Finalization Phase

Build a Data Mart in SQL (DLBDSPBDM01)

Fabian Menne – Matriculation 32008377

Tutor: Musharaf Doger

07/12/2023

Project Progress:

The focus of my Airbnb database project is to create a fully functional postgresql database system that allows a company to run their business. The database consists of three major roles, namely the administrative role for the company staff, host for people offering their accommodation and guest for people interested in staying for a holiday. In the first phase, the ERD was created to visualize the different database table entities and their relations. I've learned that it's important to have a clear ERD where the relationship lines do not overlap, this was achieved using MySQL workbench tool. My database design also includes access level which more realistically reflects security within a company and accommodates for a growing business. A few tables had to be revised, some were obsolete or turned into a many-to-many lookup table using surrogate keys.

In the development phase the entity-relationship diagram got translated to the actual database model using pgAdmin4. PostgreSQL is a state-of-the-art database management system and allows for easy querying, viewing of tables and much more. After all the tables were created I started to create realistic table entries. Small tables were done manually, however to capture more complex structure between tables (e.g. when data is joined) I wrote my own python scripts to build sql insert statements that would then be used to add it to the database.

I've learned a lot about selecting, joins, statements, creating views or aggregating data in SQL that allowed me to build the database. It has been tested that all the tables properly work, can be joined using primary and foreign keys, and interesting economic aspects like tax and revenue can be calculated. The project made me understand better how SQL databases work and prepared me to more efficiently function in future projects.

Database Metadata:

To retrieve metadata from pgAdmin4 it is possible to use a SELECT statement for the pg_class where it is located. First, I run ANALYZE for each table to have the database update it's metadata statistics and reflect the current state. Then the following SQL statement it used:

```
SELECT oid as Object_ID, relname as table_name, relpages, reltuples as rows, relallvisible,
reltoastrelid as toast_object_id, relhasindex as table_has_indexes, relnatts as
num_of_columns FROM pg_class WHERE relnamespace = 2200 AND reltype != 0;
```

The database size is 34.74 MB and metadata output is shown below:

							table_has	
	oid	**table_name**	**relpages**	**rows**	**relallvisible**	**toast_oid**	**_indexes**	**#_columns**
1	16408	Allowed Operations	1	52	0	16411	True	3
2	16549	User	84	1250	84	16552	True	14
3	16395	Activity Log	7	500	0	16398	True	5
4	16423	Amenity Categories	1	30	0	0	True	2
5	16417	Amenity	1	150	0	0	True	3
6	16451	Guest	5	1000	0	0	True	2
7	16429	Booking	641	22275	481	0	True	17
8	16535	Review	48	4476	38	16539	True	7
9	16457	Host	1	200	0	0	True	1
10	16479	Payment	55	5265	55	0	True	9
11	16510	Promo	1	42	0	0	True	5
12	16502	Preferences	30	1000	30	0	True	9
13	16469	Message	1725	66716	1725	16474	True	7
14	16494	Payment Status	1	20	0	16497	True	3
15	16527	Property Type	1	20	0	16530	True	3
16	16463	Language	1	50	0	0	True	2
17	16445	Country	1	49	0	0	True	3
18	16563	Media	15	1250	15	16566	True	5
19	16544	Review Type	1	3	0	0	True	2
20	16557	User Type	1	3	0	0	True	2
21	16575	Property Amenity	14	3045	14	0	True	2
22	16590	Preferences Amenity	9	1972	9	0	True	2
23	16595	Preferences Property Type	9	2032	9	0	True	2
24	16857	Access Level	1	10	0	0	True	2
25	16873	Admin	1	50	0	0	True	2
26	16884	AccessLevelAllowedOperations	1	30	0	0	True	2
27	16585	Preferences Country	10	2062	10	0	True	2
28	16972	Property Images	2	399	0	0	False	2
29	16518	Property	27	399	27	16523	True	24
30	16992	Chat	241	22275	241	0	True	3
31	16487	Payment Method	21	1250	21	0	True	5
32	17018	Payment Method Name	1	20	0	0	True	2
33	16580	Property Review	226	41796	225	0	True	3
34	17043	property_view	0	-1	0	0	False	25

Installation instructions: Download PostgreSQL and pgAdmin4 from their official website. Next, for Linux you can use my installation script located on my github repo and follow the instructions there. Other operating systems can follow the alternative installation commands, create a test database (e.g. test) and import the database dump file from pgAdmin4 to initialize the database. All the .sql files are in the Finalization folder called /sql and python files used to generate in /python. For more information it’s highly suggested to visit the github page and read in more detail.