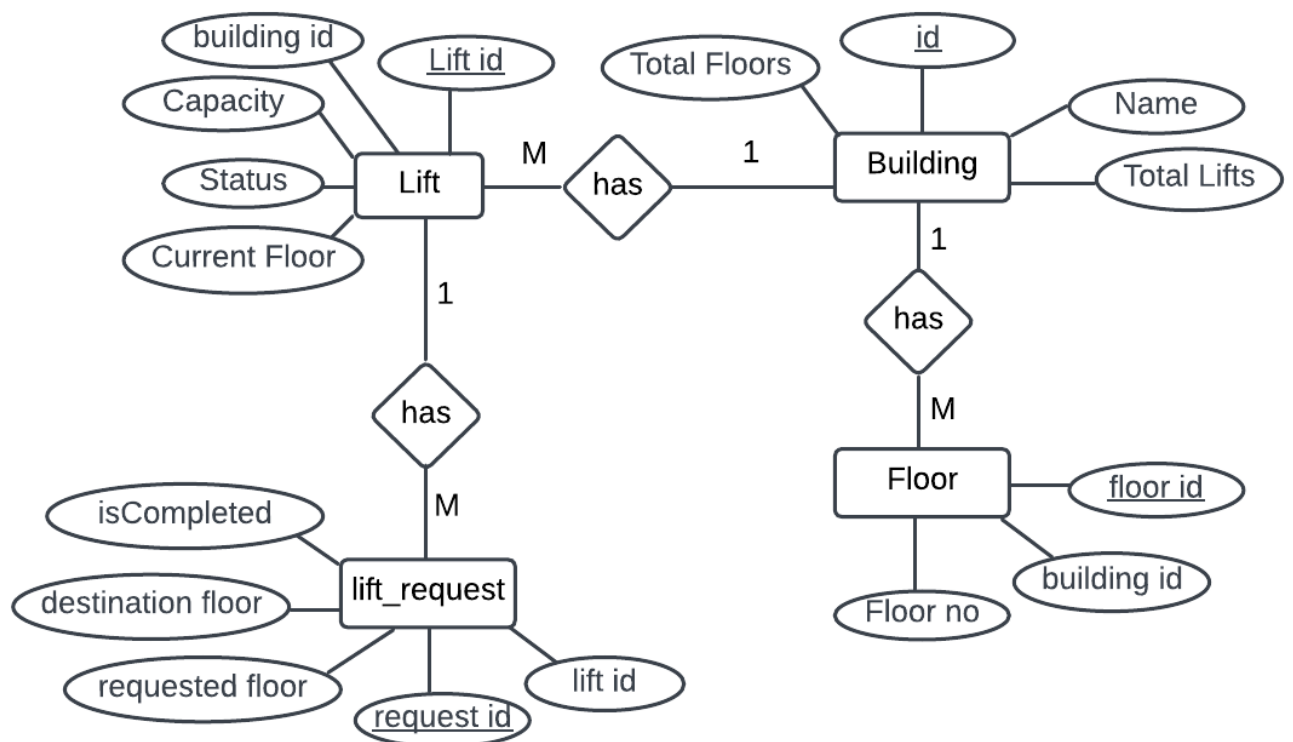


Assignment : Creating database for lift Management System

Coordinator : Suresh Burde

Submitted by : Bhagyashree Dattatray Watpade.

Entity Relationship Diagram:



Database : PostgreSQL

Tables : Building, Floor, Lift, lift request

Creating Tables:

```
7  create table Building (  
8      building_id serial primary key,  
9      Name VARCHAR(20) NOT NULL,  
10     Total_Floors INT NOT NULL,  
11     Total_Lifts INT NOT NULL  
12  
13 );  
14 |  
15 create table Floor (  
16     floor_id serial primary key,  
17     Floor_no INT NOT NULL,  
18     building_id INT REFERENCES Building(building_id) on delete cascade  
19 );  
20  
21  
22 create table Lift (  
23     lift_id serial primary key,  
24     Capacity INT,  
25     Status VARCHAR(50),  
26     Current_Floor INT,  
27     building_id INT REFERENCES Building(building_id) on delete cascade  
28  
29 );  
30
```

```
30  
31 create table Lift_Request (  
32     request_id serial primary key,  
33     requested_floor int,  
34     destination_floor int,  
35     lift_id int REFERENCES Lift(lift_id) on delete cascade,  
36     isCompleted BOOLEAN DEFAULT FALSE  
37 );  
38
```

Insert Data Into Building Table :

```
39  
40 insert into Building(Name, Total_Floors, Total_Lifts) values('Building_A', 10, 5);  
41 insert into Building(Name, Total_Floors, Total_Lifts) values('Building_B', 15, 10);  
42 insert into Building(Name, Total_Floors, Total_Lifts) values('Building_C', 40, 15);  
43
```

Insert Data Into Floor Table :

```
48
49 insert into Floor (Floor_no, building_id) values (1, 1);
50 insert into Floor (Floor_no, building_id) values (1, 2);
51 insert into Floor (Floor_no, building_id) values (2, 2);
52 insert into Floor (Floor_no, building_id) values (2, 3);
53 insert into Floor (Floor_no, building_id) values (3, 1);
54 insert into Floor (Floor_no, building_id) values (3, 3);
55
```

Insert data into Lift Table :

```
58
59 insert into Lift (Capacity, Status, Current_Floor, building_id) values (1000, 'Idle', 1, 1);
60 insert into Lift (Capacity, Status, Current_Floor, building_id) values (3000, 'Moving Up', 2, 2);
61 insert into Lift (Capacity, Status, Current_Floor, building_id) values (5000, 'Moving Up', 1, 3);
62 insert into Lift (Capacity, Status, Current_Floor, building_id) values (1500, 'Moving Down', 3, 1);
63
64
```

Insert Data Into Lift Request Table:

```
58
59 insert into Lift (Capacity, Status, Current_Floor, building_id) values (1000, 'Idle', 1, 1);
60 insert into Lift (Capacity, Status, Current_Floor, building_id) values (3000, 'Moving Up', 2, 2);
61 insert into Lift (Capacity, Status, Current_Floor, building_id) values (5000, 'Moving Up', 1, 3);
62 insert into Lift (Capacity, Status, Current_Floor, building_id) values (1500, 'Moving Down', 3, 1);
63
64
```

Select * from Building;

Data Output					
Messages					
Notifications					
	building_id [PK] integer	name character varying (20)	total_floors integer	total_lifts integer	
1	1	Building_A	10	5	
2	2	Building_B	15	10	
3	3	Building_C	40	15	

select * from Floor;

	floor_id [PK] integer	floor_no integer	building_id integer
1	1	1	1
2	2	1	2
3	3	2	2
4	4	2	3
5	5	3	1
6	6	3	3

select * from Lift;

	lift_id [PK] integer	capacity integer	status character varying (50)	current_floor integer	building_id integer
1	1	1000	Idle	1	1
2	2	3000	Moving Up	2	2
3	3	5000	Moving Up	1	3
4	4	1500	Moving Down	3	1

select * from Lift Request;

	request_id [PK] integer	requested_floor integer	destination_floor integer	lift_id integer	iscompleted boolean
1	1	1	9	1	false
2	2	2	8	2	true
3	3	3	7	3	false
4	4	1	4	4	true

Retrieve all lifts in a specific building:

SELECT * FROM Lift WHERE building_id = 2;

	lift_id [PK] integer	capacity integer	status character varying (50)	current_floor integer	building_id integer
1	2	3000	Moving Up	2	2

Code :

```
drop table Building cascade
drop table Floor
drop table Lift cascade
drop table Lift_Request
```

```
create table Building (
    building_id serial primary key,
    Name VARCHAR(20) NOT NULL,
    Total_Floors INT NOT NULL,
    Total_Lifts INT NOT NULL
```

```
);
```

```
create table Floor (
    floor_id serial primary key,
    Floor_no INT NOT NULL,
    building_id INT REFERENCES Building(building_id) on delete cascade
);
```

```
create table Lift (
    lift_id serial primary key,
    Capacity INT,
    Status VARCHAR(50),
    Current_Floor INT,
    building_id INT REFERENCES Building(building_id) on delete cascade
);
```

```
create table Lift_Request (
    request_id serial primary key,
    requested_floor int,
    destination_floor int,
    lift_id int REFERENCES Lift(lift_id) on delete cascade,
    isCompleted BOOLEAN DEFAULT FALSE
);
```

```
insert into Building(Name, Total_Floors, Total_Lifts) values('Building_A', 10, 5);
insert into Building(Name, Total_Floors, Total_Lifts)values('Building_B', 15, 10);
```

```
insert into Building(Name, Total_Floors, Total_Lifts) values('Building_C', 40, 15);
```

```
select * from Building;
```

```
/*insert data into Floor table */
```

```
insert into Floor (Floor_no, building_id) values (1, 1);
insert into Floor (Floor_no, building_id) values (1, 2);
insert into Floor (Floor_no, building_id) values (2, 2);
insert into Floor (Floor_no, building_id) values (2, 3);
insert into Floor (Floor_no, building_id) values (3, 1);
insert into Floor (Floor_no, building_id) values (3, 3);
```

```
/*insert data into Lift table */
```

```
insert into Lift (Capacity, Status, Current_Floor, building_id) values (1000, 'Idle', 1, 1);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (3000, 'Moving Up', 2, 2);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (5000, 'Moving Up', 1, 3);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (1500, 'Moving Down', 3, 1);
```

```
/* Insert data into lift request table */
```

```
insert into Lift_Request (requested_floor, destination_floor, lift_id, isCompleted) values (1, 9, 1, FALSE);
insert into Lift_Request (requested_floor, destination_floor, lift_id, isCompleted) values (2, 8, 2, TRUE);
insert into Lift_Request (requested_floor, destination_floor, lift_id, isCompleted) values (3, 7, 3, FALSE);
insert into Lift_Request (requested_floor, destination_floor, lift_id, isCompleted) values (1, 4, 4, TRUE);
```

```
select * from Building;
select * from Floor;
select * from Lift;
select * from Lift_Request;
```

```
/*Retrieve all lifts in a specific building*/
```

```
select *
from Lift
where building_id = 2;
```

