Database for Renting Apartments and Bedrooms
Concept Phase: The Definition and Requirement Specification

Software for processing: Word, exported to Pdf, Tool to draw ER-Models: drawsql, DBMS: MySQL.

Roles (Person/ User groups):

- **Host** serve with their properties (Green colour in ER Model).
- **Guest** make booking and payment (Yellow colour in ER Model).
- **Marketplace** as bridge between Host and Guest, handling the booking and payment through Credit Card (Red colour in ER Model), and pay the host after cut by commissions.

property_host:
Each host can be related to any number of booking/ bookings.
Each host can own at least one property. Each host can have more than one property.

property:
Each property has at least one room. Can be several rooms.
Property is available, if property has at least one room table with room_availability column = TRUE
.

property_Type:
Each type haz any property. Each property has one type.

room:
Each room is child of one property. Each property has at least one room.
Guest can book the room belongs to property, through the booking handling by the marketplace.
By default, room_availability is TRUE. But if it's booked through booking table, the room_availability will become FALSE.

room_Type: Each type has any room. Each room has one type.

property_commission:
Each property has each own percentage of commission cut by the marketplace, depend on negotiation and agreement between host and the marketplace.

facility: Each property has each own facility, such as internet, parking, swimming pool, etc. I don't list too many facility options here, to make it simple.

amenities: Each room has own amenity list, such as internet, aircon, heater, washer, television, kettle, refrigerator, hair dryer, welcome drink, towel, soap, shampoo, etc.

country: Each country has any number of cities. Each city located in one country.

city: Each country has any number of cities. Each city located in one country.

neighborhood:
Each neighbourhood has any properties. Each property location located in one neighbourhood.

property_review: Guest only can review the property if in payment table, payment_date is NOT NULL. If average review from more than certain number of guests are below the minimum standard rate, property can be banned, since it will affect the marketplace's image.

guest_review: Host can rate the guest 1 to 5. (1 = Terrible, 2 = Bad, 3 = Okay, 4 = Good, 5 = Excellent). If average review from more than certain number of hosts are below 3, guest can be banned by the marketplace.

guest_commission: Beside take commission from host, market place take commission from guest.

guest: Guest have guest_level. Can be level up through higher rating and loyalty.

booking:
Booking table is the centre/ heart of this business. It has relation to host, room, guest, payment, cancelation, property_review. Not every booking will move on to payment table. Some of them will go to cancelation table.

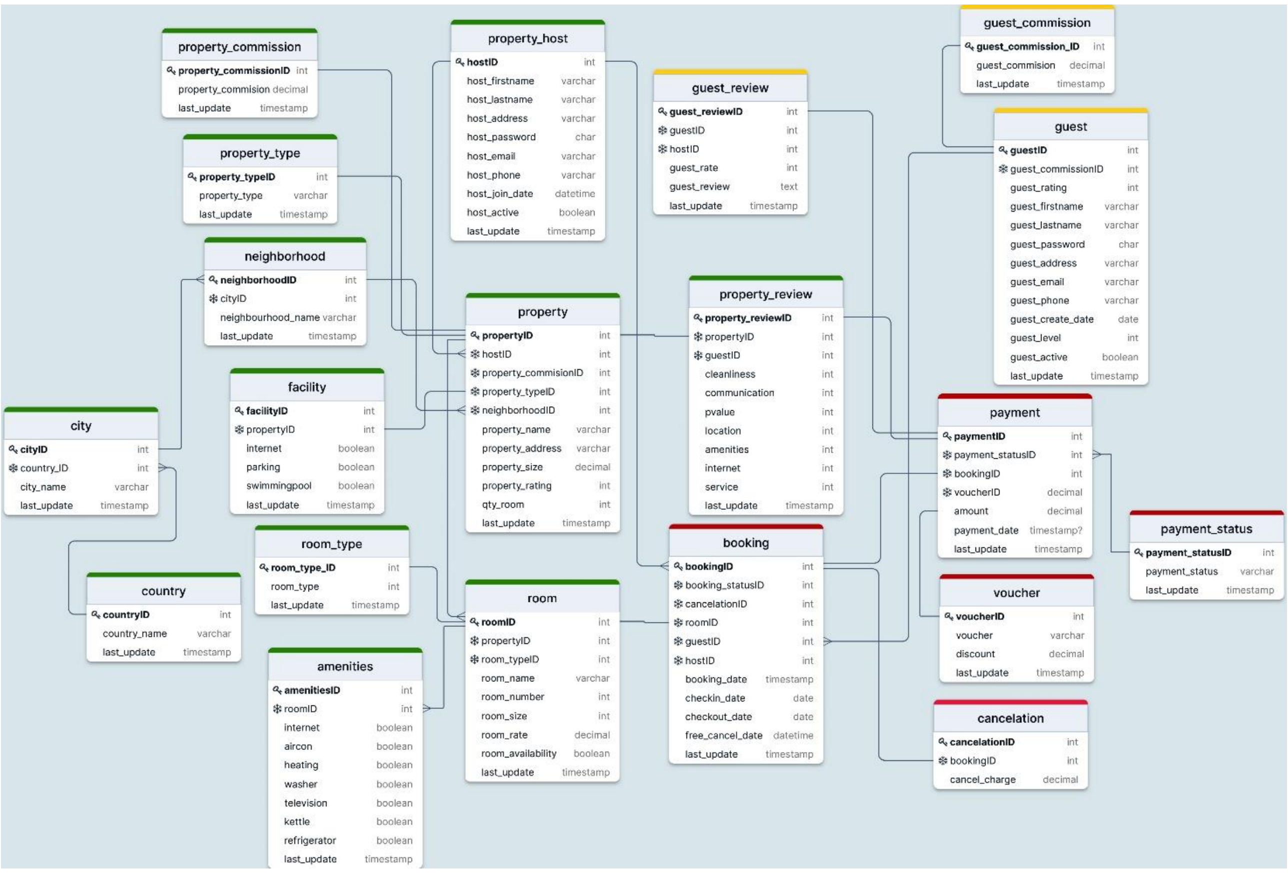
voucher: Voucher can be given as promotion to certain guest or location, related to event/ ads.

payment: Guest pay through Credit Card. Marketplace hold the payment for 24 hours after the check-in date.

payment_Status: Status of payment, either on-hold period, paid, or having problem.

cancelation:
Each booking has zero or 1 cancelation. Each cancelation related to one bookingID.
Cancelation before free_cancel_date time column in booking table is free of charge. Otherwise guest will be charged by amount indicated in cancel_charge column inside Cancelation table.

ER Diagram:


property_commission
property_commissionID int
property_commision decimal
last_update timestamp
property_type
property_typeID int
property_type varchar
last_update timestamp
neighborhood
neighborhoodID int
cityID int
neighbourhood_name varchar
last_update timestamp
city
cityID int
country_ID int
city_name varchar
last_update timestamp
facility
facilityID int
propertyID int
internet boolean
parking boolean
swimmingpool boolean
last_update timestamp
country
countryID int
country_name varchar
last_update timestamp
room_type
room_type_ID int
room_type int
last_update timestamp
amenities
amenitiesID int
roomID int
internet boolean
aircon boolean
heating boolean
washer boolean
television boolean
kettle boolean
refrigerator boolean
last_update timestamp
property_host
hostID int
host_firstname varchar
host_lastname varchar
host_address varchar
host_password char
host_email varchar
host_phone varchar
host_join_date datetime
host_active boolean
last_update timestamp
property
propertyID int
hostID int
property_commisionID int
property_typeID int
neighborhoodID int
property_name varchar
property_address varchar
property_size decimal
property_rating int
qty_room int
last_update timestamp
room
roomID int
propertyID int
room_typeID int
room_name varchar
room_number int
room_size int
room_rate decimal
room_availability boolean
last_update timestamp
guest_review
guest_reviewID int
guestID int
hostID int
guest_rate int
guest_review text
last_update timestamp
property_review
property_reviewID int
propertyID int
guestID int
cleanliness int
communication int
pvalue int
location int
amenities int
internet int
service int
last_update timestamp
booking
bookingID int
booking_statusID int
cancelationID int
roomID int
guestID int
hostID int
booking_date timestamp
checkin_date date
checkout_date date
free_cancel_date datetime
last_update timestamp
guest_commission
guest_commission_ID int
guest_commision decimal
last_update timestamp
guest
guestID int
guest_commissionID int
guest_rating int
guest_firstname varchar
guest_lastname varchar
guest_password char
guest_address varchar
guest_email varchar
guest_phone varchar
guest_create_date date
guest_level int
guest_active boolean
last_update timestamp
payment
paymentID int
payment_statusID int
bookingID int
voucherID decimal
amount decimal
payment_date timestamp?
last_update timestamp
payment_status
payment_statusID int
payment_status varchar
last_update timestamp
voucher
voucherID int
voucher varchar
discount decimal
last_update timestamp
cancelation
cancelationID int
bookingID int
cancel_charge decimal

iubh INTERNATIONALE HOCHSCHULE

Build a DataMart

dlbdspbdm01

32009177

Development

with **MySQL**

DataMart Development

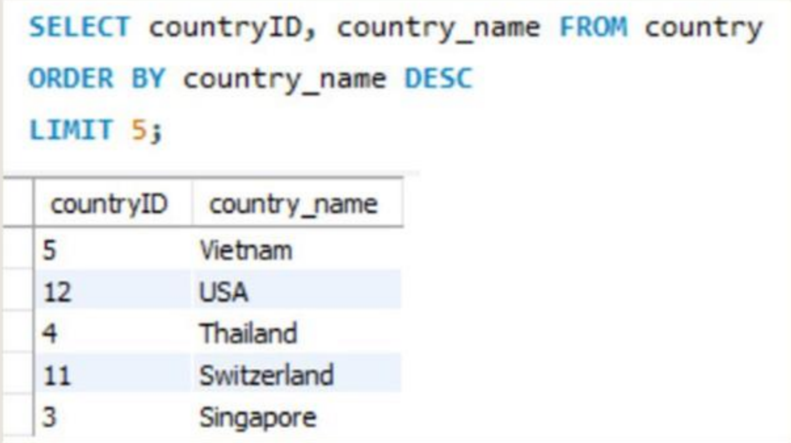
In this development phase, all SQL-Statement of each of 20 attributes creation and screenshot of test result will be recorded and presented.

Create Database :

```
DROP DATABASE IF EXISTS datamart;
CREATE DATABASE IF NOT EXISTS datamart;
USE datamart;

DROP TABLE IF EXISTS
city, country, property_commission, facility, property_type,neighborhood,
amenities, room_type, property_host, property, room,property_review,
guest_review, guest_commission, guest,booking, payment, voucher,
cancelation, payment_status;
```

country

CREATE TABLE country (countryID INTEGER AUTO_INCREMENT,
country_name VARCHAR(50),
PRIMARY KEY (countryID),last_update TIMESTAMP DEFAULT now());

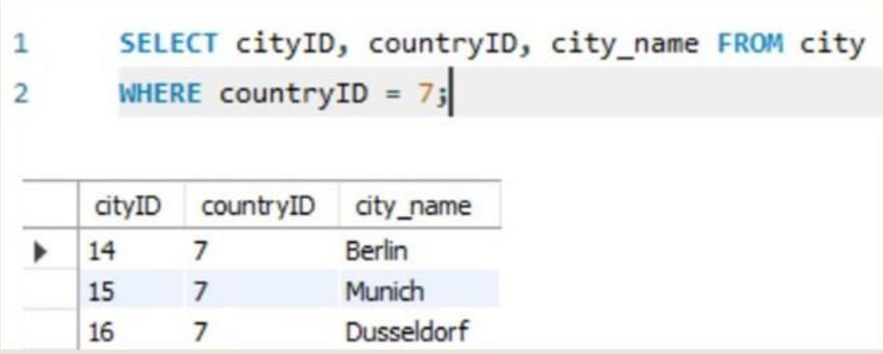


```
SELECT countryID, country_name FROM country
ORDER BY country_name DESC
LIMIT 5;
```

countryID	country_name
5	Vietnam
12	USA
4	Thailand
11	Switzerland
3	Singapore

city

CREATE TABLE city (cityID INTEGER AUTO_INCREMENT,
countryID INTEGER,city_name VARCHAR(50) NOT NULL,
PRIMARY KEY (cityID),FOREIGN KEY (countryID) REFERENCES
country(countryID), last_update TIMESTAMP DEFAULT now());

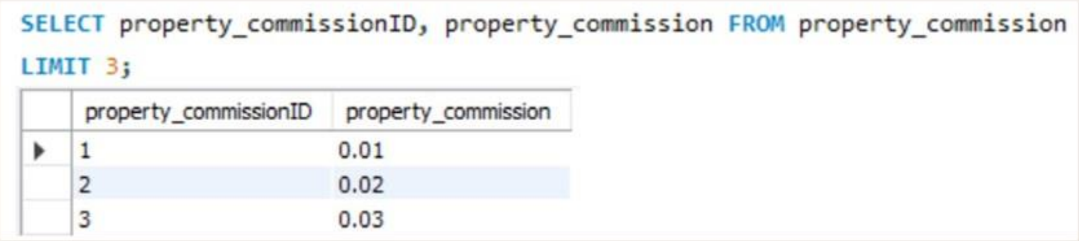


```
SELECT cityID, countryID, city_name FROM city
WHERE countryID = 7;
```

cityID	countryID	city_name
14	7	Berlin
15	7	Munich
16	7	Dusseldorf

property_commission

```
CREATE TABLE property_commission (
property_commissionID INTEGER AUTO_INCREMENT,
property_commission DECIMAL(3,2) NOT NULL,
PRIMARY KEY (property_commissionID),
last_update TIMESTAMP DEFAULT now() );
```

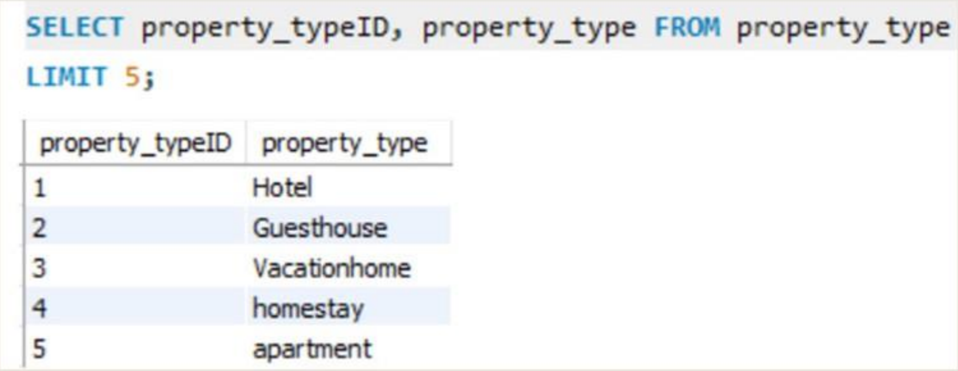


```
SELECT property_commissionID, property_commission FROM property_commission
LIMIT 3;
```

	property_commissionID	property_commission
▶	1	0.01
	2	0.02
	3	0.03

property_type

```
CREATE TABLE property_type (
property_typeID INTEGER AUTO_INCREMENT, property_type
VARCHAR(50), PRIMARY KEY (property_typeID),
last_update TIMESTAMP DEFAULT now() );
```

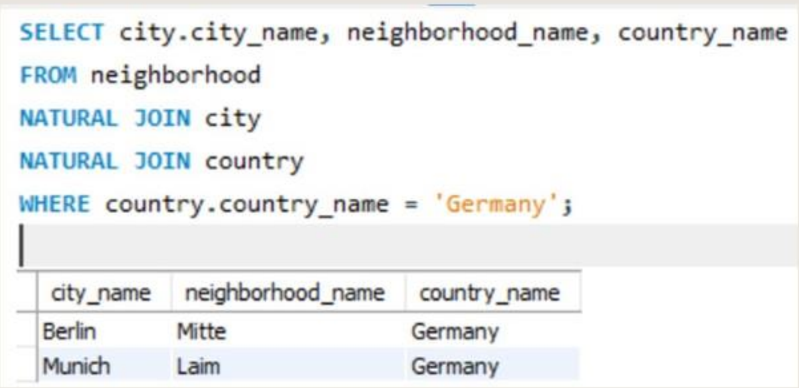


```
SELECT property_typeID, property_type FROM property_type
LIMIT 5;
```

property_typeID	property_type
1	Hotel
2	Guesthouse
3	Vacationhome
4	homestay
5	apartment

neighborhood

CREATE TABLE neighborhood(neighborhoodID INTEGER AUTO_INCREMENT, cityID INTEGER, neighborhood_name VARCHAR(50), PRIMARY KEY (neighborhoodID), FOREIGN KEY (cityID) REFERENCES city(cityID), last_update TIMESTAMP DEFAULT now());

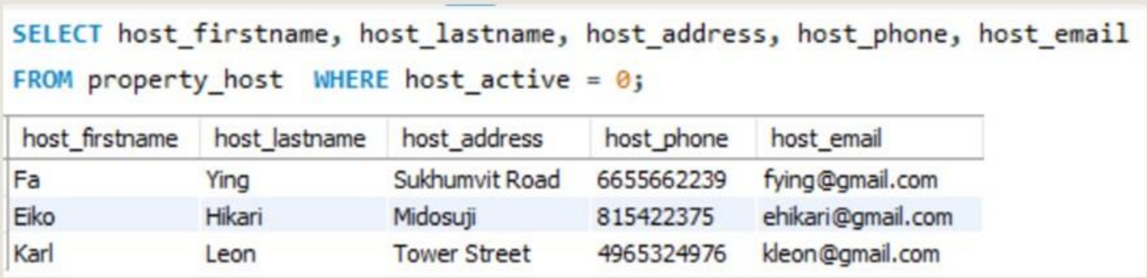


```
SELECT city.city_name, neighborhood_name, country_name
FROM neighborhood
NATURAL JOIN city
NATURAL JOIN country
WHERE country.country_name = 'Germany';
```

city_name	neighborhood_name	country_name
Berlin	Mitte	Germany
Munich	Laim	Germany

property_host

CREATE TABLE property_host (
hostID INTEGER AUTO_INCREMENT,
host_firstname VARCHAR(50), host_lastname VARCHAR(50),
host_address VARCHAR(200),
host_email VARCHAR(50), host_password CHAR(8),
host_phone VARCHAR(20), host_join_date DATETIME,
host_active BOOLEAN, PRIMARY KEY (hostID),
last_update TIMESTAMP DEFAULT now());



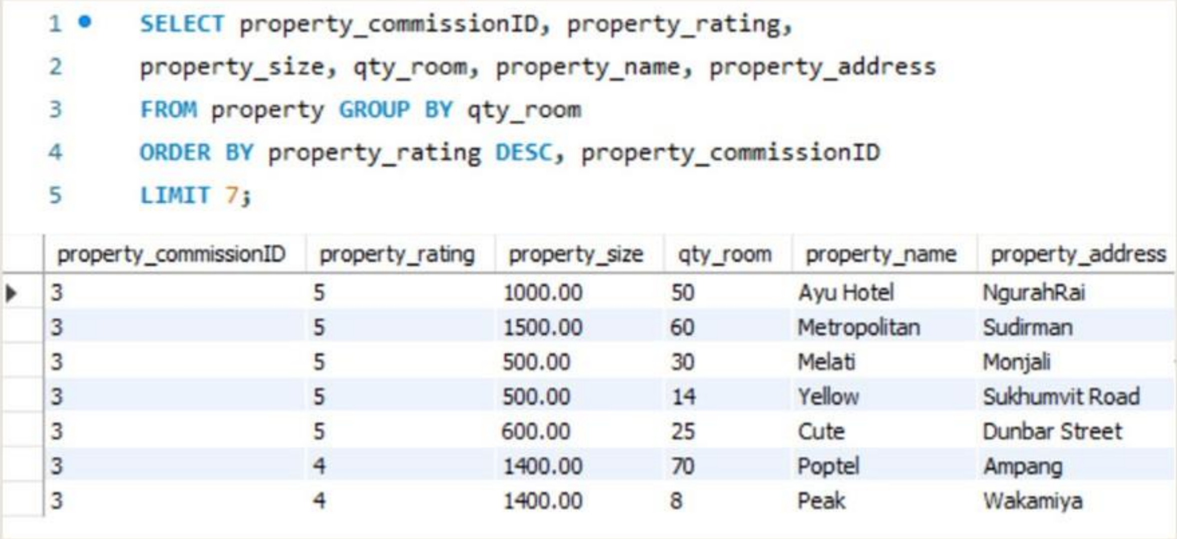
```
SELECT host_firstname, host_lastname, host_address, host_phone, host_email
FROM property_host  WHERE host_active = 0;
```

host_firstname	host_lastname	host_address	host_phone	host_email
Fa	Ying	Sukhumvit Road	6655662239	fying@gmail.com
Eiko	Hikari	Midosuji	815422375	ehikari@gmail.com
Karl	Leon	Tower Street	4965324976	kleon@gmail.com

property

```
CREATE TABLE property (
propertyID INTEGER AUTO_INCREMENT,
hostID INTEGER, property_typeID INTEGER,
neighborhoodID INTEGER,
property_commissionID INTEGER,
property_name VARCHAR(50),
property_address VARCHAR(200),
property_size DECIMAL(10,2),
property_rating INTEGER,
qty_room INTEGER,
PRIMARY KEY (propertyID), FOREIGN KEY (property_commissionID)
REFERENCES property_commission(property_commissionID),
FOREIGN KEY (hostID) REFERENCES property_host(hostID),
FOREIGN KEY (property_typeID ) REFERENCES
property_type(property_typeID ), FOREIGN KEY (neighborhoodID)
REFERENCES neighborhood(neighborhoodID),
last_update TIMESTAMP  DEFAULT now() );
```

property

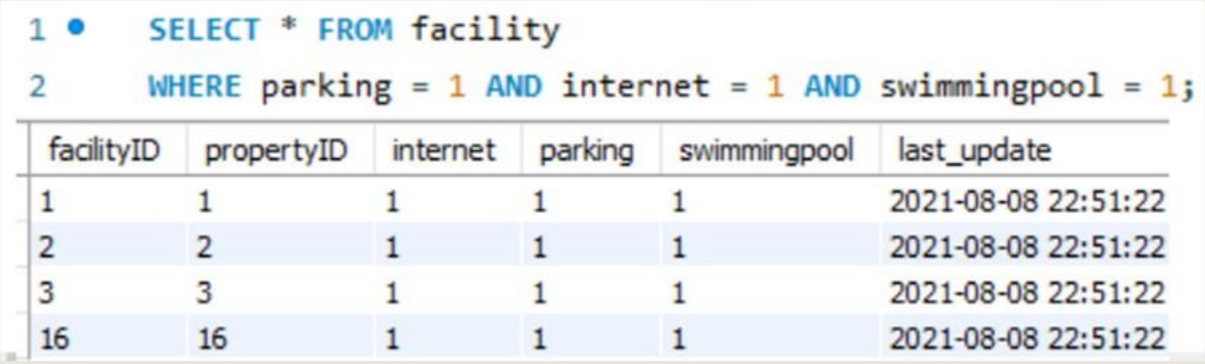


```
SELECT property_commissionID, property_rating,
property_size, qty_room, property_name, property_address
FROM property GROUP BY qty_room
ORDER BY property_rating DESC, property_commissionID
LIMIT 7;
```

property_commissionID	property_rating	property_size	qty_room	property_name	property_address
3	5	1000.00	50	Ayu Hotel	NgurahRai
3	5	1500.00	60	Metropolitan	Sudirman
3	5	500.00	30	Melati	Monjali
3	5	500.00	14	Yellow	Sukhumvit Road
3	5	600.00	25	Cute	Dunbar Street
3	4	1400.00	70	Poptel	Ampang
3	4	1400.00	8	Peak	Wakamiya

facility

```
CREATE TABLE facility (
facilityID INTEGER AUTO_INCREMENT,
propertyID INTEGER, internet BOOLEAN,
parking BOOLEAN, swimmingpool BOOLEAN,
PRIMARY KEY (facilityID),
FOREIGN KEY (propertyID) REFERENCES property(propertyID),
last_update TIMESTAMP DEFAULT now() );
```

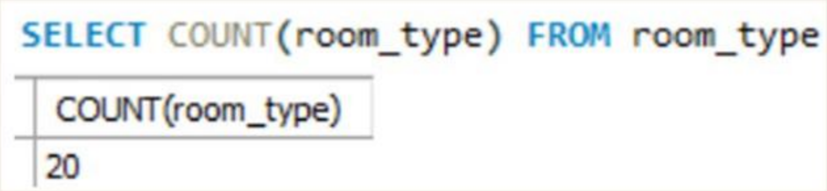


```
SELECT * FROM facility
WHERE parking = 1 AND internet = 1 AND swimmingpool = 1;
```

facilityID	propertyID	internet	parking	swimmingpool	last_update
1	1	1	1	1	2021-08-08 22:51:22
2	2	1	1	1	2021-08-08 22:51:22
3	3	1	1	1	2021-08-08 22:51:22
16	16	1	1	1	2021-08-08 22:51:22

room_type

```
CREATE TABLE room_type (
room_typeID INTEGER AUTO_INCREMENT,
room_type INTEGER,
PRIMARY KEY (room_typeID),
last_update TIMESTAMP DEFAULT now() );
```

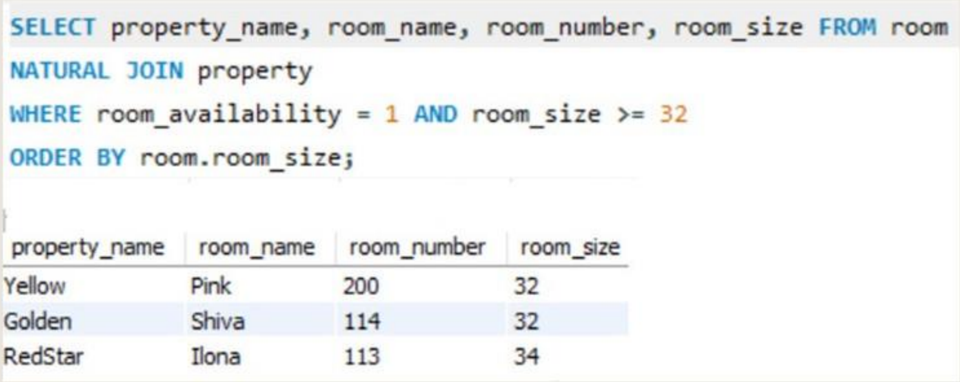


```
SELECT COUNT(room_type) FROM room_type
```

COUNT(room_type)
20

room

CREATE TABLE room(roomID INTEGER AUTO_INCREMENT, room_typeID INTEGER,propertyID INTEGER, room_name VARCHAR(50), room_number INTEGER, room_size INTEGER,room_rate DECIMAL, oom_availability BOOLEAN, PRIMARY KEY (roomID), FOREIGN KEY (room_typeID) REFERENCES room_type (room_typeID), FOREIGN KEY (propertyID) REFERENCES property(propertyID), last_update TIMESTAMP DEFAULT now());););

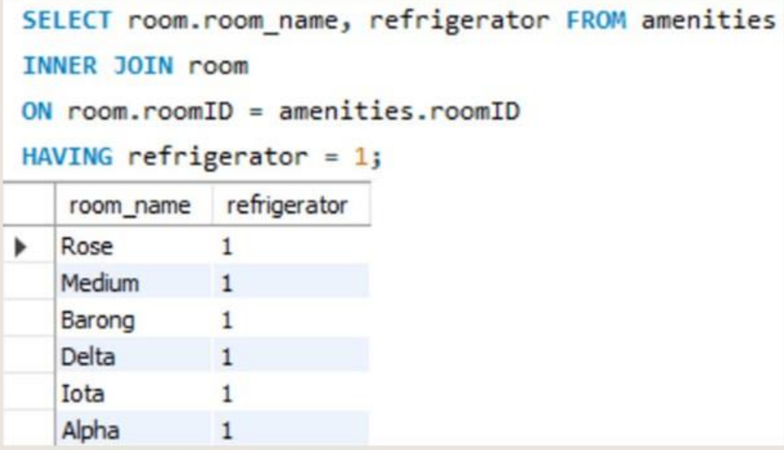


```
SELECT property_name, room_name, room_number, room_size FROM room
NATURAL JOIN property
WHERE room_availability = 1 AND room_size >= 32
ORDER BY room.room_size;
```

property_name	room_name	room_number	room_size
Yellow	Pink	200	32
Golden	Shiva	114	32
RedStar	Ilona	113	34

amenities

CREATE TABLE amenities (
amenitiesID INTEGER AUTO_INCREMENT,
roomID INTEGER, internet BOOLEAN, aircon BOOLEAN,
heating BOOLEAN, washer BOOLEAN, television BOOLEAN,
kettle BOOLEAN, refrigerator BOOLEAN,
PRIMARY KEY (amenitiesID),
FOREIGN KEY (roomID) REFERENCES room(roomID),
last_update TIMESTAMP DEFAULT now());

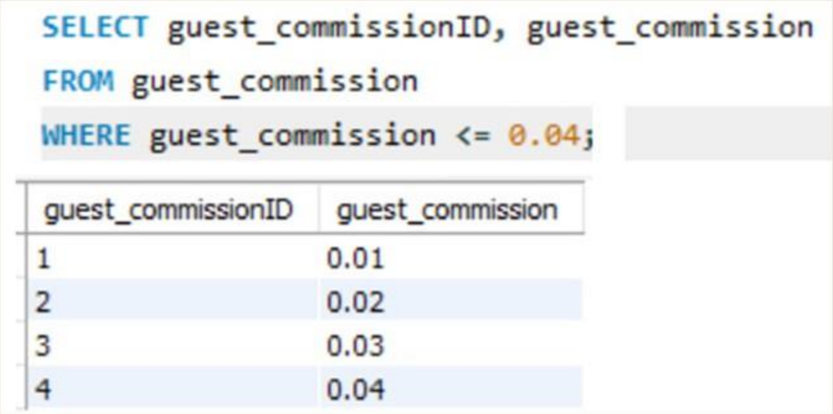


```
SELECT room.room_name, refrigerator FROM amenities
INNER JOIN room
ON room.roomID = amenities.roomID
HAVING refrigerator = 1;
```

	room_name	refrigerator
▶	Rose	1
	Medium	1
	Barong	1
	Delta	1
	Iota	1
	Alpha	1

guest_commission

CREATE TABLE guest_commission(guest_commissionID INTEGER AUTO_INCREMENT,guest_commission DECIMAL(3,2), PRIMARY KEY (guest_commissionID), last_update TIMESTAMP DEFAULT now());

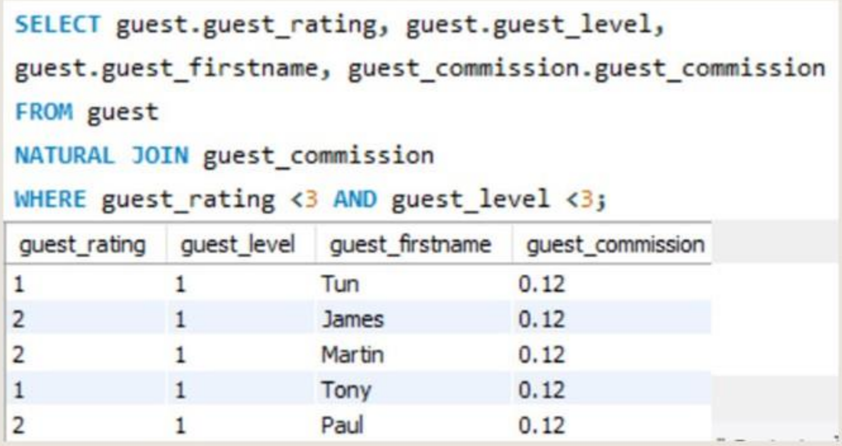


```
SELECT guest_commissionID, guest_commission
FROM guest_commission
WHERE guest_commission <= 0.04;
```

guest_commissionID	guest_commission
1	0.01
2	0.02
3	0.03
4	0.04

guest

CREATE TABLE guest (guestID INTEGER AUTO_INCREMENT, guest_commissionID INTEGER, guest_rating INTEGER, guest_firstname VARCHAR(50), guest_lastname VARCHAR(50), guest_address VARCHAR(50), guest_email VARCHAR(50), guest_password CHAR(8), guest_phone VARCHAR(20), guest_create_date DATETIME, guest_active BOOLEAN, guest_level INTEGER, PRIMARY KEY (guestID), FOREIGN KEY (guest_commissionID) REFERENCES guest_commission (guest_commissionID),last_update TIMESTAMP DEFAULT now());

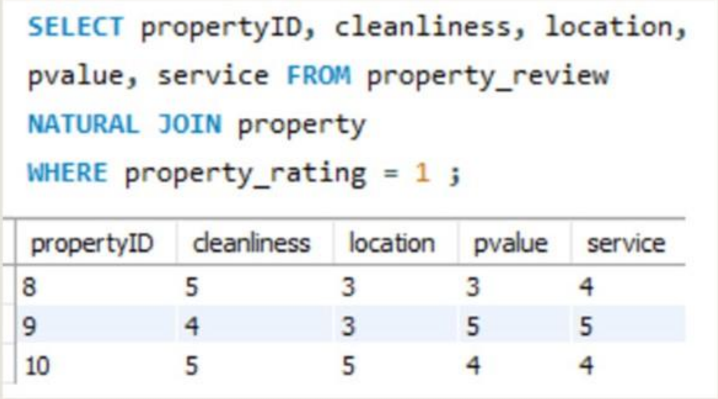


```
SELECT guest.guest_rating, guest.guest_level,
guest.guest_firstname, guest_commission.guest_commission
FROM guest
NATURAL JOIN guest_commission
WHERE guest_rating <3 AND guest_level <3;
```

guest_rating	guest_level	guest_firstname	guest_commission
1	1	Tun	0.12
2	1	James	0.12
2	1	Martin	0.12
1	1	Tony	0.12
2	1	Paul	0.12

property_review

CREATE TABLE property_review (property_reviewID INTEGER AUTO_INCREMENT, propertyID INTEGER, guestID INTEGER, cleanliness INTEGER, communication INTEGER, pvalue INTEGER, location INTEGER, amenities INTEGER, internet INTEGER, service INTEGER, PRIMARY KEY (property_reviewID),
FOREIGN KEY (guestID) REFERENCES guest(guestID),
FOREIGN KEY (propertyID) REFERENCES property (propertyID),
last_update TIMESTAMP DEFAULT now());

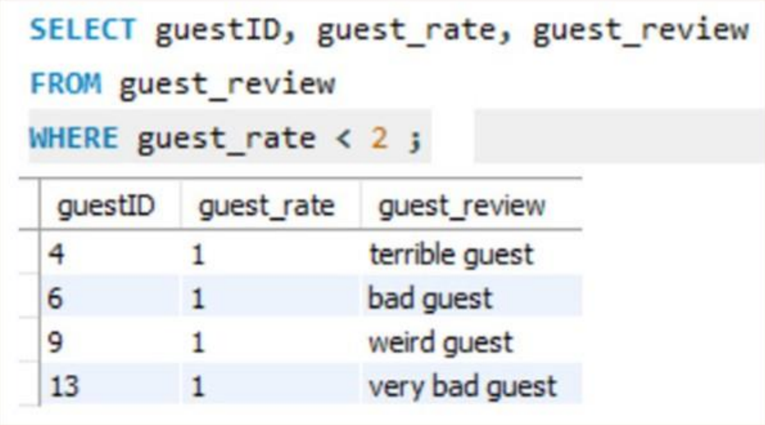


```
SELECT propertyID, cleanliness, location,
pvalue, service FROM property_review
NATURAL JOIN property
WHERE property_rating = 1 ;
```

propertyID	cleanliness	location	pvalue	service
8	5	3	3	4
9	4	3	5	5
10	5	5	4	4

guest_review

CREATE TABLE guest_review(
guest_reviewID INTEGER AUTO_INCREMENT,
guestID INTEGER, hostID INTEGER, guest_rate INTEGER,
guest_review TEXT, PRIMARY KEY (guest_reviewID),
FOREIGN KEY (guestID) REFERENCES guest(guestID),
FOREIGN KEY (hostID) REFERENCES property_host(hostID),
last_update TIMESTAMP DEFAULT now());

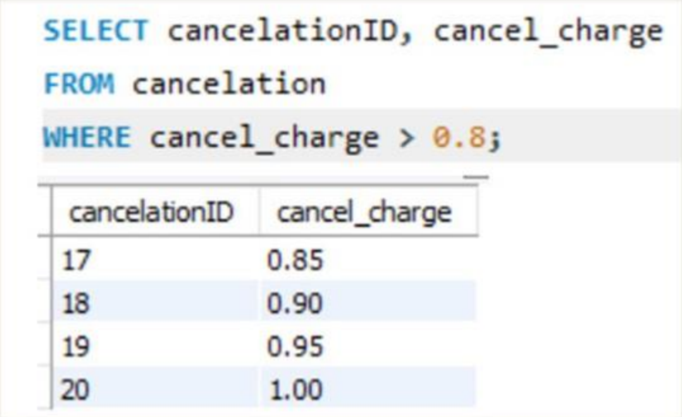


```
SELECT guestID, guest_rate, guest_review
FROM guest_review
WHERE guest_rate < 2 ;
```

guestID	guest_rate	guest_review
4	1	terrible guest
6	1	bad guest
9	1	weird guest
13	1	very bad guest

cancelation

```
CREATE TABLE cancelation (
cancelationID INTEGER AUTO_INCREMENT,
cancel_charge DECIMAL(10,2),
PRIMARY KEY (cancelationID),
last_update TIMESTAMP DEFAULT now() );
```

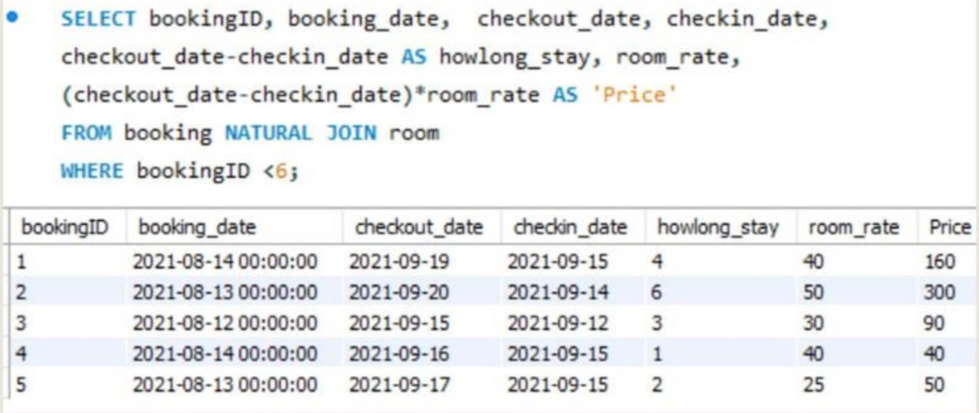


```
SELECT cancelationID, cancel_charge
FROM cancelation
WHERE cancel_charge > 0.8;
```

cancelationID	cancel_charge
17	0.85
18	0.90
19	0.95
20	1.00

booking

```
CREATE TABLE booking (bookingID INTEGER AUTO_INCREMENT,
guestID INTEGER,hostID INTEGER, booking_date TIMESTAMP,
checkin_date DATE, checkout_date DATE,free_cancel_date DATETIME,
PRIMARY KEY (bookingID), FOREIGN KEY (roomID) REFERENCES
room(roomID),FOREIGN KEY (guestID) REFERENCES guest(guestID),
FOREIGN KEY (hostID) REFERENCES property_host(hostID),
FOREIGN KEY (cancelationID) REFERENCES cancelation(cancelationID),
CHECK (booking_date <= free_cancel_date <= checkin_date < checkout_date),
last_update TIMESTAMP DEFAULT now() );
```

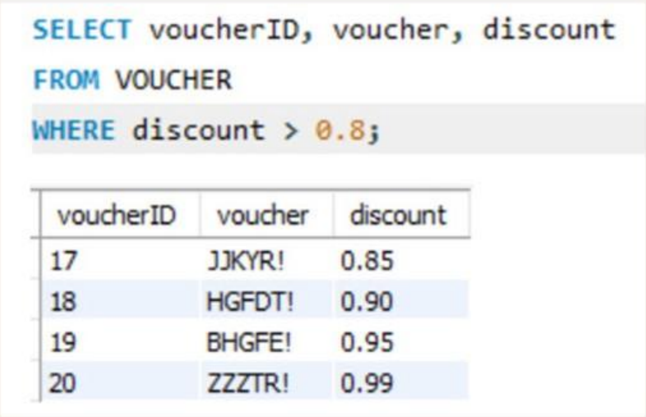


```
SELECT bookingID, booking_date,  checkout_date, checkin_date,
checkout_date-checkin_date AS howlong_stay, room_rate,
(checkout_date-checkin_date)*room_rate AS 'Price'
FROM booking NATURAL JOIN room
WHERE bookingID <6;
```

bookingID	booking_date	checkout_date	checkin_date	howlong_stay	room_rate	Price
1	2021-08-14 00:00:00	2021-09-19	2021-09-15	4	40	160
2	2021-08-13 00:00:00	2021-09-20	2021-09-14	6	50	300
3	2021-08-12 00:00:00	2021-09-15	2021-09-12	3	30	90
4	2021-08-14 00:00:00	2021-09-16	2021-09-15	1	40	40
5	2021-08-13 00:00:00	2021-09-17	2021-09-15	2	25	50

voucher

```
CREATE TABLE voucher (voucherID INTEGER AUTO_INCREMENT,
voucher VARCHAR(20),
discount DECIMAL(10,2),
PRIMARY KEY (voucherID),
last_update TIMESTAMP  DEFAULT  now() );
```

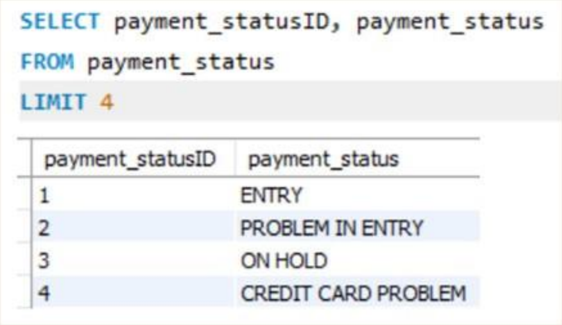


```
SELECT voucherID, voucher, discount
FROM VOUCHER
WHERE discount > 0.8;
```

voucherID	voucher	discount
17	JJKYR!	0.85
18	HGFDT!	0.90
19	BHGFE!	0.95
20	ZZZTR!	0.99

payment_status

```
CREATE TABLE payment_status (
payment_statusID INTEGER AUTO_INCREMENT,
payment_status  VARCHAR(50),
PRIMARY KEY (payment_statusID),
last_update TIMESTAMP  DEFAULT  now() );
```

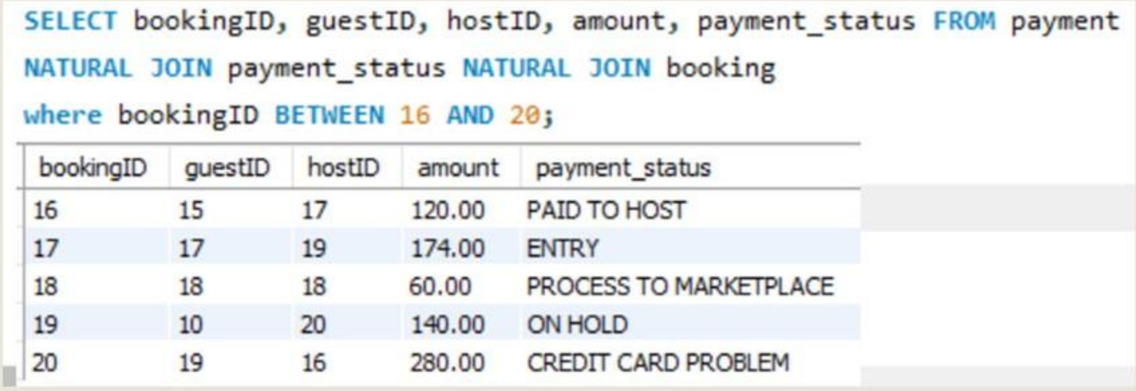


```
SELECT payment_statusID, payment_status
FROM payment_status
LIMIT 4
```

payment_statusID	payment_status
1	ENTRY
2	PROBLEM IN ENTRY
3	ON HOLD
4	CREDIT CARD PROBLEM

payment

```
CREATE TABLE payment (paymentID INTEGER
AUTO_INCREMENT,payment_statusID INTEGER,bookingID INTEGER,
voucherID INTEGER,amount DECIMAL(10,2),payment_date
TIMESTAMP,PRIMARY KEY (paymentID), FOREIGN KEY
(payment_statusID) REFERENCES
payment_status(payment_statusID),FOREIGN KEY (bookingID)
REFERENCES booking(bookingID),FOREIGN KEY (voucherID)
REFERENCES voucher(voucherID),last_update TIMESTAMP DEFAULT
now() );
```



```
SELECT bookingID, guestID, hostID, amount, payment_status FROM payment
NATURAL JOIN payment_status NATURAL JOIN booking
where bookingID BETWEEN 16 AND 20;
```

bookingID	guestID	hostID	amount	payment_status
16	15	17	120.00	PAID TO HOST
17	17	19	174.00	ENTRY
18	18	18	60.00	PROCESS TO MARKETPLACE
19	10	20	140.00	ON HOLD
20	19	16	280.00	CREDIT CARD PROBLEM

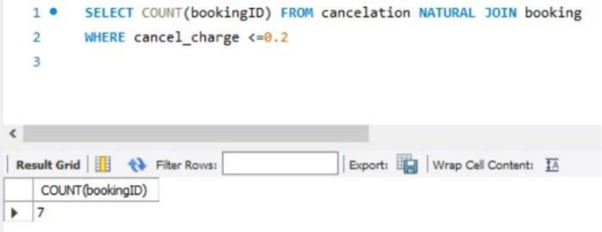
Database for Renting Apartments and Bedrooms

Finalization Phase

Database Management Functionality:

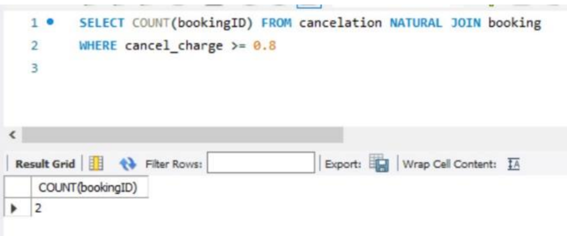
To can gain information to make decision. Several examples:

1. To analyse: Did too high cancelation charge relate to low booking quantity? Did low cancelation charge relate to high booking quantity? From data below, seems yes.
   Low cancelation charge attract 7 bookings, high cancelation charge only attract 2 bookings.



```sql
SELECT COUNT(bookingID) FROM cancelation NATURAL JOIN booking
WHERE cancel_charge <=0.2
```

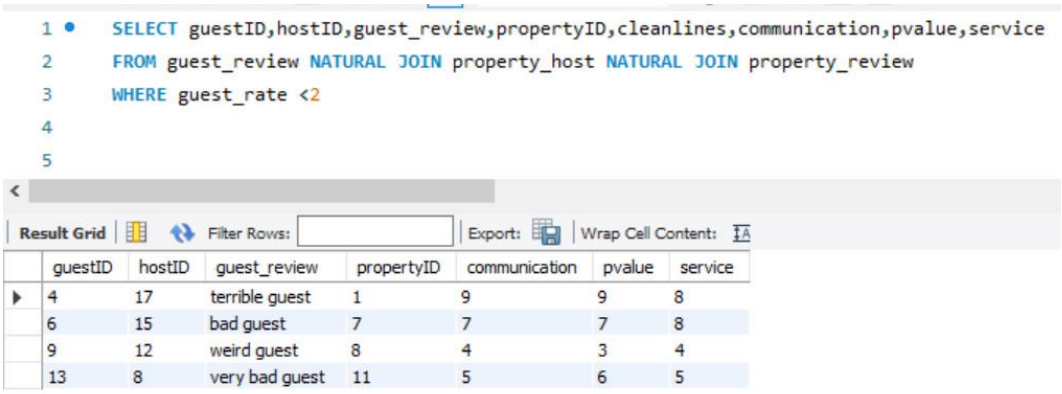
COUNT(bookingID)
7



```sql
SELECT COUNT(bookingID) FROM cancelation NATURAL JOIN booking
WHERE cancel_charge >= 0.8
```

COUNT(bookingID)
2

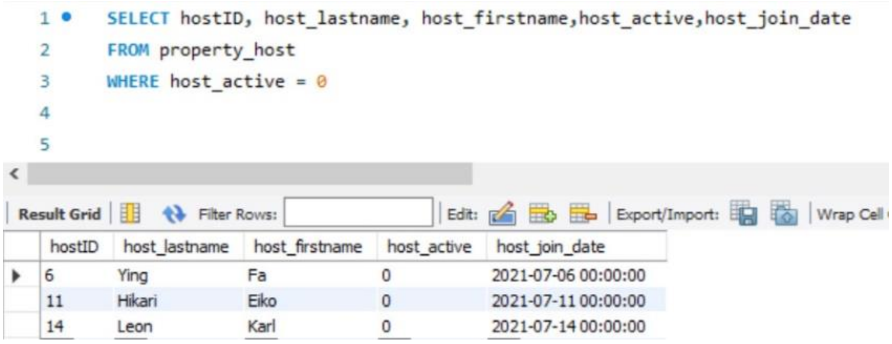
2. To ban guests who received too many bad reviews from good hosts with good property rating. From data below, we can’t trust bad review which came from bad host ID#12 for guest ID#9, since host with ID#12 bad review from other guests.
   But seems guest with ID#4 and ID#6 are really bad guests, since bad reviews came from good hosts ID #17 and #15 who received.
   If guest with ID #4 and #6 receive more negative reviews from more hosts, marketplace can ban those guests temporarily or permanently.



```sql
SELECT guestID,hostID,guest_review,propertyID,cleanlines,communication,pvalue,service
FROM guest_review NATURAL JOIN property_host NATURAL JOIN property_review
WHERE guest_rate <2
```

guestID	hostID	guest_review	propertyID	communication	pvalue	service
4	17	terrible guest	1	9	9	8
6	15	bad guest	7	7	7	8
9	12	weird guest	8	4	3	4
13	8	very bad guest	11	5	6	5

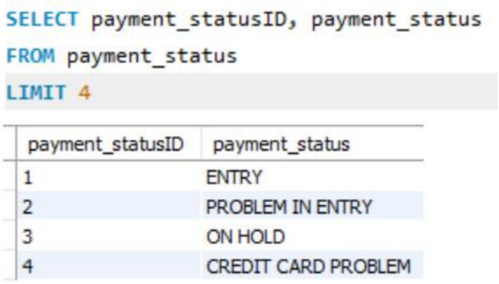
3. To decrease commission from inactive host (host_active=0) to attract them to be more active.



```sql
SELECT hostID, host_lastname, host_firstname,host_active,host_join_date
FROM property_host
WHERE host_active = 0
```

hostID	host_lastname	host_firstname	host_active	host_join_date
6	Ying	Fa	0	2021-07-06 00:00:00
11	Hikari	Eiko	0	2021-07-11 00:00:00
14	Leon	Karl	0	2021-07-14 00:00:00

4. To know in which city/ which country send marketing team to get more hosts to join the marketplace.
5. To know if there are too much payment's error within certain period.



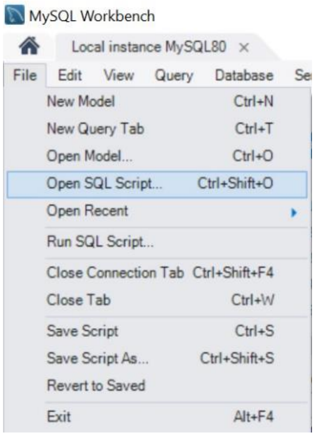
```sql
SELECT payment_statusID, payment_status
FROM payment_status
LIMIT 4
```

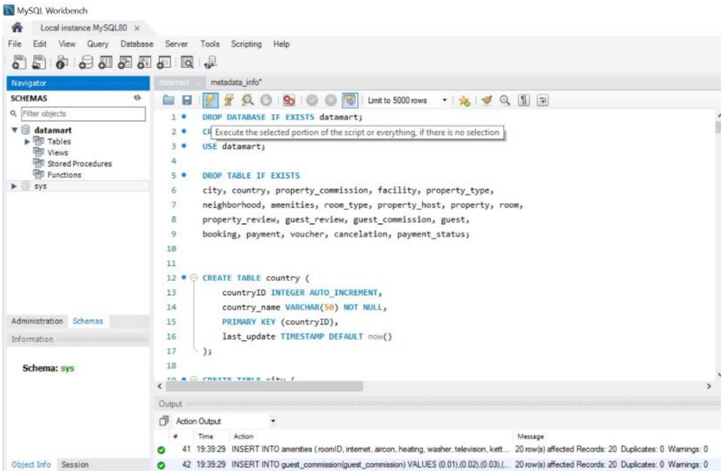
payment_statusID	payment_status
1	ENTRY
2	PROBLEM IN ENTRY
3	ON HOLD
4	CREDIT CARD PROBLEM

And many more info we can gET from the database.

Installation:

From MySQL click File, Open SQL Scripts, choose "datamart.sql" file, click "execute" icon to Run the scripts, then click "refresh" button in SCHEMAS.





Schema: sys

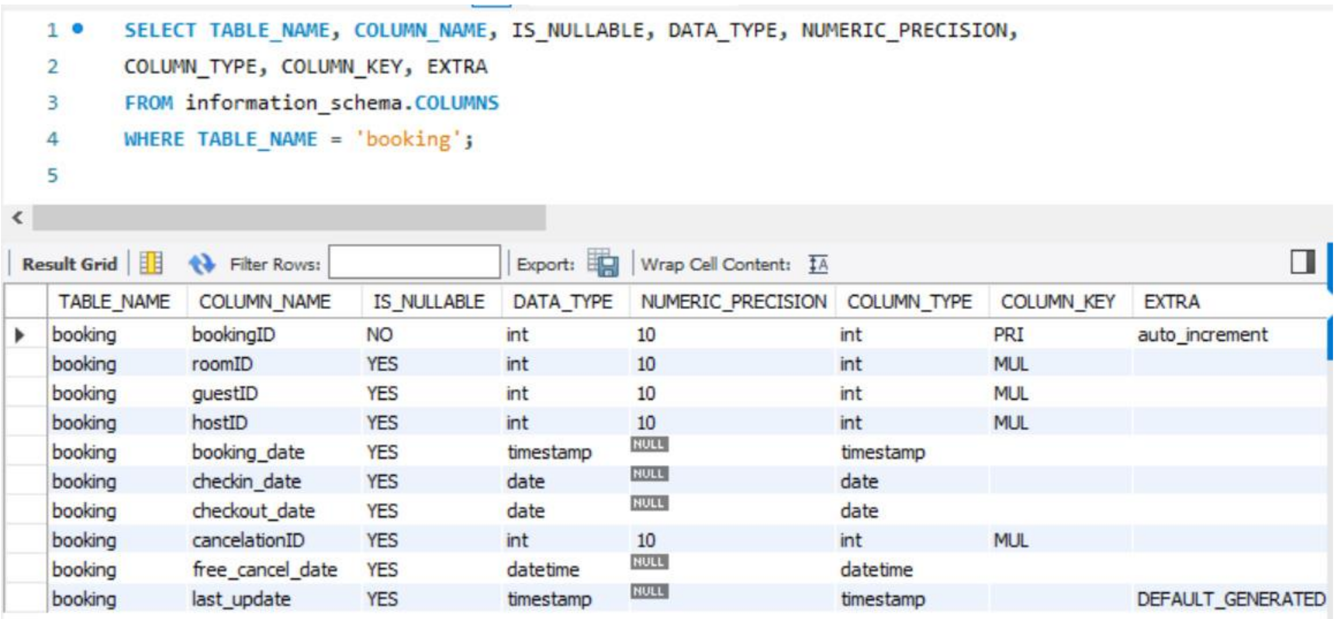
Execute the selected portion of the script or everything, if there is no selection

```sql
DROP DATABASE IF EXISTS datamart;
CR
USE datamart;

DROP TABLE IF EXISTS
city, country, property_commission, facility, property_type,
neighborhood, amenities, room_type, property_host, property, room,
property_review, guest_review, guest_commission, guest,
booking, payment, voucher, cancelation, payment_status;


CREATE TABLE country (
    countryID INTEGER AUTO_INCREMENT,
    country_name VARCHAR(50) NOT NULL,
    PRIMARY KEY (countryID),
    last_update TIMESTAMP DEFAULT now()
);

```

```
#   Time      Action                                                                       Message
41  19:39:29  INSERT INTO amenities ( roomID, internet, aircon, heating, washer, television, kett...   20 row(s) affected Records: 20  Duplicates: 0  Warnings: 0
42  19:39:29  INSERT INTO guest_commission(guest_commission) VALUES (0.01),(0.02),(0.03),(...     20 row(s) affected Records: 20  Duplicates: 0  Warnings: 0
```

Metadata:

Using scripts below to get metadata info from specific table:



```sql
SELECT TABLE_NAME, COLUMN_NAME, IS_NULLABLE, DATA_TYPE, NUMERIC_PRECISION,
COLUMN_TYPE, COLUMN_KEY, EXTRA
FROM information_schema.COLUMNS
WHERE TABLE_NAME = 'booking';

```

TABLE_NAME	COLUMN_NAME	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	COLUMN_TYPE	COLUMN_KEY	EXTRA
booking	bookingID	NO	int	10	int	PRI	auto_increment
booking	roomID	YES	int	10	int	MUL	
booking	guestID	YES	int	10	int	MUL	
booking	hostID	YES	int	10	int	MUL	
booking	booking_date	YES	timestamp	NULL	timestamp		
booking	checkin_date	YES	date	NULL	date		
booking	checkout_date	YES	date	NULL	date		
booking	cancelationID	YES	int	10	int	MUL	
booking	free_cancel_date	YES	datetime	NULL	datetime		
booking	last_update	YES	timestamp	NULL	timestamp		DEFAULT_GENERATED