**Summary/Abstract**

The problem is to meet the guest and host on the same page so that guest can track their transactions, and the host will be able to track their transactions. This project emphasizes solving this problem with a database solution so that data can be persistent and the transaction can be reported or analyzed. In this project, twenty-one tables are designed for demonstration purposes and tables are connected so that technically data can be fetched from each other. The first requirement is specified then related tables are designed with their attributes, and after that Entity relationship diagram is created in the conceptual phase. The development phase contains database generation by SQL (structured data query language) statements and test purposes data will be included in each table. In the development phase, there have been developed slides to show SQL statements for “CREATE”, “INSERT” and “SELECT” commands. In the submitted “.zip” file it may be seen SQL files for installation, each SQL file is separated by a related folder name, and the “SQL\_Backup\_AirBnB” folder contains the “sql-data.sql” file which contains create and insert statements in one file so that this file can be used for installation to a database. In the finalization phase, all tables with their metadata are added and it can be seen in the table below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **TABLE\_NAME** | **TABLE\_ROWS** | **DATA\_LENGTH** | **INDEX\_LENGTH** |
| **1** | accountpayable | 20 | 16384 | 0 |
| **2** | accountreceivable | 20 | 16384 | 0 |
| **3** | accountreceivable\_has\_facilityinvoice | 20 | 16384 | 32768 |
| **4** | bank | 20 | 16384 | 32768 |
| **5** | carrier | 20 | 16384 | 32768 |
| **6** | employee | 20 | 16384 | 16384 |
| **7** | expectedincome | 20 | 16384 | 49152 |
| **8** | facility | 20 | 16384 | 0 |
| **9** | facilityadvantages | 20 | 16384 | 16384 |
| **10** | facilityarchitecture | 20 | 16384 | 16384 |
| **11** | facilityavailability | 20 | 16384 | 16384 |
| **12** | facilityinvoice | 20 | 16384 | 49152 |
| **13** | facilityinvoice\_has\_accountpayable | 20 | 16384 | 32768 |
| **14** | facilitylocation | 20 | 16384 | 16384 |
| **15** | facilityorder | 20 | 16384 | 32768 |
| **16** | facilityoverdaystayed | 20 | 16384 | 16384 |
| **17** | facilityprice | 20 | 16384 | 16384 |
| **18** | facilityrate | 20 | 16384 | 32768 |
| **19** | facilityreview | 20 | 16384 | 16384 |
| **20** | facilitysocial | 20 | 16384 | 16384 |
| **21** | facilitytaxrate | 20 | 16384 | 16384 |
| **22** | userrole | 20 | 16384 | 0 |
| **23** | usertaxid | 20 | 16384 | 16384 |

To get meta-data SQL query is used shown below,

**SELECT**

information\_schema.TABLES.TABLE\_SCHEMA,

information\_schema.TABLES.TABLE\_NAME,

information\_schema.TABLES.TABLE\_ROWS,

information\_schema.TABLES.DATA\_LENGTH,

information\_schema.TABLES.INDEX\_LENGTH,

information\_schema.TABLES.TABLE\_COMMENT

**FROM**

information\_schema.TABLES

**WHERE**

information\_schema.TABLES.TABLE\_SCHEMA **=** 'airbnb' ;

There are twenty-three tables shown in the above table but two tables are coming from many-to-many relationships so they are not included in the counting tables in the project. This project is the backbone of the backend in an application which makes the possibility of data persistency and it has the potential to develop a user interface but developing the front-end will not be emphasized. This project helped to gain practical knowledge and gain experience to design databases, specify requirements, and knowledge for CRUD (Create, Read, Update, Delete) applications in the software industry. By doing this project, search engines on the internet are used a lot to dive deep into related topics that are required to build a database.