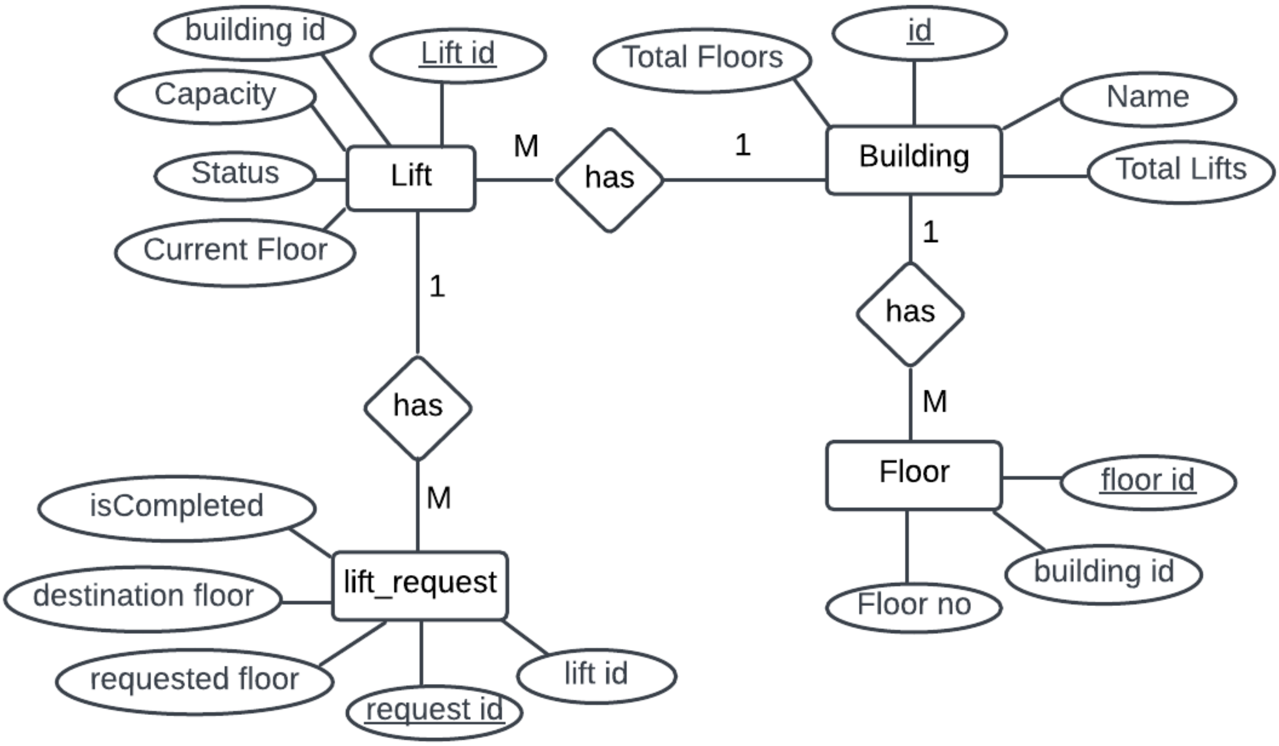
Assignment : Creating database for lift Management System

Coordinator : Suresh Burde

Submitted by : Bhagyashree Dattatray Watpade.

---

## Entity Relationship Diagram:

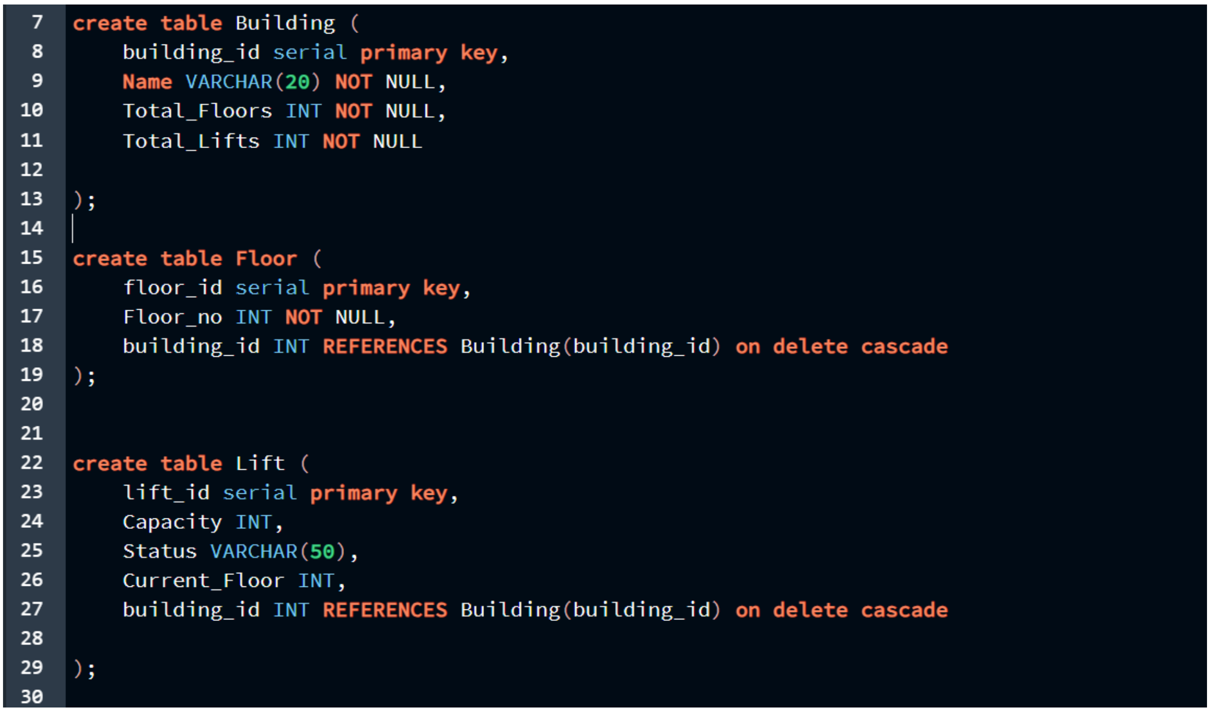


---

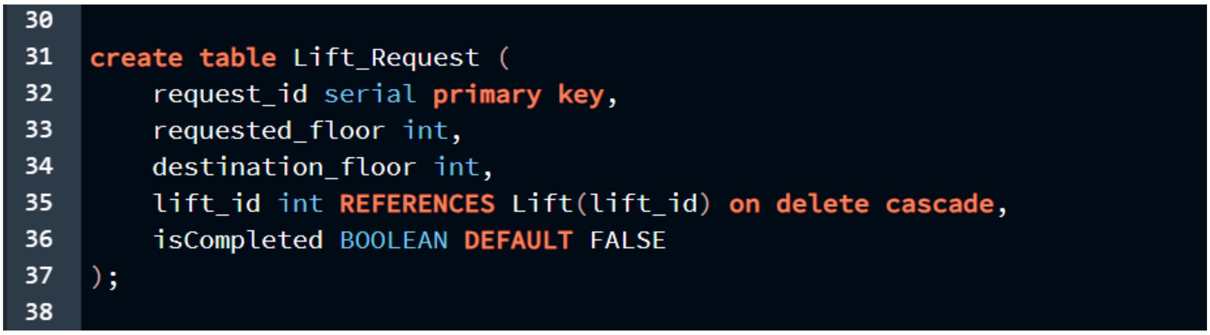
## Database : PostgreSQL

## Tables : Building, Floor, Lift, lift_request

### Creating Tables:

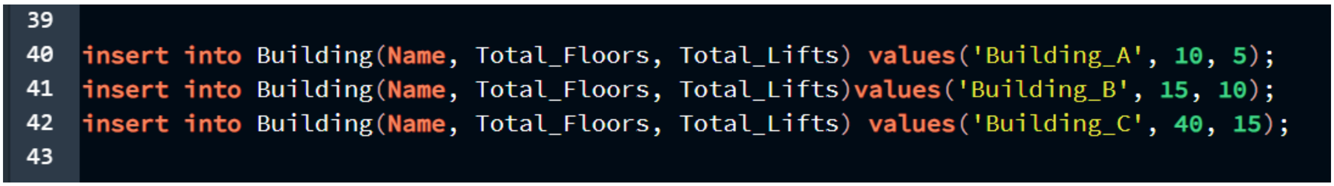


```sql
create table Building (
    building_id serial primary key,
    Name VARCHAR(20) NOT NULL,
    Total_Floors INT NOT NULL,
    Total_Lifts INT NOT NULL

);

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

);

```



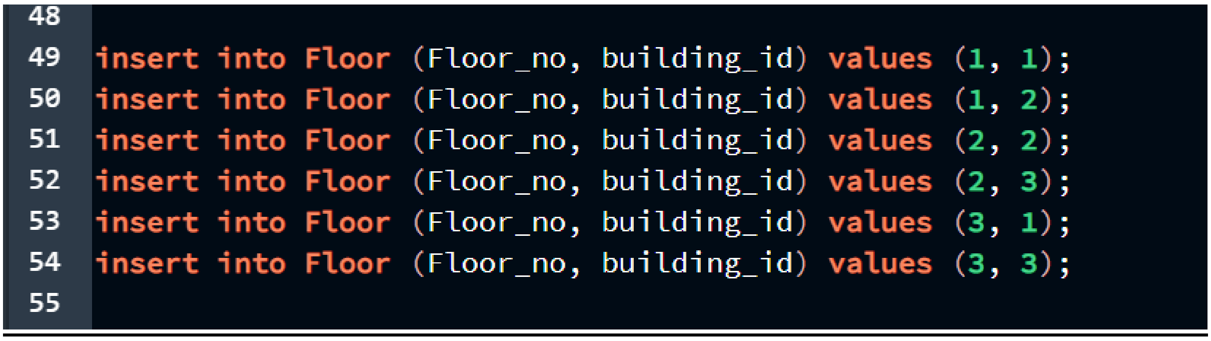
```sql

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 Data Into Building Table :



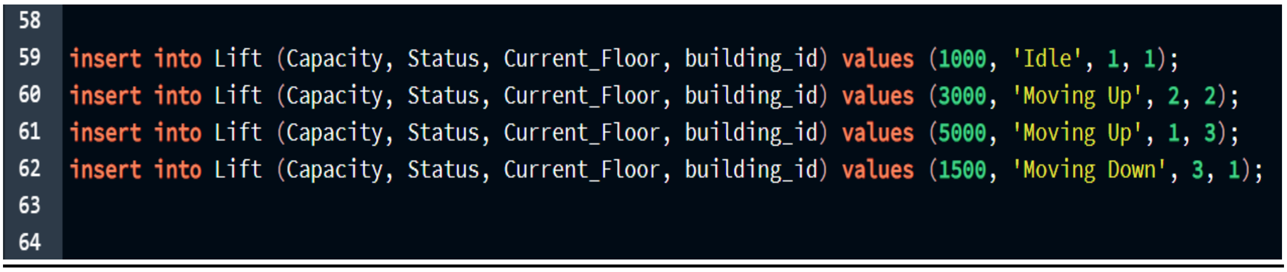
```sql

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);

```

Insert Data Into Floor Table :



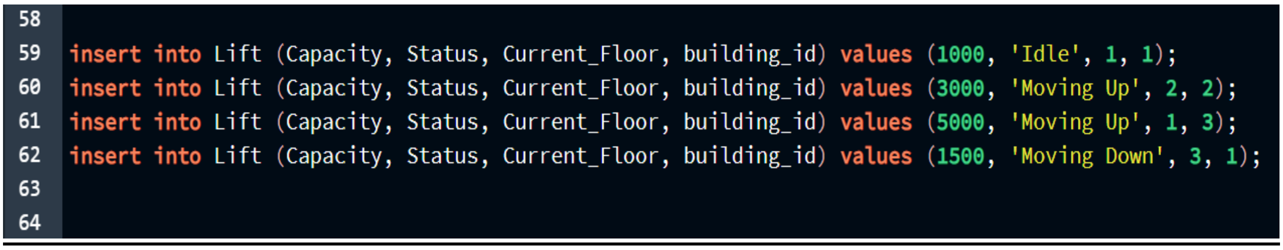
```sql
insert into Floor (Floor_no, building_id) values (1, 1);
insert into Floor (Floor_no, building_id) values (1, 2);
insert into Floor (Floor_no, building_id) values (2, 2);
insert into Floor (Floor_no, building_id) values (2, 3);
insert into Floor (Floor_no, building_id) values (3, 1);
insert into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

Insert data into Lift Table :



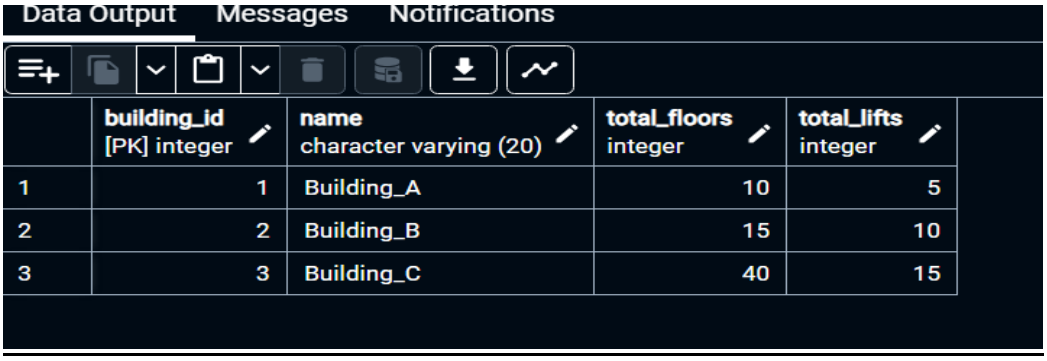
```sql
insert into Lift (Capacity, Status, Current_Floor, building_id) values (1000, 'Idle', 1, 1);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (3000, 'Moving Up', 2, 2);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (5000, 'Moving Up', 1, 3);
insert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

Insert Data Into Lift Request Table:



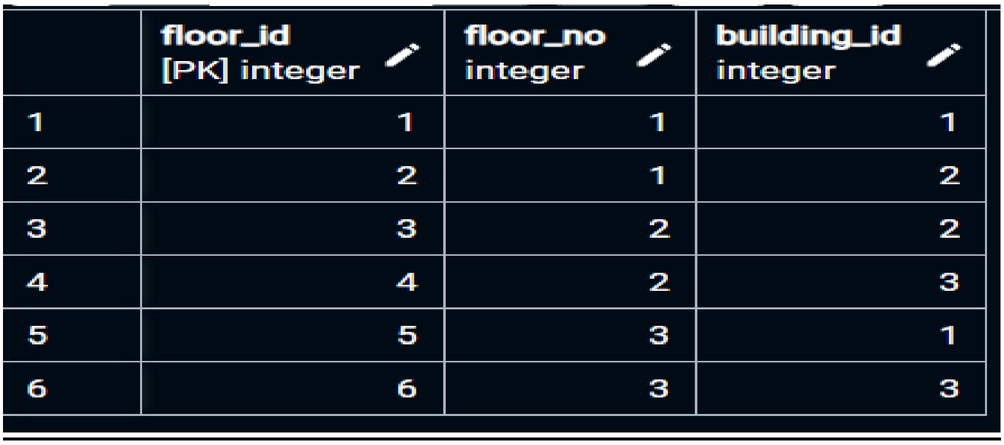
```sql
insert into Lift (Capacity, Status, Current_Floor, building_id) values (1000, 'Idle', 1, 1);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (3000, 'Moving Up', 2, 2);
insert into Lift (Capacity, Status, Current_Floor, building_id) values (5000, 'Moving Up', 1, 3);
insert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

Select * from Building;



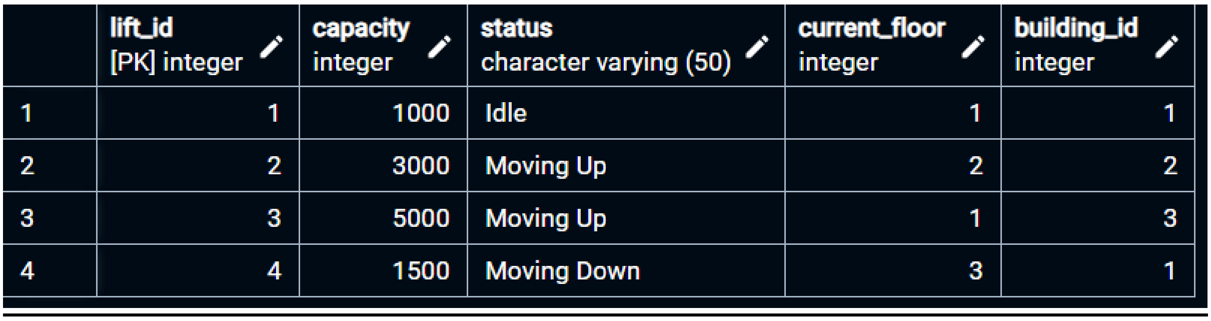
| | building_id<br>[PK] integer | name<br>character varying (20) | total_floors<br>integer | total_lifts<br>integer |
|---|---|---|---|---|
| 1 | 1 | Building_A | 10 | 5 |
| 2 | 2 | Building_B | 15 | 10 |
| 3 | 3 | Building_C | 40 | 15 |

select * from Floor;



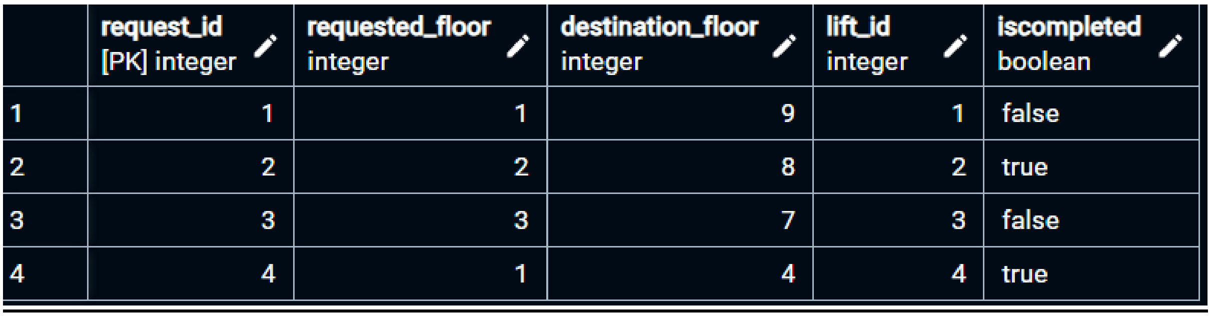
| | floor_id<br>[PK] integer | floor_no<br>integer | building_id<br>integer |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 2 |
| 3 | 3 | 2 | 2 |
| 4 | 4 | 2 | 3 |
| 5 | 5 | 3 | 1 |
| 6 | 6 | 3 | 3 |

select * from Lift;



| | lift_id<br>[PK] integer | capacity<br>integer | status<br>character varying (50) | current_floor<br>integer | building_id<br>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 |

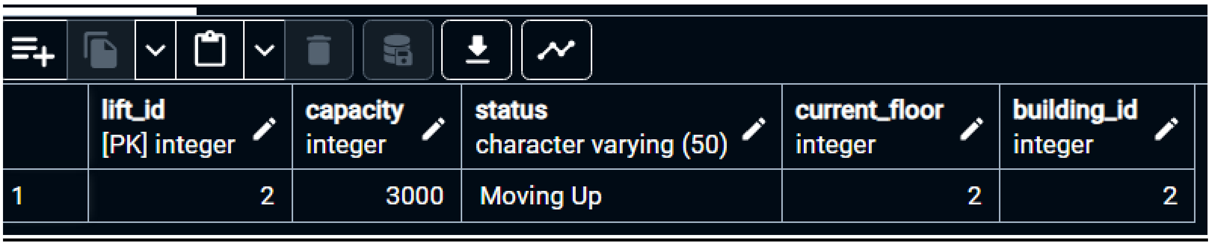
select * from Lift_Request;



| | request_id<br>[PK] integer | requested_floor<br>integer | destination_floor<br>integer | lift_id<br>integer | iscompleted<br>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<br>[PK] integer | capacity<br>integer | status<br>character varying (50) | current_floor<br>integer | building_id<br>integer |
|---|---|---|---|---|---|
| 1 | 2 | 3000 | Moving Up | 2 | 2 |

<u>Code :</u>

```
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);
```

```sql
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;
```