Database Design: AirBNB

A guide for a database design for an AirBNB website

Ben Brumm

www.databasestar.com

Database Design: AirBNB

This guide is a companion to my YouTube video on designing a database for the AirBNB website.

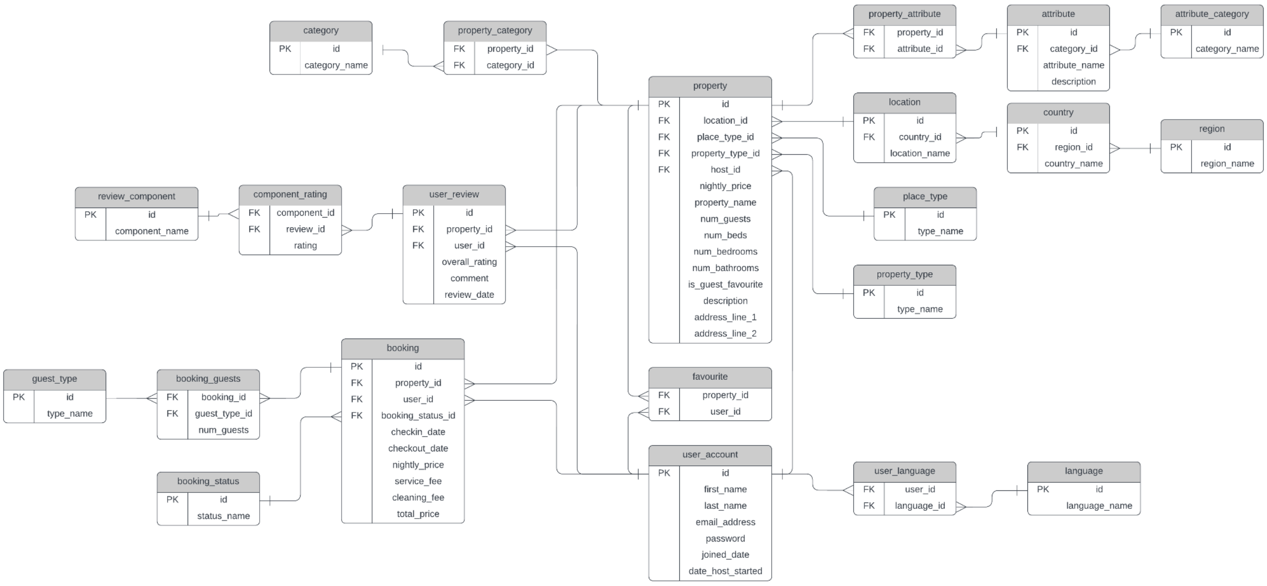
In this guide, you'll see:

- An Entity Relationship Diagram for an AirBNB website, from my YouTube video.
- An explanation of the purpose of each table and field, with sample data.
- SQL scripts to create each of these tables with some sample data.

Let's get into it.

Entity Relationship Diagram

Here's the ERD for this database:



A PNG file of this ERD is available here:

https://dbshostedfiles.s3.us-west-2.amazonaws.com/dbs/erd_airbnb.png

Database Definition

This section explains each of these tables and fields.

property

A record for a property that can be booked on the website.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
location_id	A foreign key to the location table to indicate the location for the property	2, 5, 7
place_type_id	A foreign key to the place_type table to indicate the type of place this is.	1, 2, 4
property_type_id	A foreign key to the property_type table to indicate the type of property	3, 4, 5
host_id	A foreign key to the user_account table to indicate who the host of this property is.	1, 3, 8
nightly_price	The price to stay one night in this property. It's assumed that all nights cost the same amount (no variation for weekends) and the price is in a single currency (such as US Dollars)	200 450 120
property_name	The name of the property, shown in emails and on the top of the property page	Wensley - Rustic Luxury, Great Ocean Rd Hinterland
num_guests	The maximum number of guests that can stay in the property	2, 4, 10
num_beds	The number of beds available in the property	1, 2, 4
num_bedrooms	The number of bedrooms in the property	1, 2, 6
num_bathrooms	The number of bathrooms in the property	1, 2
is_guest_favourite	Whether or not this property is a "guest favourite"	0 (No) or 1 (Yes)
description	A long description of the property, to be shown on the property page.	"Set high on the rolling hills of 80 acres The Wensley is a…"
address_line_1	The first line of the address of this	Unit 10

	property	
address_line_2	The second line of the address of this property	123 Smith St

category

A lookup table for the different categories that can be applied to a property.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
category_name	The name of the category of this property. Shown across the top of the home page.	Cabin Off-the-grid OMG Tiny Home

property_category

A joining table for properties and categories.

Column	Description	Sample Data
property_id	A foreign key to the property table to indicate which property this record relates to.	1, 2, 3
category_id	A foreign key to the category table to indicate which category this record relates to.	4, 2, 10

attribute_category

A lookup table for the different categories that attributes can belong to.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
category_name	The name of the category to group attributes. Shown on the bottom of the property page, and can be used to show different attributes in	House Rules Safety & Property Cancellation Policy

	different sections on the page.	Offerings

attribute

A list of all the attributes that are able to be applied to a property.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
category_id	A foreign key to the attribute_category table.	2, 4, 8
attribute_name	The name of the attribute that can be applied to a property	Featured In Designed By Free Cancellation Valley View Kitchen Wi-Fi
description	An optional description of the attribute	This property offers free cancellation up to 2 days before the booking date.

property_attribute

A joining table of the attributes that are applied to a property.

Column	Description	Sample Data
property_id	A foreign key to the property table.	1, 2, 3
attribute_id	A foreign key to the attribute table.	3, 6, 8

location

A lookup table for the location of a property, which may be a city or suburb.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3

country_id	A foreign key to the country table to indicate which country this location is in	1, 4, 7
location_name	The name of the location, which is shown in several places on the website.	Point Lonsdale Swansea Loceri

country

A lookup table for the country that the property is in.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
region_id	A foreign key to the region table.	3, 4, 8
country_name	The name of the country.	Australia Italy United States

region

A lookup table for the region in the world that the country is in.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
region_name	The name of the region to be shown on the website	Europe Southeast Asia Oceania

place_type

A lookup able for the "type of place", which is either a Room or an Entire Home.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
type_name	The name of this type of place.	Room Entire Home

property_type

A lookup table for the type of property, such as House or Apartment.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
type_name	The name of this type of property	House Apartment Guest House Hotel

user_account

A table for the users who access the website, make bookings, leave reviews, and host properties.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
first_name	The first name of the user, and can be used to show as the "host name" on properties.	Judy
last_name	The last name of the user	Smith
email_address	The email address for the user, which can be used to send booking details to and to log in.	john@apple.com
password	The password for the user's account on the website	An encrypted text value
joined_date	The date that the user has joined the site	2023-03-02

date_host_started	The date that the user became a host	

favourite

A record of the properties that a user has indicated as being their favourite.

Column	Description	Sample Data
property_id	A foreign key to the property table that is made a favourite.	1, 2, 3
user_id	A foreign key to the user_account table for the user who made the favourite.	2, 3, 17

user_review

A record for a review left by a user on a property.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
property_id	A foreign key to the property table that was reviewed.	1, 3, 6
user_id	A foreign key to the user_account that made the review.	1, 6, 7
overall_rating	The rating provided for the property overall for this user.	1, 2, 3, 4, 5
comment	A text comment left by the user as part of the review	"A great little property that we really enjoyed staying at…"
review_date	The date the review was made.	2023-01-04

component_rating

A joining table of the review components for a property, and the rating for each of them for the property.

Column	Description	Sample Data
component_id	A foreign key to the review_component table	1, 2, 3
review_id	A foreign key to the user_review table	1, 4, 5
rating	A number rating made for this component for this review by the user.	1, 2, 3, 4, 5

review_component

A lookup table for all of the areas that can be reviewed for a property.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
component_name	The name of the component of the review, which is displayed to the users.	Cleanliness Accuracy Check-in Communication Location Value

booking

A record for a booking for a property.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
property_id	A foreign key to the property table for the property that is booked.	2, 4, 10
user_id	A foreign key to the user_account table for the user who made the booking.	2, 4, 9
booking_status_id	A foreign key to the booking_status table to capture the status of this booking.	1, 2, 3

checkin_date	The check-in date for this booking.	2023-05-07
checkout_date	The check-out date for this booking.	2023-05-11
nightly_price	The price of a single night for this booking. It's in this table so that it is kept for a booking even if the property.nightly_price changes.	120
service_fee	The service fee for the booking.	15
cleaning_fee	The cleaning fee for the booking.	20
total_price	The total price to pay for this booking.	155

booking_guests

A joining table for the number of guests of each type for a booking.

Column	Description	Sample Data
booking_id	A foreign key to the booking table, to store the booking for this record.	1, 2, 3
guest_type_id	A foreign key to the guest_type table for the relevant guest_type.	1, 2, 3
num_guests	The number of guests of this type for this booking	2, 5, 8

guest_type

A lookup table for the type of guests on a booking, such as Adults and Children.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
type_name	The name of this guest type, which is shown to the user when making a booking.	Adults Children Infants Pets

booking_status

A lookup table for the status of a booking.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
status_name	A value that describes the status of the booking	Awaiting Approval Approved Cancelled Completed

language

A lookup table for the language spoken by a user, to assist with communication and check-in.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
language_name	The name of the spoken language	English Mandarin Italian

user_language

A joining table for the languages spoken by a user.

Column	Description	Sample Data
user_id	A foreign key to the user_account table.	1, 2, 3
language_id	A foreign key to the language table	1, 4, 5

SQL Scripts

Here is the SQL code to create the tables for this database.

The script is written for MySQL, but it can easily be modified to work on your preferred database vendor by changing the data types and removing the IF EXISTS (if your database doesn't support it).

```
CREATE DATABASE airbnb;

USE airbnb;

CREATE DATABASE airbnb;

USE airbnb;

DROP TABLE IF EXISTS booking_guests;
DROP TABLE IF EXISTS guest_type;
DROP TABLE IF EXISTS booking;
DROP TABLE IF EXISTS booking_status;
DROP TABLE IF EXISTS component_rating;
DROP TABLE IF EXISTS user_review;
DROP TABLE IF EXISTS review_component;
DROP TABLE IF EXISTS property_attribute;
DROP TABLE IF EXISTS property_category;
DROP TABLE IF EXISTS favourite;
DROP TABLE IF EXISTS property;
DROP TABLE IF EXISTS user_account;
DROP TABLE IF EXISTS property_type;
DROP TABLE IF EXISTS place_type;
DROP TABLE IF EXISTS location;
DROP TABLE IF EXISTS country;
DROP TABLE IF EXISTS region;
DROP TABLE IF EXISTS attribute;
DROP TABLE IF EXISTS attribute_category;
DROP TABLE IF EXISTS category;


CREATE TABLE category (
     id INT,
     category_name VARCHAR(200),
     CONSTRAINT pk_category PRIMARY KEY (id)
);

CREATE TABLE attribute_category (
     id INT,
     category_name VARCHAR(200),
     CONSTRAINT pk_attrcat PRIMARY KEY (id)
);

CREATE TABLE attribute (
     id INT,
     category_id INT,
     attribute_name VARCHAR(200),
     description VARCHAR(500),
     CONSTRAINT pk_attribute PRIMARY KEY (id),
     CONSTRAINT fk_att_category
     FOREIGN KEY (category_id) REFERENCES attribute_category (id)
);

CREATE TABLE region (
     id INT,
     region_name VARCHAR(200),
     CONSTRAINT pk_region PRIMARY KEY (id)
);

CREATE TABLE country (
     id INT,
     region_id INT,
     country_name VARCHAR(200),
     CONSTRAINT pk_country PRIMARY KEY (id),
      CONSTRAINT fk_country_reg
     FOREIGN KEY (region_id) REFERENCES region (id)
);

CREATE TABLE location (
     id INT,
     country_id INT,
     location_name VARCHAR(200),
     CONSTRAINT pk_location PRIMARY KEY (id),
     CONSTRAINT fk_location_country
     FOREIGN KEY (country_id) REFERENCES country (id)
);

CREATE TABLE place_type (
     id INT,
     type_name VARCHAR(200),
     CONSTRAINT pk_placetype PRIMARY KEY (id)
);

CREATE TABLE property_type (
     id INT,
     type_name VARCHAR(200),
     CONSTRAINT pk_propertytype PRIMARY KEY (id)
);
```

```
CREATE TABLE user_account (
	id INT,
	first_name VARCHAR(300),
	last_name VARCHAR(300),
	email_address VARCHAR(350),
	password VARCHAR(200),
	joined_date DATE,
	date_host_started DATE,
	CONSTRAINT pk_user PRIMARY KEY (id)
);

CREATE TABLE property (
	id INT,
	location_id INT,
	place_type_id INT,
	property_type_id INT,
	host_id INT,
	nightly_price INT,
	property_name VARCHAR(200),
	num_guests INT,
	num_beds INT,
	num_bedrooms INT,
	num_bathrooms INT,
	is_guest_favourite INT,
	description VARCHAR(2000),
	address_line_1 VARCHAR(500),
	address_line_2 VARCHAR(500),
	CONSTRAINT pk_property PRIMARY KEY (id),
	CONSTRAINT fk_property_loc
	FOREIGN KEY (location_id) REFERENCES location (id),
	CONSTRAINT fk_property_placetype
	FOREIGN KEY (place_type_id) REFERENCES place_type (id),
	CONSTRAINT fk_property_proptype
	FOREIGN KEY (property_type_id) REFERENCES property_type (id),
	CONSTRAINT fk_property_host
	FOREIGN KEY (host_id) REFERENCES user_account (id)
);

CREATE TABLE favourite (
	property_id INT,
	user_id INT,
	CONSTRAINT fk_fav_prop
	FOREIGN KEY (property_id) REFERENCES property (id),
	CONSTRAINT fk_fav_user
	FOREIGN KEY (user_id) REFERENCES user_account (id)
);

CREATE TABLE property_category (
      property_id INT,
      category_id INT,
      CONSTRAINT fk_propertycat_prop
      FOREIGN KEY (property_id) REFERENCES property (id),
      CONSTRAINT fk_propertycat_cat
      FOREIGN KEY (category_id) REFERENCES category (id)
);

CREATE TABLE property_attribute (
      property_id INT,
      attribute_id INT,
      CONSTRAINT fk_propcat_prop
      FOREIGN KEY (property_id) REFERENCES property (id),
      CONSTRAINT fk_propcat_cat
      FOREIGN KEY (attribute_id) REFERENCES attribute (id)
);

CREATE TABLE review_component (
      id INT,
      component_name VARCHAR(100),
      CONSTRAINT pk_reviewcomp PRIMARY KEY (id)
);

CREATE TABLE user_review (
      id INT,
      property_id INT,
      user_id INT,
      overall_rating INT,
      comment VARCHAR(2000),
      review_date DATE,
      CONSTRAINT pk_userreview PRIMARY KEY (id),
      CONSTRAINT fk_userreview_prop
      FOREIGN KEY (property_id) REFERENCES property (id),
      CONSTRAINT fk_userreview_user
      FOREIGN KEY (user_id) REFERENCES user_account (id)
);

CREATE TABLE component_rating (
      component_id INT,
      review_id INT,
      rating INT,
      CONSTRAINT fk_comprat_comp
      FOREIGN KEY (component_id) REFERENCES review_component (id),
     CONSTRAINT fk_comprat_review
      FOREIGN KEY (review_id) REFERENCES user_review (id)
);
```

```
CREATE TABLE booking_status (
     id INT,
     status_name VARCHAR(50),
     CONSTRAINT pk_bookingstatus PRIMARY KEY (id)
);

CREATE TABLE booking (
     id INT,
     property_id INT,
     user_id INT,
     booking_status_id INT,
     checkin_date DATE,
     checkout_date DATE,
     nightly_price INT,
     service_fee INT,
     cleaning_fee INT,
     total_price INT,
     CONSTRAINT pk_booking PRIMARY KEY (id),
     CONSTRAINT fk_booking_prop
     FOREIGN KEY (property_id) REFERENCES booking (id),
     CONSTRAINT fk_booking_user
     FOREIGN KEY (user_id) REFERENCES user_account (id),
     CONSTRAINT fk_booking_bookstatus
     FOREIGN KEY (booking_status_id) REFERENCES booking_status (id)
);


CREATE TABLE guest_type (
     id INT,
     type_name VARCHAR(50),
     CONSTRAINT pk_guesttype PRIMARY KEY (id)
);

CREATE TABLE booking_guests (
     booking_id INT,
     guest_type_id INT,
     num_guests INT,
     CONSTRAINT fk_bkguest_booking
     FOREIGN KEY (booking_id) REFERENCES booking (id),
     CONSTRAINT fk_bkguest_guesttype
     FOREIGN KEY (guest_type_id) REFERENCES guest_type (id)
);
```

Conclusion

I hope you found this guide useful. If you have any questions or issues with it, let me know at ben@databasestar.com.

Thanks,

Ben Brumm

www.DatabaseStar.com