

Abstract

When creating and implementing the Rental_Bookings database a top-down approach was used starting with the analysis of the most general matter: defining the objective of the database. Aiming to design a database that is easy to use and provides relevant details about rentals to prospective guests, is valuable to hosts and delivers business insights to internal business analysts, several roles were drafted to reflect these different user groups. These groups or roles were characterized in detail with respect to their expected activities and needs to access or manipulate.

With this information in mind the first draft of an Entity Relationship Model (ERM) was designed. Again, consistent with the top-down approach, general tables were identified first and attributes added to them. More specific tables and their attributes were derived in a next step. Subsequently, normalization was ensured. Finally, to complete the ERM draft, relationships between the tables were defined.

This drafted ERM was then implemented using MariaDB as relational database solution and DBeaver as database administration tool. During the implementation optimization potential concerning the ERM was identified and the drafted ERM as well as the data dictionary were iteratively adjusted accordingly resulting in the final ERM and database design.

First step of the implementation – and therefore the first two queries in the SQL file – is creating the database Rental_Bookings itself (after deleting a possibly existing database with identical name). This is followed by declaring independent tables in the following six queries and after these, 18 queries to create tables containing foreign key relationships finish the table creation process. After these queries the basic framework of the database is available but obviously its 24 tables are still empty.

Using the created database (27th query) the following 24 queries fulfil the purpose of filling the previously created tables with test data whereby independent tables are filled first and tables with foreign key relationships are filled second. The test data used in these queries was generated using Mockaroo Random Data Generator (<https://www.mockaroo.com/>) and partially modified. In order to satisfy foreign key integrity, the tables were initially filled in a stepwise process so that the correct foreign key values are contained in the data insert queries in the resulting SQL file.

After all tables have been populated with data, the Test Case Script can be executed. After specifying to use the created database, the next 24 queries serve to check the functionality of the database tables. Each of them is a simple select statement sometimes featuring restricting clause to select specific records only.

Finally, there are 2 more queries to display some information about the database itself. This information is also reflected in the metadata below.

Metadata:

The complete SQL Script to create this database, insert test data, provide test cases and metadata uses 78 Queries:

- 01_DDL_DML_Script:
 - 2 queries to drop and (re)create the database should it already exist
 - 1 query to specify the use of this created database
 - 24 queries are used to create each table
 - 24 queries to fill each table
- 02_Test_Cases_Script:
 - 1 query to specify the use of this created database
 - 24 queries to test every table
 - 2 queries to get information about the database itself and its tables

The resulting Rental_Bookings database includes 24 tables, each containing between 20 and 75 rows and containing between 16kB and 80kB worth of data:

table	description	row count	size in kB
banktransfer	Bank Account Details for Bank Transfers using IBAN	20	48.0000
billingaddress	Members' Billing Addresses	45	32.0000
cc_details	Credit Card Details	42	48.0000
coupons	Coupons (not member-specific)	22	16.0000
couponsjournal	Redemption Journal for Coupons	20	48.0000
currencies	Currencies with conversion rate to USD	20	32.0000
feeclasses	Fee classes for Hosts & Guests with respective fees	20	16.0000
guest	Guest details	42	48.0000
guestratings	Ratings of Guests by Hosts	24	48.0000
host	Host details	20	48.0000
invoices	Invoice Details	24	80.0000
location	Rental Location Details	24	16.0000
mainadress	Members' main addresses	42	16.0000
members	Member details	42	48.0000
paymentmeth	Payment methods including status	20	32.0000
paypal	Members' Paypal details	42	48.0000
pricing	Rental Rate details	28	48.0000
receipts	Receipt Information (Cashout to Hosts)	24	80.0000
rental	Rental details	24	64.0000
rentalbookings	Information about bookings	24	80.0000
rentalpictures	Picture Links for rentals	75	32.0000
rentalratings	Ratings of rentals by guests	24	64.0000
rentaltype	Types of rentals	20	16.0000
watchlist	Members' watchlist	21	48.0000

The whole database with its 24 tables therefore has a size of about 1 MB:

database	size in kB	table count
rental_bookings	1056.0000	24