

# Project Data Mart

## Phase 2

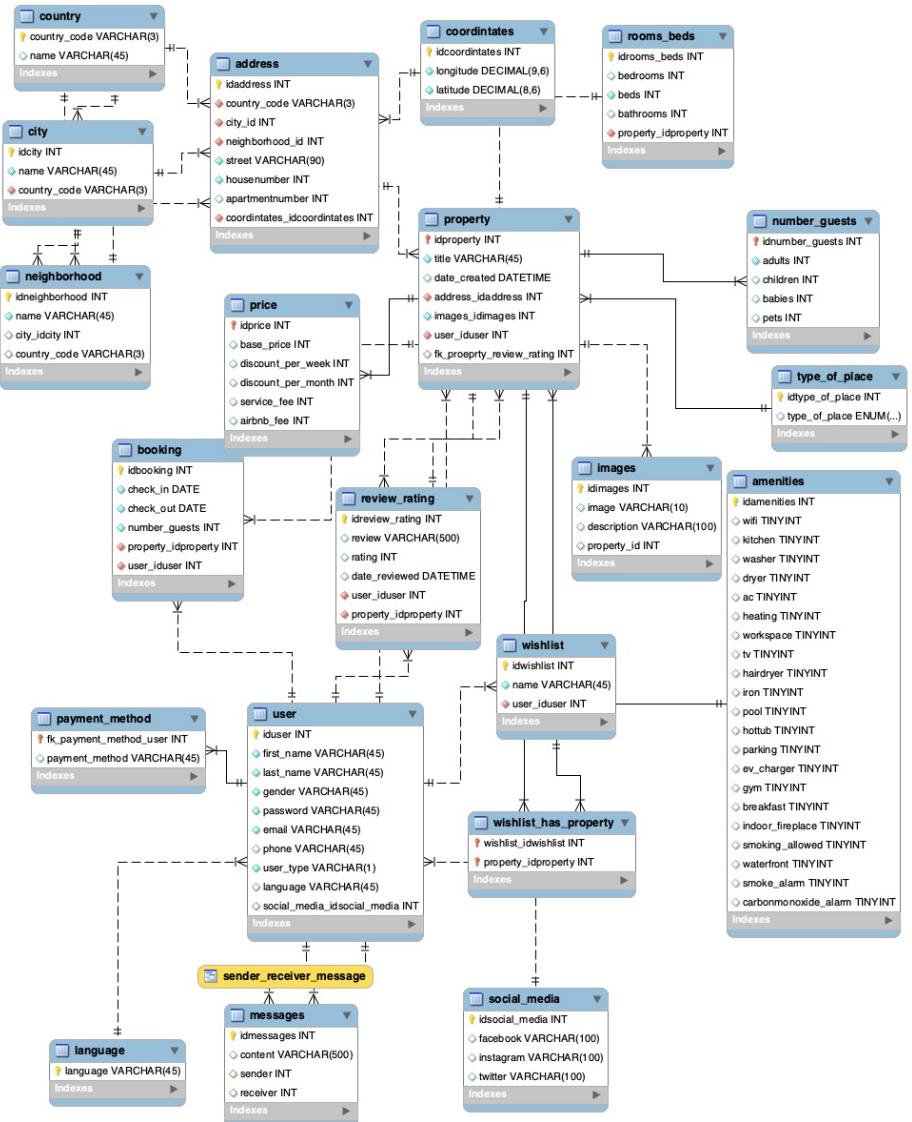
BY EMRE ERTÜRK

4TH OF MAY 2022

# Agenda

- ❑ Creating ERD
- ❑ Code for creating schema, tables, and relationships
  - ❑ Schema
  - ❑ Tables
  - ❑ Relationships (Primary Keys and Foreign Keys)
- ❑ Entry of dummy variables

# Create ERD



4

# Create Schema

```
CREATE SCHEMA IF NOT EXISTS `Airbnb` DEFAULT CHARACTER SET utf8 ;  
USE `Airbnb` ;
```

## Create Schema/Database `Airbnb`

```
1 CREATE SCHEMA IF NOT EXISTS `Airbnb` DEFAULT CHARACTER SET utf8 ;  
2 USE `Airbnb` ;
```

# Create Table

THE SQL QUERIES INCLUDE:

- CREATE TABLE
- ATTRIBUTE TYPE
- PRIMARY KEYS
- FOREIGN KEYS

THE SQL QUERIES HAVE BEEN CREATED USING  
MYSQL WORKBENCH FORWARD  
ENGINEERING

## CREATE TABLE `country`

```

20  CREATE TABLE IF NOT EXISTS `Airbnb`.`country` (
21    `country_code` VARCHAR(3) NOT NULL,
22    `name` VARCHAR(45) NULL DEFAULT NULL,
23    PRIMARY KEY (`countrycode`)
24  ENGINE = InnoDB
25  DEFAULT CHARACTER SET = utf8mb3;

```

Two attributes whereas countrycode is the international 3-letter countrycode as primary key (pk) and country name is the actual name of the country

1 • SELECT \* FROM Country

countrycode	name
AFG	Afghanistan
AUS	Australia
BB	Barbados
CA	Canada
CH	Switzerland
CHN	China
ESP	Spain
ETH	Ethiopia
FR	France
GER	Germany
GHA	Ghana
JPN	Japan
KOR	Korea
MEX	Mexico
NIG	Nigeria
NZ	New Zealand
PHL	Phillipines
RSA	South Africa
TUR	Turkey
UK	UK
USA	USA
NULL	NULL

## CREATE TABLE `city`

```

1 ▼ CREATE TABLE IF NOT EXISTS `Airbnb`.`city` (
2   `idcity` INT NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(45) NOT NULL,
4   `country_code` VARCHAR(3) NOT NULL,
5   PRIMARY KEY (`idcity`),
6   INDEX `fk_city_country_idx` (`country_code` ASC) VISIBLE,
7   CONSTRAINT `fk_city_country`
8     FOREIGN KEY (`country_code`)
9       REFERENCES `Airbnb`.`country` (`countrycode`)
10      ON DELETE CASCADE
11      ON UPDATE CASCADE)
12 ENGINE = InnoDB
13 AUTO_INCREMENT = 23
14 DEFAULT CHARACTER SET = utf8mb3;

```

3 attributes from which two identify the city by city\_id and city name. The 3<sup>rd</sup> attribute is a foreign key from the country table which gets updated on CASCADE.

1 • SELECT \* FROM City

	idcity	name	country_code
▶	1	Istanbul	TUR
	2	Berlin	GER
	3	Munich	GER
	4	Frankfurt	GER
	5	Miami	USA
	6	New York	USA
	7	Izmir	TUR
	8	London	UK
	9	Liverpool	UK
	10	Manila	PHL
	11	Johannesburg	RSA
	12	Cancun	MEX
	13	New Mexico	MEX
	14	Manchester	UK
	15	Paris	FR
	16	Marseille	FR
	17	Toronto	CA
	18	Vancouver	CA
	19	Ottawa	CA
	20	Kiyoto	JPN
	21	Tokyo	JPN
	22	Seoul	KOR

## CREATE TABLE `neighborhood`

```

1  CREATE TABLE IF NOT EXISTS `Airbnb`.`neighborhood` (
2    `idneighborhood` INT NOT NULL AUTO_INCREMENT,
3    `name` VARCHAR(45) NOT NULL,
4    `city_idcity` INT NULL DEFAULT NULL,
5    `country_code` VARCHAR(3) NULL DEFAULT NULL,
6    PRIMARY KEY (`idneighborhood`),
7    INDEX `fk_neighborhood_city1_idx` (`city_idcity` ASC) VISIBLE,
8    INDEX `fk_country_idx` (`country_code` ASC) VISIBLE,
9    CONSTRAINT `fk_country`
10      FOREIGN KEY (`country_code`)
11        REFERENCES `Airbnb`.`country` (`countrycode`)
12        ON DELETE CASCADE
13        ON UPDATE CASCADE,
14    CONSTRAINT `fk_neighborhood_city1`
15      FOREIGN KEY (`city_idcity`)
16        REFERENCES `Airbnb`.`city` (`idcity`)
17    ENGINE = InnoDB
18    AUTO_INCREMENT = 21
19    DEFAULT CHARACTER SET = utf8mb3;

```

This table follows the structure of the city table and is basically a second drill-down for location. It includes the neighborhood id (pk), name, city\_id, and country\_code. The latter two are fk from city and country table respectively.

```

1 •   SELECT neighborhood.name AS neighborhood, city.name AS city, country.name AS country
2   FROM neighborhood
3   JOIN city
4   ON neighborhood.city_idcity = city.idcity
5   JOIN country
6   ON neighborhood.country_code = country.country_code

```

	idcity	name	country_code
▶	1	Istanbul	TUR
	2	Berlin	GER
	3	Munich	GER
	4	Frankfurt	GER
	5	Miami	USA
	6	New York	USA
	7	Izmir	TUR
	8	London	UK
	9	Liverpool	UK
	10	Manila	PHL
	11	Johannesburg	RSA
	12	Cancun	MEX
	13	New Mexico	MEX
	14	Manchester	UK
	15	Paris	FR
	16	Marseille	FR
	17	Toronto	CA
	18	Vancouver	CA
	19	Ottawa	CA
	20	Kiyoto	JPN
	21	Tokyo	JPN
	22	Seoul	KOR

## CREATE TABLE `coordinates`

```

74  CREATE TABLE IF NOT EXISTS `Airbnb`.`coordinates` (
75    `idcoordinates` INT NOT NULL AUTO_INCREMENT,
76    `longitude` DECIMAL(9,6) NOT NULL,
77    `latitude` DECIMAL(8,6) NOT NULL,
78    PRIMARY KEY (`idcoordinates`)
79  ENGINE = InnoDB
80  AUTO_INCREMENT = 21
81  DEFAULT CHARACTER SET = utf8mb3;

```

This table shows the longitude and latitudes of an address. According to airbnb, the exact location is never shown but a radius is given. That's why I have impleted the DECIMAL(9,6) and DECIMAL(8,6) respectively to illustrate the shown accuracy.

```
1 •  SELECT * FROM Airbnb.coordinates;
```

idcoordinates	longitude	latitude
1	-79.198108	-43.651721
2	92.612874	82.597578
3	-159.299890	54.616588
4	-44.981903	-8.763479
5	47.669755	-39.535590
6	-8.578690	89.963046
7	135.387757	-87.643430
8	-51.295307	-53.453520
9	-12.557926	-30.015891
10	-168.205483	31.337556
11	-13.052881	-16.574861
12	-68.098655	11.580464
13	-29.817208	-24.338222
14	-115.415106	58.049883
15	140.167184	16.556647
16	-3.552596	51.423077
17	0.442208	-29.883797
18	-4.330988	0.929560
19	161.837542	-67.031980
20	151.044340	49.237891
NULL	NULL	NULL

## CREATE TABLE `address`

```

87 CREATE TABLE IF NOT EXISTS `Airbnb`.`address` (
88     `idaddress` INT NOT NULL AUTO_INCREMENT,
89     `country_code` VARCHAR(3) NOT NULL,
90     `city_id` INT NOT NULL,
91     `neighborhood_id` INT NOT NULL,
92     `street` VARCHAR(90) NOT NULL,
93     `housenumber` INT NOT NULL,
94     `apartmentnumber` INT NULL DEFAULT NULL,
95     `coordintates_idcoordintates` INT NOT NULL,
96     PRIMARY KEY (`idaddress`),
97     INDEX `fk_idx` (`country_code` ASC) VISIBLE,
98     INDEX `fk3_idx` (`neighborhood_id` ASC) VISIBLE,
99     INDEX `fk4_idx` (`coordintates_idcoordintates` ASC) VISIBLE,
100    INDEX `fk2_idx` (`city_id` ASC) VISIBLE,
101    CONSTRAINT `fk`
102        FOREIGN KEY (`country_code`)
103        REFERENCES `Airbnb`.`country` (`countrycode`)
104        ON DELETE CASCADE
105        ON UPDATE CASCADE,
106    CONSTRAINT `fk2`
107        FOREIGN KEY (`city_id`)
108        REFERENCES `Airbnb`.`city` (`idcity`)
109        ON DELETE CASCADE
110        ON UPDATE CASCADE,
111    CONSTRAINT `fk3`
112        FOREIGN KEY (`neighborhood_id`)
113        REFERENCES `Airbnb`.`neighborhood` (`idneighborhood`)
114        ON DELETE CASCADE
115        ON UPDATE CASCADE,
116    CONSTRAINT `fk4`
117        FOREIGN KEY (`coordintates_idcoordintates`)
118        REFERENCES `Airbnb`.`coordintates` (`idcoordintates`)
119        ON DELETE CASCADE
120        ON UPDATE CASCADE)
121 ENGINE = InnoDB
122 AUTO_INCREMENT = 21
123 DEFAULT CHARACTER SET = utf8mb3;

```

The address table brings everything from the last 4 slides together and includes country, city, neighborhood, and coordinates next to street names, apartment number, and house number.

The PK is the surrogate `idaddress`. The Foreign keys have all the cascade set so that updates are processed efficiently within the database and the data integrity can be upheld. The indexes help to retrieve the data faster within the tables when a query attempts to search an entry.

```

1 •   SELECT
2      country.name AS country,
3      city.name AS city,
4      neighborhood.name AS neighborhood,
5      street,
6      housenumber,
7      apartmentnumber,
8      coordintates.longitude,
9      coordintates.latitude
10     FROM Airbnb.address
11     JOIN country
12     ON address.country_code = country.country_code
13     JOIN city
14     ON address.city_id = city.idcity
15     JOIN neighborhood
16     ON address.neighborhood_id = neighborhood.idneighborhood
17     JOIN coordintates
18     ON address.coordintates_idcoordintates = coordintates.idcoordintates
19     WHERE city.name="Istanbul";

```

country	city	neighborhood	street	housenumber	apartmentnumber	longitude	latitude
Turkey	Istanbul	Beyazit	abc	13	5	-79.198108	-43.651721
Turkey	Istanbul	Fatih	Haggia Sophia Sokak	23	5	92.612874	82.597578
Turkey	Istanbul	Galata	fener sokak	433	22	-159.299890	54.616588
Turkey	Istanbul	Galata	Galata sokak	45	13	-44.981903	-8.763479

```
CREATE TABLE `amenities`
```

```
CREATE TABLE IF NOT EXISTS `Airbnb`.`amenities` (
  `idamenities` INT NOT NULL AUTO_INCREMENT,
  `wifi` TINYINT NULL DEFAULT NULL,
  `kitchen` TINYINT NULL DEFAULT NULL,
  `washer` TINYINT NULL DEFAULT NULL,
  `dryer` TINYINT NULL DEFAULT NULL,
  `ac` TINYINT NULL DEFAULT NULL,
  `heating` TINYINT NULL DEFAULT NULL,
  `workspace` TINYINT NULL DEFAULT NULL,
  `tv` TINYINT NULL DEFAULT NULL,
  `hairdryer` TINYINT NULL DEFAULT NULL,
  `iron` TINYINT NULL DEFAULT NULL,
  `pool` TINYINT NULL DEFAULT NULL,
  `hottub` TINYINT NULL DEFAULT NULL,
  `parking` TINYINT NULL DEFAULT NULL,
  `ev_charger` TINYINT NULL DEFAULT NULL,
  `gym` TINYINT NULL DEFAULT NULL,
  `breakfast` TINYINT NULL DEFAULT NULL,
  `indoor_fireplace` TINYINT NULL DEFAULT NULL,
  `smoking_allowed` TINYINT NULL DEFAULT NULL,
  `waterfront` TINYINT NULL DEFAULT NULL,
  `smoke_alarm` TINYINT NULL DEFAULT NULL,
  `carbonmonoxide_alarm` TINYINT NULL DEFAULT NULL,
  PRIMARY KEY (`idamenities`),
  ENGINE = InnoDB
) AUTO_INCREMENT = 21
DEFAULT CHARACTER SET = utf8mb3;
```

The amenities include all extras of a property. It has a surrogate PK. The type of entry is a BOOLEAN for all entries except the PK. The Default value is set to NULL bc only "1" implies that an amenity has that amenity.

```
1 •     SELECT * FROM amenities
2      LIMIT 3
```

## CREATE TABLE `number\_guest`

```

248 CREATE TABLE IF NOT EXISTS `Airbnb`.`number_guests` (
249   `idnumber_guests` INT NOT NULL AUTO_INCREMENT,
250   `adults` INT NOT NULL,
251   `children` INT NULL DEFAULT NULL,
252   `babies` INT NULL DEFAULT NULL,
253   `pets` INT NULL DEFAULT NULL,
254   PRIMARY KEY (`idnumber_guests`),
255   CONSTRAINT `fk_number_of_guests_property`
256   FOREIGN KEY (`idnumber_guests`)
257   REFERENCES `Airbnb`.`property` (`idproperty`)
258   ON DELETE CASCADE
259   ON UPDATE CASCADE;
260
261 ENGINE = InnoDB
262 AUTO_INCREMENT = 21
263 DEFAULT CHARACTER SET = utf8mb3;

```

This table sums up the hosts settings to define the maximum number of guests and the properties allowed guest types. If pets aren't allowed than the pet column will show NULL. The PK is a surrogate key. Due to its 1:1 relationship to `property` the surrogate PK in property is the FK in `number\_guest`.

```

1 *  SELECT number_guests.adults AS "number of adults", property.title
2   FROM number_guests
3   JOIN property
4   ON number_guests.idnumber_guests = property.idproperty
5   WHERE number_guests.adults > 4;

```

number of adults	title
5	lorem. Donec
5	quam vel
5	lorem, eget
5	eleifend non,
5	Mauris nulla.
6	risus varius

## CREATE TABLE `rooms\_beds`

```

290 CREATE TABLE IF NOT EXISTS `Airbnb`.`rooms_beds` (
291     `idrooms_beds` INT NOT NULL AUTO_INCREMENT,
292     `bedrooms` INT NULL DEFAULT NULL,
293     `beds` INT NOT NULL,
294     `bathrooms` INT NULL DEFAULT NULL,
295     PRIMARY KEY (`idrooms_beds`),
296     CONSTRAINT `fk_rooms_beds_property`
297         FOREIGN KEY (`idrooms_beds`)
298             REFERENCES `Airbnb`.`property` (`idproperty`)
299             ON DELETE CASCADE
300             ON UPDATE CASCADE)
301 ENGINE = InnoDB
302 AUTO_INCREMENT = 21
303 DEFAULT CHARACTER SET = utf8mb3;

```

This table sums up the hosts settings to define the maximum number of guests and the properties allowed guest types. If pets aren't allowed than the pet column will show NULL. The PK is a surrogate key. Due to its 1:1 relationship to `property` the surrogate PK in `property` is the FK in `number\_guest`.

---

```

1 • SELECT
2     ROUND(AVG(rooms_beds.bedrooms),1) AS "Avergage bedrooms",
3     ROUND(AVG(rooms_beds.beds),1) AS "Avergage beds",
4     ROUND(AVG(rooms_beds.bathrooms),1) AS "Avergage bathrooms"
5 FROM rooms_beds

```

---

	Average bedroo...	Average beds	Average bathro...
▶	1.9	4.3	4.8

## CREATE TABLE `type\_of\_place`

```

309 CREATE TABLE IF NOT EXISTS `Airbnb`.`type_of_place` (
310     `idtype_of_place` INT NOT NULL AUTO_INCREMENT,
311     `type_of_place` ENUM('apartment', 'room', 'bed', 'hotel') NULL DEFAULT NULL,
312     PRIMARY KEY (`idtype_of_place`)
313     ENGINE = InnoDB
314     AUTO_INCREMENT = 21
315     DEFAULT CHARACTER SET = utf8mb3;

```

This table shows the type of place that the host set for the property. PK is a surrogate key that is simultaneously the PK of the property table as a property they share a 1:1 relationship.

---

```

1 •   SELECT b.idproperty, b.title, a.type_of_place
2      FROM type_of_place a
3      JOIN property b
4      ON a.idtype_of_place = b.idproperty
5      WHERE type_of_place = "apartment" OR type_of_place = "hotel"
6      ORDER BY idproperty

```

	idproperty	title	type_of_place
▶	2	quam. Pellentesque	apartment
	9	Phasellus at	hotel
	10	nibh. Donec	apartment
	14	eleifend non,	apartment
	18	Phasellus dolor	hotel

## CREATE TABLE `property`

```

321 CREATE TABLE IF NOT EXISTS `Airbnb`.`property` (
322     `idproperty` INT NOT NULL AUTO_INCREMENT,
323     `title` VARCHAR(45) NOT NULL,
324     `date_created` DATETIME NULL DEFAULT NULL,
325     `address_idaddress` INT NOT NULL,
326     `price_idprice` INT NOT NULL,
327     `images_idimages` INT NOT NULL,
328     `user_iduser` INT NOT NULL,
329     `fk_proeprty_review_rating` INT NULL DEFAULT NULL,
330     PRIMARY KEY (`idproperty`),
331     INDEX `fk_property_address1_idx`(`address_idaddress` ASC) VISIBLE,
332     INDEX `fk_property_price1_idx`(`price_idprice` ASC) VISIBLE,
333     INDEX `fk_property_user1_idx`(`user_iduser` ASC) VISIBLE,
334     INDEX `fk_property_review_rating_1_idx`(`fk_proeprty_review_rating` ASC) VISIBLE,
335     CONSTRAINT `fk_property_address1`
336         FOREIGN KEY(`address_idaddress`)
337             REFERENCES `Airbnb`.`address`(`idaddress`)
338             ON DELETE CASCADE
339             ON UPDATE CASCADE,
340     CONSTRAINT `fk_property_price1`
341         FOREIGN KEY(`price_idprice`)
342             REFERENCES `Airbnb`.`price`(`idprice`)
343             ON DELETE CASCADE
344             ON UPDATE CASCADE,
345     CONSTRAINT `fk_property_review_rating_1`
346         FOREIGN KEY(`fk_proeprty_review_rating`)
347             REFERENCES `Airbnb`.`review_rating`(`property_idproperty`)
348             ON DELETE CASCADE
349             ON UPDATE CASCADE,
350     CONSTRAINT `fk_property_rooms_beds1`
351         FOREIGN KEY(`idproperty`)
352             REFERENCES `Airbnb`.`rooms_beds`(`idrooms_beds`)
353             ON DELETE CASCADE
354             ON UPDATE CASCADE,
355     CONSTRAINT `fk_property_type_of_place1`
356         FOREIGN KEY(`idproperty`)
357             REFERENCES `Airbnb`.`type_of_place`(`idtype_of_place`)
358             ON DELETE CASCADE
359             ON UPDATE CASCADE,
360     CONSTRAINT `fk_property_user1`
361         FOREIGN KEY(`user_iduser`)
362             REFERENCES `Airbnb`.`user`(`iduser`)
363             ON DELETE CASCADE
364             ON UPDATE CASCADE,
365     CONSTRAINT `fk_property_amenities`
366         FOREIGN KEY(`idproperty`)
367             REFERENCES `Airbnb`.`amenities`(`idamenities`)
368             ON DELETE CASCADE
369             ON UPDATE CASCADE

```

One of the main tables storing the property information bundled. There is a surrogate PK and all FK are updated/deleted on CASCADE to assure data integrity.

### 1 • DESCRIBE property;

Field	Type	Null	Key	Default	Extra
▶ idproperty	int	NO	PRI	NULL	auto_increment
title	varchar(45)	NO		NULL	
date_created	datetime	YES		NULL	
address_idaddress	int	NO	MUL	NULL	
price_idprice	int	NO	MUL	NULL	
images_idimages	int	NO		NULL	
user_iduser	int	NO	MUL	NULL	
fk_proeprty_review_rating	int	YES	MUL	NULL	

## CREATE TABLE `images`

```

420  CREATE TABLE IF NOT EXISTS `Airbnb`.`images` (
421      `idimages` INT NOT NULL AUTO_INCREMENT,
422      `image` BLOB NOT NULL,
423      `description` VARCHAR(100) NOT NULL DEFAULT NULL,
424      `property_id` INT NOT NULL DEFAULT NULL,
425      PRIMARY KEY (`idimages`),
426      INDEX `fk_images_property_1_idx`(`property_id` ASC) VISIBLE,
427      CONSTRAINT `fk_images_property_1`
428          FOREIGN KEY (`property_id`)
429          REFERENCES `Airbnb`.`property` (`idproperty`)
430          ON DELETE CASCADE
431          ON UPDATE CASCADE)
432  ENGINE = InnoDB
433  AUTO_INCREMENT = 51
434  DEFAULT CHARACTER SET = utf8mb3;

```

All images uploaded are uploaded to this table. The PK is a surrogate key and FK is defined and connected to properties. There is a 1:N relationship to the property table. One property can have multiple images, but one image can only belong to one property. A BLOB type makes it possible to store an image in the database.

### 1 • SELECT

```

2      images.property_id,
3      COUNT(images.image) AS "Number of pictures"
4  FROM images
5  WHERE images.property_id IN (
6      SELECT property.idproperty
7      FROM property
8      JOIN rooms_beds
9      ON property.idproperty = rooms_beds.idrooms_beds
10     WHERE rooms_beds.beds > 3)
11    GROUP BY images.property_id

```

property_id	Number of pictu...
3	6
6	4
7	2
9	1
11	2
12	1
13	3
15	1
16	5
17	1

## CREATE TABLE `review\_rating`

```

269 CREATE TABLE IF NOT EXISTS `Airbnb`.`review_rating` (
270     `idreview_rating` INT NOT NULL AUTO_INCREMENT,
271     `review` VARCHAR(500) NULL DEFAULT NULL,
272     `rating` INT NULL DEFAULT NULL,
273     `date_reviewed` DATETIME NULL DEFAULT NULL,
274     `user_iduser` INT NOT NULL,
275     `property_idproperty` INT NOT NULL,
276     PRIMARY KEY (`idreview_rating`),
277     INDEX `fk_review_rating_user1_idx`(`user_iduser` ASC) VISIBLE,
278     INDEX `fk_review_rating_property1_idx`(`property_idproperty` ASC) VISIBLE,
279     CONSTRAINT `fk_review_rating_user1`
280         FOREIGN KEY (`user_iduser`)
281             REFERENCES `Airbnb`.`user`(`iduser`)
282             ON DELETE CASCADE
283             ON UPDATE CASCADE,
284     CONSTRAINT `fk_review_rating_property1`
285         FOREIGN KEY (`property_idproperty`)
286             REFERENCES `Airbnb`.`property`(`idproperty`)
287             ON DELETE CASCADE
288             ON UPDATE CASCADE;
289 ENGINE = InnoDB
290 AUTO_INCREMENT = 51
291 DEFAULT CHARACTER SET = utf8mb3;

```

This table shows all reviews and rating of a user for a property. It has a surrogate PK and two FK connecting the relationship to user and property tables.

### 1 • SELECT

```

2     b.first_name,
3     b.last_name,
4     ROUND(AVG(a.rating),2) AS "Average_rating"
5     FROM Airbnb.review_rating a
6     JOIN user b ON a.user_iduser = b.iduser
7     GROUP BY a.user_iduser
8     ORDER BY AVG(a.rating) DESC
9     LIMIT 10;

```

	first_name	last_name	Average_rating
▶	Ferris	Sloan	5.00
◀	Dahlia	Tyson	5.00
◀	Vance	Rivas	4.50
◀	Wayne	Cleveland	4.00
◀	Igor	Alvarado	4.00
◀	Dexter	Long	4.00
◀	Destiny	Allen	3.67
◀	Fay	Graham	3.50
◀	Daria	Vang	3.50
◀	Buffy	Leon	3.40

## CREATE TABLE `social\_media`

```

160  CREATE TABLE IF NOT EXISTS `Airbnb`.`social_media` (
161    `idsocial_media` INT NOT NULL AUTO_INCREMENT,
162    `facebook` VARCHAR(100) NULL DEFAULT NULL,
163    `instagram` VARCHAR(100) NULL DEFAULT NULL,
164    `twitter` VARCHAR(100) NULL DEFAULT NULL,
165    PRIMARY KEY (`idsocial_media`)
166  ENGINE = InnoDB
167  AUTO_INCREMENT = 21
168  DEFAULT CHARACTER SET = utf8mb3;

```

The table sums up all social media profiles of a user.

---

1 • **SELECT \* FROM Airbnb.social\_media;**

idsocial_media	facebook	instagram	twitter
1	ee	ee	ee
2	ww	ww	ww
3	aa	aa	aa
4	bb	bb	bb
5	cc	cc	cc
6	dd	dd	dd
7	ff	ff	ff
8	gg	gg	gg
9	hh	hh	hh
10	ii	ii	ii
11	jj	jj	jj
12	kk	kk	kk

## CREATE TABLE `language`

```
174  CREATE TABLE IF NOT EXISTS `Airbnb`.`language` (
175    `language` VARCHAR(45) NOT NULL,
176    PRIMARY KEY (`language`))
177  ENGINE = InnoDB
178  DEFAULT CHARACTER SET = utf8mb3;
```

This table shows all spoken languages in the database. There is no surrogate need necessary as the languages itself are unique already.

```
1 •   SELECT * FROM Airbnb.language;
```

language
english
french
german
japanese
korean
spanish
turkish
twi
urdu
xhosa
NULL

## CREATE TABLE `messages`

```

440 CREATE TABLE IF NOT EXISTS `Airbnb`.`messages` (
441     `idmessages` INT NOT NULL AUTO_INCREMENT,
442     `content` VARCHAR(500) NULL DEFAULT NULL,
443     `sender` INT NULL DEFAULT NULL,
444     `receiver` INT NULL DEFAULT NULL,
445     PRIMARY KEY (`idmessages`),
446     INDEX `sender_idx` (`sender` ASC) VISIBLE,
447     INDEX `receiver_idx` (`receiver` ASC) VISIBLE,
448     CONSTRAINT `receiver`
449         FOREIGN KEY (`receiver`)
450             REFERENCES `Airbnb`.`user` (`iduser`)
451             ON DELETE CASCADE
452             ON UPDATE CASCADE,
453     CONSTRAINT `sender`
454         FOREIGN KEY (`sender`)
455             REFERENCES `Airbnb`.`user` (`iduser`)
456             ON DELETE CASCADE
457             ON UPDATE CASCADE)
458 ENGINE = InnoDB
459 AUTO_INCREMENT = 26
460 DEFAULT CHARACTER SET = utf8mb3;

```

This table shows the messages exchanged between users. It is a surrogate PK and two FK directed to the USER table. It is a recursive relationship with itself as both FKs `sender` and `receiver` use the `iduser` as FK but have different meaning.

```

1 • CREATE VIEW `sender_receiver_message` AS
2     SELECT
3         CONCAT(b.first_name, " ", b.last_name) AS "sender_name",
4         CONCAT(c.first_name, " ", c.last_name) AS "receiver_name",
5         a.content
6     FROM messages a
7     JOIN user b ON a.sender = b.iduser
8     JOIN user c ON a.receiver = c.iduser

```

sender_name	receiver_name	content
Pascale English	Leonard Benjamin	euismod ac, fermentum vel, mauris. Integer se...
Kermit Mcintosh	Abra Duke	placerat, augue.
Libby Lambert	Carissa Todd	Fusce aliquet magna a
Abra Duke	Brenden Mullen	Quisque tincidunt pede ac urna.
Rina Emerson	Aristotle Roach	amet ultricies sem magna nec quam. Curabitur...
Fay Graham	Libby Lambert	tincidunt, nunc ac mattis ornare, lectus ante dict...
Daria Vang	Whoopi Madden	In scelerisque scelerisque dui. Suspendisse
Aristotle Roach	Clare McCarthy	quis diam. Pellentesque habitant morbi tristique...
Daria Vang	Ferris Sloan	commodo hendrerit. Donec porttitor tellus non...
Abra Duke	Dahlia Tyson	faucibus id, libero. Donec consecutetur mauris i...
Daria Vang	Libby Lambert	id, mollis nec,
Whoopi Madden	Dexter Long	vestibulum. Mauris magna. Duis dignissim temp...
Destiny Allen	Ferris Sloan	ornare egestas ligula. Nullam feugiat placerat v...

```

1 • SELECT receiver_name, COUNT(*) AS "count"
2     FROM Airbnb.sender_receiver_message
3     GROUP BY receiver_name
4     ORDER BY "count" DESC
5     LIMIT 3;

```

receiver_name	count	▼
Brenden Mullen	3	
Leonard Benjamin	2	
Carissa Todd	2	

## CREATE TABLE `wishlist`

```

466 CREATE TABLE IF NOT EXISTS `Airbnb`.`wishlist` (
467   `idwishlist` INT NOT NULL AUTO_INCREMENT,
468   `name` VARCHAR(45) NOT NULL,
469   `user_iduser` INT NOT NULL,
470   PRIMARY KEY (`idwishlist`),
471   INDEX `fk_wishlist_user1_idx` (`user_iduser` ASC) VISIBLE,
472   CONSTRAINT `fk_wishlist_user1`
473     FOREIGN KEY (`user_iduser`)
474       REFERENCES `Airbnb`.`user` (`iduser`)
475       ON DELETE CASCADE
476       ON UPDATE CASCADE)
477 ENGINE = InnoDB
478 AUTO_INCREMENT = 21
479 DEFAULT CHARACTER SET = utf8mb3;

```

A wishlist is a bunch of properties saved by the guests. In this table you see a surrogate PK and a FK connecting the `user` table to wishlists.

```

1 •  SELECT
2      a.name AS "wishlist_name",
3      CONCAT(b.first_name, " ", b.last_name) AS "user_name"
4  FROM Airbnb.wishlist a
5  JOIN user b
6  ON a.user_iduser = b.iduser
7  LIMIT 10;

```

wishlist_name	user_name
eleifend. Cras sed	Dexter Long
accumsan neque et	Brenden Mullen
bibendum fermentum metus.	Anthony Kent
ipsum nunc id	Kermit Mcintosh
semper cursus. Integer	Abra Duke
luctus felis purus	Igor Alvarado
Duis gravida. Praesent	Fay Graham
egestas. Sed pharetra,	Aristotle Roach
porttitor scelerisque neque.	Rina Emerson
egestas a, dui.	Carissa Todd

## CREATE TABLE `wishlist\_has\_property`

```

485 CREATE TABLE IF NOT EXISTS `Airbnb`.`wishlist_has_property` (
486     `wishlist_idwishlist` INT NOT NULL,
487     `property_idproperty` INT NOT NULL,
488     PRIMARY KEY (`wishlist_idwishlist`, `property_idproperty`),
489     INDEX `fk_wishlist_has_property_property1_idx`(`property_idproperty` ASC) VISIBLE,
490     INDEX `fk_wishlist_has_property_wishlist1_idx`(`wishlist_idwishlist` ASC) VISIBLE,
491     CONSTRAINT `fk_wishlist_has_property_property1`
492         FOREIGN KEY (`property_idproperty`)
493             REFERENCES `Airbnb`.`property`(`idproperty`)
494             ON DELETE CASCADE
495             ON UPDATE CASCADE,
496     CONSTRAINT `fk_wishlist_has_property_wishlist1`
497         FOREIGN KEY (`wishlist_idwishlist`)
498             REFERENCES `Airbnb`.`wishlist`(`idwishlist`)
499             ON DELETE CASCADE
500             ON UPDATE CASCADE)
501 ENGINE = InnoDB
502 DEFAULT CHARACTER SET = utf8mb3;

```

In order to display the N:M relationship between `wishlist` and `property`, I created a supporting table `wishlist\_has\_property`. This table shows only two PKs which are also FKs to respective tables.

```

1 •   SELECT b.name AS "wishlist_name", c.title AS "property_title"
2   FROM Airbnb.wishlist_has_property a
3   JOIN wishlist b ON a.wishlist_idwishlist = b.idwishlist
4   JOIN property c ON a.wishlist_idwishlist = c.idproperty
5   LIMIT 10;

```

wishlist_name	property_title
▶ egestas. Sed pharetra, bibendum fermentum metus. vehicula aliquet libero.	Fusce feugiat. nonummy ipsum risus varius
▶ luctus felis purus tempus scelerisque, lorem	tincidunt dui Donec fringilla.
▶ ultrices sit amet, Duis gravida. Praesent	Mauris nulla. lorem, eget
▶ Nulla aliquet. Proin amet, consectetur adipiscing	venenatis lacus. eleifend non,
▶ egestas. Sed pharetra,	Fusce feugiat.

## CREATE TABLE `user`

```

184 CREATE TABLE IF NOT EXISTS `Airbnb`.`user` (
185     `iduser` INT NOT NULL AUTO_INCREMENT,
186     `first_name` VARCHAR(45) NOT NULL,
187     `last_name` VARCHAR(45) NOT NULL,
188     `gender` VARCHAR(45) NOT NULL,
189     `password` VARCHAR(45) NOT NULL,
190     `email` VARCHAR(45) NOT NULL,
191     `phone` VARCHAR(45) NULL DEFAULT NULL,
192     `user_type` ENUM("g", "h") NOT NULL,
193     `language` VARCHAR(45) NULL DEFAULT 'english',
194     `social_media_idsocial_media` INT NULL DEFAULT '1',
195     PRIMARY KEY (`iduser`),
196     INDEX `fk_user_social_media1_idx` (`social_media_idsocial_media` ASC) VISIBLE,
197     INDEX `language_idx` (`language` ASC) VISIBLE,
198     CONSTRAINT `fk_user_social_media1`
199         FOREIGN KEY (`social_media_idsocial_media`)
200             REFERENCES `Airbnb`.`social_media`(`idsocial_media`),
201         CONSTRAINT `language`
202             FOREIGN KEY (`language`)
203             REFERENCES `Airbnb`.`language`(`language`)
204             ON DELETE CASCADE
205             ON UPDATE CASCADE)
206 ENGINE = InnoDB
207 AUTO_INCREMENT = 31
208 DEFAULT CHARACTER SET = utf8mb3;

```

The `user` table is a very important table. It stores all information about a user and its relationship to listings. A key attribute is the classification of the `user_type` which is a BOOLEAN string disguised in the `ENUM()` command. `ENUM` allows to only choose between the given options. There are two FK coming from `'language'` and `'social_media'` connecting both tables to user.

```

1 •   SELECT user_type, COUNT(*) AS "count"
2   FROM Airbnb.user
3   GROUP BY user_type;

```

user_type	count
g	20
h	10

## CREATE TABLE `payment\_method`

```

214 CREATE TABLE IF NOT EXISTS `Airbnb`.`payment_method` (
215     `fk_payment_method_user` INT NOT NULL AUTO_INCREMENT,
216     `payment_method` VARCHAR(45) NULL DEFAULT NULL,
217     PRIMARY KEY (`fk_payment_method_user`),
218     CONSTRAINT `fk_paymentmethod_user_1`
219         FOREIGN KEY (`fk_payment_method_user`)
220             REFERENCES `Airbnb`.`user` (`iduser`)
221             ON DELETE CASCADE
222             ON UPDATE CASCADE)
223 ENGINE = InnoDB
224 AUTO_INCREMENT = 21
225 DEFAULT CHARACTER SET = utf8mb3;

```

The payment method has a FK as PK connecting to `user`. As it is a 1:1 relationship the FK and PK can be the same. To uphold data integrity, CASCades have been built in.

```

1 •   SELECT
2      CONCAT(b.first_name, " ", b.last_name) AS "user_name",
3      a.payment_method
4  FROM Airbnb.payment_method a
5  JOIN user b ON a.fk_payment_method_user = b.iduser
6  WHERE CONCAT(b.first_name, " ", b.last_name) = "Destiny Armstrong";
7

```

user_name	payment_method
Destiny Armstrong	applepay

## CREATE TABLE `price`

```

231  CREATE TABLE IF NOT EXISTS `Airbnb`.`price` (
232    `idprice` INT NOT NULL AUTO_INCREMENT,
233    `base_price` INT NULL DEFAULT NULL,
234    `discount_per_week` INT NULL DEFAULT NULL,
235    `discount_per_month` INT NULL DEFAULT NULL,
236    `service_fee` INT NULL DEFAULT NULL,
237    `airbnb_fee` INT NULL DEFAULT NULL,
238    PRIMARY KEY (`idprice`)
239    CONSTRAINT `fk_price_property_1`
240      FOREIGN KEY (`idprice`)
241        REFERENCES `Airbnb`.`property` (`idproperty`)
242        ON DELETE CASCADE
243        ON UPDATE CASCADE)
244 ENGINE = InnoDB
245 AUTO_INCREMENT = 21
246 DEFAULT CHARACTER SET = utf8mb3;

```

This table has all price related data in it. The surrogate PK makes sure that every price is identifiable. It has relationship to the `booking` and `property` tables. There is one FK in the `price` table.

```

1   SELECT
2     a.idproperty,
3     a.title,
4     b.base_price
5   FROM property a
6   JOIN price b ON a.idproperty = b.idprice
7   WHERE b.base_price < 100 AND b.base_price > 50
8   ORDER BY b.base_price ASC;

```

idproperty	title	base_price
3	nonummy ipsum	59
20	purus, accumsan	62
9	Phasellus at	67
19	venenatis lacus.	88
4	pede, nonummy	93

## CREATE TABLE `booking`

```

383 CREATE TABLE IF NOT EXISTS `Airbnb`.`booking` (
384     `idbooking` INT NOT NULL AUTO_INCREMENT,
385     `check_in` DATE NOT NULL,
386     `check_out` DATE NOT NULL,
387     `number_guests` INT NOT NULL,
388     `property_idproperty` INT NOT NULL,
389     `user_iduser` INT NOT NULL,
390     `payment_method_idpayment_method` INT NOT NULL,
391     `price_idprice` INT NOT NULL,
392     PRIMARY KEY (`idbooking`),
393     INDEX `fk_booking_property1_idx` (`property_idproperty` ASC) VISIBLE,
394     INDEX `fk_booking_user1_idx` (`user_iduser` ASC) VISIBLE,
395     INDEX `fk_booking_payment_method1_idx` (`payment_method_idpayment_method` ASC) VISIBLE,
396     INDEX `fk_booking_price1_idx` (`price_idprice` ASC) VISIBLE,
397     CONSTRAINT `fk_booking_property1`
398         FOREIGN KEY (`property_idproperty`)
399             REFERENCES `Airbnb`.`property` (`idproperty`)
400             ON DELETE CASCADE
401             ON UPDATE CASCADE,
402     CONSTRAINT `fk_booking_user1`
403         FOREIGN KEY (`user_iduser`)
404             REFERENCES `Airbnb`.`user` (`iduser`)
405             ON DELETE CASCADE
406             ON UPDATE CASCADE)
407 ENGINE = InnoDB
408 AUTO_INCREMENT = 21
409 DEFAULT CHARACTER SET = utf8mb3;

```

This table comes with all information necessary to finish the booking process. PK is surrogate and the FK connect to the most important tables.

Property_name	Guest	check_in	check_out	duration_of_st...
lorem. Donec pede, nonummy	Daria Vang	2021-05-04	2021-05-08	4
Mauris nulla.	Carissa Todd	2021-05-03	2021-05-07	4
lorem, eget	Brenden Mullen	2021-05-02	2021-05-06	4
risus varius	Anthony Kent	2021-05-03	2021-05-07	4
Donec fringilla.	Anthony Kent	2021-05-02	2021-05-06	4
Phasellus dolor	Dahlia Tyson	2021-05-02	2021-05-06	4
Phasellus dolor	Leonard Benjamin	2021-05-03	2021-05-07	4
Mauris nulla.	Aristotle Roach	2021-05-04	2021-05-08	3
lorem, eget	Whoopi Madden	2021-05-03	2021-05-06	3
	Wayne Cleveland	2021-05-05	2021-05-08	3

```

1 •  SELECT
2     b.title AS "Property_name",
3     CONCAT(c.first_name, " ", c.last_name) AS "Guest",
4     a.check_in,
5     a.check_out,
6     (a.check_out - check_in) AS "duration_of_stay"
7     FROM Airbnb.booking a
8     JOIN property b ON a.property_idproperty = b.idproperty
9     JOIN user c ON a.user_iduser = c.iduser
10    ORDER BY a.check_out - check_in DESC
11    LIMIT 10;

```

# Data entry

FOR DATA ENTRY I USED  
[GENERATEDATA.COM](#) FOR  
SIMPLER AND RANDOMIZED  
DATA ENTRY