Creating the database

First the database needs to be created. I did this simply via the create database command:

```
CREATE DATABASE airbnb;
use airbnb;
```

Structure

With the database created, it is time to create, alter and insert into the entity tables. Each entity will be presented according to the following structure:

1. Create Statements
2. Alter Statements
3. Example Insert Statements for Test Case
4. Screenshot of Test Case Data within the Database

Accessibility.

An entity containing data on how fast public transport hotspots can be reached from the property (with public transport)?

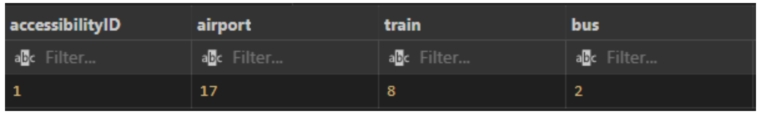
1.

```
CREATE TABLE Accessibility
(
  accessibilityID INT NOT NULL AUTO_INCREMENT,
  airport         INT NULL    ,
  train           INT NULL    ,
  bus             INT NULL    ,
  PRIMARY KEY (accessibilityID)
);
```

2. No alter statements

3.

```
INSERT INTO Accessibility (airport, train, bus)
VALUES (17, 8, 2);
```



accessibilityID	airport	train	bus
1	17	8	2

Amenities.

An entity containing what kind of amenities are available to pick from on the website.

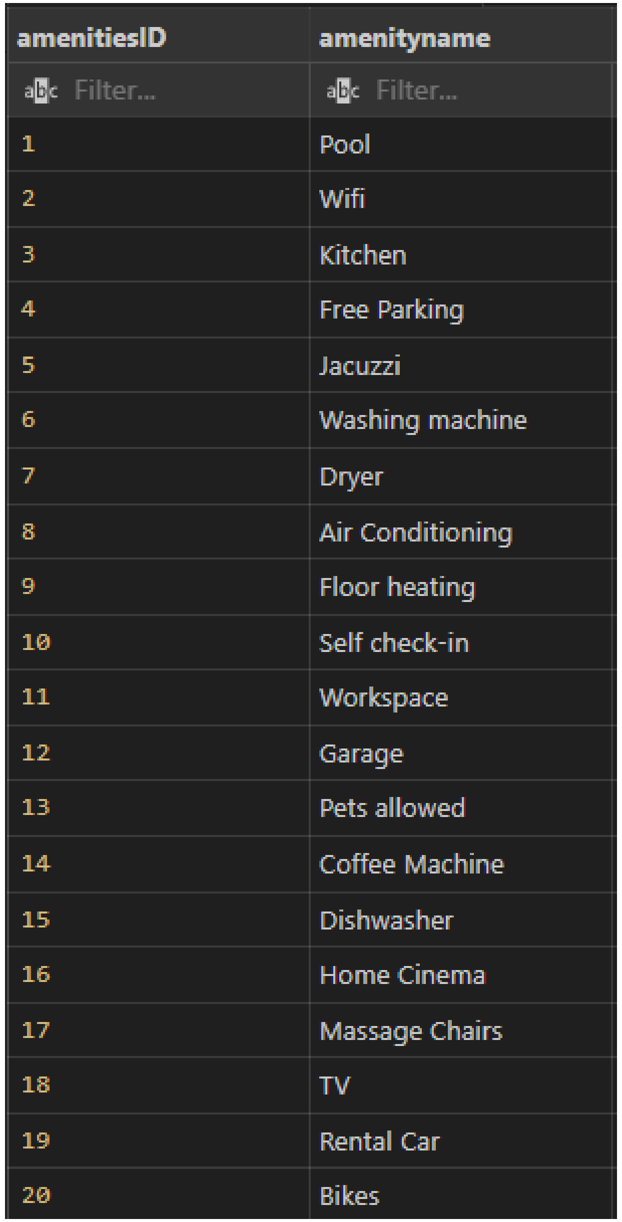
1.

```
CREATE TABLE Amenities
(
  amenitiesID INT      NOT NULL AUTO_INCREMENT,
  amenityname VARCHAR(100) NOT NULL,
  PRIMARY KEY (amenitiesID)
);
```

2. No alter statements

3.

```
INSERT INTO Amenities (amenityname)
VALUES ('Pool'), ('Wifi'), ('Kitchen'), ('Free Parking'),
('Jacuzzi'), ('Washing machine'), ('Dryer'), ('Air
Conditioning'), ('Floor heating'), ('Self check-in’),
('Workspace'), ('Garage'), ('Pets allowed'), ('Coffee
Machine'), ('Dishwasher'), ('Home Cinema'), ('Massage
Chairs'), ('TV'), ('Rental Car'), ('Bikes’);
```



amenitiesID	amenityname
Filter...	Filter...
1	Pool
2	Wifi
3	Kitchen
4	Free Parking
5	Jacuzzi
6	Washing machine
7	Dryer
8	Air Conditioning
9	Floor heating
10	Self check-in
11	Workspace
12	Garage
13	Pets allowed
14	Coffee Machine
15	Dishwasher
16	Home Cinema
17	Massage Chairs
18	TV
19	Rental Car
20	Bikes

AmenitiesAndProperties.

A relational table that links the amenities to the property.

1.

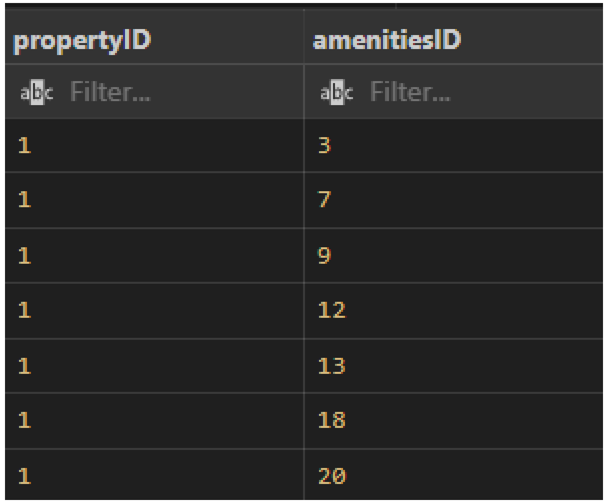
```
CREATE TABLE AmenitiesAndProperties
(
  propertyID  INT NOT NULL,
  amenitiesID INT NOT NULL
);
```

2.

```
ALTER TABLE AmenitiesAndProperties
  ADD CONSTRAINT FK_Amenities_TO_AmenitiesAndProperties
    FOREIGN KEY (amenitiesID)
    REFERENCES Amenities (amenitiesID);

ALTER TABLE AmenitiesAndProperties
  ADD CONSTRAINT FK_Properties_TO_AmenitiesAndProperties
    FOREIGN KEY (propertyID)
    REFERENCES Properties (propertyID);
```

3.

```
INSERT INTO AmenitiesAndProperties (propertyID, amenitiesID)
VALUES (1, 3), (1, 7), (1, 9), (1, 12), (1, 13), (1, 18), (1, 20);
```



propertyID	amenitiesID
1	3
1	7
1	9
1	12
1	13
1	18
1	20

Bookings.

An entity containing important booking data (property, guest and the booking dates).

1.

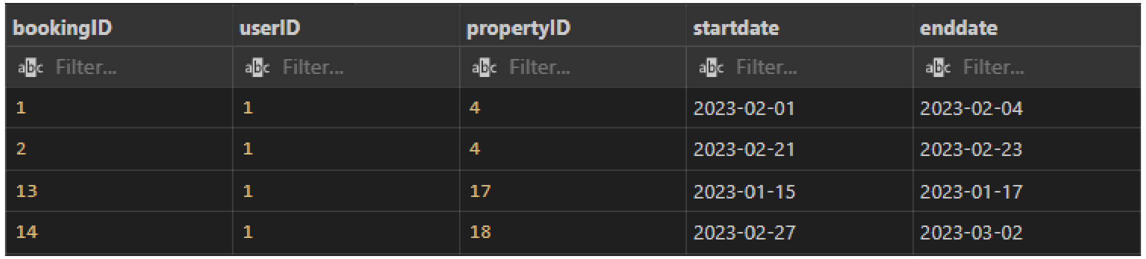
```
CREATE TABLE Bookings
(
  bookingID  INT  NOT NULL AUTO_INCREMENT,
  userID     INT  NOT NULL,
  propertyID INT  NOT NULL,
  startdate  DATE NOT NULL,
  enddate    DATE NOT NULL,
  PRIMARY KEY (bookingID)
);
```

2.

```
ALTER TABLE Bookings
  ADD CONSTRAINT FK_GuestDetails_TO_Bookings
    FOREIGN KEY (userID)
    REFERENCES GuestDetails (userID);
ALTER TABLE Bookings
  ADD CONSTRAINT FK_Properties_TO_Bookings
    FOREIGN KEY (propertyID)
    REFERENCES Properties (propertyID);
```

3.

```
INSERT INTO Bookings (userID, propertyID, startdate, enddate)

VALUES (1, 4, STR_TO_DATE('2023-02-01', '%Y-%m-%d'), STR_TO_DATE('2023-02-04',
'%Y-%m-%d')), (1, 4, STR_TO_DATE('2023-02-21', '%Y-%m-%d'), STR_TO_DATE('2023-02-23', '%Y-%m-
%d')),(1, 17, STR_TO_DATE('2023-01-15', '%Y-%m-%d'), STR_TO_DATE('2023-01-17',
'%Y-%m-%d')), (1, 18, STR_TO_DATE('2023-02-27', '%Y-%m-%d');
```



bookingID	userID	propertyID	startdate	enddate
Filter...	Filter...	Filter...	Filter...	Filter...
1	1	4	2023-02-01	2023-02-04
2	1	4	2023-02-21	2023-02-23
13	1	17	2023-01-15	2023-01-17
14	1	18	2023-02-27	2023-03-02

BuildingType.

An entity containing a collection of building types to select from.

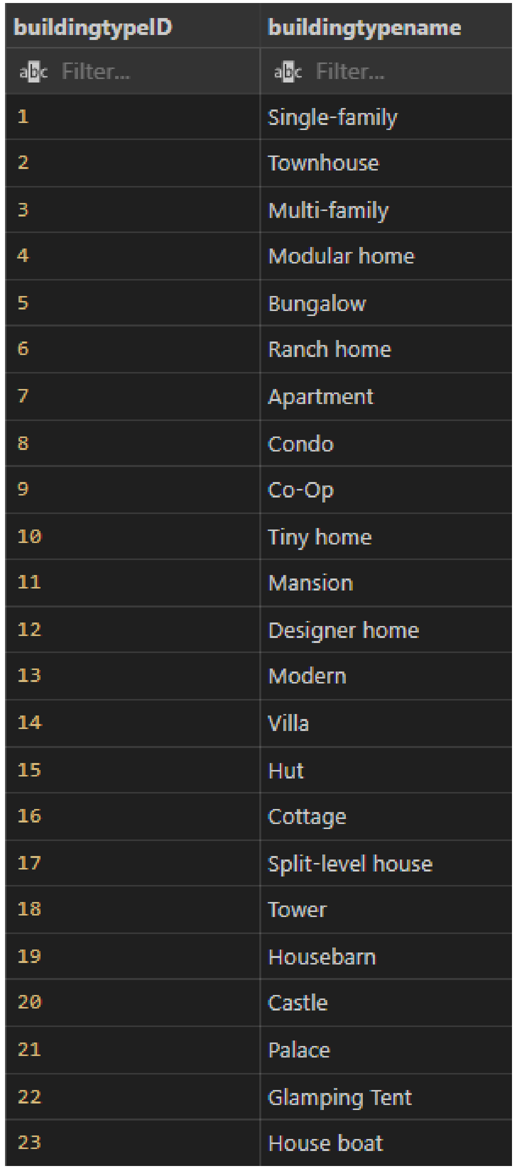
1.

```
CREATE TABLE BuildingType
(
  buildingtypeID   INT NOT NULL AUTO_INCREMENT,
  buildingtypename VARCHAR(100) NOT NULL,
  PRIMARY KEY (buildingtypeID)
);
```

2. No alter statements

3.

```
INSERT INTO BuildingType (buildingtypename)
VALUES ('Single-family'), ('Townhouse'), ('Multi-family'),
('Modular home'), ('Bungalow'), ('Ranch home'),
('Apartment'), ('Condo'), ('Co-Op'), ('Tiny home'),
('Mansion'), ('Designer home'), ('Modern'), ('Villa'),
('Hut'), ('Cottage'), ('Split-level house'), ('Tower'),
('Housebarn'), ('Castle'), ('Palace'), ('Glamping Tent'),
('House boat');
```



buildingtypeID	buildingtypename
Filter...	Filter...
1	Single-family
2	Townhouse
3	Multi-family
4	Modular home
5	Bungalow
6	Ranch home
7	Apartment
8	Condo
9	Co-Op
10	Tiny home
11	Mansion
12	Designer home
13	Modern
14	Villa
15	Hut
16	Cottage
17	Split-level house
18	Tower
19	Housebarn
20	Castle
21	Palace
22	Glamping Tent
23	House boat

BuildingTypeAndProperties.

A relational table that links the properties to their building type.

1.

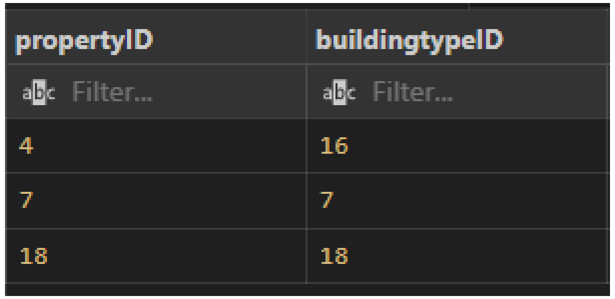
```
CREATE TABLE BuildingTypeAndProperties
(
  propertyID     INT NOT NULL,
  buildingtypeID INT NOT NULL
);
```

2.

```
ALTER TABLE BuildingTypeAndProperties
  ADD CONSTRAINT FK_Properties_TO_BuildingTypeAndProperties
    FOREIGN KEY (propertyID)
    REFERENCES Properties (propertyID);

ALTER TABLE BuildingTypeAndProperties
  ADD CONSTRAINT FK_BuildingType_TO_BuildingTypeAndProperties
    FOREIGN KEY (buildingtypeID)
    REFERENCES BuildingType (buildingtypeID);
```

3.

```
INSERT INTO BuildingTypeAndProperties (propertyID, buildingtypeID)
VALUES (4, 16), (7, 7), (18, 18);
```



propertyID	buildingtypeID
4	16
7	7
18	18

Cities.

An entity containing the names of all cities in which the business operates.

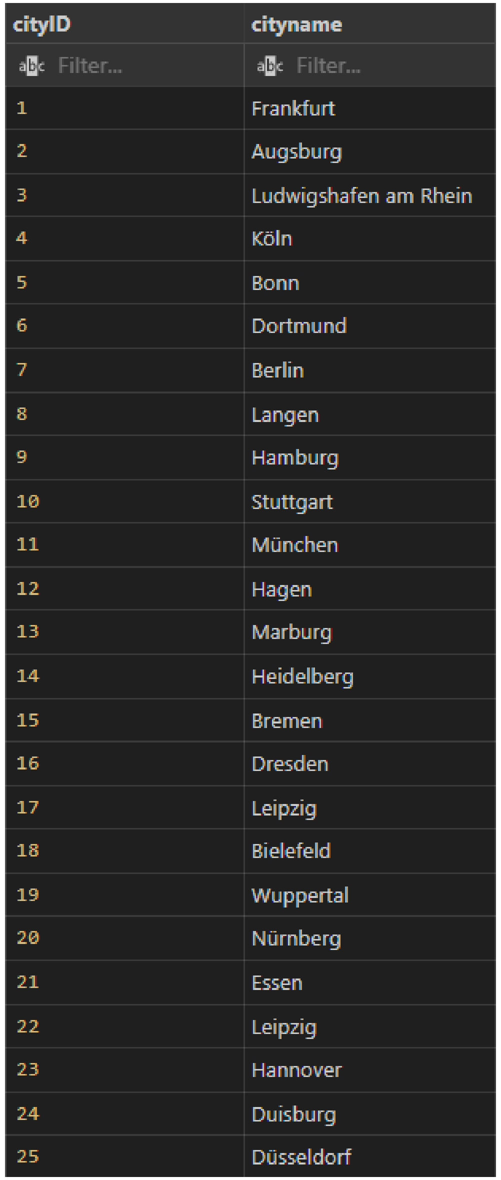
1.

```
CREATE TABLE Cities
(
  cityID   INT     NOT NULL AUTO_INCREMENT,
  cityname VARCHAR(100) NOT NULL,
  PRIMARY KEY (cityID)
);
```

2. No alter statements

3.

```
INSERT INTO Cities (cityname)
VALUES ('Frankfurt'), ('Augsburg'), ('Ludwigshafen am
Rhein'), ('Köln'), ('Bonn'), ('Dortmund'), ('Berlin'),
('Langen'), ('Hamburg'), ('Stuttgart’), ('München'),
('Hagen'), ('Marburg'), ('Heidelberg'), ('Bremen'),
('Dresden'), ('Leipzig'), ('Bielefeld'), ('Wuppertal'),
('Nürnberg'), ('Essen'), ('Leipzig'), ('Hannover'),
('Duisburg'), ('Düsseldorf');
```



cityID	cityname
Filter...	Filter...
1	Frankfurt
2	Augsburg
3	Ludwigshafen am Rhein
4	Köln
5	Bonn
6	Dortmund
7	Berlin
8	Langen
9	Hamburg
10	Stuttgart
11	München
12	Hagen
13	Marburg
14	Heidelberg
15	Bremen
16	Dresden
17	Leipzig
18	Bielefeld
19	Wuppertal
20	Nürnberg
21	Essen
22	Leipzig
23	Hannover
24	Duisburg
25	Düsseldorf

Countries.

An entity containing the names of all countries in which the business operates.

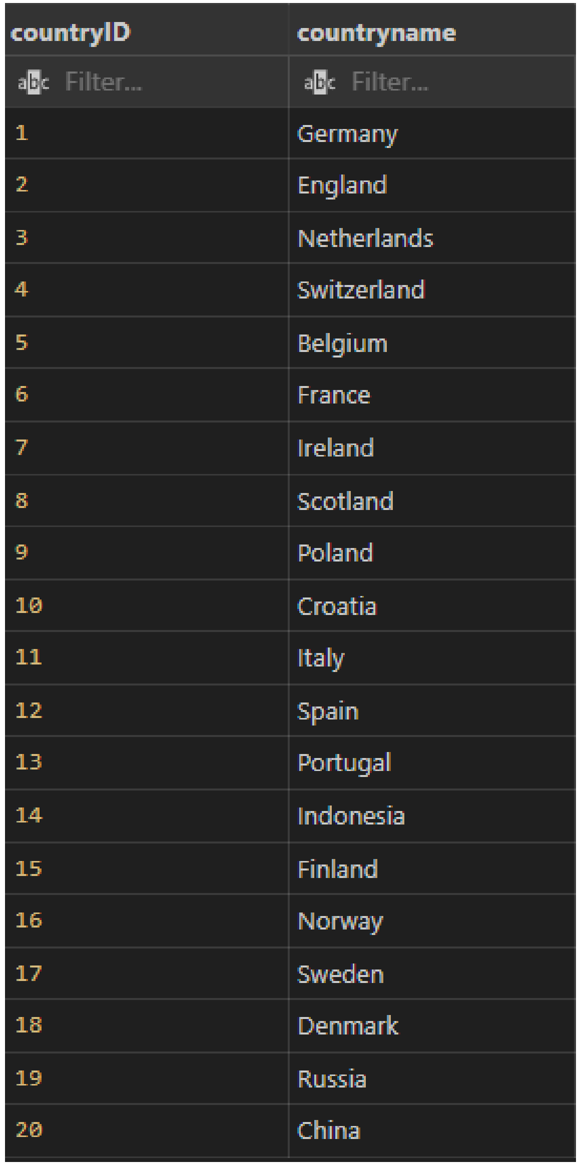
1.

```
CREATE TABLE Countries
(
  countryID   INT      NOT NULL AUTO_INCREMENT,
  countryname VARCHAR(100) NOT NULL,
  PRIMARY KEY (countryID)
);
```

2. No alter statements

3.

```
INSERT INTO Countries (countryname)
VALUES ('Germany'), ('England'),
('Netherlands'), ('Switzerland'), ('Belgium'),
('France'), ('Ireland'), ('Scotland'),
('Poland'), ('Croatia'), ('Italy'), ('Spain'),
('Portugal'), ('Indonesia'), ('Finland'),
('Norway'), ('Sweden'), ('Denmark'),
('Russia'), ('China');
```



countryID	countryname
abc Filter...	abc Filter...
1	Germany
2	England
3	Netherlands
4	Switzerland
5	Belgium
6	France
7	Ireland
8	Scotland
9	Poland
10	Croatia
11	Italy
12	Spain
13	Portugal
14	Indonesia
15	Finland
16	Norway
17	Sweden
18	Denmark
19	Russia
20	China

GuestReviews.

An entity containing reviews of the guest(s) and how they treated the property.

1.

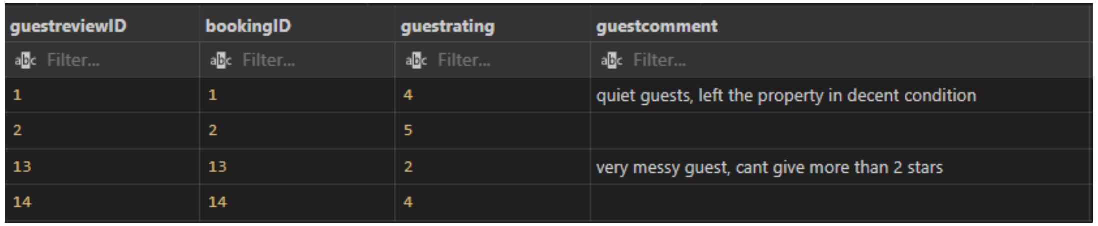
```
CREATE TABLE GuestReviews
(
  guestreviewID INT  NOT NULL AUTO_INCREMENT,
  bookingID     INT  NOT NULL,
  guestrating        INT  NOT NULL,
  guestcomment       TEXT NULL     ,
  PRIMARY KEY (guestreviewID)
);
```

2.

```
ALTER TABLE GuestReviews
  ADD CONSTRAINT FK_Bookings_TO_GuestReviews
    FOREIGN KEY (bookingID)
    REFERENCES Bookings (bookingID);
```

3.

```
INSERT INTO GuestReviews (bookingID, guestrating, guestcomment)
VALUES (1, 4, 'quiet guests, left the property in decent
condition'), (2, 5, ''), (13, 2, 'very messy guest, cant give more than 2
stars'), (14, 4, '');
```



guestreviewID	bookingID	guestrating	guestcomment
1	1	4	quiet guests, left the property in decent condition
2	2	5	
13	13	2	very messy guest, cant give more than 2 stars
14	14	4	

GuestDetails.

An entity containing additional info about users that want to book properties.

1.

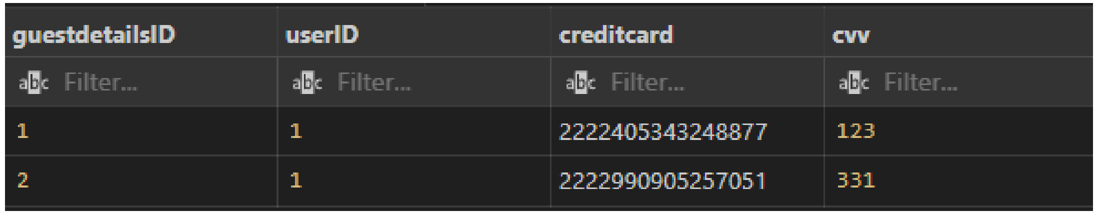
```
CREATE TABLE GuestDetails
(
  guestdetailsID INT NOT NULL AUTO_INCREMENT,
  userID      INT NOT NULL,
  creditcard BIGINT NOT NULL,
  cvv         INT NOT NULL,
  PRIMARY KEY (guestdetailsID)
);
```

2.

```
ALTER TABLE GuestDetails
  ADD CONSTRAINT FK_Users_TO_GuestDetails
    FOREIGN KEY (userID)
    REFERENCES Users (userID);
```

3.

```
INSERT INTO GuestDetails (userID, creditcard, cvv)
VALUES (1, 2222405343248877, 123), (1, 2222990905257051, 331)
```

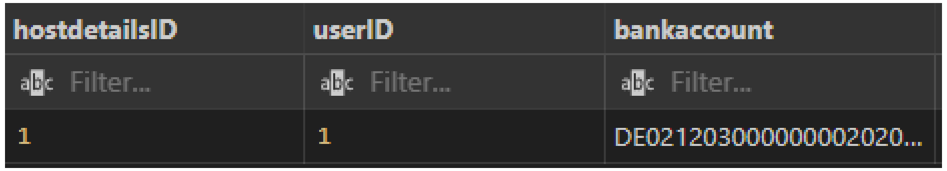


guestdetailsID	userID	creditcard	cvv
1	1	2222405343248877	123
2	1	2222990905257051	331

HostDetails.

An entity containing additional information about users that want to host properties.

```
1.
CREATE TABLE HostDetails
(
  hostdetailsID INT NOT NULL AUTO_INCREMENT,
  userID        INT      NOT NULL,
  bankaccount VARCHAR(100) NOT NULL,
  PRIMARY KEY (hostdetailsID)
);
2.
ALTER TABLE HostDetails
  ADD CONSTRAINT FK_Users_TO_HostDetails
    FOREIGN KEY (userID)
    REFERENCES Users (userID);
3.
INSERT INTO HostDetails (userID, bankaccount)
VALUES (1, 'DE02120300000000202051’);
```



hostdetailsID	userID	bankaccount
1	1	DE021203000000002020...

Languages.

An entity containing all languages available to link to your profile.

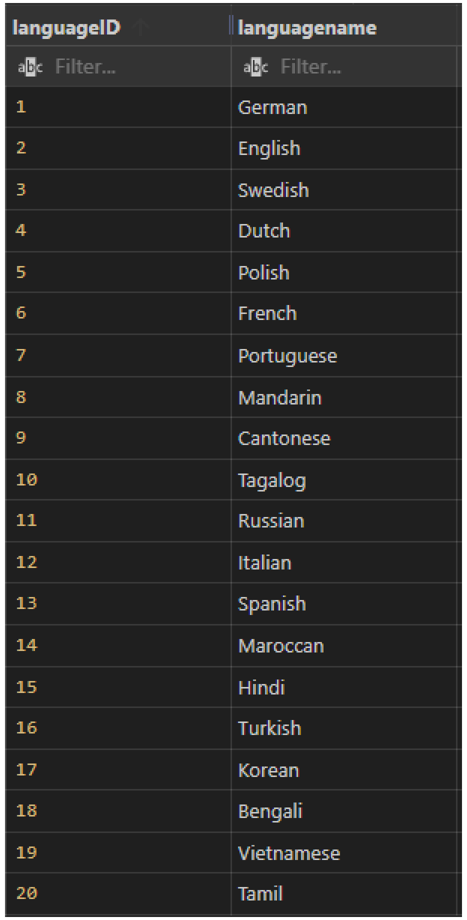
1.

```
CREATE TABLE Languages
(
  languageID   INT      NOT NULL AUTO_INCREMENT,
  languagename VARCHAR(100) NOT NULL,
  PRIMARY KEY (languageID)
);
```

2. No alter statements

3.

```
INSERT INTO Languages (languagename)
VALUES ('German'), ('English'), ('Swedish'),
('Dutch'), ('Polish'), ('French'),
('Portuguese'), ('Mandarin'), ('Cantonese'),
('Tagalog'), ('Russian'), ('Italian'),
('Spanish'), ('Maroccan'), ('Hindi'),
('Turkish'), ('Korean'), ('Bengali'),
('Vietnamese'), ('Tamil');
```

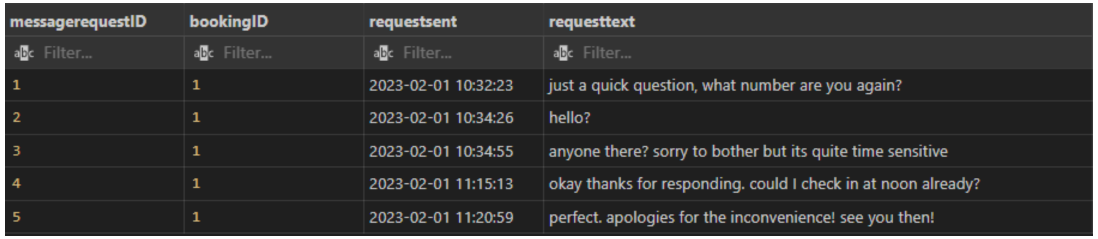


languageID	languagename
1	German
2	English
3	Swedish
4	Dutch
5	Polish
6	French
7	Portuguese
8	Mandarin
9	Cantonese
10	Tagalog
11	Russian
12	Italian
13	Spanish
14	Maroccan
15	Hindi
16	Turkish
17	Korean
18	Bengali
19	Vietnamese
20	Tamil

MessageRequests.

An entity containing messages sent from guests to their hosts.

```
1.
CREATE TABLE MessageRequests
(
  messagerequestID   INT      NOT NULL AUTO_INCREMENT,
  bookingID     INT      NOT NULL,
  requestsent     DATETIME NOT NULL,
  requesttext TEXT      NOT NULL,
  PRIMARY KEY (messagerequestID)
);
2.
ALTER TABLE MessageRequests
  ADD CONSTRAINT FK_Bookings_TO_MessageRequests
    FOREIGN KEY (bookingID)
    REFERENCES Bookings (bookingID);
3.
INSERT INTO MessageRequests (bookingID, requestsent, requesttext)
VALUES (1, STR_TO_DATE('2023-02-01 10:32:23', '%Y-%m-%d %H:%i:%s'), 'just a quick
question, what number are you again?'), (1, STR_TO_DATE('2023-02-01 10:34:26', '%Y-
%m-%d %H:%i:%s'), 'hello?'), (1, STR_TO_DATE('2023-02-01 10:34:55', '%Y-%m-%d
%H:%i:%s'), 'anyone there? sorry to bother but its quite time sensitive'), (1,
STR_TO_DATE('2023-02-01 11:15:13', '%Y-%m-%d %H:%i:%s'), 'okay thanks for responding.
could I check in at noon already?'), (1, STR_TO_DATE('2023-02-01 11:20:59', '%Y-%m-%d
%H:%i:%s'), 'perfect. apologies for the inconvenience! see you then!’);
```



messagerequestID	bookingID	requestsent	requesttext
1	1	2023-02-01 10:32:23	just a quick question, what number are you again?
2	1	2023-02-01 10:34:26	hello?
3	1	2023-02-01 10:34:55	anyone there? sorry to bother but its quite time sensitive
4	1	2023-02-01 11:15:13	okay thanks for responding. could I check in at noon already?
5	1	2023-02-01 11:20:59	perfect. apologies for the inconvenience! see you then!

MessageReplies.

An entity containing messages sent from host back to the guest, answering any questions or requests.

1.

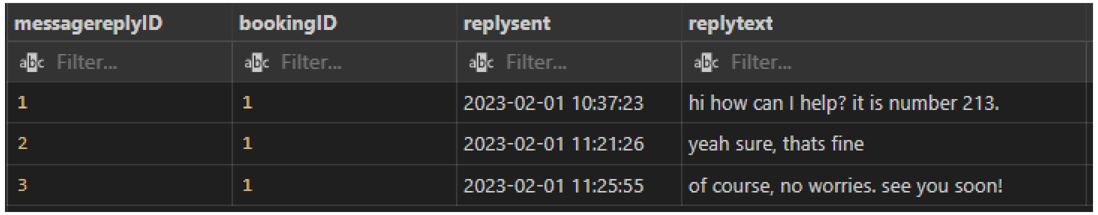
```
CREATE TABLE MessageReplies
(
  messagereplyID   INT      NOT NULL AUTO_INCREMENT,
  bookingID      INT      NOT NULL,
  replysent     DATETIME NOT NULL,
  replytext TEXT      NOT NULL,
  PRIMARY KEY (messagereplyID)
);
```

2.

```
ALTER TABLE MessageReplies
  ADD CONSTRAINT FK_Bookings_TO_MessageReplies
    FOREIGN KEY (bookingID)
    REFERENCES Bookings (bookingID);
```

3.

```
INSERT INTO MessageReplies (bookingID, replysent, replytext)

VALUES (1, STR_TO_DATE('2023-02-01 10:37:23', '%Y-%m-%d %H:%i:%s'), 'hi how
can I help? it is number 213.'), (1, STR_TO_DATE('2023-02-01 11:21:26', '%Y-
%m-%d %H:%i:%s'), 'yeah sure, thats fine'), (1, STR_TO_DATE('2023-02-01
11:25:55', '%Y-%m-%d %H:%i:%s'), 'of course, no worries. see you soon!’);
```



messagereplyID	bookingID	replysent	replytext
abc Filter...	abc Filter...	abc Filter...	abc Filter...
1	1	2023-02-01 10:37:23	hi how can I help? it is number 213.
2	1	2023-02-01 11:21:26	yeah sure, thats fine
3	1	2023-02-01 11:25:55	of course, no worries. see you soon!

Payment.

An entity containing the payment details of each booking.

1.

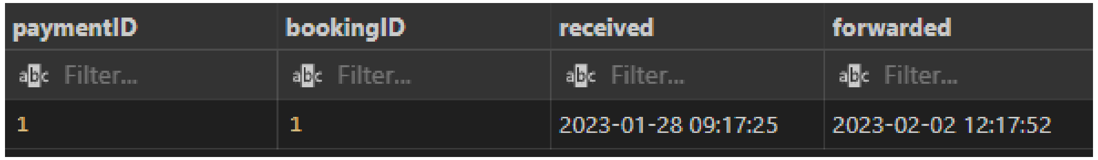
```
CREATE TABLE Payment
(
  paymentID INT      NOT NULL AUTO_INCREMENT,
  bookingID INT      NOT NULL,
  received  DATETIME NULL    ,
  forwarded DATETIME NULL    ,
  PRIMARY KEY (paymentID)
);
```

2.

```
ALTER TABLE Payment
  ADD CONSTRAINT FK_Bookings_TO_Payment
    FOREIGN KEY (bookingID)
    REFERENCES Bookings (bookingID);
```

3.

```
INSERT INTO Payment (bookingID, received, forwarded)
VALUES (1, STR_TO_DATE('2023-01-28 09:17:25', '%Y-%m-%d %H:%i:%s’),
 STR_TO_DATE('2023-02-02 12:17:52', '%Y-%m-%d %H:%i:%s’));
```



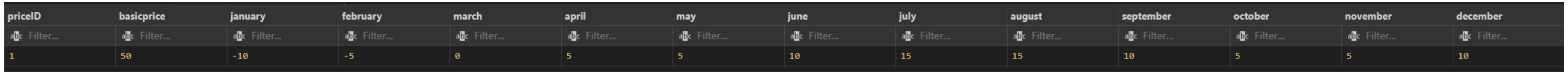
paymentID	bookingID	received	forwarded
Filter...	Filter...	Filter...	Filter...
1	1	2023-01-28 09:17:25	2023-02-02 12:17:52

Price.

An entity containing basic price per night and modifiers for each individual month.

1.

```
CREATE TABLE Price
(
  priceID    INT   NOT NULL AUTO_INCREMENT,
  basicprice INT   NOT NULL     ,
  january    INT   NULL     ,
  february   INT   NULL     ,
  march      INT   NULL     ,
  april      INT   NULL     ,
  may        INT   NULL     ,
  june       INT   NULL     ,
  july       INT   NULL     ,
  august     INT   NULL     ,
  september  INT   NULL     ,
  october    INT   NULL     ,
  november   INT   NULL     ,
  december   INT   NULL     ,
  PRIMARY KEY (priceID)
);
```



priceID	basicprice	january	february	march	april	may	june	july	august	september	october	november	december
Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	50	-10	-5	0	5	5	10	15	15	10	5	5	10

2. No alter statements

3.

```
INSERT INTO Price (basicprice, january, february, march, april, may,
june, july, august, september, october, november, december)

Values (50, -10, -5, 0, 5, 5, 10, 15, 15, 10, 5, 5, 10);
```

ProfileImages.

An entity containing the paths to all uploaded profile pictures.

1.

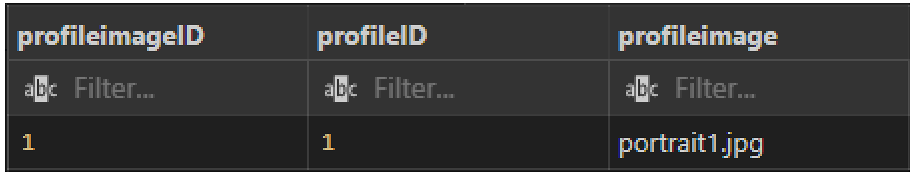
```
CREATE TABLE ProfileImages
(
  profileimageID INT        NOT NULL AUTO_INCREMENT,
  profileID      INT        NOT NULL,
  profileimage      VARCHAR(100)  NOT NULL,
  PRIMARY KEY (profileimageID)
);
```

2.

```
ALTER TABLE ProfileImages
  ADD CONSTRAINT FK_Profiles_TO_ProfileImages
    FOREIGN KEY (profileID)
    REFERENCES Profiles (profileID);
```

3.

```
INSERT INTO ProfileImages (profileID, profileimage)
VALUES(1, 'portrait1.jpg'),
```



profileimageID	profileID	profileimage
1	1	portrait1.jpg

Profiles.

An entity containing users' profiles along with a (short) bio.

1.

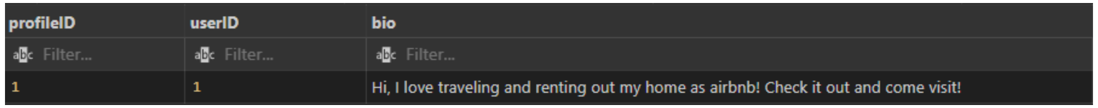
```
CREATE TABLE Profiles
(
  profileID INT  NOT NULL AUTO_INCREMENT,
  userID    INT  NOT NULL,
  bio       TEXT NULL     ,
  PRIMARY KEY (profileID)
);
```

2.

```
ALTER TABLE Profiles
  ADD CONSTRAINT FK_Users_TO_Profiles
    FOREIGN KEY (userID)
    REFERENCES Users (userID);
```

3.

```
INSERT INTO Profiles (userID, bio)
VALUES (1, 'Hi, I love traveling and renting out my home as
airbnb! Check it out and come visit!’);
```



profileID	userID	bio
Filter...	Filter...	Filter...
1	1	Hi, I love traveling and renting out my home as airbnb! Check it out and come visit!

ProfilesAndLanguages.

A relational entity that links the profile ID with the user's language capabilities.

1.

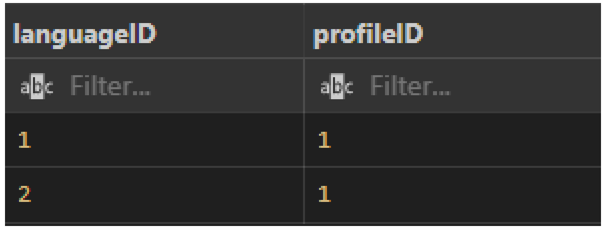
```
CREATE TABLE ProfilesAndLanguages
(
  languageID INT NOT NULL,
  profileID  INT NOT NULL
);
```

2.

```
ALTER TABLE ProfilesAndLanguages
  ADD CONSTRAINT FK_Languages_TO_ProfilesAndLanguages
    FOREIGN KEY (languageID)
    REFERENCES Languages (languageID);

ALTER TABLE ProfilesAndLanguages
  ADD CONSTRAINT FK_Profiles_TO_ProfilesAndLanguages
    FOREIGN KEY (profileID)
    REFERENCES Profiles (profileID);
```

3.

```
INSERT INTO ProfilesAndLanguages (languageID, profileID)
VALUES (1, 1), (2, 1);
```



languageID	profileID
1	1
2	1

Properties.

An entity containing links to all entities related to the property, as well as a listing name, description and size attribute.

1.

```
CREATE TABLE Properties
(
  propertyID      INT     NOT NULL AUTO_INCREMENT,
  addressID       INT     NOT NULL,
  roomsID         INT     NOT NULL,
  accessibilityID INT     NOT NULL,
  userID          INT     NOT NULL,
  priceID         INT     NOT NULL,
  propertyname    VARCHAR(100) NOT NULL,
  propertydescription     TEXT    NULL    ,
  size            INT   NULL    ,
  PRIMARY KEY (propertyID)
);
```

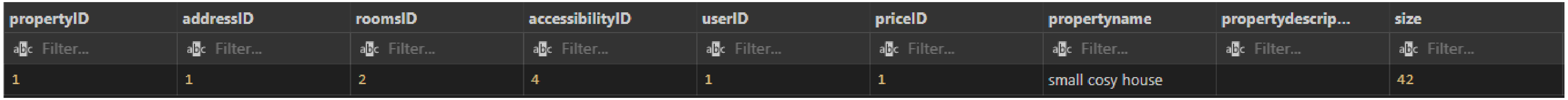
2.

```
ALTER TABLE Properties
  ADD CONSTRAINT FK_PropertyAddress_TO_Properties
    FOREIGN KEY (addressID)
    REFERENCES PropertyAddress (addressID);
ALTER TABLE Properties
  ADD CONSTRAINT FK_Rooms_TO_Properties
    FOREIGN KEY (roomsID)
    REFERENCES Rooms (roomsID);
ALTER TABLE Properties
  ADD CONSTRAINT FK_Accessibility_TO_Properties
    FOREIGN KEY (accessibilityID)
    REFERENCES Accessibility (accessibilityID);
ALTER TABLE Properties
  ADD CONSTRAINT FK_HostDetails_TO_Properties
    FOREIGN KEY (userID)
    REFERENCES HostDetails (userID);
ALTER TABLE Properties
  ADD CONSTRAINT FK_Price_TO_Properties
    FOREIGN KEY (priceID)
    REFERENCES Price (priceID);
```

Properties. P2.

3.

```
INSERT INTO Properties (addressID, roomsID,
accessibilityID, userID, priceID, propertyname,
propertydescription, size)

VALUES (1, 2, 4, 1, 1, 'small cosy house', '',
42),
```



propertyID	addressID	roomsID	accessibilityID	userID	priceID	propertyname	propertydescrip...	size
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
1	1	2	4	1	1	small cosy house		42

PropertyAddress.

An entity containing the address of a property.

1.

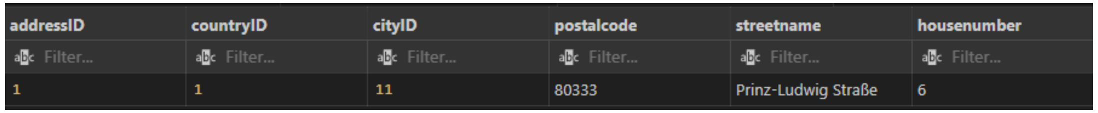
```
CREATE TABLE PropertyAddress
(
  addressID   INT     NOT NULL AUTO_INCREMENT,
  countryID   INT     NOT NULL,
  cityID      INT     NOT NULL,
  postalcode  VARCHAR(10) NOT NULL,
  streetname  VARCHAR(100) NOT NULL,
  housenumber VARCHAR(10) NOT NULL,
  PRIMARY KEY (addressID)
);
```

2.

```
ALTER TABLE PropertyAddress
  ADD CONSTRAINT FK_Countries_TO_PropertyAddress
    FOREIGN KEY (countryID)
    REFERENCES Countries (countryID);

ALTER TABLE PropertyAddress
  ADD CONSTRAINT FK_Cities_TO_PropertyAddress
    FOREIGN KEY (cityID)
    REFERENCES Cities (cityID);
```

3.

```
INSERT INTO PropertyAddress (countryID, cityID, postalcode, streetname, housenumber)
VALUES (1, 11, '80333', 'Prinz-Ludwig Straße', '6'),
```



addressID	countryID	cityID	postalcode	streetname	housenumber
1	1	11	80333	Prinz-Ludwig Straße	6

PropertyImages.

An entity containing the file path to the property image.

1.

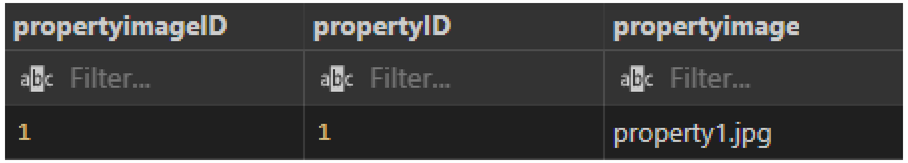
```
CREATE TABLE PropertyImages
(
  propertyimageID INT          NOT NULL AUTO_INCREMENT,
  propertyID      INT          NOT NULL,
  propertyimage   VARCHAR(100) NOT NULL,
  PRIMARY KEY (propertyimageID)
);
```

2.

```
ALTER TABLE PropertyImages
  ADD CONSTRAINT FK_Properties_TO_PropertyImages
    FOREIGN KEY (propertyID)
    REFERENCES Properties (propertyID);
```

3.

```
INSERT INTO PropertyImages (propertyID, propertyimage)
VALUES (1, 'property1.jpg'),
```



propertyimageID	propertyID	propertyimage
abc Filter...	abc Filter...	abc Filter...
1	1	property1.jpg

PropertyReviews.

An entity containing reviews of properties including rating and comments.

1.

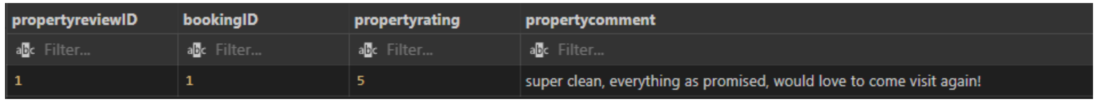
```
CREATE TABLE PropertyReviews

(

  propertyreviewID INT  NOT NULL AUTO_INCREMENT,

  bookingID        INT  NOT NULL,

  propertyrating   INT  NOT NULL,

  propertycomment  TEXT NULL    ,

  PRIMARY KEY (propertyreviewID)

);
```

2.

```
ALTER TABLE PropertyReviews

  ADD CONSTRAINT FK_Bookings_TO_PropertyReviews

    FOREIGN KEY (bookingID)

    REFERENCES Bookings (bookingID);
```

3.

```
INSERT INTO PropertyReviews (bookingID, propertyrating,
propertycomment)

VALUES (1, 5, 'super clean, everything as promised, would love to come
visit again!’);
```



propertyreviewID	bookingID	propertyrating	propertycomment
1	1	5	super clean, everything as promised, would love to come visit again!

Rooms.

An entity containing the number of rooms and beds of a property.

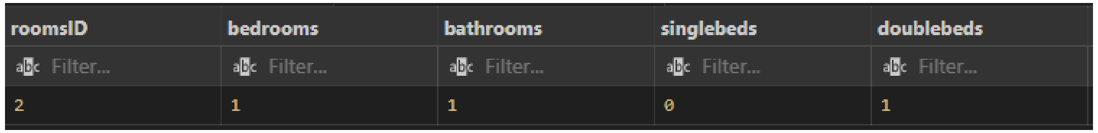
1.

```
CREATE TABLE Rooms
(
  roomsID     INT NOT NULL AUTO_INCREMENT,
  bedrooms    INT NOT NULL,
  bathrooms   INT NOT NULL,
  singlebeds INT NOT NULL,
  doublebeds INT NOT NULL,
  PRIMARY KEY (roomsID)
);
```

2. No alter statements

3.

```
INSERT INTO Rooms (bedrooms, bathrooms, 
singlebeds, doublebeds)
VALUES (1, 1, 1, 0);
```



roomsID	bedrooms	bathrooms	singlebeds	doublebeds
2	1	1	0	1

Users.

An entity containing the basic user details.

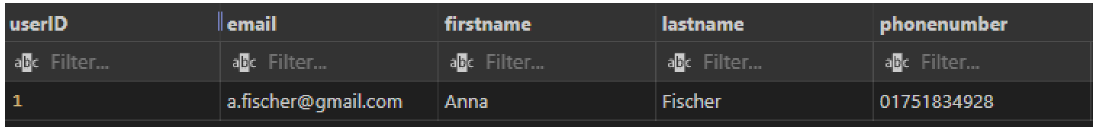
1.

```
CREATE TABLE Users
(
  userID      INT      NOT NULL AUTO_INCREMENT,
  email       VARCHAR(100) NOT NULL,
  firstname   VARCHAR(100) NOT NULL,
  lastname    VARCHAR(100) NOT NULL,
  phonenumber VARCHAR(100) NOT NULL,
  PRIMARY KEY (userID)
);
```

2. No alter statements

3.

```
INSERT INTO Users (email, firstname, lastname,
phonenumber)
VALUES ('a.fischer@gmail.com', 'Anna',
'Fischer', '01751834928'),
```



userID	email	firstname	lastname	phonenumber
1	a.fischer@gmail.com	Anna	Fischer	01751834928

VacationTypeAndProperties.

A relational entity linking a property and its vacationtype.

1.

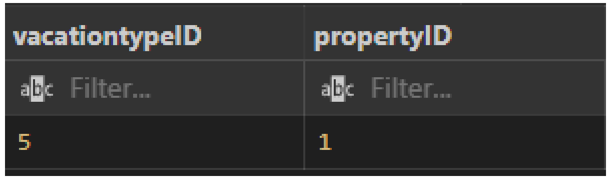
```
CREATE TABLE VacationTypeAndProperties
(
  vacationtypeID INT NOT NULL,
  propertyID     INT NOT NULL
);
```

2.

```
ALTER TABLE VacationTypeAndProperties
  ADD CONSTRAINT FK_VacationType_TO_VacationTypeAndProperties
    FOREIGN KEY (vacationtypeID)
    REFERENCES VacationType (vacationtypeID);

ALTER TABLE VacationTypeAndProperties
  ADD CONSTRAINT FK_Properties_TO_VacationTypeAndProperties
    FOREIGN KEY (propertyID)
    REFERENCES Properties (propertyID);
```

3.

```
INSERT INTO VacationTypeAndProperties (vacationtypeID, propertyID)
VALUES (5, 1);
```



vacationtypeID	propertyID
5	1

VacationType.

An entity containing all offered vacationtypes.

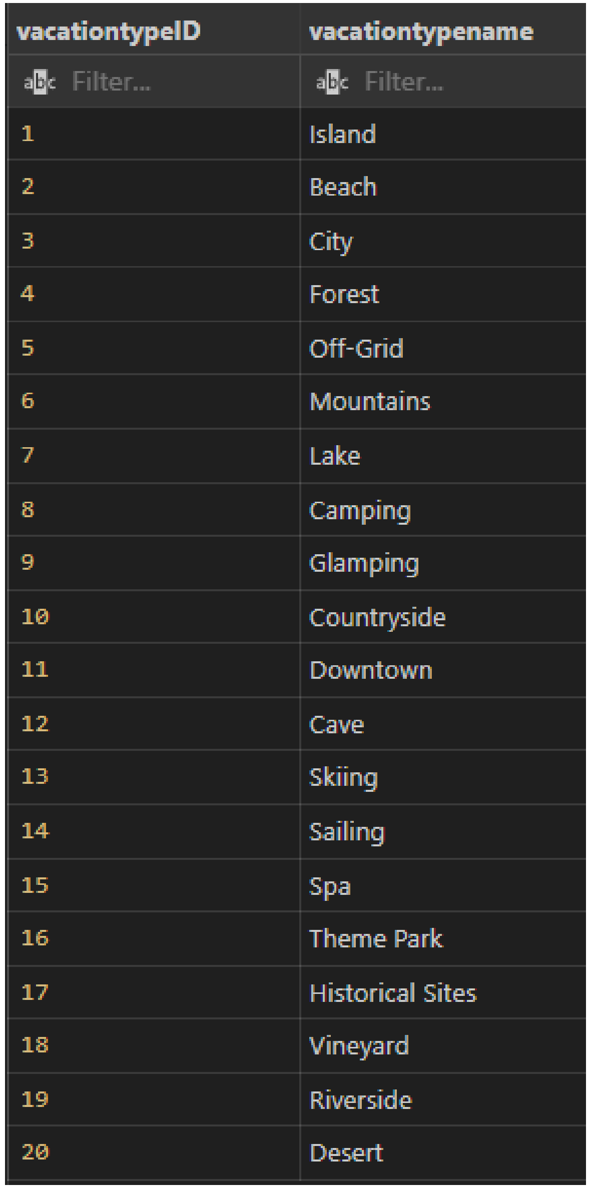
1.

```
CREATE TABLE VacationType
(
  vacationtypeID   INT NOT NULL AUTO_INCREMENT,
  vacationtypename VARCHAR(100) NOT NULL,
  PRIMARY KEY (vacationtypeID)
);
```

2. No alter statements

3.

```
INSERT INTO VacationType (vacationtypename)
VALUES ('Island'), ('Beach'), ('City'),
('Forest'), ('Off-Grid'), ('Mountains'), ('Lake'),
('Camping'), ('Glamping'), ('Countryside'),
('Downtown'), ('Cave'), ('Skiing'), ('Sailing'),
('Spa'), ('Theme Park'), ('Historical Sites'),
('Vineyard'), ('Riverside'), ('Desert');
```



vacationtypeID	vacationtypename
Filter...	Filter...
1	Island
2	Beach
3	City
4	Forest
5	Off-Grid
6	Mountains
7	Lake
8	Camping
9	Glamping
10	Countryside
11	Downtown
12	Cave
13	Skiing
14	Sailing
15	Spa
16	Theme Park
17	Historical Sites
18	Vineyard
19	Riverside
20	Desert