Juliana	Manhattan	Upper East Side	40.76021	-73.9616
6027266	ACADIA Spacious 2 Bedroom Apt - Close to Hospitals	9.59E+10	verified	Juliana	Manhattan	Upper East Side	40.76021	-73.9616

Deleting the duplicates:

```
-- DELETING DUPLICATES --

WITH duplicates AS
             (SELECT airbnb_id,
                    ROW_NUMBER() OVER (PARTITION BY airbnb_id, airbnb_host_id ORDER BY airbnb_id) AS
                    ROW_NUM
             FROM airbnbdatacopy)
DELETE FROM duplicates WHERE ROW_NUM > 1;

(541 rows affected)
```

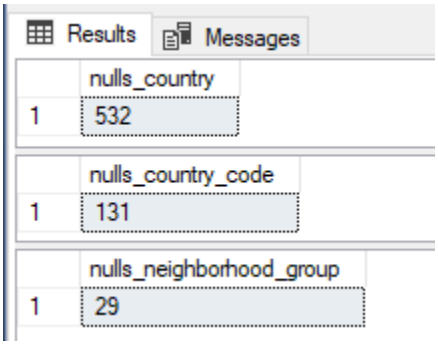
Nulls

Handling NULL values in the following columns:

- country
- country_code
- neighborhood_group

Checking the number of NULL values in each of these columns

```
-- CHECKING THE NUMBER OF NULL VALUES EXIST IN THE country, country_code --
-- AND neighborhood_group COLUMNS --

SELECT COUNT(*) AS 'nulls_country'
FROM airbnbdatacopy
WHERE country IS NULL;

SELECT COUNT(*) AS 'nulls_country_code'
FROM airbnbdatacopy
WHERE country_code IS NULL;

SELECT COUNT(*) AS 'nulls_neighborhood_group'
FROM airbnbdatacopy
WHERE neighborhood_group IS NULL;
```



Fixing the NULL values in the country and country_code columns:

Since this dataset contains only information from Airbnb location in New York, all NULL values in the country and country_code columns will be set to 'United States' and 'US', respectively.

```
-- FIXING THE NULL VALUES IN THE country AND country_code COLUMNS --

UPDATE airbnbdatacopy
SET country = ISNULL(country, 'United States') FROM airbnbdatacopy;

UPDATE airbnbdatacopy
SET country_code = ISNULL(country_code, 'US') FROM airbnbdatacopy;
```

Fixing the NULL values in the neighborhood_group column:

Creating a temp table that groups together neighborhood_groups and neighborhood where the values are not null and then using it to join onto the airbnbdata table to fill the null values in the neighborhood_group column.

Creating the temp table 'neighbors':

```
-- CREATE TEMP TABLE CALLED neighborhoods --

SELECT * INTO #neighborhoods
FROM
(SELECT neighborhood_group,
       neighborhood
FROM airbnbdatacopy
WHERE neighborhood_group IS NOT NULL AND neighborhood IS NOT NULL
GROUP BY neighborhood_group, neighborhood) AS #neighborhoods;
```

Checking the first 10 rows:

```
SELECT *
FROM #neighborhoods
ORDER BY neighborhood_group, neighborhood;
```

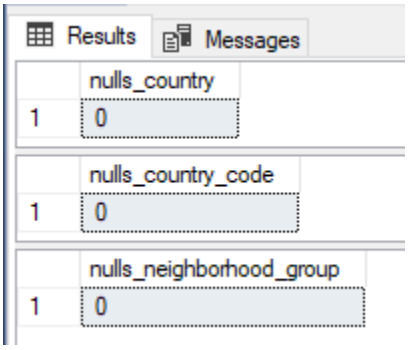
neighborhood_group	neighborhood
Bronx	Allerton
Bronx	Baychester
Bronx	Belmont
Bronx	Bronxdale
Bronx	Castle Hill
Bronx	City Island
Bronx	Claremont Village
Bronx	Clason Point
Bronx	Concourse
Bronx	Concourse Village

Now to set the NULL values in the neighborhood_group from the airbnbdata table by joining it with the neighborhoods temp table.

```
-- SETTING THE NULL VALUES IN THE neighborhood_group COLUMN --

UPDATE a
SET neighborhood_group = ISNULL(a.neighborhood_group, n.neighborhood_group)
                                                FROM airbnbdatacopy a
                                                LEFT JOIN #neighborhoods n
                                                ON a.neighborhood = n.neighborhood;
```

Checking the updates in the country, country_code, and neighborhood_group columns:

```
SELECT COUNT(*) AS 'nulls_country'
FROM airbnbdatacopy
WHERE country IS NULL;

SELECT COUNT(*) AS 'nulls_country_code'
FROM airbnbdatacopy
WHERE country_code IS NULL;

SELECT COUNT(*) AS 'nulls_neighborhood_group'
FROM airbnbdatacopy
WHERE neighborhood_group IS NULL;
```



Data Manipulation

Adding columns for Smoking, Pets, and Wi-fi – Here I will create a column for each that will return 'Yes', 'No' or 'Unknown' on whether these are allowed or not and/or available.

Adding column to indicate a non-smoking location

The smoking column will show whether smoking was mentioned in the house rules of each Airbnb location in the dataset. Where the house rules have any text that is like 'no smoke' or 'no smoking', 'No' will be returned. Likewise, if house rules mention 'smoking allowed', 'Yes' will be returned. Anything else will return 'Unknown'.

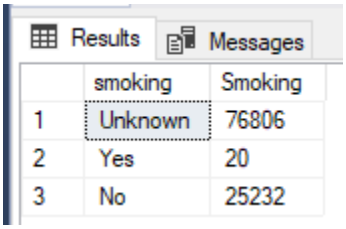
```
-- SELECTING LOCATIONS THAT MENTION NO SMOKING --

SELECT airbnb_id,
       house_rules
FROM airbnbdatacopy
WHERE house_rules LIKE '%no smoke%' OR house_rules LIKE '%no smoking%'
ORDER BY airbnb_id

-- SELECTING LOCATIONS THAT MENTION SMOKING ALLOWED --

SELECT airbnb_id,
       house_rules
FROM airbnbdatacopy
WHERE house_rules LIKE 'smoking allowed%'
ORDER BY airbnb_id;
```

Adding and setting smoking column:

```
-- ADDING smoking COLUMN --

ALTER TABLE airbnbdatacopy
ADD smoking NVARCHAR(10);

-- SETTING smoking COLUMN WITH CASE STATEMENT --

UPDATE airbnbdatacopy
SET smoking = CASE
                  WHEN house_rules LIKE '%no smoke%' OR house_rules LIKE '%no smoking%' THEN 'No'
                  WHEN house_rules LIKE 'smoking allowed%' THEN 'Yes'
                     ELSE 'Unknown'
              END
```

Checking smoking column update:

```
-- CHECKING smoking COLUMN UPDATE --

SELECT smoking,
       COUNT(smoking) AS 'Smoking'
FROM airbnbdatacopy
GROUP BY smoking;
```



Results | Messages

	smoking	Smoking
1	Unknown	76806
2	Yes	20
3	No	25232

```
SELECT TOP(3) airbnb_id,
              house_rules,
              smoking
FROM airbnbdatacopy
WHERE smoking = 'Yes';
```

airbnb_id	house_rules	smoking
14378042	Smoking allowed outside, on grounds, pets under 10 lb sitting available additional charge and events possibly pre aproved by owners,	Yes
26248074	Smoking allowed outside, on grounds, pets under 10 lb sitting available additional charge and events possibly pre aproved by owners,	Yes
10421365	Smoking allowed outside, on grounds, pets under 10 lb sitting available additional charge and events possibly pre aproved by owners,	Yes

Adding column to indicate a no pets location

The pets column will show whether pets were mentioned in the house rules of each Airbnb location in the dataset. Where the house rules have any text that is like 'no pets', 'No' will be returned. Likewise, if house rules mention 'pets allowed', 'Yes' will be returned. Anything else will return 'Unknown'.

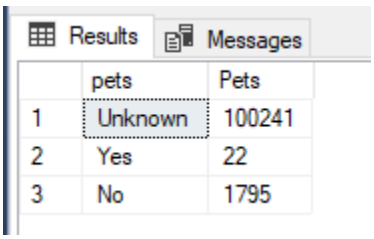
```
-- SELECTING LOCATIONS THAT MENTION NO PETS ALLOWED --

SELECT airbnb_id,
       house_rules
FROM airbnbdatacopy
WHERE house_rules LIKE 'no pets%'
ORDER BY airbnb_id

-- SELECTING LOCATIONS THAT MENTION PETS ALLOWED --

SELECT airbnb_id,
       house_rules
FROM airbnbdatacopy
WHERE house_rules LIKE 'pets allowed%'
ORDER BY airbnb_id
```

Adding and setting pets column:

```
-- ADDING pets COLUMN --

ALTER TABLE airbnbdatacopy
ADD pets NVARCHAR(10);

-- SETTING pets COLUMN --

UPDATE airbnbdatacopy
SET pets = CASE
              WHEN house_rules LIKE 'no pets%' THEN 'No'
              WHEN house_rules LIKE 'pets allowed%' THEN 'Yes'
                     ELSE 'Unknown'
              END
              FROM airbnbdatacopy
```

Checking pets column update:

```
-- CHECKING pets COLUMN UPDATE --

SELECT pets,
       COUNT(pets) AS 'Pets'
FROM airbnbdatacopy
GROUP BY pets;
```


Results Messages

	pets	Pets
1	Unknown	100241
2	Yes	22
3	No	1795

```
SELECT TOP(3) airbnb_id,
              house_rules,
              pets
FROM airbnbdatacopy
WHERE pets = 'Yes';
```

airbnb_id	house_rules	pets	smoking
5265644	Pets Allowed (With Fee) NO Smoking	Yes	No
3287306	Pets Allowed (With Fee) NO Smoking	Yes	No
11200660	Pets Allowed (With Fee) NO Smoking	Yes	No

Adding column to indicate whether wi-fi is mentioned

The wi-fi column will show whether wi-fi was mentioned in the house rules of each Airbnb location in the dataset. Where the house rules have any text that is like 'free wifi', 'Yes' will be returned. Anything else will return 'Unknown'.

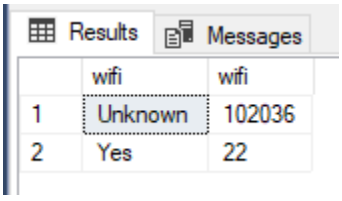
```
-- SELECTING LOCATIONS THAT MENTION FREE WIFI --
SELECT house_rules
FROM airbnbdatacopy
WHERE house_rules LIKE 'free wifi%'
```

Adding and setting wi-fi column:

```
-- ADDING wifi COLUMN --

ALTER TABLE airbnbdatacopy
ADD wifi NVARCHAR(10);

-- SETTING wifi COLUMN --

UPDATE airbnbdatacopy
SET wifi = CASE WHEN house_rules LIKE 'free wifi%' THEN 'Yes'
              ELSE 'Unknown'
              END
```

Checking Wi-fi column update:

```
-- CHECKING wifi COLUMN UPDATE --

SELECT wifi,
       COUNT(wifi) AS 'wifi'
FROM airbnbdatacopy
GROUP BY wifi;
```



	wifi	wifi
1	Unknown	102036
2	Yes	22

```
SELECT TOP(3) airbnb_id,
              house_rules,
              wifi
FROM airbnbdatacopy
WHERE wifi = 'Yes';
```

airbnb_id	house_rules	wifi
4982314	Free WiFi! * No smoking anywhere on property. *No out-door shoes to be worn in the house. We ask that guests be willing to be considerate and quiet while coming and going from 10 pm to 6 am, please. This is a non-smoking property. There is no smoking anywhere on the property, inside nor on the porch. We also ask guests to bring inside shoes/slippers. Thank you. As this is our home, we request you not to move the furniture nor our belongings around and to treat our home as you would wish your home to be treated. Thank you.	Yes
1025637	Free WiFi! * No smoking anywhere on property. *No out-door shoes to be worn in the house. We ask that guests be willing to be considerate and quiet while coming and going from 10 pm to 6 am, please. This is a non-smoking property. There is no smoking anywhere on the property, inside nor on the porch. We also ask guests to bring inside shoes/slippers. Thank you. As this is our home, we request you not to move the furniture nor our belongings around and to treat our home as you would wish your home to be treated. Thank you.	Yes
8938992	Free WiFi! * No smoking anywhere on property. *No out-door shoes to be worn in the house. We ask that guests be willing to be considerate and quiet while coming and going from 10 pm to 6 am, please. This is a non-smoking property. There is no smoking anywhere on the property, inside nor on the porch. We also ask guests to bring inside shoes/slippers. Thank you. As this is our home, we request you not to move the furniture nor our belongings around and to treat our home as you would wish your home to be treated. Thank you.	Yes

Dropping unnecessary columns – Final clean-up

Deleting instant_bookable column:

```
-- DELETING instant_bookable COLUMN --

ALTER TABLE airbnbdatacopy
DROP COLUMN instant_bookable
```

Deleting neighborhoods temp table:

```
-- DELETING neighborhoods TEMP TABLE --

DROP TABLE #neighborhoods
```