PRACTICAL ASSIGNMENT 1 PHASE 2 - LS EIRBIANBI

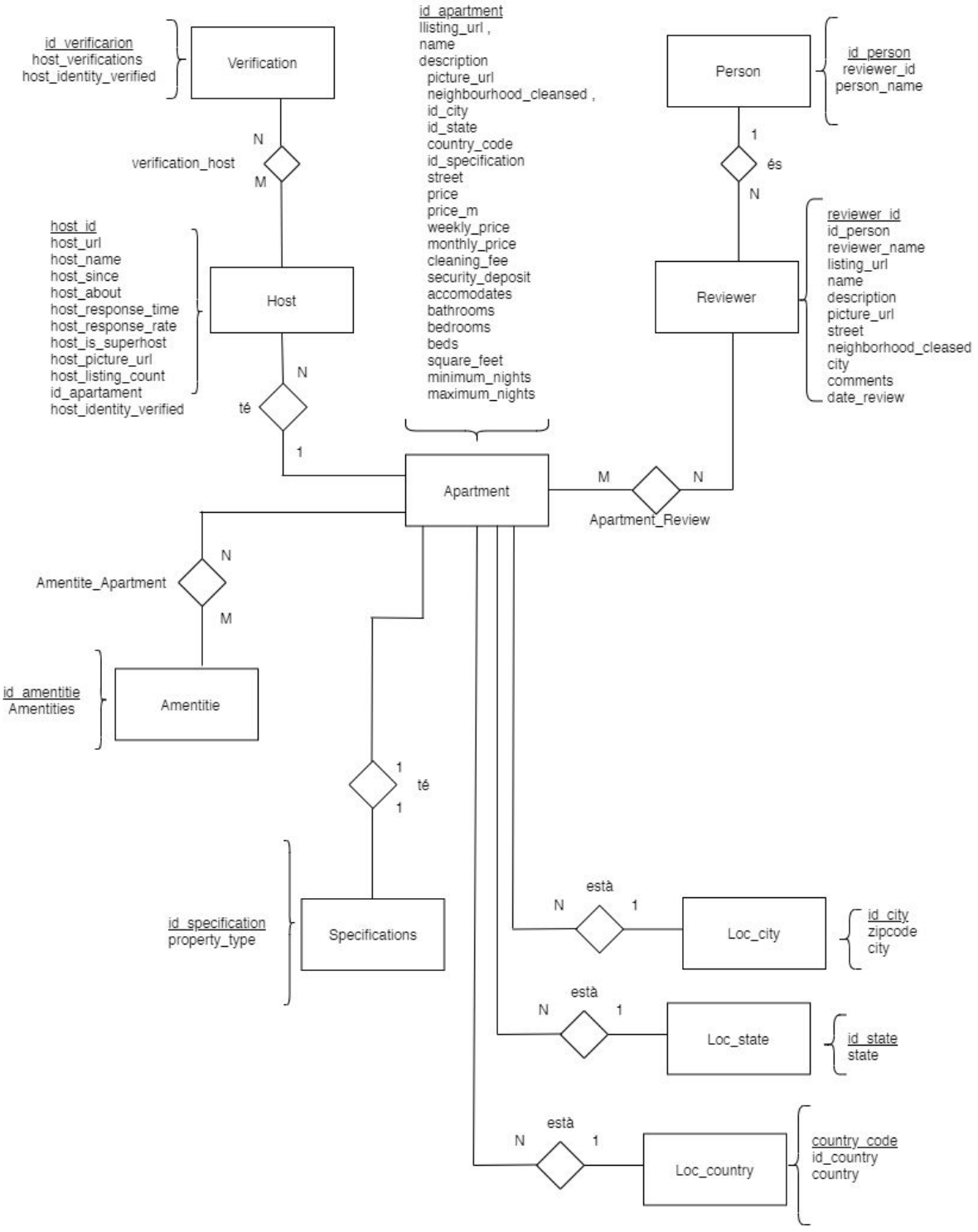
17 de juny del 2020

Database Systems

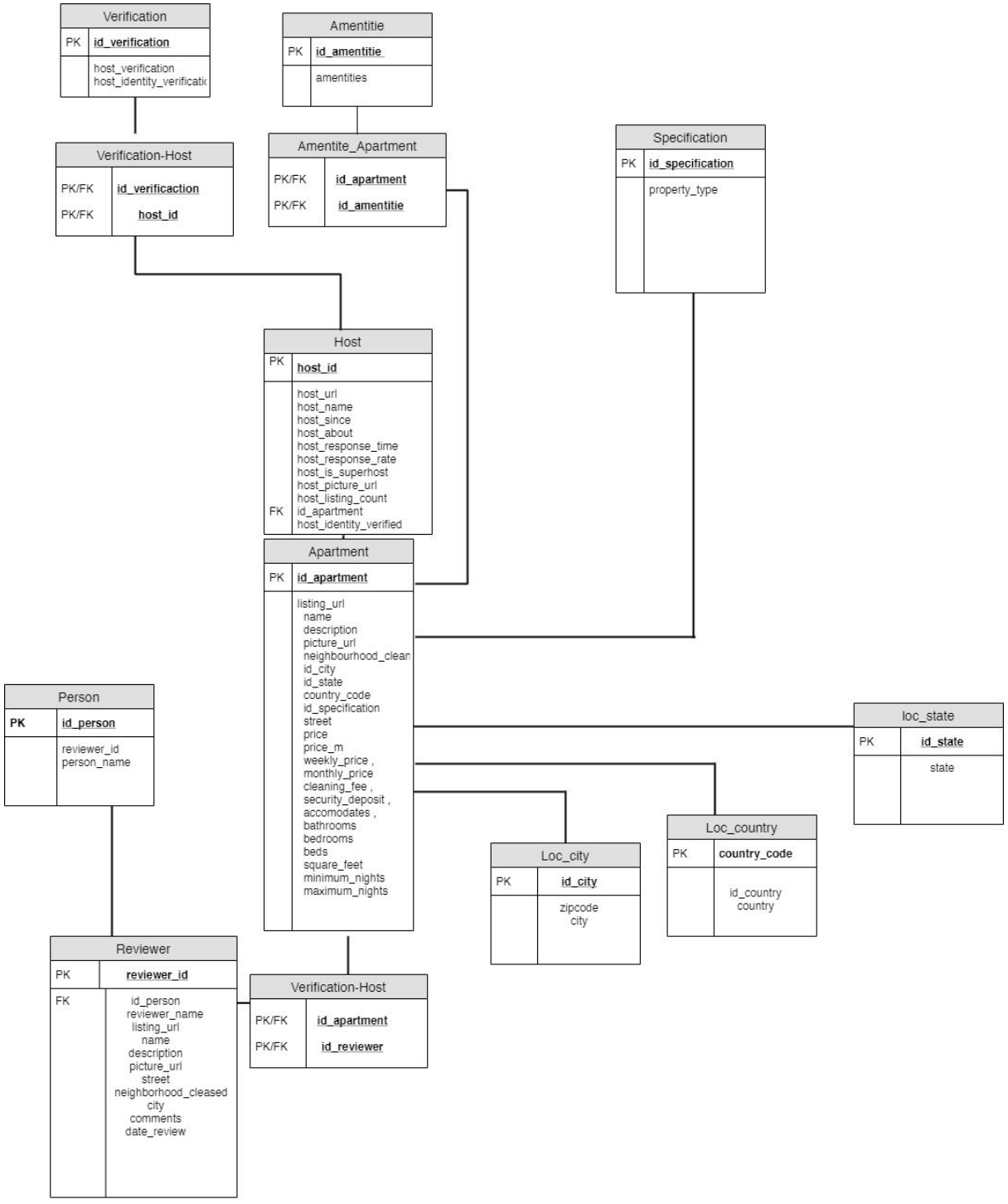
Alejandor Dapena I Silvia Miralles
alejandro.dapena silvia.miralles

Table of contents:

Conceptual Model



Relational Model



Physical Model

Taules Importació:

imp_apartament(id PK, listing_url,name, description, picture_url, street, neighbourhood_cleansed, city, state, zipcode, country_code, country, property_type, accomodates, bathrooms, bedrooms, beds, amentities, square_feet, price, weekly_price, monthly_price, security_deposit, cleaning_fee ,minimum_nights, maximum_nights)

imp_host(listing_url PK, name, description, picture_url, host_id, host_url, host_name,host_since,host_about, host_response_time, host_response_rate, host_is_superhost, host_picture_url, host_listing_cout, host_verifications, host_identity_verified)

imp_review(id, listing_url, name, description, picture_url, street, neighborhood_cleased, city, date_review, reviewer_id, reviewer_name, comments)

Taules:

Loc_country(id_country, country_code PK, country)

Loc_state(id_state PK , state)

Loc_city (id_city PK, zipcode, city)

Specification(id_specification PK, property_type)

Apartment (id_apartment PK, listing_url, name, description, picture_url, neighbourhood_cleansed, id_city FK,id_state FK, country_code FK, id_specification, street, price, price_m, weekly_price, monthly_price, cleaning_fee, security_deposit, accomodates, bathrooms, bedrooms, beds, amentities, square_feet, minimum_nights, maximum_nights)

Host(host_id PK, host_url, host_name, host_since, host_about, host_response_time, host_response_rate, host_is_superhost, host_picture_url, host_listing_cout, host_identity_verified, id_apartment FK)

Verification(id_verification PK, host_verifications, host_identity_verified)

Verification_Host(id_verification PK/FK, host_id PK/FK)

Amentite(id_amentitie PK, Amentities)

Amentite_Apartment(id_amentitie PK/FK, id_apartment PK/FK)

Person(id_person PK, reviewer_id,person_name)

Reviewer(reviewer_id PK, id_person FK, reviewer_name, listing_url, name, description, picture_url, comments, date_review)

Apartment_Reviewer(id_apartment PK/FK,id_reviewer PK/FK)

Importation verification

Inserts de totes les taules:

--Fem l'insert de la localització de country.
INSERT INTO Loc_country (country_code, country)
SELECT DISTINCT country_code, country
FROM imp_apartament;

--Comprovem que s'ompli.
SELECT * FROM Loc_country;

	id_country integer	country_code [PK] character (2)	country character varying (255)
1	1	AU	Australia

--Fem l'insert de la localització de state.
INSERT INTO Loc_state (state)
SELECT DISTINCT state
FROM imp_apartament;

--Comprovem que s'ompli.
SELECT * FROM Loc_state

	id_state [PK] integer	state character varying (255)
1	1	维多利亚州
2	2	VI
3	3	維多利亞 VIC
4	4	victoria
5	5	Melbourne
6	6	维多利亚
7	7	[null]
8	8	VICTORIA
9	9	Victory
10	10	Wheelers Hill VIC 3150
11	11	Doncaster VIC 3108
12	12	Vic

--Fem l'insert de la localització de city.
INSERT INTO Loc_city (city, zipcode)
SELECT DISTINCT city, zipcode
FROM imp_apartament;

--Comprovem que s'ompli.
SELECT * FROM Loc_city;

	id_city [PK] integer	zipcode character varying (255)	city character varying (255)
1	1	3753	Beveridge
2	2	3978	Clyde North
3	3	3206	St Kilda West
4	4	3070	Northcote
5	5	3175	Dandenong

--Fem l'insert de specification on hi ha els tipus d'apartament que pot haver-hi.
INSERT INTO Specification (property_type)
SELECT DISTINCT property_type
FROM imp_apartament;

--Comprovem que s'ompli.
SELECT * FROM Specification;

	id_specification [PK] integer	property_type character varying (255)
1	1	Apartment
2	2	Aparthotel
3	3	Boutique hotel
4	4	Train
5	5	Barn

/*Fem l'insert d'apartment, com tenim una relación 1:N
amb les taules Loc_city, Loc_state i Loc_country passem les pk's com fk´s
i en el cas de specification també posem con a fk la seva pk perquè tenim una relació 1:1
*/
INSERT INTO Apartment(listing_url, name, description, picture_url,
neighbourhood_cleansed, id_city, id_state, country_code, id_specification, street,price,
price_m, weekly_price, monthly_price, cleaning_fee, security_deposit, accomodates,
bathrooms, bedrooms, beds, amentities, square_feet, minimum_nights, maximum_nights)
SELECT ia.listing_url, ia.name, ia.description, ia.picture_url, ia.neighbourhood_cleansed,
lci.id_city, ls.id_state, lc.country_code, s.id_specification,ia.street, ia.price, ia.price::money,
ia.weekly_price, ia.monthly_price, ia.cleaning_fee, ia.security_deposit, ia.accomodates,
ia.bathrooms, ia.bedrooms, ia.beds, ia.amentities, ia.square_feet, ia.minimum_nights,
ia.maximum_nights
FROM imp_apartament AS ia, Loc_city AS lci, Loc_state AS ls, Loc_country AS lc,
Specification AS s
WHERE (lci.city = ia.city AND lci.zipcode = ia.zipcode) AND (ls.state = ia.state) AND
(lc.country_code = ia.country_code) AND (s.property_type LIKE ia.property_type);

--Comprovem que s'ompli.
SELECT * FROM Apartment;

	id_apartment [PK] integer	listing_url character varying (255)	name character varying (255)	description text	picture_url character varying (255)	neighbourhood_cleansed character varying (255)
1	1	https://www.airbnb.com/ro...	Heart		https://a0.muscache.com/i...	Darebin
2	2	https://www.airbnb.com/ro...	Donvale Townhouse for Ren...	A brand ne...	https://a0.muscache.com/i...	Whitehorse
3	3	https://www.airbnb.com/ro...	Private spacious loft bedro...		https://a0.muscache.com/i...	Port Phillip
4	4	https://www.airbnb.com/ro...	big room in clayton near mo...	minimum 1 y...	https://a0.muscache.com/i...	Monash
5	5	https://www.airbnb.com/ro...	Marina Cottage	Beach at your...	https://a0.muscache.com/i...	Port Phillip

```
/*Fem l'insert de host, com tenim una relación 1:N
amb la taula Apartment posem la pk com fk i utilitzem
productes cartesianes per vincularlos.
*/
INSERT INTO Host( host_url, host_name, host_since,
host_about, host_response_time, host_response_rate, host_is_superhost,
host_picture_url,host_listing_cout, host_identity_verified, id_apartment)
SELECT DISTINCT ih.host_url, ih.host_name, ih.host_since,
ih.host_about, ih.host_response_time, ih.host_response_rate, ih.host_is_superhost,
ih.host_picture_url, ih.host_listing_cout, ih.host_identity_verified, a.id_apartment
FROM imp_host AS ih, Apartment AS a, imp_apartament AS ia
WHERE a.listing_url = ia.listing_url AND  ia.listing_url = ih.listing_url;

--Comprovem que s'ompli.
SELECT * FROM Host;
```

	host_id [PK] integer	host_url character varying (255)	host_name character varying (255)	host_since date	host_about text	host_response_time character varying (255)
1	5826	https://www.airbnb.com/us...	Geraldine, Richard And Fam...	2017-12-19		-1
2	12626	https://www.airbnb.com/us...	Eddie	2015-04-20		-1
3	12628	https://www.airbnb.com/us...	Carla	2015-04-20		-1
4	12629	https://www.airbnb.com/us...	Carla	2015-04-20		-1
5	12630	https://www.airbnb.com/us...	Ken	2015-04-20		within an hour

```
/*Insertem el host_verification dues vegades
per poder tenir l'array net i el buit relacionats,
ademes afegim el boolea.
*/
INSERT INTO Verification_table ( host_verifications, host_verifications_x,
host_identity_verified)
SELECT DISTINCT regexp_split_to_table(host_verifications, ','), host_verifications,
host_identity_verified
FROM imp_host;

/*Del array que volem netejar anem treien
tots els signes innecessaris com son els [] i les cometes.
*/
UPDATE Verification_table
SET host_verifications = TRIM('[]'  FROM  host_verifications);

UPDATE Verification_table
SET host_verifications = REPLACE(host_verifications, '"', '');
```

--Comprovem que es guarda l'array net i el brut.
SELECT * FROM Verification_table;

	id_veri_brut integer	host_verifications text	host_verifications_x text	host_identity_verified boolean
1	1469	phone	['email', 'phone', 'google']	false
2	1617	phone	['email', 'phone', 'reviews']	false
3	2431	email	['email', 'phone', 'facebo…	true
4	2662	email	['email', 'phone', 'reviews']	false
5	576	government_id	['phone', 'jumio', 'govern…	false

--Insertem només l'array net.
INSERT INTO Verification_table_1 (host_verifications)
SELECT DISTINCT vt.host_verifications
FROM Verification_table AS vt ;

--Comprovem que es guarda l'array net.
SELECT * FROM Verification_table_1;

	id_veri_net integer	host_verifications text
1	1	email
2	2	jumio
3	3	sent_id
4	4	government_id
5	5	identity_manual

/*Fem l'insert de la verification utilitzant les dos tables anteriors
per passar l'array net i utilitzem els productes cartesians per vincular
host, imp_host, la verificacion_table (ambb l'array net i el brut)
i la taula verification_table_1 amb l'array net.
*/
INSERT INTO Verification (host_url,host_verifications ,host_identity_verified)
SELECT DISTINCT ih.host_url,v1.host_verifications, ih.host_identity_verified
FROM imp_host AS ih, Verification_table AS vt, Host AS h, Verification_table_1 AS v1
WHERE (ih.host_url = h.host_url) AND (vt.host_verifications_x = ih.host_verifications)
AND (vt.host_verifications = v1.host_verifications)AND (h.host_identity_verified =
vt.host_identity_verified);

--Comprovem que s'omple la tuala verification.
SELECT DISTINCT * FROM Verification;

	id_verification [PK] integer	host_url character varying (255)	host_verifications text	host_identity_verified boolean
1	1	https://www.airbnb.com/us…	government_id	true
2	2	https://www.airbnb.com/us…	jumio	true
3	3	https://www.airbnb.com/us…	phone	true
4	4	https://www.airbnb.com/us…	reviews	true
5	5	https://www.airbnb.com/us…	work_email	true

```
--Insertem el id de la taula verification i el id de la taula host fent productes cartesians.
INSERT INTO Verification_Host (id_verification,host_id)
SELECT  DISTINCT v.id_verification,h.host_id
FROM Verification AS v, imp_host AS ih, Host AS h
WHERE v.host_url = ih.host_url AND ih.host_picture_url= h.host_picture_url;

--Comprovem que s'omple la tuala verification_host.
SELECT * FROM Verification_Host;
```

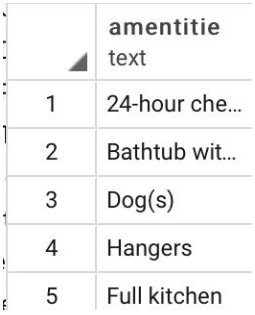
	id_verification [PK] integer	host_id [PK] integer
1	1225	341
2	37927	12413
3	68685	21188
4	48995	15479
5	59729	18669

```
/*Insertem amentities dues vegades
per poder tenir l'array net
*/
INSERT INTO amentiti_table(amentities, amenti_array)
SELECT DISTINCT regexp_split_to_table(amentities, ','), a.amentities
FROM imp_apartament AS a;

/*Del array que volem netejar anem treien
tots els signes innecessaris com son els {} i les cometes.
*/
UPDATE amentiti_table
SET amentities = TRIM('{}'  FROM  amentities);

UPDATE amentiti_table
SET amentities = TRIM('"'  FROM  amentities);

--Comprovem que es guarda l'array net i el brut.
SELECT * FROM amentiti_table;
```

	amentities text	amenti_array text
1	Free street p...	{TV,Wifi,Pool,Kit...
2	Self check-in	{"Family/kid frie...
3	Smoke detec...	{"Air conditionin...
4	Smoke detec...	{TV,"Air conditio...
5	Hangers	{TV,Wifi,Kitchen,...

--Insertem només l'array net.
INSERT INTO amentitie_table_1 (amentitie)
SELECT DISTINCT amentities
FROM amentiti_table;

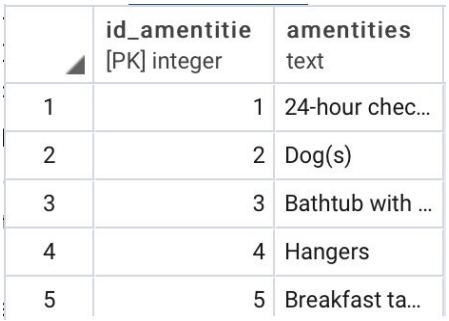
--Comprovem que es guarda l'array net.
SELECT* FROM amentitie_table_1;



	amentitie text
1	24-hour che...
2	Bathtub wit...
3	Dog(s)
4	Hangers
5	Full kitchen

/*Fem l'insert de la amentites utilitzant les dos tables anteriors
per passar l'array net i utilitzem els productes cartesians per vincular.
*/
INSERT INTO Amentite(amentities)
SELECT DISTINCT amentitie
FROM amentitie_table_1 AS am1;

--Comprovem que s'omple la taula amentite.
SELECT * FROM Amentite;



	id_amentitie [PK] integer	amentities text
1	1	24-hour chec...
2	2	Dog(s)
3	3	Bathtub with ...
4	4	Hangers
5	5	Breakfast ta...

/*Fem l'insert dels dos id de les taules amentite i apartment
fent productes cartesians per vincular cada fragment de les amentities amb el seu apartament.
*/
INSERT INTO Amentite_Apartment (id_amentitie,id_apartment)
SELECT DISTINCT am.id_amentitie,ap.id_apartment
FROM Amentite AS am, imp_apartament AS ia, Apartment AS ap
WHERE ('%'|| ap.amentities ||'%') LIKE ('%'|| am.Amentities ||'%') AND ia.listing_url= ap.listing_url;

--Comprovem que s'omple la taula amentite_apartment.
SELECT * FROM Amentite_Apartment;

	id_amentitie [PK] integer	id_apartment [PK] integer
1	80	10178
2	62	20661
3	120	21812
4	57	17710
5	156	15725

/*Fem l'insert de la tabla person per agrupar els reviews per persona*/
INSERT INTO Person (person_name, reviewer_id)
SELECT DISTINCT reviewer_name, reviewer_id
FROM imp_review;

--Comprovem que s'omple la taula person.
SELECT * FROM Person;

	id_person [PK] integer	reviewer_id integer	person_name character varying (255)
1	1	44243405	
2	2	31313319	"Ann
3	3	8250847	"後63
4	4	163478465	'Hainy'
5	5	32094964	'Sam

/*Fem l'insert de la tabla reviewer*/
INSERT INTO Reviewer (id_person, reviewer_name, listing_url, name, description,
picture_url,comments,date_review)
SELECT p.id_person, ir.reviewer_name, ir.listing_url, ir.name, ir.description,
ir.picture_url,ir.comments,ir.date_review
FROM imp_review AS ir, person AS p
WHERE ir.reviewer_id = p.reviewer_id;

--Comprovem que s'omple la taula reviewer.
SELECT * FROM Reviewer
ORDER BY id_person;

	reviewer_id [PK] integer	id_person integer	reviewer_name character varying (255)	listing_url character varying (255)	name character varying (255)	description text
1	369267	1		https://www.airbnb.com/ro...	Rustic Stable House	Unique, rustic...
2	380670	2	"Ann	https://www.airbnb.com/ro...	Heart of CBD Top Floor Vie...	Premium Loc...
3	414771	3	"後63	https://www.airbnb.com/ro...	Modern Apartment on Fam...	Clean & mode...
4	184473	4	'Hainy'	https://www.airbnb.com/ro...	Second gem in the burbs	Second room ...

/*Fem l'insert de la tabla apartment_reviewer per vincula
els reviews a quin apartament correspon*/
INSERT INTO Apartment_Reviewer(id_apartment,id_reviewer)

```
SELECT DISTINCT a.id_apartment ,r.reviewer_id
FROM Apartment AS a, Reviewer AS r, imp_review AS ir
WHERE r.picture_url = ir.picture_url AND a.listing_url = r.listing_url;

--Comprovem que s'omple la taula apartment_reviewer.
SELECT * FROM Apartment_Reviewer;
```

id_apartment [PK] integer	id_reviewer [PK] integer
8	5090
8	74981
8	110513
8	110514

```
--Eliminem les taules que ja no ens calen.
DROP TABLE amentiti_table;
DROP TABLE amentitie_table_1;
DROP TABLE Verification_table;
DROP TABLE Verification_table_1;

--Eliminem les columnes que no son necesaries.
ALTER TABLE Verification DROP COLUMN host_url;
ALTER TABLE Apartment DROP COLUMN amentities;
```

Queries and their verification

Query 1

En aquesta query hem agrupat totes aquelles ciutats i mirem el que ens estalviem al pagar set dies individuals en comptes de pagar la tarifa setmanal. Per fer-ho restem els valors. També ens assegurem de que el host està verificat. Per comprovar-ho mirem la taula de host.

```
--#&&
/* QUERY 1:
 correcte
*/
SELECT lc.city AS name, AVG((100*(7*a.price::float - a.weekly_price::float))/(7*a.price::float)) AS savings_percentage
FROM Apartment AS a, Loc_city AS lc,Host AS h
WHERE a.id_city = lc.id_city AND
h.id_apartment = a.id_apartment AND h.host_identity_verified = 't' AND
a.weekly_price IS NOT NULL AND
a.price IS NOT NULL
GROUP BY lc.city
ORDER BY savings_percentage DESC
LIMIT 3;
```

nel de Salida

Salida de datos | Comentar | Mensajes | Historial

	name character varying(255)	savings_percentage double precision
1	Cranbourne South	79.2207792207792
2	Wattle Glen	64.2857142857143
3	Pakenham	56.043956043956

Verificació

Per realitzar la verifiacio hem fet la mateixa query, però aquest cop buscant directament a la taula imp_apartment.

```
--Comprovacion 1)
SELECT a.city AS name, AVG((100*(7*a.price::float - a.weekly_price::float))/(7*a.price::float)) AS savings_percentage
FROM imp_apartament AS a
WHERE a.city LIKE 'Cranbourne South' AND
a.weekly_price IS NOT NULL AND
a.price IS NOT NULL
GROUP BY a.city;

--Comprovacion 2)
SELECT lc.city AS name, AVG((100*(7*a.price::float - a.weekly_price::float))/(7*a.price::float)) AS savings_percentage
```

nel de Salida

Salida de datos | Comentar | Mensajes | Historial

	name character varying(255)	savings_percentage double precision
1	Cranbourne South	79.2207792207792

Query 2

En aquesta query mirem quin és l'apartment més car en proporció als seus metres quadrats. Per fer-ho dividim el seu preu per els seus metres quadrats. Ordenem els pisos en funció d'aquest preu. També mostrem el nombre de reviews d'aquell apartament.

```
40  /* QUERY 2:
41   Tal com es pot veure a la comprovacio coincideix el nom del apartament i el preu,
42   tan amb les nostres taules com ambb les taules d'importacio.
43  */
44  SELECT a.name, (a.price_m/a.square_feet) AS price_m2, COUNT( ar.id_apartment) AS reviews
45  FROM  Apartment AS a,  Apartment_Reviewer AS ar,Reviewer AS r, Specification AS s
46  WHERE (a.id_apartment = ar.id_apartment) AND (ar.id_reviewer = r.reviewer_id)
47  AND (s.id_specification = a.id_specification) AND (s.property_type ='Guesthouse') AND(a.square_f
48  GROUP BY a.name, price_m2 HAVING COUNT( ar.id_apartment) > 200
49  ORDER BY price_m2
50  LIMIT 1;
51
52  --Comprovacion 1)
53  SELECT a.name, a.price,a.square_feet, (a.price_m/a.square_feet) AS price_m2, COUNT(ar.id_apartm
```

Explain | Data Output | Messages | Query History

	name character varying (255)	price_m2 money	reviews bigint
1	The Stables, Island of Rich...	$0.37	538

Verificació

Per realitzar la verificació demanem que faci els mateixos càlculs i que ens mostri totes apartats que utilitzem a la query, demanant-li específicament que ho faci del apartament que ens ha sortit a la query. Com podem veure tots els aparts coincideixen amb el de la query.

```
51
52  --Comprovacion 1)
53  SELECT a.name, a.price,a.square_feet, (a.price_m/a.square_feet) AS price_m2, COUNT(ar.id_apartm
54  FROM Apartment AS a,Apartment_Reviewer AS ar
55  WHERE a.name LIKE 'The Stables, Island of Richmond' AND a.id_apartment = ar.id_apartment
56  GROUP BY a.id_apartment;
57
58
59
60
```

Explain | Data Output | Messages | Query History

	name character varying (255)	price integer	square_feet integer	price_m2 money	contador bigint
1	The Stables, Island of Rich...	100	269	$0.37	538

Query 3

Dins de la taula Apartment comprovem i mostrem aquells apartaments que compleixin les característiques demanades a l'enunciat. Per comprovar si te balcó, busquem aquell pis en la taula amentitie apartment i ens assegurem.

```
/* QUERY 3:*/
SELECT a.name, a.listing_url, ((price::money *6*5)+a.cleaning_fee::money +(0.1*a.security_deposit::money )) AS price
FROM Apartment AS a, Host AS h, Amentite_Apartment AS aa
WHERE a.neighbourhood_cleansed  LIKE 'Port Phillip' AND
price IS NOT NULL AND
a.bathrooms > 1.5 AND
a.accomodates = 6 AND
a.minimum_nights <=5 AND
h.id_apartment = a.id_apartment AND
h.host_response_rate > 90 AND
aa.id_apartment = a.id_apartment AND
aa.id_amentitie = 164
GROUP BY a.id_apartment
ORDER BY price ASC LIMIT 1;
```

l de Salida

lida de datos | Comentar | Mensajes | Historial

	name character varying(255)	listing_url character varying(255)	price money
1	Spacious Designer Apartment near Albert Park	https://www.airbnb.com/rooms/12528535	10.050,00 €

Verificació

Per realitzar la verificació hem d'utilitzar les taules imp_apartament i imp_host i mostrem tots els caps per assegurar-nos que aquest apartament es el correcte.

```
*/
SELECT a.name, a.listing_url,a.bathrooms,a.accomodates,a.minimum_nights, ((price::money *6*5)+a.cleaning_fee::money +(0.1*a.security_deposit::money )) AS price
FROM imp_apartament AS a, imp_host AS h
WHERE a.neighbourhood_cleansed  LIKE 'Port Phillip' AND
price IS NOT NULL AND
a.bathrooms >= 1.5 AND
a.accomodates = 6 AND
a.minimum_nights <=5 AND
h.listing_url = a.listing_url AND
(h.host_response_rate > 90) AND
a.name LIKE '%Spacious Designer Apartment near Albert Park%'
ORDER BY price ASC LIMIT 1;
```

nel de Salida

Salida de datos | Comentar | Mensajes | Historial

	name character varying(255)	listing_url character varying(255)	bathrooms double precision	accomodates integer	minimum_nights integer	price money
1	Spacious Designer Apartment near Albert Park	https://www.airbnb.com/rooms/12528535	2	6	2	10.050,00 €

Query 4

Superhost: Comprovem aquells usuaris que es van registrar fa més de 5 anys, considerant la nostre data actual com 01/01/2020. Utilitzem el update per associar un true quan la data del host_since coincideix amb el que desitgem. Llavors comptem quants host hi ha que compleix la condició.

```
90  --#&&
91  /* QUERY 4:
92  Com no sabem quina data es va considera com actual per realitzar
93  la query hem decidit fer un cas generic, es a dir considera l'any qu estem com a actual
94  i el dia el 1 de gener perque es l'inici del any. Per tan fa 5 anys seria el 1 de gener del 2015.
95  */
96  UPDATE Host
97  SET host_is_superhost = 't'
98  WHERE (host_since) < '01/01/2015';
99
100 SELECT COUNT( distinct host_id) AS superhosts
101 FROM Host
102 WHERE host_is_superhost = 't'; --MOSTREM EL SUPERHOST
103
```

Explain | Data Output | Messages | Query History

	superhosts bigint
1	7513

Normal_host: Comprovem aquells usuaris que es van registrar fa menys de 5 anys, considerant la nostre data actual com 01/01/2020. Utilitzem el update per associar un false quan la data del host_since coincideix amb el que desitgem. Llavors comptem quants host hi ha que compleix la condició.

```
103
104  UPDATE Host
105  SET host_is_superhost = 'f'
106  WHERE (host_since) >= '01/01/2015';
107
108  SELECT COUNT( distinct host_id) AS normal_hosts
109  FROM Host
110  WHERE host_is_superhost = 'f'; --MOSTREM EL NORMALHOST
111
```

Explain | Data Output | Messages | Query History

	normal_hosts bigint
1	15213

Verificació superhost

Per realitzar la verificació comptem quantes línies té la taula host fa més de 5 anys.

```
113
114  SELECT COUNT(*) FROM Host WHERE (host_since) < '01/01/2015';
115
```

Explain | Data Output | Messages | Query History

	count bigint
1	7513

Verificació normal_host

Per realitzar la verificació comptem quantes línies té la taula host fa menys de 5 anys.

```
117  SELECT COUNT(*) FROM Host WHERE (host_since) >= '01/01/2015';
118
```

Explain | Data Output | Messages | Query History

	count bigint
1	15213

Query 5

Mostrem els carrers que tenen un major nombre d'apartaments i que ademés el seu preu mig no supera 100,i també mostrem el numero de apartaments del carrer i el preu.

```
110  --#&&
111  /* QUERY 5:
112  */
113  SELECT DISTINCT a.street, COUNT(a.street) AS num, AVG(a.price)::money AS price
114  FROM Apartment AS a
115  GROUP BY a.street HAVING  AVG(a.price)<=100
116  ORDER BY num DESC
117  LIMIT 3;
118
```

Explain | Data Output | Messages | Query History

	street text	num bigint	price money
1	Brunsw...	437	$94.80
2	Footscr...	159	$90.69
3	Presto...	140	$94.29

Verificació

Mostrem lo mateix que anteriorment per aquest cop buscant el carrer que ens ha sortir amb major nombre d'apartament i que no superava de mitja el preu de 100. Però tot desde la tabla imp_apartament on no fa falta fer productes cartesians perquè hi ha tota la informació necesaria.

```
118
119  --Verification
120  SELECT street, COUNT(street) AS num, AVG(price)::money AS price
121  FROM imp_apartament
122  WHERE street LIKE '%Brunswick, VIC, Australia%'
123  GROUP BY street;
124
125  --#&&
126  /* QUERY 6:
127  En comparacio a la query del enunciat ens surt diferent perque la Laura te el mat
```

Explain | Data Output | Messages | Query History

	street text	num bigint	price money
1	Brunsw...	437	$94.80

Query 6

Per buscar les persones que estan creant missatges falsos, busquem el id de la persona que ha fet més reviews. Per tan comptem el número de reviews i fem un order by per ordenar-ho de més reviews a menys. Ens surt diferent a la query de l'enunciat perquè la Laurie ha realitzar el mateix nombre de reviews que el Michel, és a dir, 23. I com nomes mostrem les tres persones que han realitzat més review el programa ens mostra la Laurie entre elles.

```
134 --#&&
135 /* QUERY 6:
136 En comparacio a la query del enunciat ens surt diferent perque la Laurie te el matixex
137 numero de reviews que el Michel, i tal com es pot veure en la comprovacio a les taules import
138 també aparteix la Laurie abans que el Michel. Si en lloc de un limit 3 fem un limit 4
139 podem veure que el seguent amb el mateix numero de reviews es el Michel.
140 */
141 SELECT distinct p.person_name, a.listing_url, COUNT(r.reviewer_id) AS num_reviews
142 FROM Apartment_Reviewer AS ar, Reviewer AS r,Apartment as a, Person AS p
143 WHERE r.reviewer_id = ar.id_reviewer AND a.id_apartment = ar.id_apartment
144 AND p.id_person = r.id_person
145 GROUP BY ( p.id_person, p.person_name, a.id_apartment)
146 ORDER BY num_reviews DESC
147 LIMIT 3;
148
```

Explain | Data Output | Messages | Query History

	person_name character varying (255)	listing_url character varying (255)	num_reviews bigint
1	Cameron	https://www.airbnb.com/ro...	81
2	Therese	https://www.airbnb.com/ro...	24
3	Laurie	https://www.airbnb.com/ro...	23

Verificació

Per realitzar la comprovació busquem els mateixos apartants però aquest cop de les taules imp_apartmen i imp_review ho comptavilitzem de la mateixa manera i ens surten els mateixos resultats.

```
149
150 --Comprovacion 1)
151 SELECT distinct r.reviewer_name, a.listing_url, COUNT(r.reviewer_id) AS num_reviews
152 FROM imp_apartament AS a, imp_review AS r
153 WHERE a.listing_url = r.listing_url
154 GROUP BY (r.reviewer_id, r.reviewer_name, a.id)
155 ORDER BY num_reviews DESC
156 LIMIT 3;
157
```

Explain | Data Output | Messages | Query History

	reviewer_name character varying (255)	listing_url character varying (255)	num_reviews bigint
1	Cameron	https://www.airbnb.com/ro...	81
2	Therese	https://www.airbnb.com/ro...	24
3	Laurie	https://www.airbnb.com/ro...	23

Query 7

En aquesta query busquem tots aquells apartaments que compleixin amb les característiques del enunciat.
Per comprovar que el número de llits siguin els correctes mirem la taula Apartment. Per mirar que estigui en la ciutat demanda, realitzem un producte cartesià entre la taula lc_city i apartment. I ens assegurem que totes les altres condicions es compleixin.

```
162
163 SELECT DISTINCT a.id_apartment, a.name, ((a.price*a.accomodates)+a.cleaning_fee+(0.1*a
164 FROM Apartment AS a,Loc_city AS lc, Amentite_Apartment AS aa, Amentite AS am, Host AS h
165 WHERE a.accomodates >=2 AND a.id_city = lc.id_city AND
166 lc.city LIKE 'Saint Kilda' AND aa.id_apartment = a.id_apartment AND aa.id_amentitie = am
167 am.amentities LIKE '%Kitchen%' AND a.beds >= 2 AND a.maximum_nights >= 2 AND a.minimum_
168 h.id_apartment = a.id_apartment AND vh.host_id = h.host_id AND
169 vh.id_verification = v.id_verification AND v.host_verifications LIKE '%phone%'
170 GROUP BY  a.id_apartment HAVING ((a.price*a.accomodates)+a.cleaning_fee+(0.1*security_d
171 ORDER BY price DESC;
172
173
```

Explain | Data Output | Messages | Query History

	id_apartment integer	name character varying (255)	price money
1	12773	★ Beach at your Doorstep: …	$4,166.00
2	4364	Huge, stylish, central, 3 br a…	$3,436.00
3	15905	Premium property in great l…	$2,900.00
4	4519	Spacious Designer Apartme…	$2,250.00
5	5828	Huge St Kilda Beach PentH…	$2,240.00

Verificació

Comprovem que tots els caps compleixen les especificacions de l'enunciat desde les tables importació, per exemple el Huge, stylish, central 3 br apartment with pool te 12 accomodates, city Saint Kailda, amntitie = kitchen, 8 beds, maximum_nights 1125, minimum_nights 2, host verification = phone i el price inferior a 5000.

```
174 /*Verificació: Comprovem que tots els caps compleixen les especificacions de l'enunciat d
175 per exemple el Huge, stylish, central 3 br apartment with pool te 12 accomodate, city Sai
176 amntitie = kitchen, 8 beds, maximum_nights 1125, minimum_nights 2, host verification = ph
177 */
178 SELECT DISTINCT a.id, a.name, a.accomodates, (0.1*a.security_deposit) AS security_depo
179 a.beds, a.maximum_nights, a.minimum_nights, (h.host_verifications LIKE '%phone%') AS hos
180 FROM imp_apartament AS a, imp_host AS h
181 WHERE a.accomodates >=2 AND
182 a.city LIKE 'Saint Kilda' AND
183 a.amentities LIKE '%Kitchen%' AND a.beds >= 2 AND a.maximum_nights >= 2 AND a.minimum_n
184 h.listing_url = a.listing_url
185 AND h.host_verifications LIKE '%phone%'
186 GROUP BY  a.id, h.host_verifications HAVING ((a.price*a.accomodates)+a.cleaning_fee+(0.
187 ORDER BY price DESC;
188
```

Explain | Data Output | Messages | Query History

	id bigint	name character varying (255)	accomodates integer	security_deposit numeric	city character varying (255)
1	30479523	★ Beach at your Doorstep: Enjoy Summer in St Kilda	4	30.0	Saint Kilda
2	11802035	Huge, stylish, central, 3 br apartment with pool.	12	0.0	Saint Kilda
3	15737004	Premium property in great location	6	20.0	Saint Kilda
4	12528535	Spacious Designer Apartment near Albert Park	6	120.0	Saint Kilda
5	16416097	Huge St Kilda Beach PentHouse Beauty	6	40.0	Saint Kilda

Query 8

Per realitzar el càlcul de de la fórmula hem tingut que crear una table en la qual insertem el número de verificacions i el número apartament per després poder fer el sumatori utilitzats aquest valors, ademés com volem el resultat en decimals hem canviat el price a float.

```
212 INSERT INTO Score_Table (host_id, id_apartment, apart_price, is_super_host, num_verifi, num_apar
213 SELECT DISTINCT h.host_id, a.id_apartment, a.price, h.host_is_superhost, COUNT(h.host_id), h.hc
214 FROM Host AS h, Apartment AS a, Verification_Host AS vh, Verification AS v
215 WHERE h.host_id = vh.host_id AND h.id_apartment = a.id_apartment AND vh.id_verification = v.id_v
216 GROUP BY h.host_id, a.id_apartment;
217
218 SELECT * FROM Score_Table ;
219
220 SELECT DISTINCT h.host_name, SUM((1/st.apart_price::float)*(1+(CASE WHEN st.is_super_host = '
221 FROM Host AS h, Score_Table  AS st
222 WHERE h.host_id = st.host_id AND (st.apart_price <> '0')
223 GROUP BY h.host_name
224 ORDER BY score DESC
225 LIMIT 3;
226
```

Explain | Data Output | Messages | Query History

	host_name character varying (255)	score double precision
1	Valeria	1146.37501926569
2	Jared SSP	1026.97467532596
3	Sabrina	504.460779791291

Verificació

Per realitzar la verificació hem tingut que calcula el número de verificacions de la Valeria, després calcula el número de apartament que en els que s'ha hospedat la valeria i finalment fer el càlcul de tot junt per veure si el score ens donava correcte.

Query Editor | Notifications

```
218 --Comprovacio 1) Valeria
219
220 --Numero d'apa  Execute options (accesskey + X)
221 Select COUNT(h.host_id)
222 FROM imp_host AS h, imp_apartament AS a
223 WHERE h.listing_url= a.listing_url AND host_response_time <> '-1'
224 AND h.host_name LIKE 'Valeria';
225
226 -- NUm de verificacions de la Valeria
227 Select distinct h.host_name, v.host_verifications
228 FROM Host AS h, Apartment AS a, Verification AS v, Verification_host AS vh
229 WHERE h.host_id = vh.host_id AND vh.id_verification = v.id_verification AND
230 h.id_apartment = a.id_apartment AND h.host_name LIKE 'Valeria';
231
232 --Comprobacio del calcul:
233 SELECT SUM((1/a.price::float)*(1+(CASE WHEN h.host_is_superhost = 't' then 1 else 0
```

Explain | Data Output | Messages | Query History

	count bigint
1	89

Hem intentat fer tots els càlculs desde les taules importació menys la del número de verificacions, ja que a la taula import les verificacions estan en un array i no les podem comptar.

```
226  -- Num de verificacions de la Valeria
227  Select distinct h.host_name, v.host_verifications
228  FROM Host AS h, Apartment AS a, Verification AS v, Verification_host AS vh
229  WHERE h.host_id = vh.host_id AND vh.id_verification = v.id_verification AND
230  h.id_apartment = a.id_apartment AND h.host_name LIKE 'Valeria';
231
232  --Comprobacio del calcul:
233  SELECT SUM((1/a.price::float)*(1+(CASE WHEN h.host_is_superhost = 't' then 1 else 0
```

Explain | Data Output | Messages | Query History

	host_name character varying (255)	host_verifications text
1	Valeria	facebook
2	Valeria	government_id
3	Valeria	jumio
4	Valeria	offline_government_id
5	Valeria	phone

```
238
239
240  --#&&
241  /* QUERY 9:
```

Explain | Data Output | Messages | Query History

	score double precision
1	1146.05922979201

Query 9

Ens demanen mostrar els deu millors comentaris associant uns punts a cada comentar, per tan si un comentari té més de 100 caràcters rep 15 punts i si en té menys rep 10 punts. Així hem de trobar a la persona que té una puntuació més alta.

Hem tingut que creat dues taules només, una per assosciar el spunts al comentaris fent uns updates que depenent de la dimensió del comentari associant una puntuació de 10 o 15.

La segona table ens a servir per agrupa les puntuacions per persona i finalment hem ordenat les persones segons la seva puntuació.

```
287     id_person INT,
288     puntos_c INT,
289     PRIMARY KEY (reviewer_name, id_person),
290     FOREIGN KEY (id_person) REFERENCES Person (id_person)
291 );
292 INSERT INTO tabla_sum (reviewer_name, id_person, puntos_c)
293 SELECT tf.reviewer_name, p.id_person, SUM(tf.puntos)
294 FROM tabla_final AS tf, Reviewer AS r, Person AS p
295 WHERE tf.reviewer_id = r.reviewer_id AND r.id_person = p.id_person
296 GROUP BY tf.reviewer_name,p.id_person;
297
298 SELECT DISTINCT * FROM tabla_sum
299 ORDER BY puntos_c DESC
300 LIMIT 10;
301
```

Explain | Data Output | Messages | Query History

	reviewer_name [PK] character varying (255)	id_person [PK] integer	puntos_c integer
1	Laurie	209773	1520
2	Cameron	52436	1395
3	Jessica	162525	780
4	Michael	250204	715

Verificació

Per realitzar la verificacio de la Laurie hem tingut que fer dos pasos:
Primer hem comptat les reviews que havia fet la Laurie que tenen més de 100 caracters, que per tan valien 15 punts. => 94 reviews * 15 punts = 1410 punts
Després hem comprovat quantes reviws de la Laurie tenien menys de 100 caracters i aixi hem obtingut => 11 reviews * 10 punts = 110 punts
Per tan si sumem la puntuació de la Laurie obtenim 1410 + 110 = 1520 punts que son els punts que ens donava la query que hem fet.

```
297
298 SELECT DISTINCT * FROM tabla_sum
299 ORDER BY puntos_c DESC
300 LIMIT 10;
301
302 --Verificacio: Laurie
303
304 SELECT DISTINCT Count(r.reviewer_id)
305 FROM Person AS p, Reviewer AS r
306 WHERE p.id_person = 209773 AND
307 p.id_person = r.id_person AND CHAR_LENGTH(r.comments) >= 100;
308
309 -- 94 reviews a 15 punts la review obtenim 1410 punts.
310
311 SELECT DISTINCT Count(r.reviewer_id)
```

Explain | Data Output | Messages | Query History

	count bigint
1	94

```
311 SELECT DISTINCT Count(r.reviewer_id)
312 FROM Person AS p, Reviewer AS r
313 WHERE p.id_person = 209773 AND
314 p.id_person = r.id_person AND CHAR_LENGTH(r.comments) < 100;
315
316 -- 11 reviews a 10 punts la review obtenim 110 punts
317 --Es a dir, que en total tenim 1520 punts
318
```

Explain | Data Output | Messages | Query History

	count bigint
1	11

Query 10

Aquesta query voliem trobar informació sobre un apartament que estigui en una ciutat que comenci per F, que sigui una casa, ademés volíem que estiguessin ordenats de més a menys reviews.

```
319  --#&&
320  /* QUERY 10: */
321
322  SELECT a.name, a.price_m AS price, s.property_type, COUNT(r.reviewer_id) AS review, lc.
323  FROM  Apartment AS a,  Apartment_Reviewer AS ar,Reviewer AS r, Specification AS s, Loc_c
324  WHERE a.id_apartment = ar.id_apartment
325  AND (ar.id_reviewer = r.reviewer_id)
326  AND (s.id_specification = a.id_specification)
327  AND (lc.id_city = a.id_city)
328  AND (lc.city LIKE 'F%')
329  AND (s.property_type ='House')
330  AND(a.square_feet >0)
331  GROUP BY a.name, a.price_m , a.square_feet ,  s.property_type, lc.city HAVING COUNT(r.r
332  ORDER BY review DESC;
333
```

Explain | Data Output | Messages | Query History

	name character varying (255)	price money	property_type character varying (255)	review bigint	city character varying (255)
1	sunlit studio down a quiet l...	$140.00	House	91	Fitzroy
2	Stay Belgravia 9 bedroom H...	$795.00	House	59	Footscray
3	Explore Fitzroy from Delight...	$251.00	House	46	Fitzroy

Verificació

Per fer la verifiacio hem bucat la mateixa informació però aquest com de les taules imp_apartament i imp_review, per assegurar-nos que la nostre query funciona igual en les nostre taules que en les d'importació.

```
334  --Comprobacio 1)
335
336  SELECT a.name, CONCAT('$', + a.price* 1.00) , a.property_type, COUNT(r.reviewer_id) AS
337  FROM  imp_apartament AS a, imp_review AS r
338  WHERE a.listing_url = r.listing_url
339  AND (a.city LIKE 'F%')
340  AND (a.property_type ='House')
341  AND(a.square_feet >0)
342  GROUP BY a.name, a.price , a.square_feet ,  a.property_type, a.city HAVING COUNT(r.rev
343  ORDER BY review DESC;
```

Explain | Data Output | Messages | Query History

	name character varying (255)	concat text	property_type character varying (255)	review bigint	city character varying (255)
1	sunlit studio down a quiet l...	$140.00	House	91	Fitzroy
2	Stay Belgravia 9 bedroom H...	$795.00	House	59	Footscray
3	Explore Fitzroy from Delight...	$251.00	House	46	Fitzroy

Conclusions

Aquesta pràctica ens ha servit molt per entendre l'importància de la normalització, l'importància de tenir uns models ben organitzats i estructurats, i com aquests afecten a l'execució de queries. Ademés, gràcies a les verificacions hem entens la diferència entre executar una query sobre un model normalitzat i sobre un model desnormalitzat.

Degut a consideracions personals que hem fet a l'hora de preparar i estructurar el model, el resultat de les queries no són idèntiques al resultat adjunt al enunciat de la pràctica. Tot i així, després de fer diverses verificacions amb les nostres taules d'importació, hem comprovat que els resultats són correctes, ja que compleixen tots els requisits demanats en el enunciat.

També hem après ha tractar el format de sortida d'un camp. Per exemple, en moltes queries hem hagut de mostrar el resultat final en un tipus diferent al tipus guardat en la taula, per alterar el tipus de sortida cal afegir un "::tipus" al final del camp.

A més a més ens hem trobat en múltiples reptes en el moment de fer les queris, ja que en algunes ocasions hem hagut de crear taules per poder donar un valor a un nou cap i després fer el sumatori de tot el cap. Sobretot ens hem descobert que a vegades es necessita molta imaginació per ingeniar-te-las per organitzar la informació de manera correcte.

La part més difícil per a nosaltres va ser escollir un bona organització de les taules per tal de mantenir un sistema normalitzat que al mateix temps fós àgil i eficient quan executem les queries.