BOOKINGS PROJECT

TECHNICAL DOCUMENTATION

Project created by Andres Cano

Tutors: Cesar Aracena, Jennifer Goldfeld

Index

Project Description Page 3

DER Diagrams Page 4

Table Descriptions Page 6

Project Link Page 11

Views Page 12

Functions Page 15

Stored Procedures Page 16

Triggers Page 18

Bussines Intelligence Page 20

Technologies Page 21

Project Description

Business Model

The booking platform is a peer-to-peer marketplace, connecting property owners (hosts) with travelers (guests). The platform facilitates bookings and a wide variety of properties for secure rent anywhere around the world.

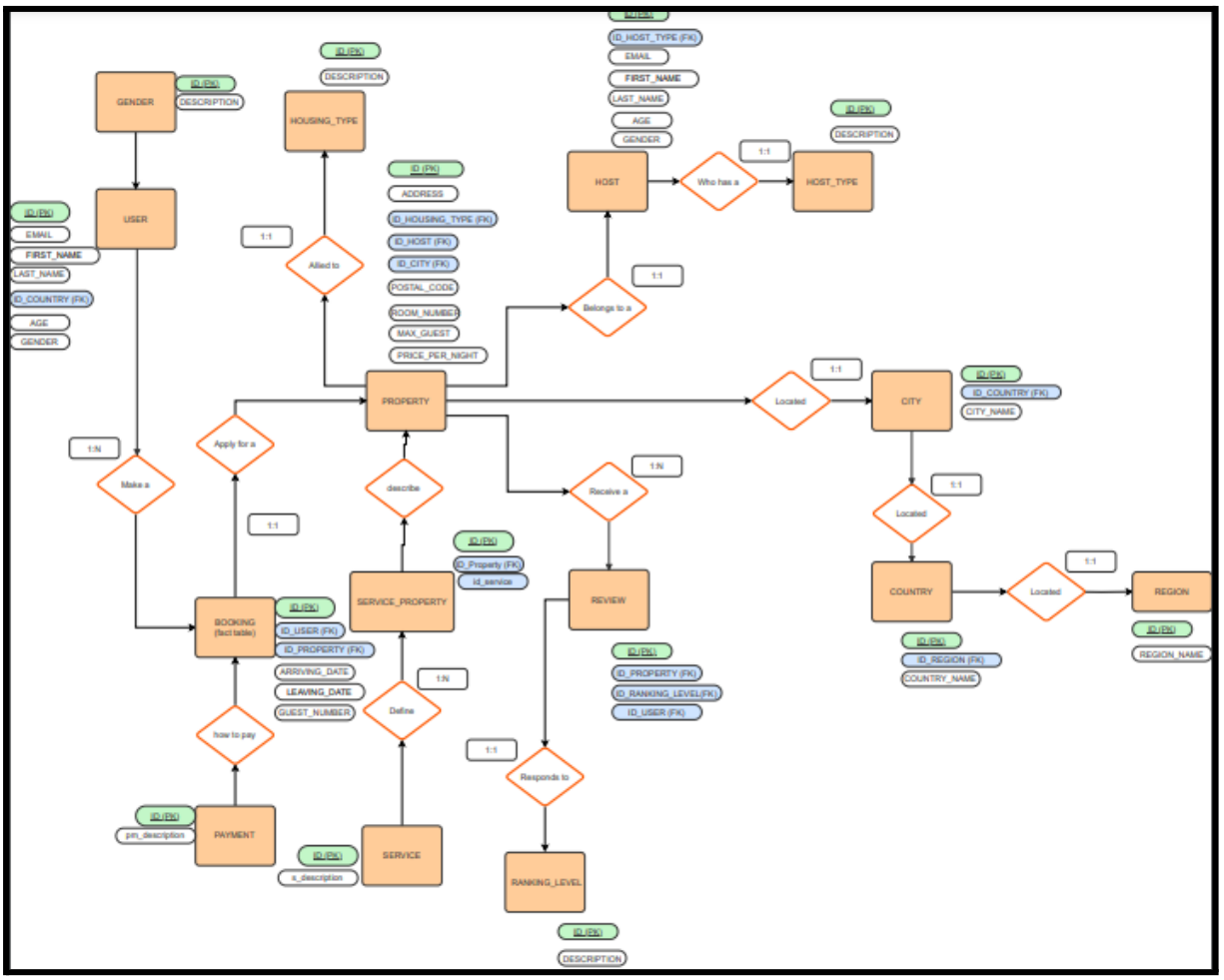
Objectives

The objective of the booking platform is to provide a convenient and secure platform for travelers to discover and book accommodations offered by property owners. It aims to streamline the booking process, promoting trust and transparency.

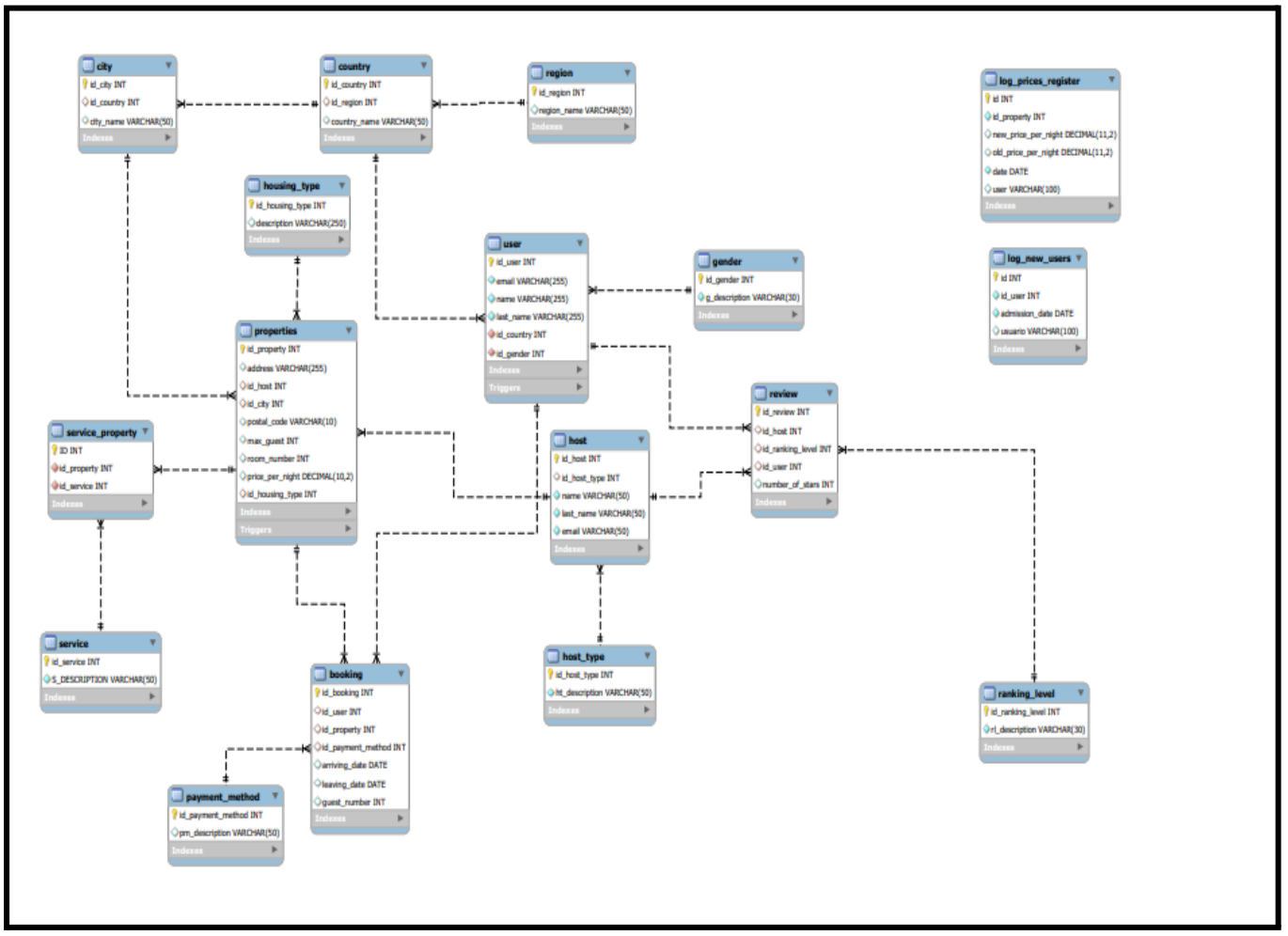
Need to be covered

The need to be covered in the platform is the management of properties, users, and reservations. Making a didactic interface to connect clients with properties according to their needs on a given date and value.

DER Diagrams



ERD Schema



Tables Description

Tabla	USER

Descripción	Description of the platform users						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_USER	INT		TRUE	TRUE	AUTO_INCREMENT	User ID in the platform
CAND	EMAIL	VARCHAR	50	TRUE			email
	NAME	VARCHAR	50	TRUE			name of the user
	LAST_NAME	VARCHAR	50	TRUE			last name of the user
FK	ID_COUNTRY	INT		TRUE			Id of the country

Tabla	HOST						
Descripcion	Description of the person who owns the apartment or house						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_HOST	host		TRUE	TRUE	AUTO_INCREMENT	host id in the platform
FK	ID_HOST_TYPE	INT		TRUE			User ID in the platform
	NAME	VARCHAR	50	TRUE			Host name
	LAST_NAME	VARCHAR	50	TRUE			Host last name
	EMAIL	VARCHAR	50	TRUE			host email

Tabla	HOST_TYPE						
Descripcion	Description of the person who owns the apartment or house						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_HOST_TYPE	INT		TRUE	TRUE	AUTO_INCREMENT	host id type
	DESCRIPTION	VARCHAR	50	TRUE	TRUE		ranking in opinion of the host

Tabla	PROPERTIES
Descripcion	Properties description inside the platform

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_PROPERTY	INT		TRUE	TRUE	AUTO_INCREMENT	id of the property
	ADDRESS	VARCHAR	50	TRUE			property address
FK	ID_HOST	INT		TRUE			host id inside the platform
FK	ID_CITY	INT		TRUE			id of the city in the platform
MUL	POSTAL_CODE	VARCHAR	50	TRUE			postal code registered of the property
	SURFACE	INT		TRUE			surface of the propery in m2
	ROOM_NUMBER	INT		TRUE			number of rooms of the property
	PROPERTY_TYPE	VARCHAR		TRUE			kind of propery (house/apartment)
	PRICE_PER_NIGHT	DEC	11,2	TRUE			price per day of the property
	MAX_GUEST	INT		TRUE			max guest capacity of the propery
FK	ID_HOUSING_TYPE	INT		TRUE			id house tYpe

Tabla	CITY						
Descripcion	City characteristics of the property						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_CITY	INT		TRUE	TRUE	AUTO_INCREMENT	city id in the platfrom
FK	ID_COUNTRY	INT		TRUE			id country in the platform
	CITY_NAME	VARCHAR	50	TRUE			name of the city

Tabla	COUNTRY						
Descripción	Country where the property is located						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_COUNTRY	INT		TRUE	TRUE	AUTO_INCREMENT	country id in the platform
FK	ID_REGION	INT		TRUE			region id in the platform
FK	ID_CITY	INT		TRUE			city id in the platform

	COUNTRY_NAME	VARCHAR	50	TRUE			country name

Tabla	REGION						
Descripción	Region in the world						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_REGION	INT		TRUE	TRUE	AUTO_INCREMENT	id region
	REGION_NAME	VARCHAR	50	TRUE			name of the region

Tabla	BOOKING						
Descripción	fact table about bookings in some specific property						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_BOOKING	INT		TRUE	TRUE	AUTO_INCREMENT	id of the booking
FK	ID_USER	INT		TRUE			Id user on the platform
FK	ID_PROPERTY	INT		TRUE			id property on the platform
	ARRIVING_DATE	DATE		TRUE			day when the user arrive
	LEAVING_DATE	DATE		TRUE			day when the user leave the property
	GUEST_NUMBER	INT		TRUE			number of booking of that propery

Tabla	DATE						
Descripción	when it took place the booking						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK(CONCAT)	ID_DATE	INT		TRUE	TRUE		concatenacion (USER+ARRIVING DATE)
FK	ID_BOOKING	INT		TRUE	TRUE		
FK	ID_USER	INT		TRUE	TRUE		
	MONTH	DATE		TRUE	TRUE		
	YEAR	DATE		TRUE	TRUE		
	MONTH_NAME						

Tabla	REVIEW						
Descripción	Indicates satisfaction level of the user						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_REVIEW	INT		TRUE	TRUE	AUTO_INCREMENT	
FK	ID_HOST	INT		TRUE			
FK	ID_RANKING_LEVEL	INT		TRUE			
FK	ID_USER	INT		TRUE			
	NUMBER_OF_STARS	INT		TRUE			how satisfied is the user

Tabla	RANKING_LEVEL						
Descripción	Description table about the different kind of puntuation a user can give						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_RANKING_LEVEL	INT		TRUE	TRUE	AUTO_INCREMENT	
	DESCRIPTION	VARCHAR	50	TRUE			Opinion ranking of the specific booking

Tabla	HOUSING_TYPE						
Descripción	Ranking of the host						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES
PK	ID_housing_type	INT		TRUE	TRUE	AUTO_INCREMENT	
	DESCRIPTION	VARCHAR	50	TRUE			

Tabla	GENDER						
Descripción	Definition feminin or masculin						
KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	NOTES

PK	id_gender	INT		TRUE	TRUE	AUTO_INCREMENT	
	G_DESCRIPTION	VARCHAR	50	TRUE			

Tabla	**payment_method**						
Descripción	HOw the user pay for the booking						
KEY	**COLUMN**	**TYPE**	**LENGHT**	**NOT NULL**	**UNIQUE**	**DEFAULT**	**NOTES**
PK	id_payment_method	INT		TRUE	TRUE	AUTO_INCREMENT	
	pm_description	VARCHAR	50	TRUE			in cash, bank transfer, credit card

Tabla	SERVICES						
Desc	Services offered by the property						
KEY	**COLUMN**	**TYPE**	**LENGHT**	**NOT NULL**	**UNIQUE**	**DEFAULT**	**NOTES**
PK	ID	INT		NOT NULL	UNIQUE	AUTO_INCREMENT	
CAND	S_DESCRIPTION	VARCHAR	50	NOT NULL			What is the infrastructure of the property

Tabla	SERVICE_PROPERTY						
Desc	Relation between property and service						
KEY	**COLUMN**	**TYPE**	**LENGHT**	**NOT NULL**	**UNIQUE**	**DEFAULT**	**NOTES**
PK	ID	INT		NOT NULL	UNIQUE	AUTO_INCREMENT	
FK	ID_PROPERTY	INT		NOT NULL			
FK	ID_SERVICE	INT		NOT NULL			

Tabla	log_new_users						
Desc	New users registration table						
KEY	**COLUMN**	**TYPE**	**LENGHT**	**NOT NULL**	**UNIQUE**	**DEFAULT**	**NOTES**
PK	ID	INT		NOT NULL	UNIQUE	AUTO_INCR	

						EMENT	
	id_user	INT		NOT NULL			
	admission_date	DATE		NOT NULL			
	usuario	VARCHAR	100	NOT NULL			

Tabla	log_prices_register						
Desc	table that records price updates in properties						
KEY	**COLUMN**	**TYPE**	**LENGHT**	**NOT NULL**	**UNIQUE**	**DEFAULT**	**NOTES**
PK	id	INT		NOT NULL	UNIQUE	AUTO_INCREMENT	
	id_property	INT		NOT NULL			
	new_price_per_night	DECIMAL	(11,2)	NOT NULL			
	old_price_per_night	DECIMAL	(11,2)	NOT NULL			
	date	DATE	(11,2)	NOT NULL			
	user	VARCHAR	100	NOT NULL			

Project Link

Below is the link to the entire project.

CREATE DATABASE BOOKING_PROJECT;

https://drive.google.com/file/d/1n3hzhP1hK21kHVEth7LEf5rG53ho-ERl/view?usp=drive_link

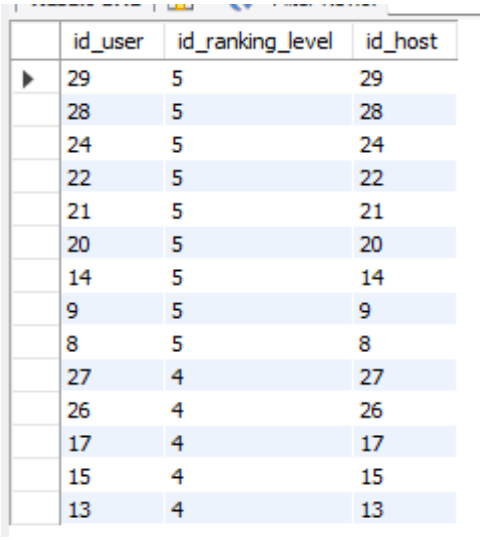
Views

Below a list for the views with their respective comments.

1) VW_BEST_RATED: View for the best host without names.

```
CREATE OR REPLACE VIEW vw_best_rated AS
      SELECT id_user,id_ranking_level,id_host
      FROM review
      WHERE id_ranking_level >= 4
      ORDER BY id_ranking_level DESC;
```

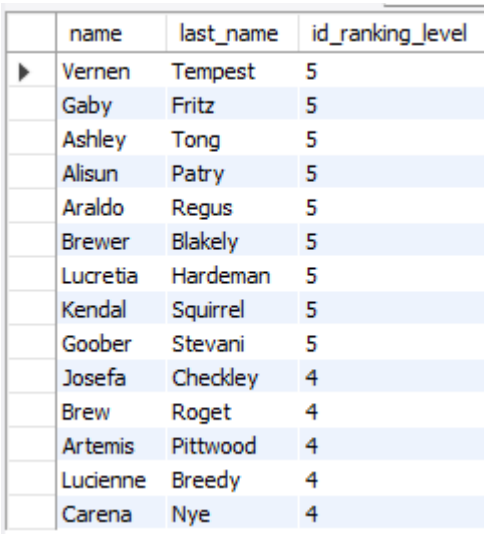
```
SELECT * FROM vw_best_rated;
```



id_user	id_ranking_level	id_host
29	5	29
28	5	28
24	5	24
22	5	22
21	5	21
20	5	20
14	5	14
9	5	9
8	5	8
27	4	27
26	4	26
17	4	17
15	4	15
13	4	13

2) VW_BEST_RATED_HOST: Best ranking host view

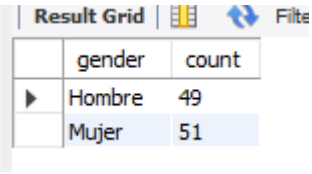
```
CREATE OR REPLACE VIEW vw_best_rated_host AS
SELECT h.name, h.last_name, v.id_ranking_level
FROM host h
JOIN vw_best_rated v ON  v.id_host= h.id_host
ORDER BY id_ranking_level DESC;
SELECT * FROM vw_best_rated_host;
```



name	last_name	id_ranking_level
Vernen	Tempest	5
Gaby	Fritz	5
Ashley	Tong	5
Alisun	Patry	5
Araldo	Regus	5
Brewer	Blakely	5
Lucretia	Hardeman	5
Kendal	Squirrel	5
Goober	Stevani	5
Josefa	Checkley	4
Brew	Roget	4
Artemis	Pittwood	4
Lucienne	Breedy	4
Carena	Nye	4

3) VW_GENDERS_COUNT: This view show the quantity of mens and womens in the database

```
CREATE OR REPLACE VIEW vw_genders_count AS
      SELECT
    CASE
        WHEN id_gender = 1 THEN 'Hombre'
        WHEN id_gender = 2 THEN 'Mujer'
        ELSE 'Otro'
    END AS gender,
    COUNT(*) AS count
FROM user
GROUP BY id_gender;

SELECT * FROM vw_genders_count;
```



Result Grid | Filte

	gender	count
▶	Hombre	49
	Mujer	51

4) VW_TOTAL_DAYS: View for total days and total income per night

```
CREATE OR REPLACE VIEW vw_total_days AS
SELECT                          b.id_user,b.id_property,b.arriving_date,
b.leaving_date,DATEDIFF(b.leaving_date,b.arriving_date)*p.price_per_nig
ht AS total_income, p.price_per_night
FROM booking b
INNER JOIN properties p  ON b.id_property=p.id_property;

SELECT * FROM vw_total_days;
```

	id_user	id_property	arriving_date	leaving_date	total_income	price_per_night
▶	2	2	2022-08-11	2023-05-25	441018.55	1536.65
	3	3	2022-10-25	2023-06-12	419064.60	1822.02
	4	4	2022-12-07	2023-02-16	81965.95	1154.45
	5	5	2022-12-05	2023-02-24	56511.27	697.67
	6	6	2022-09-26	2023-05-11	76269.73	335.99
	7	7	2023-01-02	2023-03-14	175229.42	2468.02

5) VW_TOP_PRICES_PROPERTIES: View that show the top 10 properties by price and the proprietor's name.

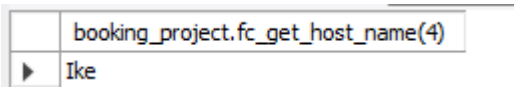
```
CREATE OR REPLACE VIEW vw_top_prices_properties AS
      SELECT p.address, p.price_per_night,p.max_guest,CONCAT(h.name, '
', h.last_name) AS Host_name
    FROM properties p
    INNER JOIN host h ON p.id_host=h.id_host
    ORDER BY price_per_night DESC
    LIMIT 10;
SELECT * from vw_top_prices_properties;
```

address	price_per_night	max_guest	Host_name
36878 Sunfield Pass	2468.02	6	Tessy Hartmann
03 Sachs Road	1822.02	3	Webster Tampin
017 South Place	1536.65	2	Alejandro Bedle
6 Del Sol Parkway	1447.46	5	Goober Stevani
81980 Mcguire Place	1154.45	9	Melanie Watters
42274 Red Cloud Center	1147.52	7	Maxwell Haskett
5 Pierstorff Lane	1098.43	6	Arin Kynder
2566 Estrada	1000.00	2	Ike Staining
3 Emmet Lane	697.67	9	Dosi Softley
611 Dayton Way	650.10	7	Ike Staining

FUNCTIONS

1) Host_Name: Function thats brings the name of the host related to the ID in the database.

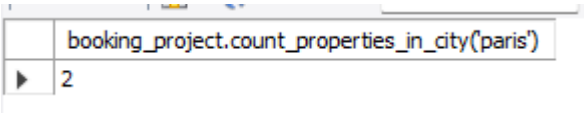
```
DELIMITER $$

CREATE FUNCTION fc_get_host_name (id_host INT)

RETURNS varchar(100)

Reads SQL data

BEGIN

      DECLARE host_name varchar(100);

     SELECT `name` INTO host_name FROM host WHERE id_host= id_host LIMIT
1;

    RETURN host_name;

END$$

select booking_project.fc_get_host_name(4);
```



booking_project.fc_get_host_name(4)
Ike

2) COUNT_PROPERTIES_PER_CITY: Function with a JOIN statement between city and properties that brings how many bookings has an specific city.

```
CREATE FUNCTION count_properties_in_city(city_name VARCHAR(50))
RETURNS INT
reads sql data
BEGIN
    DECLARE property_count INT;

    SELECT COUNT(*) INTO property_count
    FROM properties p
    JOIN city c ON p.id_city = c.id_city
    WHERE c.city_name = city_name;

    RETURN property_count;
END$$
DELIMITER ;

select booking_project.count_properties_in_city('paris');
```



booking_project.count_properties_in_city('paris')
2

STORED PROCEDURES

1) Stored procedure which sorts the user table according to a sort field and a sort direction(name and last_name columns and whether the sort order is ASC or DESC).

```
USE `booking_project`;
DROP procedure IF EXISTS `sp_get_user_names_order`;

DELIMITER $$
USE `booking_project`$$
CREATE PROCEDURE `sp_get_user_names_order` (IN order_column CHAR
(20),IN order_direction CHAR(4))
```

```
BEGIN
      DECLARE order_query VARCHAR (100);
    IF order_column <> '' THEN
           SET order_query = concat('ORDER BY',' ',order_column,' ',
order_direction);
      ELSE
           SET order_query = '';
    END IF;
           SET @consulta = CONCAT('SELECT * FROM booking_project.user',
' ', order_query);
    PREPARE `query` FROM @consulta;
    EXECUTE `query`;
    DEALLOCATE PREPARE `query`;
END$$

-- Test results
call booking_project.sp_get_user_names_order('name', 'ASC');
call booking_project.sp_get_user_names_order('last_name', 'DESC');
```

	id_user	email	name	last_name	id_country	id_gender
▶	71	ataudevin1z@topsy.com	Adela	Taudevin	4	2
	53	amatthaus1h@goodreads.com	Adelbert	Matthaus	7	1
	38	aoldroyde12@mozilla.com	Alaric	Oldroyde	19	1
	4	adyott4@mayoclinic.com	Alix	Dyott	20	1
	83	atoulch2b@acquirethisname.com	Alphard	Toulch	12	1
	13	adyd@list-manage.com	Amos	Dy	20	1
	57	adyter1l@time.com	Ardelle	Dyter	9	2
	39	ahellwich13@histats.com	Astrid	Hellwich	13	2
	67	bchominski1v@dmoz.org	Barbra	Chominski	14	2

2) HOST_DATA_INSERT: Stored procedure that insert data in the table HOST

```
DELIMITER $$
CREATE PROCEDURE `sp_data_insert_host` (IN
 id_host_type int,
 `name` VARCHAR (50),
 last_name VARCHAR (50),
 email VARCHAR (50) )
BEGIN
INSERT INTO `host` (id_host_type, `name`, last_name,email)
VALUES (id_host_type, `name`,last_name,email);
END$$

DELIMITER ;
```

```
-- Example to a data insertion
CALL booking_project.sp_data_insert_host(1, 'Charly', 'Garcia',
'charlygarcia@gmail.com');
SELECT * FROM host ORDER BY id_host DESC;
```

id_host	id_host_type	name	last_name	email
49	1	Charly	Garcia	charlygarcia@gmail.com
48	2	francisco	gomez	fran.gomez@gmail
47	3	Chrysa	Henrion	chenrion1a@paypal.com
46	3	Rakel	Adamsen	radamsen19@behance.net
45	1	Zorine	Giacomozzo	zgiacomozzo18@soundcloud.com
44	2	Gunter	Wombwell	gwombwell17@about.me
43	2	Jude	Klloran	killoran16@goodreads.com
42	2	Cos	Arnason	carnason15@skyrock.com
41	2	Flory	Aggett	faggett14@cocolog-nifty.com

TRIGGERS

For my trigger section i create two Log tables to register my new users and another to Log my prices updates and have a record to previous prices.

```
CREATE TABLE IF NOT EXISTS booking_project.log_new_users (
      id INT AUTO_INCREMENT,
    id_user INT NOT NULL,
    admission_date DATE NOT NULL,
    usuario VARCHAR(100),
    PRIMARY KEY (id)
);


CREATE TABLE IF NOT EXISTS log_prices_register(
      id INT AUTO_INCREMENT,
    id_property INT NOT NULL,
      new_price_per_night DECIMAL (11,2),
    old_price_per_night DECIMAL (11,2),
    `date` DATE NOT NULL,
    `user` VARCHAR(100),
    PRIMARY KEY (id)
```

1) NEW_USER: Trigger to register every new client in my database

```
CREATE TRIGGER `tr_new_user`
AFTER INSERT ON `user`
FOR EACH ROW
      INSERT INTO `log_new_users` (id_user, admission_date, usuario)
      VALUES (NEW.id_user, CURDATE(), CURRENT_USER());
```

```
-- Insert data in my table User and confirm the record in my Log table.
INSERT INTO `booking_project`.`user` (`id_user`, `email`, `name`,
`last_name`, `id_country`, `id_gender`) VALUES
 ('102', 'carlitos@gmail.com', 'Carlos', 'Sanches', '6', '1');
SELECT * FROM log_new_users;
```

id	id_user	admission_date	usuario
1	102	2023-07-16	root@localhost

2) PROPERTY_PRICE_UPDATE: Trigger to register prices updates and get record of all old prices in the database.

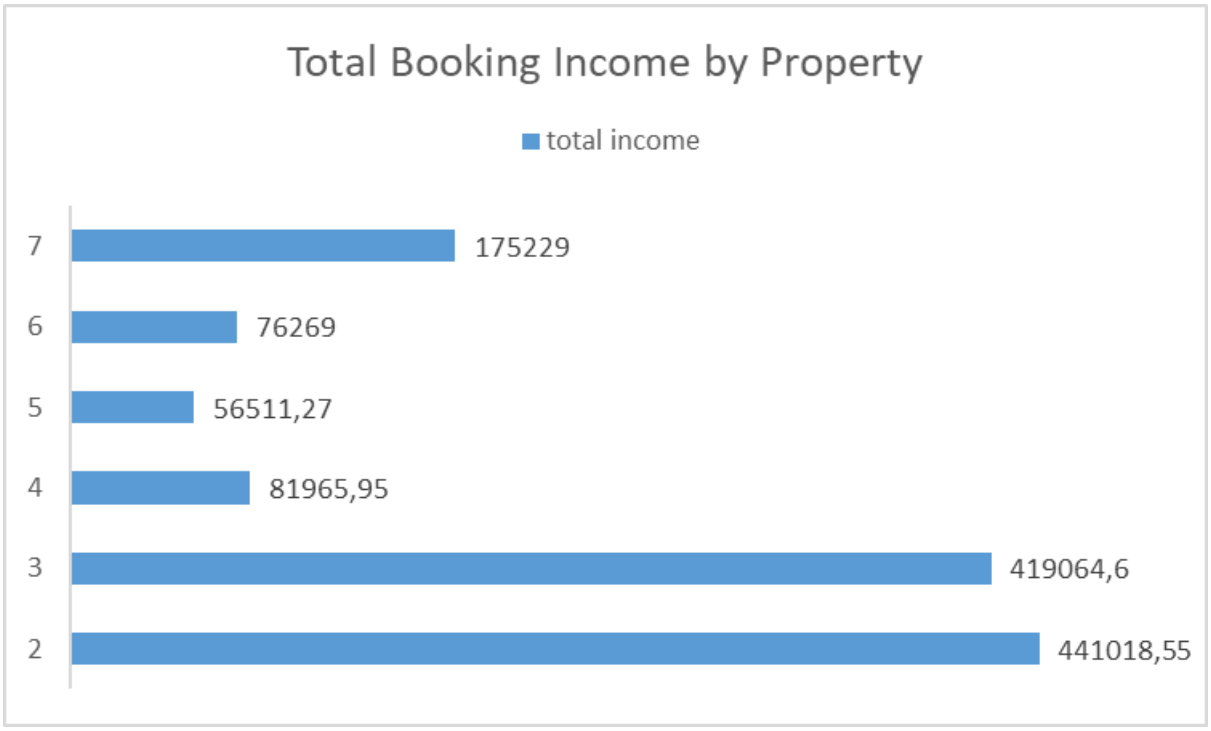
```
DELIMITER $$
CREATE TRIGGER `tr_properties_updated_prices`
AFTER UPDATE ON `properties`
FOR EACH ROW BEGIN
 IF NEW.price_per_night <> OLD.price_per_night THEN
      INSERT INTO log_prices_register (id_property,
new_price_per_night,old_price_per_night,`date`,`user`)
      VALUES (NEW.id_property, NEW.price_per_night,
OLD.price_per_night, CURDATE(), CURRENT_USER());
END IF;
END$$
DELIMITER ;

-- Confirm my results
UPDATE `booking_project`.`properties` SET `price_per_night` = '650.10'
WHERE (`id_property` = '1');
SELECT * FROM log_prices_register;
```

id	id_property	new_price_per_night	old_price_per_night	date	user
1	1	650.10	640.38	2023-07-16	root@localhost
NULL	NULL	NULL	NULL	NULL	NULL

Business Intelligence

Below are graphical reports obtained from the views vw_top_prices_properties and vw_total_days.


Top Owners properties price
3000
2500
2000
1500
1000
500
0
Tessy Hartmann
Webster Tampin
Alejandro Bedle
Goober Stevani
Melanie Watters
Maxwell Haskett
Arin Kynder
Ike Staining
Dosi Softley
Ike Staining


Total Booking Income by Property
total income
7
175229
6
76269
5
56511,27
4
81965,95
3
419064,6
2
441018,55

Technology Inside the Project

System	Product	Version
Database	MySQL Community Server - GPL	8.0.31 for Win64 on x86_64
BI	MS Excel	2016 MSO 32bits