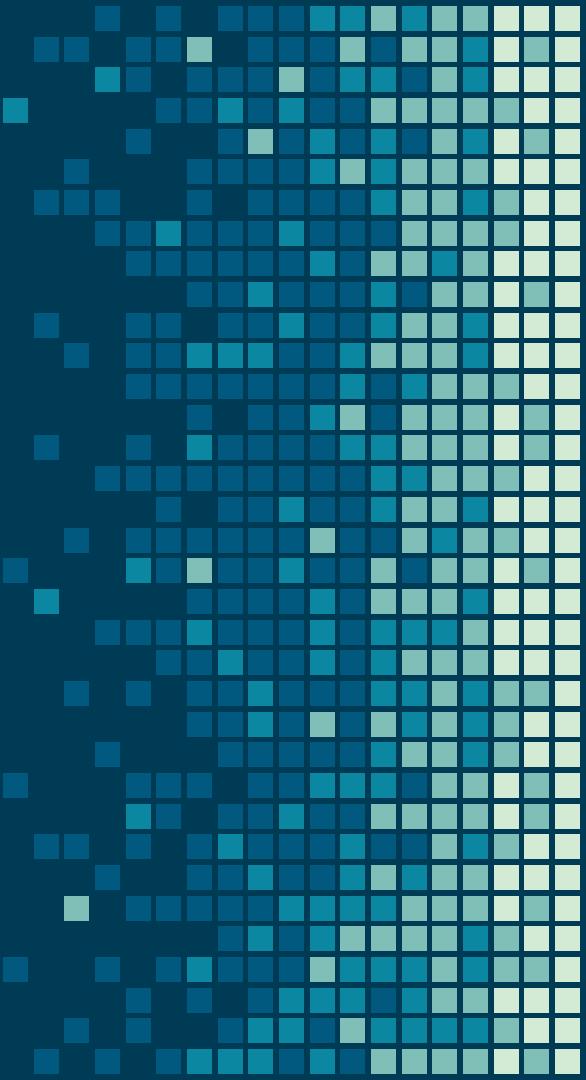


Expedia & Airbnb SQL Database Project

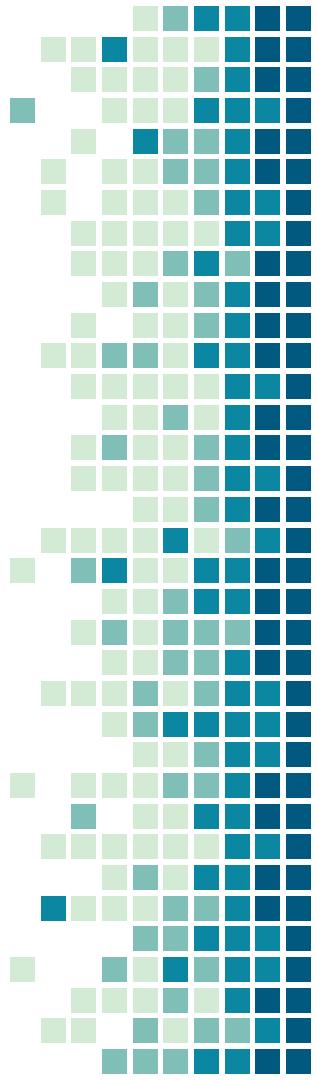
By
Juan Moctezuma-Flores





What is the goal of this database?

The purpose of this project is to create a relational database that stores Expedia (flight information only) and Airbnb data on behalf of the fictitious low-cost third party known as "JLMFCompany". This fictional enterprise allows the user to book flights and/or reserve Airbnb listings throughout the non-existent "Expedia Airbnb App (EApp)" for a fraction of the cost.





Data Tools & Software

A) Microsoft Office:

- Microsoft Excel
- Microsoft PowerPoint
- Microsoft Word

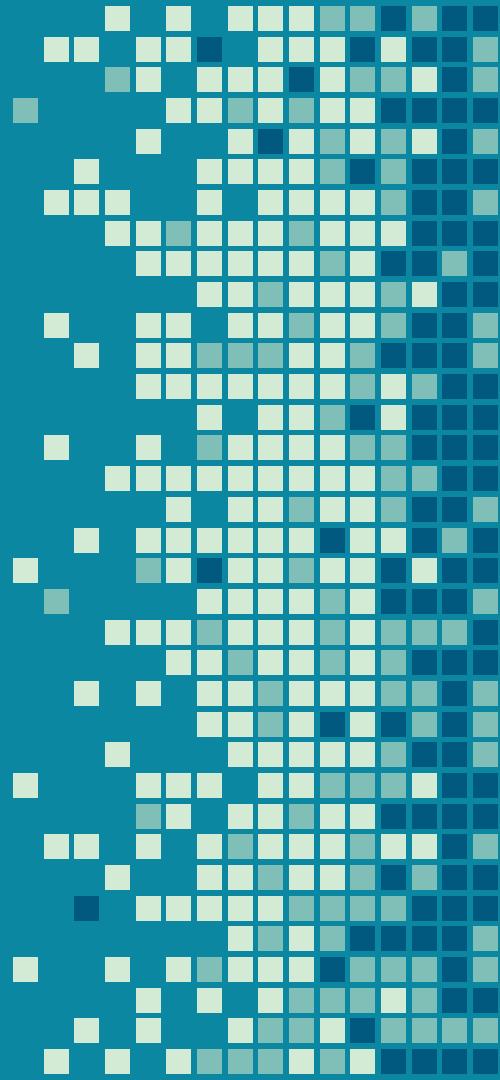
B) SQL Server:

- Azure Data Studio (platform)
- Extra tools:
 - Docker Desktop
 - Kitematic

C) Python 3:

- PyCharm CE – Platform
- Uncommon libraries/modules:
 - xlwt
 - Faker

D) Draw.io – Online Diagramming Software



Tables by Category

- JLMFCompany's database is composed of 11 tables (structures that collect or store data from Microsoft Excel → CSV files). These tables are stored in Azure Data Studio (SQL Server).
- List of tables by category:

A) User and records data:

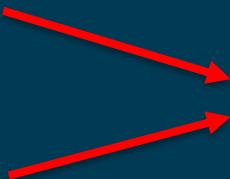
- BOOKINGS
- USER_ACCOUNT
- USER_RECORDS
- CARD_DETAILS

B) Airbnb data:

- AIRBNB_BOOKING_DATA
- AIRBNB_RESERVATION_DETAILS
- HOST_INFO
- REVIEWS

C) Expedia data:

- EXPEDIA_PASSENGER_DATA
- EXPEDIA_PASSENGER_TICKETS
- EXPEDIA_TRIP_DETAILS



Microsoft
SQL Server

Tables by Category (Part 2)

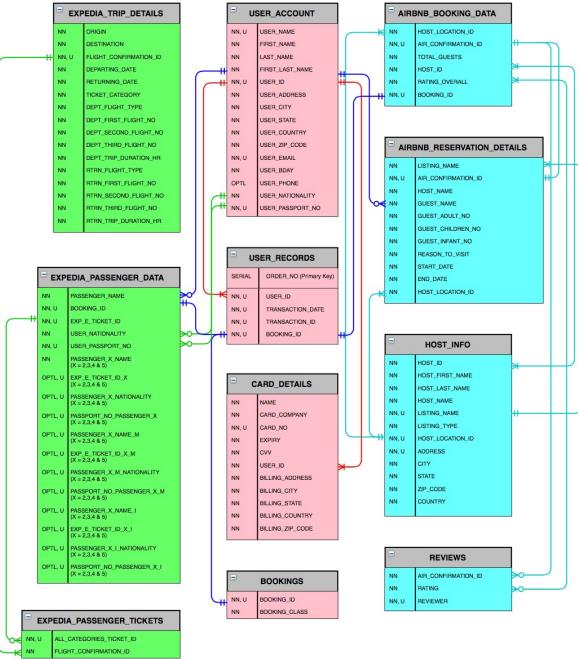
Each table has columns or fields; these relate with fields of other tables. For example, the user's id (Excel/CSV column) from **USER_ACCOUNT** relate with the user's id field from **USER_RECORDS**.

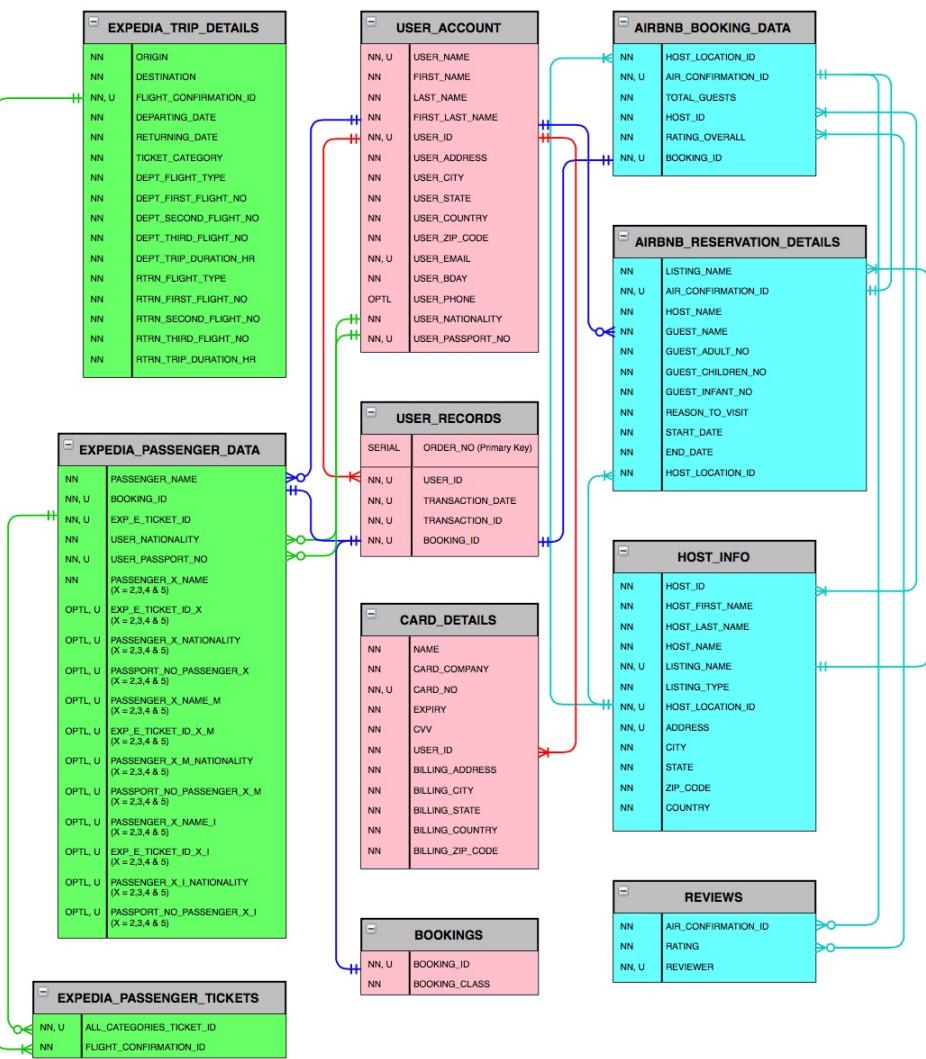
A	B	C	D	E
ORDER_NO	USER_ID	TRANSACTION_DATE	TRANSACTION_ID	BOOKING_ID
55	AH53CBSPZOG	2017-11-05	TXN-80V7ujQ60S-1whzHNwlo-65906326	AE-pgQbzek-2YNqtDle-dTxjH53
44	AK1ODDW77JF	2017-05-19	TXN-HHW5GzehFQ-hTgqWOMQ8-333907	AE-Bxn51e-v0UTqyGU-whSd4H
93	APZCN38B7SGQ	2019-08-02	TXN-9H8o6zORK9-rrn7jn6yCl-47202744	AE-WoP2My5-Pm1DwKiz-ZAkPj3h
25	AT4PQ5KIVX2A	2016-12-02	TXN-s2jx9MBul-9R9a6skt-48083515	AE-SiDphBh-MMx48pva-2qcgrZ
29	BFBFXKK0XTID	2017-01-07	TXN-slygc5Or4-c76xxVkmQ-44191341	AE-lBxTLQ4-69PzsIt-XQSzUL4
21	BGJ8U6PV2NPW	2016-09-07	TXN-YrRF1ElzI-lslwfHwku-24654425	AE-5O6Epri-QQWZBy4R-4GJzZHU
40	C717NB4BJ7PU	2017-04-06	TXN-FtnZwsI8yQ-7hUGPYAOs-56497044	AE-nBVjqzv-Mlanalpu-bk4ifFM
59	C7D0FR8HSPSU	2018-01-04	TXN-M74PLA51XF-cXLKhZNfa-14748359	AE-qqiru40-ElaLSf1Z-goAX4ra
51	CLZ2OX12Q4T1	2017-08-21	TXN-sDVjtVt6TgM-bKsqqGESV-62841738	AE-S4ZrZLM-rVm1QVx-eleATHf

A	B	C	D	E	F
USER_NAME	FIRST_NAME	LAST_NAME	FIRST_LAST_NAME	USER_ID	USER_ADDRESS
DCareyZ4C	Darren	Carey	Darren Carey	AH53CBSPZOG	221 Parker Plaza Suite 985 New Angela, WA 48931
JHale452	Justin	Hale	Justin Hale	AK1ODDW77JF	403 Brown Expressway Suite 885 East Michelleview, SC 09961
TAndersonXNJ	Todd	Anderson	Todd Anderson	APZCN38B7SGQ	PSC 746, Box 9786 APO AE 32875
DTranDUR	Daniel	Tran	Daniel Tran	AT4PQ5KIVX2A	USCC Smith FPO AE 38751
PConner1ET	Patrick	Conner	Patrick Conner	BFBFXKK0XTID	743 Hood Avenue Markport, HI 53492
CCchapmanUPS	Casey	Chapman	Casey Chapman	BGJ8U6PV2NPW	3329 Evans Cliff Apt. 084 East Heather, ME 98516
CWilliamsTXC	Carly	Williams	Carly Williams	C717NB4BJ7PU	976-3 Vincent Neck New Marisborough, AB A8K 3C8
ERivera3BM	Eloy	Rivera	Eloy Rivera	C7D0FR8HSPSU	Embajador Emiratos Árabes Unidos 273 816 Vieja República Popular Democrática de Corea, JAL 14974
AWilliamsCDB	Alex	Williams	Alex Williams	CLZ2OX12Q4T1	7634 Freeman Valleys Apt. 646 Annetteburgh, SK T5J 9M5
SBrown2VE	Stephanie	Brown	Stephanie Brown	CX8QZINTM7KU	75648 Green Shore Apt. 221 Lake Robertson, TX 64628

JLMF Company's Data Model: Entity Relationship (ER) Diagram

- The image contained in this slide represents the database's tables. Next slide (Page 6) displays a bigger image of the same ER Diagram (ERD).
- Tables contain a list of their respective columns. Meaning that every field seen in each Excel file, get reflected appropriately in SQL Server (Azure Data Studio), as long as these were loaded with the correct format.
- Page 7 contains the ERD's detailed description.





Entity Relationship (ER) Diagram Symbols & Description

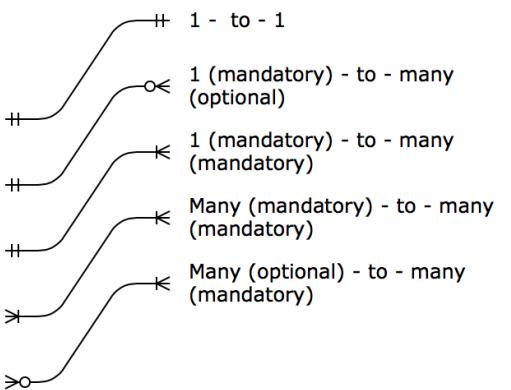
Cardinality & ordinality:

—+ One (AND only one)

—o< Zero OR many

—< One OR many

Entity Relation:



Meaning of colors:

Pink tables: Represent user, transaction and recordkeeping data.

Green tables: Represent Expedia data related with flight information only.

Blue tables: Represent Airbnb data.

Red line: For fields (columns) that have a relationship among pink tables only.

Dark blue line: For fields that relate pink, green, & blue tables.

Green line: For fields that relate among green tables; and relates green tables with pink tables.

Light blue line: For fields that relate among blue tables only.

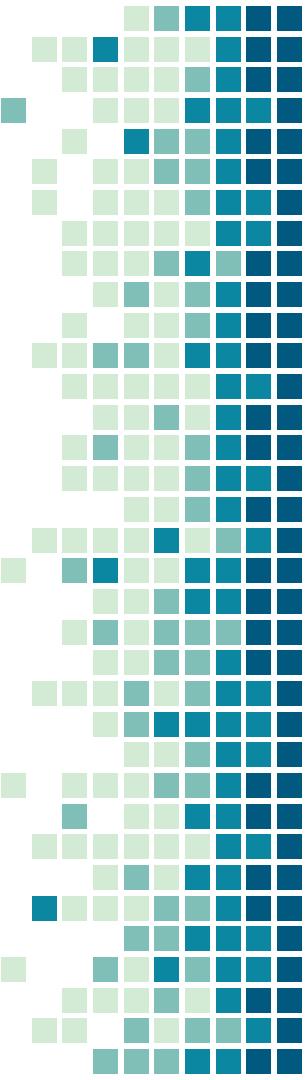
Abbreviations & Acronyms:

NN: Not Null. Represents any column that can't have empty cells.

U: Unique. Represents fields (columns) with non-repeatable reads.

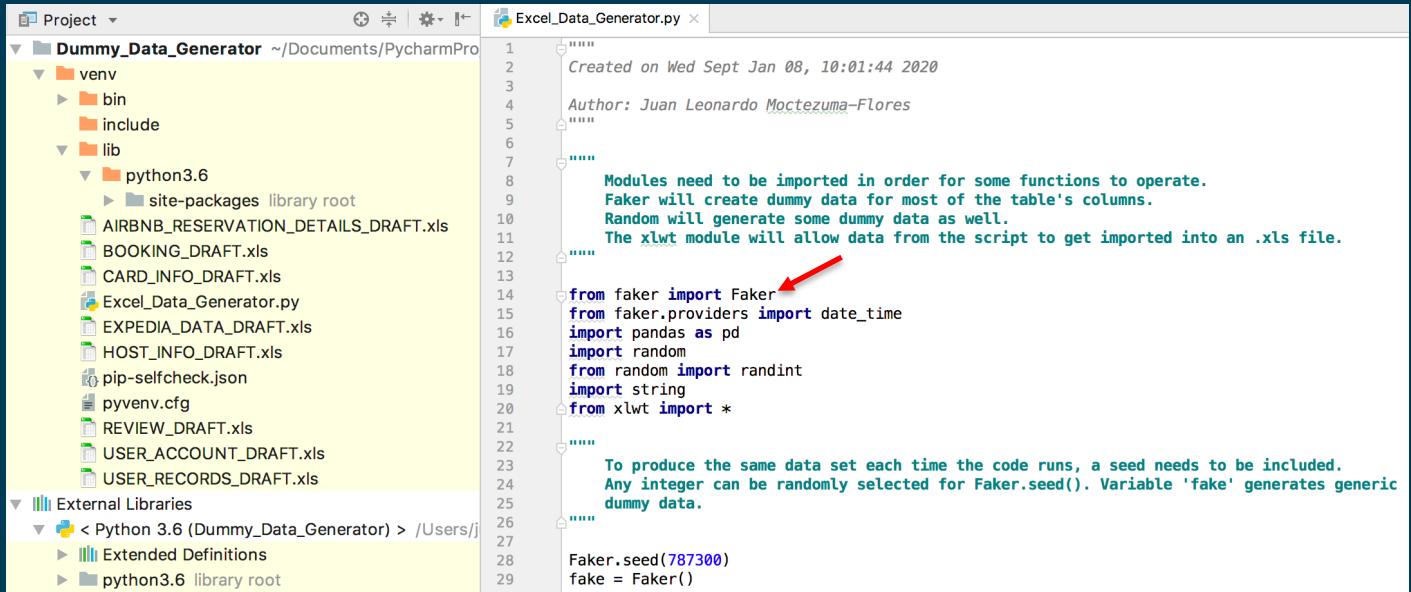
OPTL: Optional or non-essential data. For instance, phone numbers.

SERIAL: Sequential integer. Serves as the primary key in this project.



Dummy Data: Python's role

- Most columns were filled with fake information; in addition, the rest of the columns required manual cleansing or manipulation for consistency purposes.
- Unfortunately, the *Faker* module (package: bogus data generator) Is not able to generate consistent data for every Microsoft Excel column.



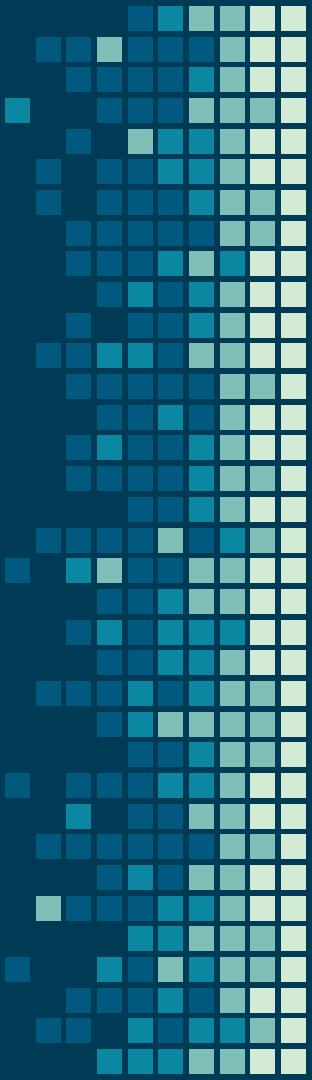
The screenshot shows the PyCharm IDE interface with the project structure on the left and the code editor on the right. The project structure includes a virtual environment folder 'venv' containing 'bin' and 'lib' subfolders, and a 'lib' folder with 'python3.6' and 'site-packages' subfolders. Several Excel files are listed under the project root: AIRBNB_RESERVATION_DETAILS_DRAFT.xls, BOOKING_DRAFT.xls, CARD_INFO_DRAFT.xls, Excel_Data_Generator.py, EXPEDIA_DATA_DRAFT.xls, HOST_INFO_DRAFT.xls, pip-selfcheck.json, pyvenv.cfg, REVIEW_DRAFT.xls, USER_ACCOUNT_DRAFT.xls, and USER_RECORDS_DRAFT.xls. The code editor displays 'Excel_Data_Generator.py' with the following content:

```
1  #####
2  # Created on Wed Sept Jan 08, 10:01:44 2020
3  #
4  # Author: Juan Leonardo Moctezuma-Flores
5  #
6  #####
7  #
8  # Modules need to be imported in order for some functions to operate.
9  # Faker will create dummy data for most of the table's columns.
10 # Random will generate some dummy data as well.
11 # The xlwt module will allow data from the script to get imported into an .xls file.
12 #
13 #
14 from faker import Faker
15 from faker.providers import date_time
16 import pandas as pd
17 import random
18 from random import randint
19 import string
20 from xlwt import *
21 #
22 #
23 # To produce the same data set each time the code runs, a seed needs to be included.
24 # Any integer can be randomly selected for Faker.seed(). Variable 'fake' generates generic
25 # dummy data.
26 #
27 Faker.seed(787300)
28 fake = Faker()
```

A red arrow points to the first line of code, 'from faker import Faker'.

Dummy Data: Python's role (Part 2)

Most of the data generated by the Python script/code involves information regarding full names, credit/debit cards, unique identifiers, dates, emails, users' phone numbers, citizenships, addresses, etc.



```
Excel_Data_Generator.py x
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
```

....
The following domains will randomly get attached to the username. For the sake of simplicity, the username matches the client's email address, since the goal is to use dummy data.

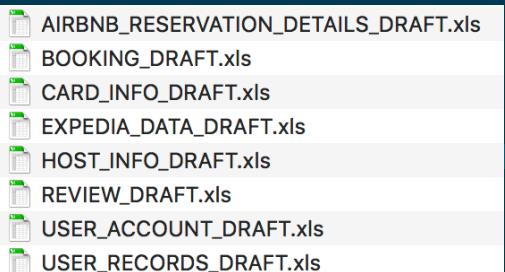
e_domains = ['@gmail.com', '@yahoo.com', '@outlook.com', '@aol.com', '@hotmail.com']
Whitespace variable.
space = " "
Airbnb Data - Motives for traveling.
motives = ['Leisure', 'Ecotourism', 'Business or work', 'Religious Tourism', 'Family Tourism', 'Health/Medical Tourism',
 'Sports Tourism', 'Education Tourism', 'Sports Tourism', 'Personal', 'Other']
Airbnb Data - Host listing types.
listing_types = ['ENTIRE_PLACE', 'PRIVATE_ROOM', 'SHARED_ROOM']
Booking Data - Booking types.
booking_class = ['CLASS_A', 'CLASS_B', 'CLASS_C']
Expedia Data - Nationality.
exp_passenger_nationality = ["UNITED STATES OF AMERICA", "AUSTRALIAN", "CANADIAN",
 "BRITISH CITIZEN", "FRANCAIS", "NEW ZEALAND", "DEUTSCH"]
Credit/Debit Card Company.
credit_card_company = ['VISA', 'MASTERCARD', 'DISCOVER']
expiry_month = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12']
expiry_year = ['21', '22', '23', '24', '25', '26', '27', '28']
Birthdays date range.
b_date1 = '1957-01-01'
b_date2 = '2000-01-01'
birth_date_dates = pd.date_range(b_date1, b_date2).tolist()

Dummy Data: Python's role (Part 3)

Different data types or categories (based on their respective tables) were generated and compiled into distinct XLS files. These were then carefully reviewed and manually manipulated, before being loaded into the database.

```
Excel_Data_Generator.py x

324     """
325         The following code lines will serve the purpose of
326         generating dummy credit/debit card data for CARD_INFO_DRAFT.xls
327     """
328
329     for x in range(len(card_h)):
330         sheet7.write(0, x, card_h[x])
331
332     for x in range(1,96):
333         sheet7.write(x, 0, random.choice(credit_card_company))
334         sheet7.write(x, 1, ran_gen(4,us_pass_no) + '-' + ran_gen(4,us_pass_no) + '-'
335                     + ran_gen(4,us_pass_no) + '-' + ran_gen(4,us_pass_no))
336         sheet7.write(x, 2, random.choice(expiry_month) + '/' + random.choice(expiry_year))
337         sheet7.write(x, 3, ran_gen(3,us_pass_no))
338
339     wb_7.save('CARD_INFO_DRAFT.xls')
340
341     """
342         The following code lines will serve the purpose of
343         generating dummy transaction data for USER_RECORDS_DRAFT.xls
344     """
345
346     for x in range(len(user_records_h)):
347         sheet8.write(0, x, user_records_h[x])
348
349     for x in range(1,41):
350         sheet8.write(x, 0, random.choice(transaction_dates))
351         sheet8.write(x, 1, 'TXN-' + ran_gen(10,alpha_num_v2) + '-' + ran_gen(9,alpha_num_v2) + '-' + ran_gen(8,us_pass_no))
352
353     wb_8.save('USER_RECORDS_DRAFT.xls')
354
355 #END OF CODE
```



Note: Although there are 11 tables, only 8 files with dummy data were produced.

SQL Server: Database Script

Users > juanleonardomoctezuma > Desktop > JLMFCompany_DATABASE.sql

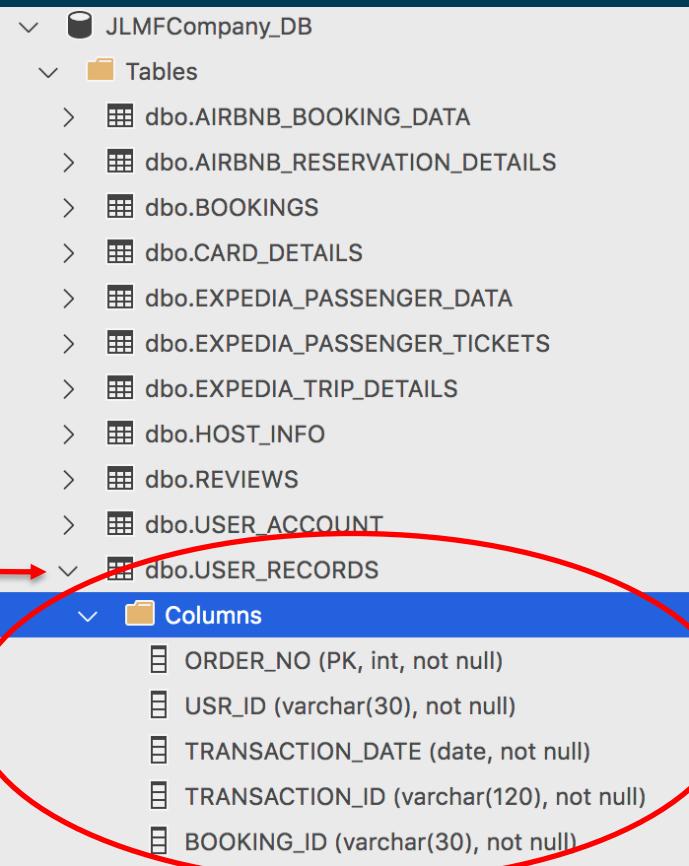
```
▶ Run □ Cancel ⏪ Connect ⏪ Change Connection Select Database ▾ |  
1  -- This script creates a new database called 'JLMFCompany_DB' --  
2  -- This database (DB) is connected to the 'master' database --  
3  USE master  
4  GO  
5  
6  -- This query will create a new DB if it doesn't exist already --  
7  IF NOT EXISTS (  
8      SELECT [name]  
9          FROM sys.databases  
10         WHERE [name] = 'JLMFCompany_DB'  
11    )  
12  
13  CREATE DATABASE JLMFCompany_DB  
14  GO
```

SQL Server: Tables Script

Users > juanleonardomocetzuma > Desktop > JLMFCompany_TABLES.sql

Run Cancel Connect Change Connection Select Database

```
1  /* This script contains every table created for this project.
2   JLMFCompany's database is composed of 11 tables.*/
3
4  CREATE TABLE USER_ACCOUNT
5  (
6      USER_NM VARCHAR(150) NOT NULL UNIQUE,
7      FIRST_NAME VARCHAR(80) NOT NULL,
8      LAST_NAME VARCHAR(150) NOT NULL,
9      FIRST_LAST_NAME VARCHAR(230) NOT NULL,
10     USR_ID VARCHAR(30) NOT NULL UNIQUE,
11     USER_ADDRESS VARCHAR(400) NOT NULL,
12     USER_CITY VARCHAR(100) NOT NULL,
13     USER_STATE VARCHAR(100) NOT NULL,
14     USER_COUNTRY VARCHAR(100) NOT NULL,
15     USER_ZIP_CODE VARCHAR(10) NOT NULL,
16     USER_EMAIL VARCHAR(200) NOT NULL UNIQUE,
17     USER_BDAY DATE NOT NULL,
18     USER_PHONE VARCHAR(100),
19     USER_NATIONALITY VARCHAR(100) NOT NULL,
20     USER_PASSPORT_NO VARCHAR(70) NOT NULL UNIQUE
21 )
22
23 CREATE TABLE USER_RECORDS
24 (
25     ORDER_NO INT NOT NULL PRIMARY KEY,
26     USR_ID VARCHAR(30) NOT NULL,
27     TRANSACTION_DATE DATE NOT NULL,
28     TRANSACTION_ID VARCHAR(120) NOT NULL UNIQUE,
29     BOOKING_ID VARCHAR(30) NOT NULL UNIQUE
30 )
31
32 CREATE TABLE AIRBNB_BOOKING_DATA
33 (
34     HOST_LOCATION_ID VARCHAR(30) NOT NULL UNIQUE,
35     AIR_CONFIRMATION_ID VARCHAR(30) NOT NULL UNIQUE,
36     TOTAL_GUESTS INT NOT NULL,
37     HOST_ID_NO VARCHAR(30) NOT NULL,
38     RATING_OVERALL DECIMAL(3,2) NOT NULL,
```



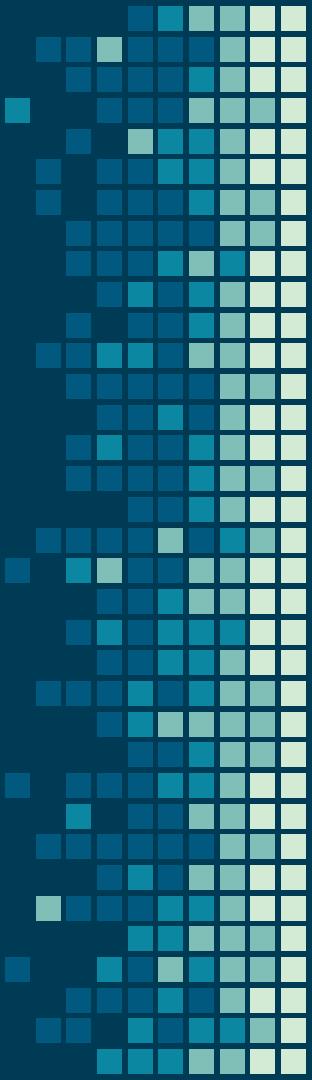
What happens when you load data?

Please observe the EXPEDIA_TRIP_DETAILS table (or file), which was created in Excel. The picture below represents columns A – F.

ORIGIN	DESTINATION	FLIGHT_CONFIRMATION_ID	DEPARTING_DATE	RETURNING_DATE	TICKET_CATEGORY
Denver, CO (DEN-Denver Intl.)	Rio de Janeiro, Brazil (GIG-Galeao - Antonio Carlos Jobim Intl.)	AVSJO9	2017-04-12	2017-05-01	ROUND-TRIP
Sacramento, CA (SMF-Sacramento Intl.)	San Diego, CA (SAN-San Diego Intl.)	BGA0R8H	2019-04-11	2019-04-14	ROUND-TRIP
Tokyo, Japan (NRT-Narita Intl.)	Hanoi, Vietnam (HAN-Noi Bai Intl.)	CMFN2Q	2019-06-03		ONE-WAY
Newark, NJ (EWR-Liberty Intl.)	Singapore, Singapore (SIN-Changi)	CMWVHH3	2019-01-10		ONE-WAY
Hong Kong, Hong Kong SAR (HKG-Hong Kong Intl.)	Taipei, Taiwan (TPE-Taoyuan Intl.)	DXCQP3	2017-03-22		ONE-WAY
Dublin, Ireland (DUB)	Los Angeles, CA (LAX-Los Angeles Intl.)	DZQ98T	2017-04-01	2017-04-18	ROUND-TRIP
Paris, France (CDG-Roissy-Charles de Gaulle)	Shanghai, China (PVG-Pudong Intl.)	EQU05B	2018-02-24	2018-03-24	ROUND-TRIP
Chicago, IL (ORD-O'Hare Intl.)	Santiago, Chile (SCL-Arturo Merino Benitez)	FBCTUS	2017-12-12	2017-12-18	ROUND-TRIP
New York, NY (JFK-John F. Kennedy Intl.)	Stockholm, Sweden (STO-All Airports)	FEFE5A	2017-03-20	2017-04-05	ROUND-TRIP
New York, NY (JFK-John F. Kennedy Intl.)	Madrid, Spain (MAD-Adolfo Suarez Madrid-Barajas)	FMBVKX	2019-07-20		ONE-WAY
Tijuana, Baja California Norte, Mexico (TIJ-General Abelardo L. Rodriguez Intl.)	Cancun, Quintana Roo, Mexico (CUN-Cancun Intl.)	FNAMSU	2018-12-25	2019-01-01	ROUND-TRIP
Orlando, FL (MCO-Orlando Intl.)	Baltimore, MD (BWI-Baltimore Washington Intl.)	FUWZFM	2018-08-22		ONE-WAY
San Francisco, CA (SFO-San Francisco Intl.)	Frankfurt, Germany (FRA-Frankfurt Intl.)	GHOCA8	2019-09-29	2019-10-17	ROUND-TRIP
Madrid, Spain (MAD-Adolfo Suarez Madrid-Barajas)	Dublin, Ireland (DUB)	GUUQE8	2017-05-03	2017-05-07	ROUND-TRIP
San Diego, CA (SAN-San Diego Intl.)	Tokyo, Japan (NRT-Narita Intl.)	HFZH50	2019-05-31	2019-06-21	ROUND-TRIP
Bangkok, Thailand (BKK-Suvarnabhumi Intl.)	Tashkent, Uzbekistan (TAS-Tashkent Intl.)	HHFR81	2019-06-03	2019-06-07	ROUND-TRIP
Los Angeles, CA (LAX-Los Angeles Intl.)	Budapest, Hungary (BUD-Ferenc Liszt Intl.)	IERZMW	2019-12-12	2019-12-21	ROUND-TRIP
Vancouver, BC, Canada (YVR-Vancouver Intl.)	Quebec, QC, Canada (YQB-Jean Lesage Intl.)	IIO0IU	2018-03-11		ONE-WAY
Tijuana, Baja California Norte, Mexico (TIJ-General Abelardo L. Rodriguez Intl.)	Aguascalientes, Aguascalientes, Mexico (AGU-Licenciado Jesus Teran Peredo Intl.)	JKI2J8	2018-11-23	2018-11-27	ROUND-TRIP
Mexico City, Distrito Federal, Mexico (MEX-Mexico City Intl.)	Rio de Janeiro, Brazil (GIG-Galeao - Antonio Carlos Jobim Intl.)	JOFOU9	2018-01-09	2018-01-26	ROUND-TRIP
Albuquerque, NM (ABQ-Albuquerque Intl. Sunport)	Boston, MA (BOS-All Airports)	LMYX1L	2016-09-30		ONE-WAY
San Diego, CA (SAN-San Diego Intl.)	London, England, UK (LHR-Heathrow)	LPD3LK	2018-05-09	2018-05-17	ROUND-TRIP
Milwaukee, WI (MKE-General Mitchell Intl.)	Boston, MA (BOS-Logan Intl.)	MEM11T	2016-06-11	2016-06-16	ROUND-TRIP
San Diego, CA (SAN-San Diego Intl.)	Oslo, Norway (OSL-Gardermoen)	NLQFJX	2019-12-18	2020-01-09	ROUND-TRIP
Miami, FL (MIA-Miami Intl.)	San Diego, CA (SAN-San Diego Intl.)	OGMJ11	2018-08-15	2018-08-19	ROUND-TRIP
Auckland, New Zealand (AKL-Auckland Intl.)	Doha, Qatar (DOH-Hamad Intl.)	ONJK95	2018-12-03		ONE-WAY
Sydney, NSW, Australia (SYD-Kingsford Smith Intl.)	Rotterdam, Netherlands (RTM-Rotterdam The Hague)	OPTAB0	2016-11-09	2016-11-15	ROUND-TRIP
Minneapolis, MN (MSP-Minneapolis - St. Paul Intl.)	Amsterdam, Netherlands (AMS-Schiphol)	QYP6TM	2017-03-20		ONE-WAY
Mexico City, Distrito Federal, Mexico (MEX-Mexico City Intl.)	Dallas, TX (DFW-Dallas-Fort Worth Intl.)	RBS0UC	2019-03-16		ONE-WAY
Johannesburg, South Africa (JNB-O.R. Tambo Intl.)	San Diego, CA (SAN-San Diego Intl.)	RRYXDE	2018-05-29	2018-06-21	ROUND-TRIP
Barcelona, Spain (BCN-Barcelona Intl.)	Edinburgh, Scotland, UK (EDI)	SHQ1Q0	2017-11-19	2017-11-13	ROUND-TRIP
Lima, Peru (LIM-Jorge Chavez Intl.)	London, England, UK (LHR-Heathrow)	TBHD4R	2020-01-20		ONE-WAY
Calgary, AB, Canada (YYC-Calgary Intl.)	Kelowna, BC, Canada (YLW-Kelowna Intl.)	TBWAA4V	2020-01-10	2020-01-10	ROUND-TRIP
Mexico City, Distrito Federal, Mexico (MEX-Mexico City Intl.)	Santiago, Chile (SCL-Arturo Merino Benitez)	TIY3M5	2017-03-10	2017-03-14	ROUND-TRIP
Hong Kong, Hong Kong SAR (HKG-Hong Kong Intl.)	Bishkek, Kyrgyzstan (FRU-Manas Intl.)	TXNOYH	2017-03-20	2017-03-29	ROUND-TRIP
Kansas City, MO (MCI-Kansas City Intl.)	Rome, Italy (FCO-Fiumicino - Leonardo da Vinci Intl.)	UPGP7N	2019-05-27	2019-06-27	ROUND-TRIP

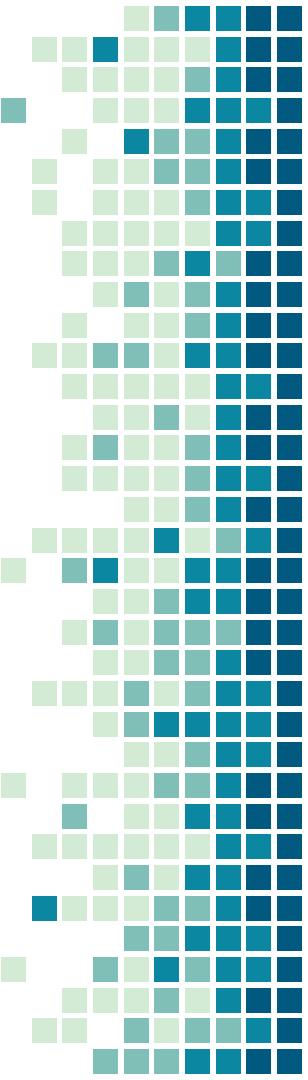
What happens when you load data? (Part 2)

Please observe the EXPEDIA_TRIP_DETAILS table. Data gets loaded (as a CSV file) into Azure Data Studio. The picture below represents columns A – E (Excel file: EXPEDIA_TRIP_DETAILS.xlsx).



A screenshot of the Azure Data Studio interface showing a query results grid. The top bar includes 'Run', 'Cancel', 'Disconnect', 'Change Connection' (set to 'JLMFCompany_DB'), and a dropdown menu. The query entered is 'SELECT * FROM EXPEDIA_TRIP_DETAILS;'. The results grid has columns: ORIGIN, DESTINATION, FLIGHT_CONFIRMATION_ID, DEPARTING_DATE, and RETURNING_DATE. The data consists of approximately 30 rows of flight information.

ORIGIN	DESTINATION	FLIGHT_CONFIRMATION_ID	DEPARTING_DATE	RETURNING_DATE
1 Denver, CO (DEN-Denver Intl.)	Rio de Janeiro, Brazil (GIG-Galeao ...	AVSJ09	2017-04-12	2017-05-01
2 Sacramento, CA (SMF-Sacramento Intl.)	San Diego, CA (SAN-San Diego Intl.)	BGA0RH	2019-04-11	2019-04-14
3 Tokyo, Japan (NRT-Narita Intl.)	Hanoi, Vietnam (HAN-Noi Bai Intl.)	CMFN2Q	2019-06-03	NULL
4 Newark, NJ (EWR-Liberty Intl.)	Singapore, Singapore (SIN-Changi)	CMWHH3	2019-01-10	NULL
5 Hong Kong, Hong Kong SAR (HKG-Hong K...	Taipei, Taiwan (TPE-Taoyuan Intl.)	DXCQP3	2017-03-22	NULL
6 Dublin, Ireland (DUB)	Los Angeles, CA (LAX-Los Angeles In...	DZQ98T	2017-04-01	2017-04-18
7 Paris, France (CDG-Roissy-Charles de...	Shanghai, China (PVG-Pudong Intl.)	EQV05B	2018-02-24	2018-03-24
8 Chicago, IL (ORD-O'Hare Intl.)	Santiago, Chile (SCL-Arturo Merino ...	FBCTUS	2017-12-12	2017-12-18
9 New York, NY (JFK-John F. Kennedy In...	Stockholm, Sweden (STO-All Airports)	FEFE5A	2017-03-20	2017-04-05
... New York, NY (JFK-John F. Kennedy In...	Madrid, Spain (MAD-Adolfo Suarez Ma...	FMBVKX	2019-07-20	NULL
... Tijuana, Baja California Norte, Mexi...	Cancun, Quintana Roo, Mexico (CUN-C...	FNAMSU	2018-12-25	2019-01-01
... Orlando, FL (MCO-Orlando Intl.)	Baltimore, MD (BWI-Baltimore Washin...	FUWZFM	2018-08-22	NULL
... San Francisco, CA (SFO-San Francisco...	Frankfurt, Germany (FRA-Frankfurt I...	GHOCAB8	2019-09-29	2019-10-17
... Madrid, Spain (MAD-Adolfo Suarez Ma...	Dublin, Ireland (DUB)	GUUQEB	2017-05-03	2017-05-07
... San Diego, CA (SAN-San Diego Intl.)	Tokyo, Japan (NRT-Narita Intl.)	HFZH50	2019-05-31	2019-06-21
... Bangkok, Thailand (BKK-Suvarnabhumi ...	Tashkent, Uzbekistan (TAS-Tashkent ...	HHFR81	2019-06-03	2019-06-07
... Los Angeles, CA (LAX-Los Angeles Int...	Budapest, Hungary (BUD-Ferenc Liszt...	IERZMW	2019-12-12	2019-12-21
... Vancouver, BC, Canada (YVR-Vancouver...	Quebec, QC, Canada (YQB-Jean Lesage...	II001U	2018-03-11	NULL
... Tijuana, Baja California Norte, Mexi...	Aguascalientes, Aguascalientes, Mex...	JKJ2J8	2018-11-23	2018-11-27
... Mexico City, Distrito Federal, Mexic...	Rio de Janeiro, Brazil (GIG-Galeao ...	JOF0U9	2018-01-09	2018-01-26
... Albuquerque, NM (ABQ-Albuquerque Int...	Boston, MA (BOS-All Airports)	LMYX1L	2016-09-30	NULL
... San Diego, CA (SAN-San Diego Intl.)	London, England, UK (LHR-Heathrow)	LPD3LK	2018-05-09	2018-05-17
... Milwaukee, WI (MKE-General Mitchell ...	Boston, MA (BOS-Logan Intl.)	MEMI1T	2016-06-11	2016-06-16
... San Diego, CA (SAN-San Diego Intl.)	Oslo, Norway (OSL-Gardermoen)	NLQFJX	2019-12-18	2020-01-09
... Miami, FL (MIA-Miami Intl.)	San Diego, CA (SAN-San Diego Intl.)	OGMJ11	2018-08-15	2018-08-19



What else can be done with the stored data?

- Run SQL queries for verification purposes.
- Observe trends and patterns.
- Calculate numerical data.
- Combine or join data that corresponds to different tables.
- Research.

The following slides (pages 17 – 23) consist of a series of research questions that were written by the author of this presentation. These questions include the queries (and their respective results) that were used to find the answers.

Research: Observation

```
▶ Run □ Cancel ⚙ Disconnect ☰ Change Connection JLMFCompany_DB ▾ |  
1  /*Question 1: Who are the users that opened an account but have not booked  
2  any flights or reserved any Airbnb listing through the company's platform - "EApp"?*/  
3  
4  SELECT  
5      UA.USER_ID,  
6      UA.FIRST_LAST_NAME,  
7      UR.TRANSACTION_DATE,  
8      UR.TRANSACTION_ID  
9  FROM  
10     USER_ACCOUNT UA  
11     FULL OUTER JOIN USER_RECORDS UR  
12         ON UA.USER_ID = UR.USER_ID  
13     FULL OUTER JOIN CARD_DETAILS CD  
14         ON UA.USER_ID = CD.USER_ID  
15  WHERE UR.TRANSACTION_DATE IS NULL AND UR.TRANSACTION_DATE IS NULL  
16 ;
```

Results Messages

	USER_ID	FIRST_LAST_NAME	TRANSACTION_DATE	TRANSACTION_ID
1	CX8QZINTM7KU	Stephanie Brown	NULL	NULL
2	V7H1CT77K0HK	Stephen Lewis	NULL	NULL
3	DYTBTXJ65W0B	Tracy Mendoza	NULL	NULL
4	V1D5VH6KEXGY	Wanda Rodgers	NULL	NULL

Research: Record lookup

Run Cancel ⚙ Disconnect Change Connection JLMFCompany_DB ▾ |

```
1  /*Question 2: What is the booking id, credit/debit card number, and transaction
2  identifier linked to order number 95?*/
3
4  SELECT
5      CD.CARD_NO,
6      UR.ORDER_NO,
7      B.BOOKING_ID,
8      UR.TRANSACTION_ID
9  FROM
10     BOOKINGS B
11     JOIN USER_RECORDS UR
12         ON UR.BOOKING_ID = B.BOOKING_ID
13     JOIN CARD_DETAILS CD
14         ON CD.USER_ID = UR.USER_ID
15 WHERE UR.ORDER_NO < 96 AND UR.ORDER_NO > 94
16 ;
```

Results Messages

	CARD_NO	ORDER_NO	BOOKING_ID	TRANSACTION_ID
1	6011-3703-2772-7455	95	AE-Q6jR9EV-JD391Emb-2adg2ch	TXN-NPGWnrEvbg-ZZSs1kCYi-12383996

Research: Date Range & Filtering

Run Cancel ⚙ Disconnect ⚙ Change Connection JLMFCompany_DB ▾ |

```
1  /*Question 3: Who are the users (account holders) and their respective phone numbers
2   linked to every, and linked to every purchase or transaction that occurred after April 20, 2016?*/
3
4  SELECT
5      UA.FIRST_LAST_NAME,
6      UA.USER_ID,
7      UA.USER_PHONE
8  FROM
9      USER_ACCOUNT UA
10     INNER JOIN EXPEDIA_PASSENGER_DATA EPD
11         ON EPD.PASSENGER_NAME = UA.FIRST_LAST_NAME
12     INNER JOIN USER_RECORDS UR
13         ON UR.USER_ID = UA.USER_ID
14     WHERE UR.TRANSACTION_DATE > '4/20/2016'
15 ;
```

Results Messages

	FIRST_LAST_NAME	USER_ID	USER_PHONE
1	Adam Cook	LUF120L7WI1H	426-991-2522 x689
2	Anthony Freeman	MNNW81GPFWTJ	+1-763-849-8641x846
3	Andrew Rodriguez	XRBVDN0B5BDB	001-030-260-4078x780
4	Brian Clark	03Q2KN2ZD6A5	(308) 738-4945
5	Chad Francis	EI6379UB7MR9	(970)395-1487x175
6	Cody Green	TLTXRA0E2UQX	+1-446-005-5950x4917
7	Daniel Tran	AT4PQ5KIVX2A	+1-081-971-4790x712

Research: Numerical - Average

Run Cancel ⚙ Disconnect ☰ Change Connection JLMFCompany_DB ▾ |

```
1  /*Question 4: What is the average overall rating for every Airbnb listing
2   in each city within the United States?*/
3
4  SELECT
5      FORMAT(ROUND(AVG(ABD.RATING_OVERALL),2),'0.0#') AS AVG_RATING_X_CITY,
6      HI.CITY
7  FROM
8      AIRBNB_BOOKING_DATA ABD
9  JOIN
10     HOST_INFO HI
11     ON HI.HOST_LOCATION_ID = ABD.HOST_LOCATION_ID
12 WHERE
13     HI.COUNTRY LIKE '%UNITED STATES%'
14 GROUP BY
15     HI.CITY
16 ORDER BY
17     AVG_RATING_X_CITY DESC
18 ;
```

Results Messages

	AVG_RATING_X_CITY	CITY
1	5.0	CHULA VISTA
2	5.0	HAWAII
3	5.0	LOS ANGELES
4	5.0	NEW YORK
5	4.92	SAN DIEGO
6	4.63	CHICAGO

Research: Numerical - Percentage

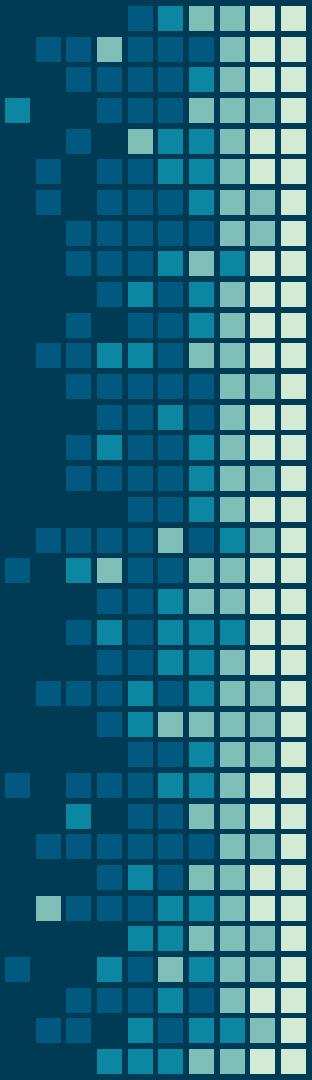
Run □ Cancel ⌂ Disconnect ⌂ Change Connection JLMCompany_DB |

```
1 /*Question 5: What is the percentage of guests per category (adults, minors and children) in each Airbnb
2 reservation (AIR_CONFIRMATION_ID)? What are the host names and arrival dates linked to every reservation?*/
3
4 SELECT
5     HI.HOST_NAME,
6     ARD.AIR_CONFIRMATION_ID,
7     ARD.START_DATE,
8     ARD.END_DATE,
9     FORMAT(ROUND(SUM(ARD.GUEST_ADULT_NO) * 100.0 / SUM(ABD.TOTAL_GUESTS),3),'0.0#') AS GUEST_ADULT_PERCENT,
10    FORMAT(ROUND(SUM(ARD.GUEST_CHILDREN_NO) * 100.0 / SUM(ABD.TOTAL_GUESTS),3),'0.0#') AS GUEST_CHILDREN_PERCENT,
11    FORMAT(ROUND(SUM(ARD.GUEST_INFANT_NO) *100.0 / SUM(ABD.TOTAL_GUESTS),3),'0.0#') AS GUEST_INFANT_PERCENT
12 FROM
13     AIRBNB_RESERVATION_DETAILS ARD
14 JOIN
15     AIRBNB_BOOKING_DATA ABD
16     ON ABD.AIR_CONFIRMATION_ID = ARD.AIR_CONFIRMATION_ID
17 JOIN
18     HOST_INFO HI
19     ON HI.HOST_ID = ABD.HOST_ID
20 WHERE
21     ARD.START_DATE > '12/31/2018' AND ARD.START_DATE < '1/1/2020'
22 GROUP BY
23     ARD.AIR_CONFIRMATION_ID, HI.HOST_NAME, ARD.START_DATE, ARD.END_DATE
24 ORDER BY
25     HI.HOST_NAME ASC
26 ;
27
```

Results Messages

	HOST_NAME	AIR_CONFIRMATION_ID	START_DATE	END_DATE	GUEST_ADULT_PERCENT	GUEST_CHILDREN_PERCENT	GUEST_INFANT_PERCENT
1	Benjamin Murray	Kh6ntql2-ZsTItI1	2019-04-02	2019-04-05	100.0	0.0	0.0
2	Caitlin Taylor	3FmV1yDU-dCkSvQ0	2019-04-12	2019-04-16	33.33	33.33	33.33
3	Caitlin Taylor	wQJecAIF-LFFYJxQ	2019-10-13	2019-10-15	25.0	75.0	0.0
4	Joseph Ritter	8NyAfaCb-KV3rYf4	2019-04-22	2019-04-28	40.0	40.0	20.0
5	Kelly Wagner	R72BrCpt-1DHrQQN	2019-01-09	2019-01-13	33.33	66.67	0.0
6	Mark Collins	d60sFwL6-5sl70St	2019-11-21	2019-11-24	100.0	0.0	0.0

Research: Numerical - Sum



```
▶ Run □ Cancel ⚙ Disconnect ☰ Change Connection JLMFCompany_DB |  
1  /*Question 6: What is the sum of every flight passenger per flight confirmation id? */  
2  
3  SELECT  
4      EPT.FLIGHT_CONFIRMATION_ID,  
5      COUNT(EPT.FLIGHT_CONFIRMATION_ID) AS PASSENGER_COUNT  
6  FROM  
7      EXPEDIA_PASSENGER_TICKETS EPT  
8  GROUP BY EPT.FLIGHT_CONFIRMATION_ID  
9  
10 UNION ALL  
11  
12 SELECT  
13     'SUM' FLIGHT_CONFIRMATION_ID,  
14     COUNT(FLIGHT_CONFIRMATION_ID)  
15  FROM  
16      EXPEDIA_PASSENGER_TICKETS  
17 ;
```

Results Messages

FLIGHT_CONFIRMATION_ID	PASSENGER_COUNT
...	2
WLUK9A	2
W0013U	7
XEAUNW	1
XHCEAD	2
XVNT9P	2
YCJYK6	2
YOK20R	5
ZEIMTV	2
ZFNCYO	3
ZGEK5M	2
ZHIWJS	2
ZRNQXW	2
ZWDDAT	2
SUM	132

Research: Trends & Patterns

Run Cancel Disconnect Change Connection JLMFCompany_DB |

```
1 /*Question 7: What are the longest round-trips and nonstop flights that have been booked?*/
2
3 SELECT
4     FLIGHT_CONFIRMATION_ID,
5     DEPT_FIRST_FLIGHT_NO,
6     DEPT_TRIP_DURATION_HR,
7     RTRN_FIRST_FLIGHT_NO,
8     RTRN_TRIP_DURATION_HR,
9     FORMAT(SUM(DEPT_TRIP_DURATION_HR) + SUM(RTRN_TRIP_DURATION_HR), '0.0#') AS TOTAL_TRIP_DURATION
10
11 FROM
12     EXPEDIA_TRIP_DETAILS
13 WHERE
14     RTRN_FLIGHT_TYPE != 'NA' AND RTRN_FLIGHT_TYPE != 'LAYOVER'
15 GROUP BY
16     FLIGHT_CONFIRMATION_ID,
17     DEPT_FIRST_FLIGHT_NO,
18     RTRN_FIRST_FLIGHT_NO,
19     DEPT_TRIP_DURATION_HR,
20     RTRN_TRIP_DURATION_HR
21 HAVING
22     COUNT(DEPT_FIRST_FLIGHT_NO) = 1 AND COUNT(DEPT_SECOND_FLIGHT_NO) = 0 AND COUNT(DEPT_THIRD_FLIGHT_NO) = 0
23 ORDER BY RTRN_TRIP_DURATION_HR DESC
24 ;
```

Results Messages

	FLIGHT_CONFIRMATION_ID	DEPT_FIRST_FLIGHT_NO	DEPT_TRIP_DURATION_HR	RTRN_FIRST_FLIGHT_NO	RTRN_TRIP_DURATION_HR	TOTAL_TRIP_DURATION
1	EQV05B	CI 836	11.33	CI 833	12.42	23.75
2	LPD3LK	AA 6204	10.5	BA 273	11.17	21.67
3	DZQ98T	EI 69	11.08	EI 68	10.25	21.33
4	HFZH50	JL 65	11.58	JL 66	10	21.58
5	TIY3M5	AM 10	8	AM 11	8.75	16.75
6	OGMJ11	AA 2674	5.15	AA 1209	5.07	10.22
7	UXI05Z	DL 1818	4.05	DL 1802	3.73	7.78
8	JKJ2J8	Y4 490	2.82	Y4 491	2.97	5.79

CONCLUSION

- The ERD's purpose is to demonstrate the relationship among the user & records, Airbnb and Expedia data tables.
- This data model is designed to maintain every data set organized, and hence avoid inconsistencies or data duplication.
- Testing the database throughout SQL queries is necessary to confirm that data was loaded correctly.
- Analyzing some research questions might help enterprises to improve their overall business based on insights, trends, or patterns.

Reference Sources

Docs.microsoft.com. (2020). *[MS-SQLPROTLP]: Technical Documents*. [online] Available at: https://docs.microsoft.com/en-us/openspecs/sql_server_protocols/ms-sqlprotlp/3432cb95-3f91-446d-86d7-b9ea0816464e [Accessed 19 Feb. 2020].

Docs.microsoft.com. (2020). *Microsoft Azure Documentation*. [online] Available at: <https://docs.microsoft.com/en-us/azure/> [Accessed 19 Feb. 2020].

Docs.microsoft.com. (2020). *SQL Server technical documentation - SQL Server*. [online] Available at: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15> [Accessed 19 Feb. 2020].

Docs.python.org. (2020). *3.8.2rc2 Documentation*. [online] Available at: <https://docs.python.org/3/> [Accessed 19 Feb. 2020].

Inside Airbnb. (2020). *Inside Airbnb. Adding data to the debate*. [online] Available at: <http://insideairbnb.com/get-the-data.html> [Accessed 19 Feb. 2020].

Faker.readthedocs.io. (2020). *Welcome to Faker's documentation! — Faker 4.0.1 documentation*. [online] Available at: <https://faker.readthedocs.io/en/master/> [Accessed 19 Feb. 2020].

Lucidchart.com. (2020). [online] Available at: <https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning> [Accessed 19 Feb. 2020].

Openflights.org. (2020). *OpenFlights: Airport and airline data*. [online] Available at: <https://openflights.org/data.html> [Accessed 19 Feb. 2020].

Oracle Help Center. (2020). *Database Documentation - Database Documentation*. [online] Available at: <https://docs.oracle.com/en/database/> [Accessed 19 Feb. 2020].

W3schools.com. (2020). *SQL Tutorial*. [online] Available at: <https://www.w3schools.com/sql/> [Accessed 19 Feb. 2020].

Worldpostalcode.com. (2020). *World Postal Code - free zip/postal code lookup*. [online] Available at: <https://worldpostalcode.com> [Accessed 19 Feb. 2020].