

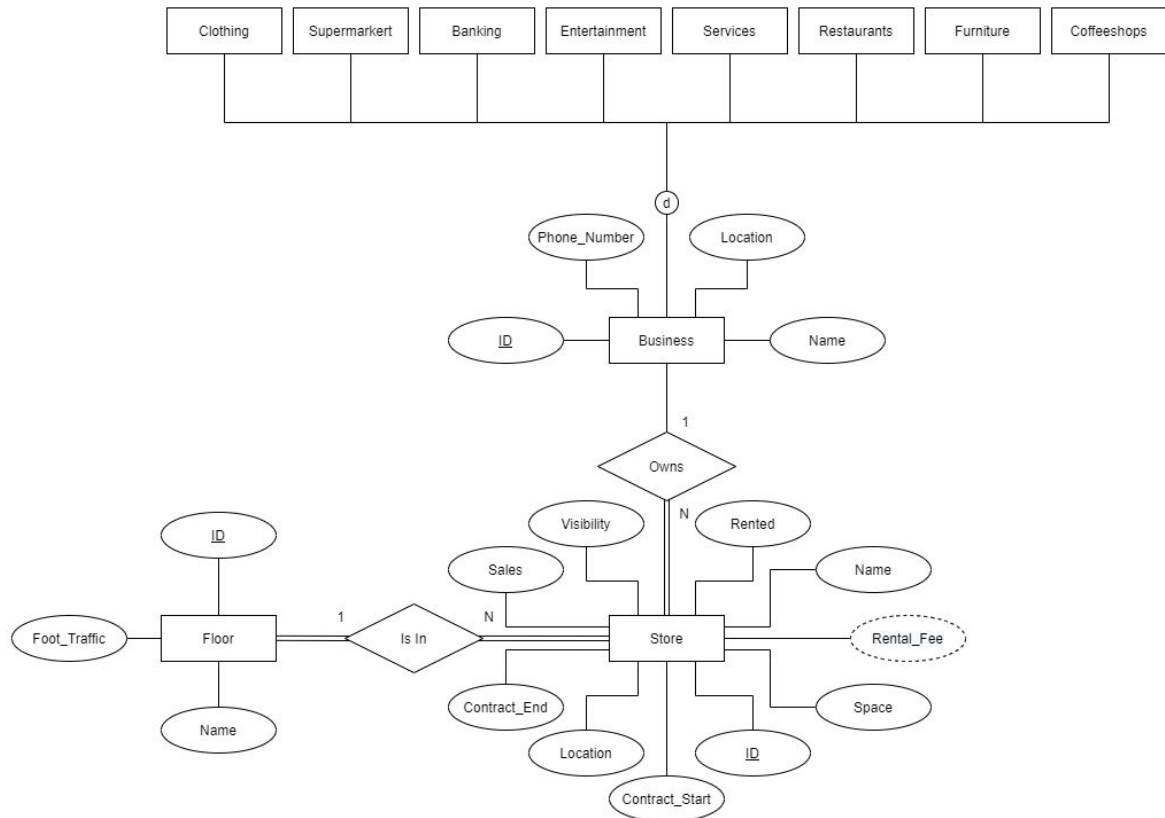
Database Project

Bashar Al Sharaf (20170089)

Omar Alshuha (20180032)

Our data base is called **City mall**.

1- Conceptual schema (ER/EER):



Assumptions:

1-A business may own more than 1 store

2-A business may be in the database if it does not own any stores currently.

Figure 1: ER/EER Diagram

The EER diagram contains of three entities : store,floor and business.

We have two relationships , one of them is called **Is_In that** between floor and store , and the other is called **Owns** between store and business.

{clothing , supermarket ,banking ,entertainment,services,restaurants,furniture,coffeeshops} is a specialization of business based upon **business_type**.

$$rental_fee = visibility \times space \times \frac{sales}{foot_traffic} \times 365$$

This equation below explains how to calculate the rental fee for each store .

Rental_fee parameters:

Visibility: a number inside this [0-1] , the highest number, the highest visible.

0 : completely hidden, 1: completely visible.

Space: area of the store

Sales:the average of transactions/exchanging between the buyer and a seller.

Foot_traffic:average of the people that visit the store per day

365: a constant number , to multiply the whole equation with this number to find the rental store per year.

If the store is to be rented for the 1st time, rental_fee = **25000JD**.

2- Relational Schema:

Relational Mapping

Business

<u>ID</u>	Phone_Number	Location	Name	Business_Type
-----------	--------------	----------	------	---------------

Store

<u>ID</u>	Location	Contract_Start	Contract_End	Name	Space	Rented	Visibility	Sales	Rental_Fee	Floor_ID	Business_ID
-----------	----------	----------------	--------------	------	-------	--------	------------	-------	------------	----------	-------------

Floor

<u>ID</u>	Foot_Traffic	Name
-----------	--------------	------

In relational schema “business” , **ID** is primary key of this relation , The **Business_Type** is called “discriminating attribute” that indicates the subclass to which each tuple belongs, **Name** indicates the owner of the store.

In relational schema “store” , **ID** is primary key of this relation, **location** is the zone of the mall the store belongs to.

Business_ID : a foreign key refers to primary key in the business relation that define which business type of the store.

In relational schema “floor” , **ID** is primary key of this relation.

03-SQL Schema and the Database state:

We installed the postgresql server in our kali, We connected to postgresql database via **psql** command, then we listed all databases that created in postgresql server.

```
postgres=# create database city_mall
```

Figure.2 : Creating the city mall's database.

```
root@kali:~# systemctl start postgresql
root@kali:~# su postgres
postgres@kali:/root$ psql
psql (13.1 (Debian 13.1-1+b1))
Type "help" for help.

postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
city_mall	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
project	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=CTc/postgres
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=CTc/postgres

```
(5 rows)

postgres=#
```

We connected to city mall data base via **\c** command

```
postgres=# \c city_mall
You are now connected to database "city_mall" as user "postgres".
city_mall=#
```

```
city_mall=# select * from floor;
id | foot_traffic | name
---+-----+---
0  |      238.21 | P1
1  |      892.67 | GF
2  |      381.34 | F1
3  |      345.91 | F2
4  |     405.102 | F3
(5 rows)
```

Figure.3 : Floor's table.

```
city_mall=# select * from business;
id | phone_number | location | name | business_type
---+-----+-----+-----+-----
3  | 1317139872  | Uptown  | Inditex group | clothes
4  | 2315612291  | Abdoun  | The Movie Masters cinema group | entertainment
6  | 7712144421  | Tla_Ali | TBC corpration | furniture
5  | 7712334515  | Abdoun  | Abdul Hameed Shoman | banking
7  | 1317139872  | Uptown  | Chris Kempczinski | restaurant
1  | 7812345612  | Khalda  | ALSHARYA | coffeeshops
0  | 7788821214  | Sweileh | Majid Al Futtaim Group | supermarket
2  | 7712345901  | Dahiyat Al Rasheed | huawei investement and holding | services
(8 rows)
```

Figure.4 : Business's table.

```
city_mall=# select * from store;
id | location | contract_start | contract_end | name | space | rented | visibility | sales | rental_fee | floor_id | business_id
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
35 | GF-35 | 2013-01-01 | 2024-01-01 | Midas | 250 | t | 0.5 | 500 | 25555.355 | 1 | 6
39 | GF-39 | 2013-02-03 | 2020-12-30 | starbucks | 306 | f | 0.6475 | 322 | 306.9 | 1 | 1
27 | F3-27 | 2012-04-02 | 2024-04-02 | macdonald | 450 | t | 0.4451 | 74 | 13354.58 | 4 | 7
16 | P1-16 | 2010-04-01 | 2024-04-01 | Arab Bank | 908 | t | 0.1185 | 299 | 49295.61 | 0 | 5
14 | F3-14 | 2014-09-01 | 2019-01-01 | Zara | 578 | f | 0.2023 | 127 | 13379.99 | 4 | 3
21 | GF-21 | 2013-08-15 | 2024-08-15 | grand cinema | 847 | t | 0.9062 | 258 | 178424.97 | 1 | 4
1 | GF-01 | 2006-08-15 | 2030-08-15 | carrefour | 146 | t | 0.4957 | 695 | 20566.41 | 1 | 0
6 | GF-06 | | | | 316 | f | 0.4882 | | 25000 | 1 |
(8 rows)
```

Figure.5 : Store's table.

SQL queries:

1-floor queries:

```
drop table if exists floor cascade;
create table floor(id serial primary key,foot_traffic float(15),name varchar(255) unique not
null);
INSERT INTO FLOOR(Foot_Traffic, Name) VALUES (238.21,'P1');
INSERT INTO FLOOR(Foot_Traffic, Name) VALUES (892.67,'GF');
INSERT INTO FLOOR(Foot_Traffic, Name) VALUES (381.34,'F1');
INSERT INTO FLOOR(Foot_Traffic, Name) VALUES (345.91,'F2');
INSERT INTO FLOOR(Foot_Traffic, Name) VALUES (405.102,'F3');
```

2-Business queries:

```
drop table if exists business cascade;
create table business (id serial primary key,phone_number varchar(10),location
varchar(255),name varchar(255) unique not null,business_type varchar(255)not null);

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(0778882121,
'Sweileh','Majid Al Futtaim Group','supermarket'); #carrefour.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(0781234561,
'Downtown', 'ALSHARYA','coffeeshops');#starbucks.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(0771234561,
'Dahiyat Al Rasheed', 'Telenor','services');#huawei.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(1317139872,
'Uptown', 'Inditex group','clothes');#ZARA.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(2315612291,
'Abdoun', 'The Movie Masters cinema group','entertainment');#grandcinema.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(0771233451,
'Abdoun', 'Abdul Hameed Shoman','banking');#arab bank.

INSERT INTO Business(Phone_Number, Location, Name,business_type ) VALUES(0771214442,
'Tla_Ali', 'TBC corpration','furniture');#Midas.
```

```
INSERT INTO Business(Phone_Number, Location, Name,business_type )
VALUES(1317139872,'Uptown', 'Chris Kempczinski','furniture');#MACDONALD.
```

3-Store queries:

```
drop table if exists store cascade;
```

```
create table store (id serial primary key,location varchar(255), contract_start date,
contract_end date, name varchar(255) unique, space float(15),rented boolean,visibility
float(10), sales int , rental_fee float(15));
```

```
alter table store add column floor_id int;
```

```
alter table store add constraint fk foreign key (Floor_id) references floor (id);
```

```
alter table store add column business_id int ;
```

```
alter table store add constraint fk_1 foreign key (Business_id) references business (id);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Ground Floor', ' 2013-02-03 ' , '2020-12-30
','starbucks',306,false,0.6475,322,306.90,2,2);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Ground Floor', '2013-01-01',
'2024-01-01','Midas',250,true, 0.5,500,25555.355 ,2,7);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Third Floor', '2012-04-02' ,
'2024-04-02','macdonald',450,true,0.4451,74,13354.58,5,8);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Parking Floor', '2010-04-01 ' , ' 2024-04-01 ' ,'Arab
Bank',908,true,0.1185,299,49295.61 ,1,6);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Third Floor', '2014-09-01' ,
'2019-01-01','Zara',578,false,0.2023,127,13379.99 ,5,4);
```



```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Ground Floor', '2013-08-15' , '2024-08-15
','grand cinema',847,true,0.9062,258,178424.97,2,5);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Ground Floor', Null ,
Null,Null,316,false,0.4882,Null,Null,2,Null);
```

```
INSERT INTO store (location, contract_start,contract_end ,name,space,rented,visibility,
sales,rental_fee,Floor_id,Business_id) VALUES('Ground Floor', '2006-08-15' ,
'2030-08-15','carrefour',146,true,0.4957,695,20566.41,2,1);
```

SQL data discussions:

```
city_mall=# select * from store;
```

id	location	contract_start	contract_end	name	space	rented	visibility	sales	rental_fee	floor_id	business_id
35	GF-35	2013-01-01	2024-01-01	Midas	250	t	0.5	500	25555.355	1	6
39	GF-39	2013-02-03	2020-12-30	starbucks	306	f	0.6475	322	306.9	1	1
27	F3-27	2012-04-02	2024-04-02	macdonald	450	t	0.4451	74	13354.58	4	7
16	P1-16	2010-04-01	2024-04-01	Arab Bank	908	t	0.1185	299	49295.61	0	5
14	F3-14	2014-09-01	2019-01-01	Zara	578	f	0.2023	127	13379.99	4	3
21	GF-21	2013-08-15	2024-08-15	grand cinema	847	t	0.9062	258	178424.97	1	4
1	GF-01	2006-08-15	2030-08-15	carrefour	146	t	0.4957	695	20566.41	1	0
6	GF-06				316	f	0.4882		25000	1	

(8 rows)

In the figure above , we see the store's table and we will discuss some queries.

Notes:

- 1- We added **serial** constraint in sql queries when we create the tables , and we used serial to define auto-increment columns in tables.
- 2- We added also **unique** constraint to ensures that all values in a column are different.

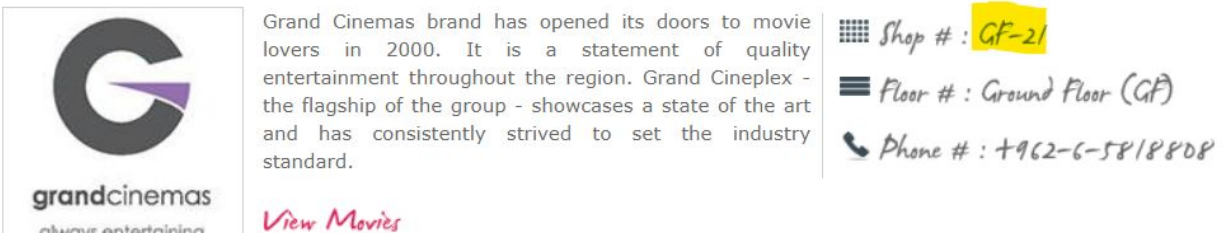


Figure.6 : Grand cinema 's store

Let's take an example , Grand cinema is a store where it is located in zone 2. Grand cinema contract started at **2013-08-15** and it will finish three four years , grand cinema 's space equals to **847** meter square, and the rented flag is false because the grand cinema is still renting as we see in the end contract.

Grand cinema 's visibility equals **0.9062** and total its sales equal **258**, so the rental fee per year for this store equals **178424.97 JD**.

Grand cinema has two foreign keys , one of them is floor_id and its value is 1 , that means the foot_traffic in this store has **892.67 people** per day and this store located in **GF** floor. the other foregin key is Business_id that means grand cinema is an **entertainment** business type and the owner of the store is **The Movie Masters cinema group** .

In the business table , we have three stores are not rented due to some reasons:

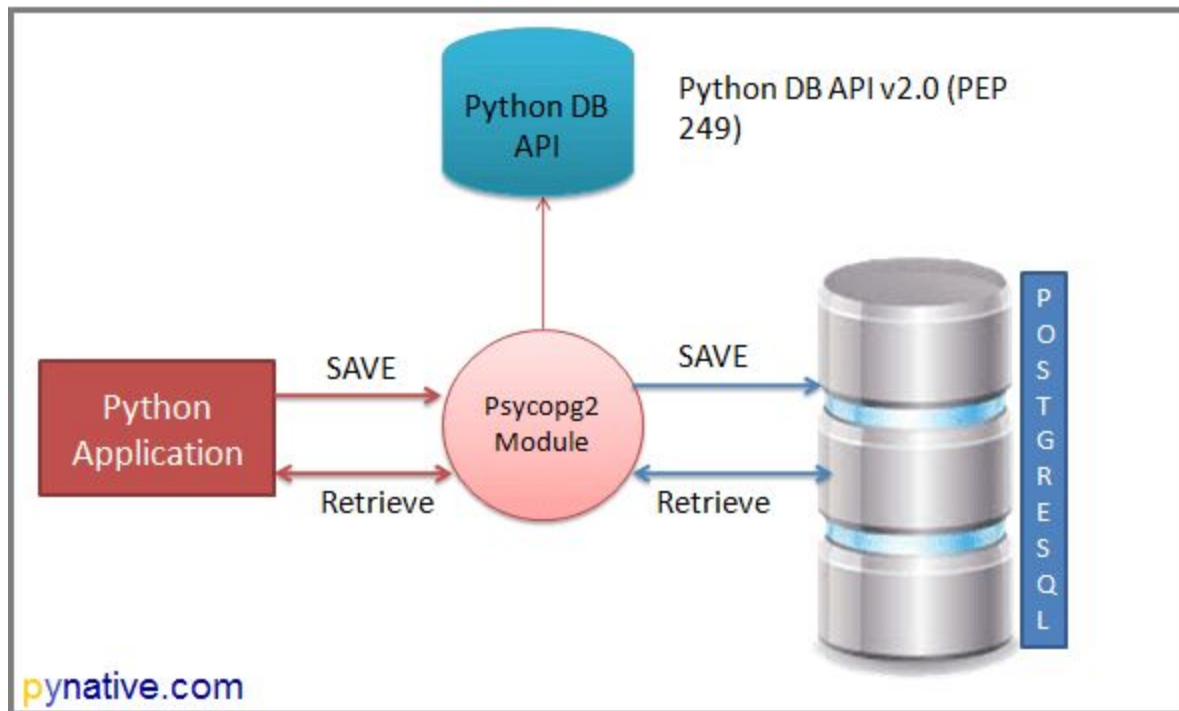
- 1-The lease of the store is expired.
- 2-the store has no these Necessary attributes : name , start contract, end contract , name, sales and business_type.

Examples of the stores are not rented:

- 1- **Zara** , the lease of the contract is expired at 2019.
- 2-**Starbucks** , the lease of the contract is expired at 2020.
- 3- A store has a location "**Grand floor**" and it has no name , no start and end contract , so theres's no sales , it's anonymous store and we can't classify it to business_store .

4- The DBMS and the programming environment (The interface) :

For the interface, we have decided to use python as the programming language for it. We chose python for its flexibility, and ease of use. For the DBMS we chose PostgreSQL. And to connect them together we used the psycopg2 library which offers lots of functionalities that were useful for our interface



Psycopg2 uses the API to communicate with the postgresSQL database. It connects directly to the database.

5- Tasks team members:

Student name:	The Tasks
Omar Alshuha	1- Conceptual schema (ER/EER). 2- Relational Schema. 3-Designing the interface. 4-Writing the report.
Bashar Sharaf	1- Conceptual schema (ER/EER). 2- Relational Schema. 3-SQL Schema and the Database state. 4-Writing the report.