

---

# Airbnb Data Mart

## Abstract

# 1 Introduction

The goal of the course **Build a Data Mart in SQL** was to design and implement an application for the temporal renting of lodgings in the form of a self-contained relational database. The purpose of the project was to further research relational databases and consequently the improvement of programming skills in SQL. This two-page abstract summarizes the design, the functionality and the metadata of the application.

## 1.1 Description of the Habit Tracking Application

The Airbnb data mart is a self-contained SQL database, which is developed for the use case of temporal renting of lodgings. The application features an efficient database structure and an administration interface for controlling the system. The data mart has built-in functionality for the creation of guest and host users, the management of lodgings for hosts and a reservation system for guests. Additionally the application supports multiple currencies, the writing of reviews and an extensive search functionality. To use the features of the data mart, a comprehensive set of stored procedures is supplied with the installation.

## 2 Management Functionality of the Data Mart

Like already explained, the management of the data mart is realized with the help of stored procedures. The procedures can thereby be called from the administration interface. To summarize it, the data mart has seventeen procedures for controlling all of the implemented functionality.

### 2.1 User Administration and Management of Lodgings

For basic user administration, the data mart features two procedures for the creation and deletion of users. For the creation and design of lodgings, six additional procedures are supplied. These six procedures contain functionality for the creation, deletion and the structuring of lodgings. Thereby structuring refers to the addition of rooms, rule, policies and furniture to a respective lodging.

### 2.2 Searching for a Lodging and Making a Reservation

To find a suitable lodging for a stay, the guest users can either use one of two procedures to search by a lodging's attributes or location. Two supplementary procedures allow for searching nearby locations with the help of coordinates or to display detailed information on a lodging, for example furniture or the price. When a lodging has been found, the guest can make a reservation with yet another procedure.

## 2.3 Writing Reviews and Payment Management

After a stay, an additional procedure implements functionality for writing a review and issuing a rating. Finally, the displaying and closing of open payments is realized with the last three procedures.

## 3 Data Mart Structure Metadata

To implement the aforementioned functionality, the application features seventeen different stored procedures, which operate on twenty-two unique tables. Following is a short summary of the database regarding the individual tables sizes, number of entries and the overall size of the data mart considering the test data which is supplied with the initial installation of the application.

Table Name	Description	Entries	Size
booking	Stores all reservations	40	48,0 KiB
city	List of countries from the MONDIAL database	3044	222,0 KiB
continent	List of continents from the MONDIAL database	5	32,0 KiB
country	List of countries from the MONDIAL database	238	32,0 KiB
currency	Stores the available currencies for users	20	32,0 KiB
furnishing	List of available furniture for a lodging	20	16,0 KiB
location	Stores locations of lodgings, sights and public transport	80	80,0 KiB
lodging	Stores all host-created lodgings	20	80,0 KiB
lodging_furnishing	Junction table for lodging furniture	200	32,0 KiB
lodging_policy	Junction table for lodging policies	60	32,0 KiB
lodging_room	Junction table for lodging rooms	83	32,0 KiB
lodging_rule	Junction table for lodging rules	60	32,0 KiB
paymentoption	List of all payment options	20	32,0 KiB
policy	List of all available policies for a lodging	20	16,0 KiB
publictransport	Stores unique names of public transport stations	40	32,0 KiB
review	Store all reviews from guest users	40	32,0 KiB
room	List of all available rooms for a lodging	20	16,0 KiB
rule	List of all available rules for a lodging	20	16,0 KiB
sight	Stores sights	20	48,0 KiB
state	List of states from the MONDIAL database	1434	160,0 KiB
transactions	Stores the financial information of a booking	40	48,0 KiB
users	Stores all user-related information	20	64,0 KiB

Summing the information for the individual tables yields a total of 5544 entries which allocate 1,1 MiB of disk space.

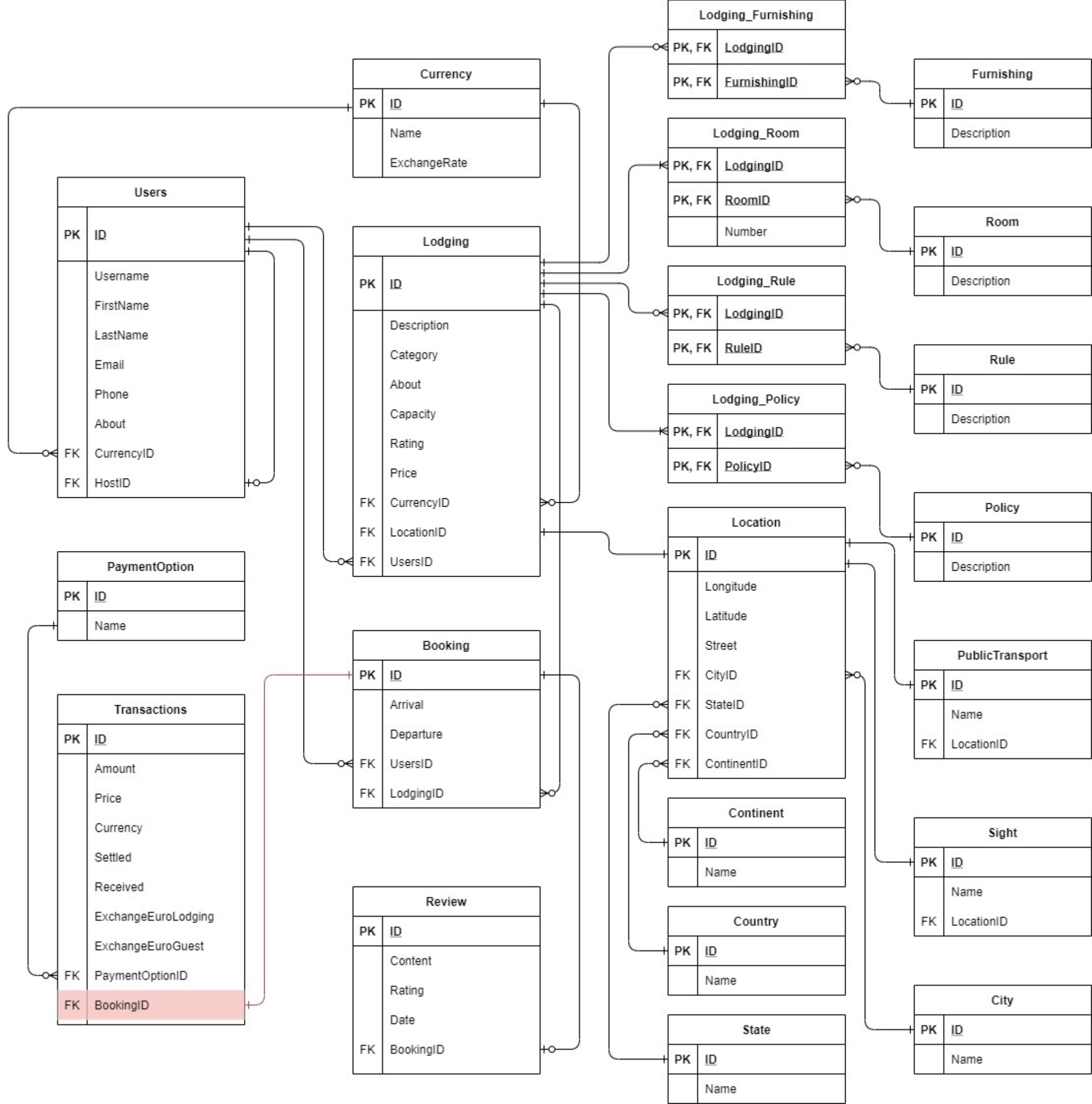


Table Name	Attribute Name	Content	Type	Constraints	PK/FK	
<b>Users</b>	ID	PK of the Users table	INT	NOT NULL AUTO_INCREMENT	PK	
	Username	Unique username of a customer	VARCHAR(128)	NOT NULL UNIQUE		
	FirstName	First name of a customer	VARCHAR(128)	NOT NULL		
	LastName	Last name of a customer	VARCHAR(128)	NOT NULL		
	Email	Customer email	VARCHAR(128)	NOT NULL		
	Phone	Customer phone number	VARCHAR(20)	NOT NULL		
	About	Short description for a host	TEXT			
	CurrencyID	Currency for price display and payments	INT	NOT NULL	FK	
HostID	Recursive FK used to identify a host user	INT		FK		
<b>Lodging</b>	ID	PK of the Lodging table	INT	NOT NULL AUTO_INCREMENT	PK	
	Description	Short description of the lodging with key details	VARCHAR(128)	NOT NULL <b>UNIQUE</b>		
	Category	Lodging category, for example apartment, flat, etc	VARCHAR(64)	NOT NULL		
	About	Detailed description about the lodging	TEXT	NOT NULL		
	Capacity	Maximal number of residents	INT	NOT NULL		
	Rating	Average rating calculated from all guest reviews	DECIMAL(2,1)			
	Price	Price per night in the hosts chosen currency	DECIMAL(7,2)	NOT NULL		
	CurrencyID	Currency of the price per night	INT	NOT NULL		FK
	LocationID	Junction to the Location table, location of the Lodging	INT	NOT NULL		FK
	UsersID	Junction to the Users table, Host/Owner of the Lodging	INT	NOT NULL	FK	
<b>Currency</b>	ID	PK of the Currency table	INT	NOT NULL AUTO_INCREMENT	PK	
	Name	Name of the currency	VARCHAR(64)	NOT NULL UNIQUE		
	ExchangeRate	Exchange rate of the currency to Euro (Updated regularly)	DECIMAL(20,9)	NOT NULL		
<b>Booking</b>	ID	PK of the Booking table	INT	NOT NULL AUTO_INCREMENT	PK	
	Arrival	Date of the check-in / arrival	DATE	NOT NULL		
	Departure	Date of the check-out / departure	DATE	NOT NULL		
	UsersID	Junction to the Users table	INT	NOT NULL		FK
	<del>TransactionID</del>	<del>Junction to the Transaction table</del>	<del>INT</del>	<del>NOT NULL</del>		<del>FK</del>
	LodgingID	Junction to the Lodging table	INT	NOT NULL	FK	
<b>Transaction</b>	ID	PK of the Transaction table	INT	NOT NULL AUTO_INCREMENT	PK	
	Amount	Summed price of the booking plus 5% provision	DECIMAL(19,2)	NOT NULL		
	Price	Lodging price per night at the time of booking	DECIMAL(19,2)	NOT NULL		
	Currency	Chosen currency of the <b>host</b>	VARCHAR(64)	NOT NULL		
	Received	Flag if the money has arrived on the bank account	BOOLEAN	NOT NULL DEFAULT FALSE		
	Settled	Flag if the money was paid to the host	BOOLEAN	NOT NULL DEFAULT FALSE		
	ExchangeEuroLodging	Exchange rate from lodging currency to Euro on the day of payment	DECIMAL(20,9)	NOT NULL		
	ExchangeEuroGuest	Exchange rate from Euro to guest currency on the day of payment	DECIMAL(20,9)	NOT NULL		
	PaymentOptionID	User chosen payment option for the transaction	INT	NOT NULL		FK

	BookingID	Booking which corresponds to the transaction	INT	NOT NULL	FK
Table Name	Attribute Name	Content	Type	Constraints	PK/FK
<b>Review</b>	ID Content Rating BookingID	PK of the Review table Content of the review written by a guest Rating of the lodging from 1-5 stars Junction to the booking which corresponds to the review	INT TEXT DECIMAL(2,1) INT	NOT NULL AUTO_INCREMENT NOT NULL NOT NULL NOT NULL	PK   FK
<b>Furnishing</b>	ID Description	PK of the Furnishing table Type of furnishing e.g. fridge, air conditioning, wlan, etc	INT VARCHAR(64)	NOT NULL AUTO_INCREMENT NOT NULL	PK
<b>Lodging_Furnishing</b>	LodgingID FurnishingID	Junction to the Lodging table Junction to the Furnishing table	INT INT	NOT NULL NOT NULL	PK/FK PK/FK
<b>Room</b>	ID Description	PK of the Room table Description of the room e.g. bathroom, livingroom, etc	INT VARCHAR(64)	NOT NULL AUTO_INCREMENT NOT NULL	PK
<b>Lodging_Room</b>	LodgingID RoomID Number	Junction to the Lodging table Junction to the Room table Number of a specific room in a lodging, e.g 2 bathrooms	INT INT TINYINT	NOT NULL NOT NULL NOT NULL <b>DEFAULT 1</b>	PK/FK PK/FK
<b>Rule</b>	ID Description	ID of the Rule table Description of a house rule, e.g. No smoking, no pets, check-in time	INT VARCHAR(64)	NOT NULL AUTO_INCREMENT NOT NULL	PK
<b>Lodging_Rule</b>	LodgingID RuleID	Junction to the Lodging table Junction to the Rule table	INT INT	NOT NULL NOT NULL	PK/FK PK/FK
<b>Policy</b>	ID Description	PK of the Policy table Contains information like damage clauses, cancellation conditions, etc	INT TEXT	NOT NULL AUTO_INCREMENT NOT NULL	PK
<b>Lodging_Policy</b>	LodgingID PolicyID	Junction to the Lodging table Junction to the Policy table	INT INT	NOT NULL NOT NULL	PK/FK PK/FK
<b>Location</b>	ID Longitude Latitude Street CityID StateID CountryID ContinentID	PK of the Location table Longitude of a location Latitude of a location Street name and house number/flat number, etc. City or Town of the location State or Province of the location Country of the location Continent of the location	INT <b>DECIMAL(17,14)</b> <b>DECIMAL(17,14)</b> VARCHAR(128) INT INT INT INT	NOT NULL AUTO_INCREMENT NOT NULL NOT NULL NOT NULL NOT NULL NOT NULL NOT NULL NOT NULL	PK    FK FK FK FK
<b>Continent</b>	ID	PK of the Continent table	INT	NOT NULL AUTO_INCREMENT	PK

Table Name	Attribute Name	Content	Type	Constraints	PK/FK
<b>Country</b>	ID	PK of the Country table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the country	VARCHAR(64)	NOT NULL	
<b>State</b>	ID	PK of the State table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the state or province	VARCHAR(128)	NOT NULL	
<b>City</b>	ID	PK of the City table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the city/town	VARCHAR(128)	NOT NULL	
<b>Sight</b>	ID	PK of the Sight table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the sight	VARCHAR(128)	NOT NULL	
	LocationID	Location of the sight	INT	NOT NULL	
<b>PublicTransport</b>	ID	PK of the PublicTransport table	INT	NOT NULL AUTO_INCREMENT	PK
	Description	Description of the transport, e.g. central train station, bus stop, etc	VARCHAR(64)	NOT NULL	
	LocationID	Location of the public transport	INT	NOT NULL	
<b>PaymentOption</b>	ID	PK of the PaymentOption table	INT	NOT NULL AUTO_INCREMENT	PK
	Name	Name of the payment option, e.g. Paypal, Credit Card	VARCHAR(64)	NOT NULL UNIQUE	