Co-working Space Management System

Data Management for Analytics

Milestone: Project Report

Group 24

Aditi Chadha

Harsh Shingala

857-391-4247

781-985-1057

chadha.ad@northeastern.edu

shingala.h@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Aditi Chadha

Signature of Student 2: Harsh Shingala

Submission Date: 12/09/22

Use Case Study Report

Group No. 24

Student Names: Aditi Chadha & Harsh Shingala

I. Executive Summary:

The principal objective of this study was to design, execute a relational database and write queries on the implemented database. Many studies suggest that creativity amongst workers can be sufficiently enhanced with appropriate physical environments having features like freedom, opportunity to interact with diverse set of individuals along with a supportive environment [1]. There are many research-intensive companies who are struggling to find spaces for their employees who work in remote locations, away from headquarters and unable to travel to the location. Setting up local Co-working space is a recent solution to this problem, allowing a shared, collaborative environment to foster creativity for small enterprises as well as allow an opportunity for freelancers to make new connections and allow them to work from anywhere [2]. However, to operate the coworking space we need to have a database to store the bookings, visitors, spaces available, bill and manage invoices etc. In this project we, aim to build a **Coworking Space Management system** for 'SmartSpaces', an upcoming provider of coworking spaces aiming to expand across Boston, Massachusetts.

The database was modelled based on the assumptions and requirements researched for managing spaces, booking system, employee management system and billing systems. After successfully identifying the major entity types and their attribute types, the EER diagram was made. The UML diagram was also made to further refine the database by refining on the limitations of EER diagram. The first step in implementing the database was to generate the data as per the requirements. For the purpose of populating data, Mockaroo was used. Additionally, the data was manually refined in excel to fine tune the format required by MySQL. The generated data was stored in csv file. Using DDL commands, we created the database named 'coworking' and their respective tables. Datatypes for each column, Foreign key, Primary Key as well as Not Null constraints were specified while creating a table. Used Import csv option to upload the data into respective tables. The database was further refined on few columns of 'Invoice' table such as 'SpaceCost' and 'ServiceCost' based on the update statements using the 'PerSeatCost' and 'SeatsBooked'.

The created database 'coworking' was connected to Python in Jupyter Notebook and few analytics were depicted, giving insights as to how 'SmartSpaces' can expand their services across Boston**.** Few metrics like 'Recurring Users', Capacity of Neighborhoods and revenue generated can assist in understanding the locations where the organization should focus on building their services.

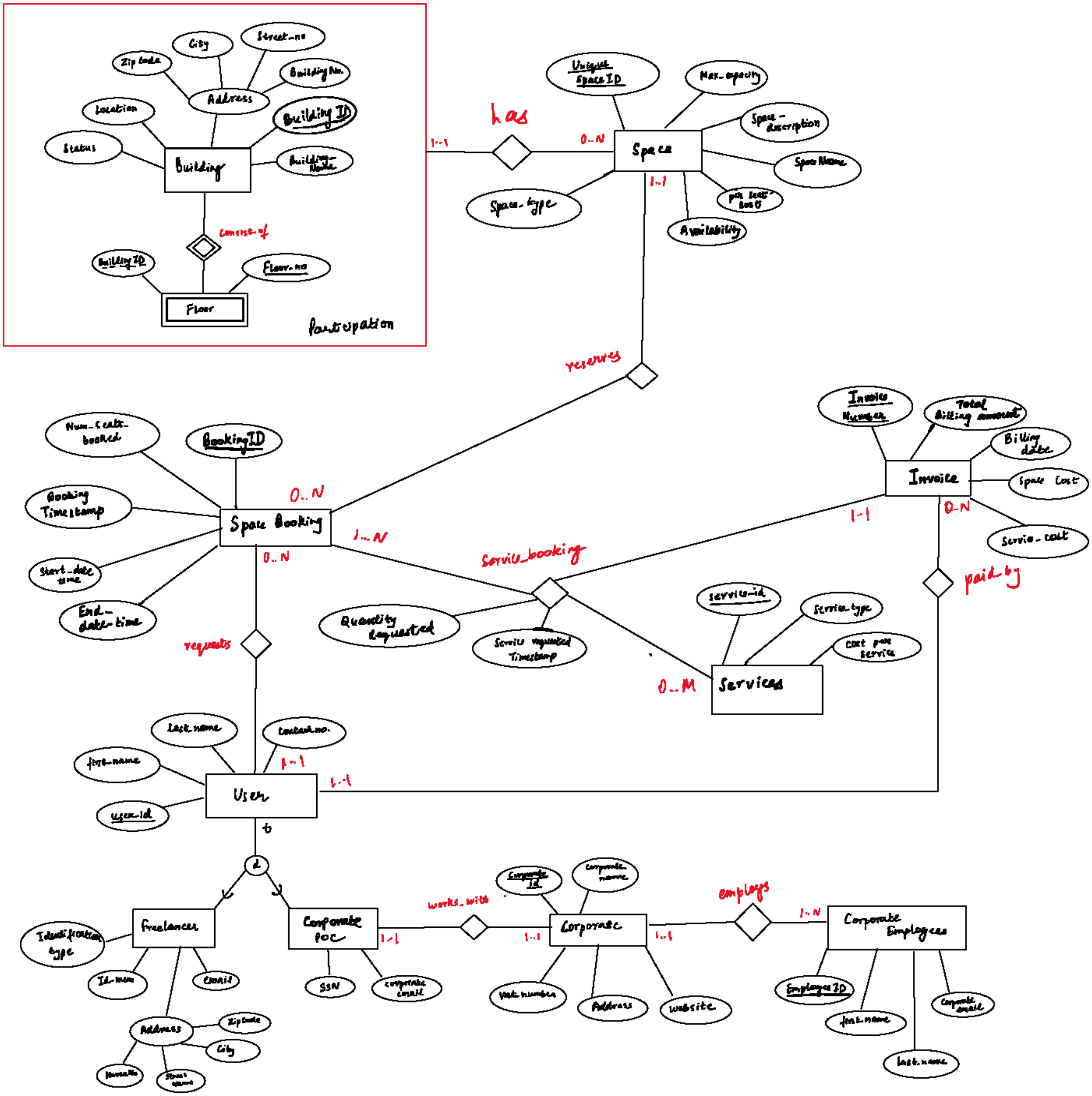
II. Introduction

According to a study, amongst the Small-Scale Businesses, which can be defined as organizations with up to 500 employees, firms with 20-99 employees had the highest share of small businesses. This necessitates designated spaces for work for the growing number of businesses within reasonable costs. Especially during the early days of small businesses, it is imperative that interaction and collaboration amongst the employees and team members is effective and efficient. Studies suggest that creativity amongst workers can be sufficiently enhanced with appropriate physical environments having features like freedom, opportunity to interact with diverse set of individuals along with a supportive environment [1]. There are many research-intensive companies who are struggling to find spaces for their employees who work in remote locations, away from headquarters and unable to travel to the location. This necessitates in building spaces to work in local to different areas which have easy access to transportation services and urban resources. There are already many startups like 'WeWork', 'ImpactHub', 'Office Evolution' etc. working to solve the infrastructure availability problem for the remote workers, new startups, or freelancers.

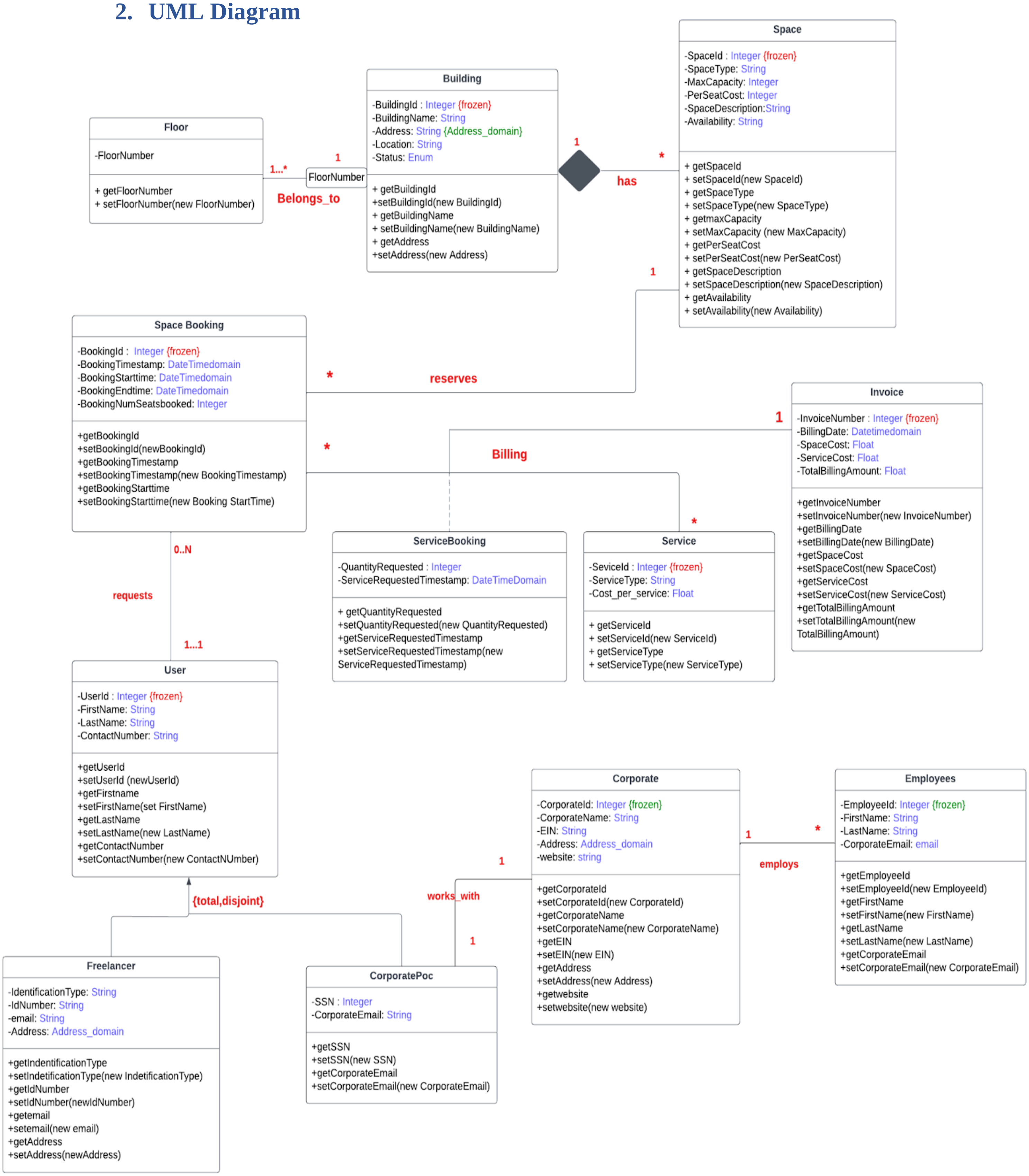
With this project, we want to enable 'SmartSpaces' to keep track & monitor their services in different locations throughout Boston as well as assist them in further expanding their business. We will use the coworking space management system to understand the types of users and their behaviors as well as recognize the neighborhoods across Boston where the business is currently flourishing and can be further expanded.

III. Conceptual Data Modelling

1. EER Diagram



2. UML Diagram


Space
-SpaceId : Integer {frozen}
-SpaceType: String
-MaxCapacity: Integer
-PerSeatCost: Integer
-SpaceDescription:String
-Availability: String
+ getSpaceId
+ setSpaceId(new SpaceId)
+ getSpaceType
+ setSpaceType(new SpaceType)
+ getmaxCapacity
+ setMaxCapacity (new MaxCapacity)
+ getPerSeatCost
+ setPerSeatCost(new PerSeatCost)
+ getSpaceDescription
+ setSpaceDescription(new SpaceDescription)
+ getAvailability
+ setAvailability(new Availability)
Building
-BuildingId : Integer {frozen}
-BuildingName: String
-Address: String {Address_domain}
-Location: String
-Status: Enum
+ getBuildingId
+setBuildingId(new BuildingId)
+ getBuildingName
+ setBuildingName(new BuildingName)
+ getAddress
+setAddress(new Address)
Floor
-FloorNumber
+ getFloorNumber
+ setFloorNumber(new FloorNumber)
1...*
1
FloorNumber
Belongs_to
1
*
has
1
Space Booking
-BookingId : Integer {frozen}
-BookingTimestamp: DateTimedomain
-BookingStarttime: DateTimedomain
-BookingEndtime: DateTimedomain
-BookingNumSeatsbooked: Integer
+getBookingId
+setBookingId(newBookingId)
+getBookingTimestamp
+setBookingTimestamp(new BookingTimestamp)
+getBookingStarttime
+setBookingStarttime(new Booking StartTime)
*
reserves
Invoice
-InvoiceNumber : Integer {frozen}
-BillingDate: Datetimedomain
-SpaceCost: Float
-ServiceCost: Float
-TotalBillingAmount: Float
+getInvoiceNumber
+setInvoiceNumber(new InvoiceNumber)
+getBillingDate
+setBillingDate(new BillingDate)
+getSpaceCost
+setSpaceCost(new SpaceCost)
+getServiceCost
+setServiceCost(new ServiceCost)
+getTotalBillingAmount
+setTotalBillingAmount(new TotalBillingAmount)
1
*
Billing
*
ServiceBooking
-QuantityRequested : Integer
-ServiceRequestedTimestamp: DateTimeDomain
+ getQuantityRequested
+setQuantityRequested(new QuantityRequested)
+getServiceRequestedTimestamp
+setServiceRequestedTimestamp(new ServiceRequestedTimestamp)
Service
-SeviceId : Integer {frozen}
-ServiceType: String
-Cost_per_service: Float
+ getServiceId
+ setServiceId(new ServiceId)
+ getServiceType
+ setServiceType(new ServiceType)
0..N
requests
1...1
User
-UserId : Integer {frozen}
-FirstName: String
-LastName: String
-ContactNumber: String
+getUserId
+setUserId (newUserId)
+getFirstname
+setFirstName(set FirstName)
+getLastName
+setLastName(new LastName)
+getContactNumber
+setContactNumber(new ContactNUmber)
{total,disjoint}
Freelancer
-IdentificationType: String
-IdNumber: String
-email: String
-Address: Address_domain
+getIndentificationType
+setIndetificationType(new IndetificationType)
+getIdNumber
+setIdNumber(newIdNumber)
+getemail
+setemail(new email)
+getAddress
+setAddress(newAddress)
CorporatePoc
-SSN : Integer
-CorporateEmail: String
+getSSN
+setSSN(new SSN)
+getCorporateEmail
+setCorporateEmail(new CorporateEmail)
works_with
1
1
Corporate
-CorporateId: Integer {frozen}
-CorporateName: String
-EIN: String
-Address: Address_domain
-website: string
+getCorporateId
+setCorporateId(new CorporateId)
+getCorporateName
+setCorporateName(new CorporateName)
+getEIN
+setEIN(new EIN)
+getAddress
+setAddress(new Address)
+getwebsite
+setwebsite(new website)
1
*
employs
Employees
-EmployeeId: Integer {frozen}
-FirstName: String
-LastName: String
-CorporateEmail: email
+getEmployeeId
+setEmployeeId(new EmployeeId)
+getFirstName
+setFirstName(new FirstName)
+getLastName
+setLastName(new LastName)
+getCorporateEmail
+setCorporateEmail(new CorporateEmail)

IV. Mapping Conceptual Model to Relational Model

Primary Key – Bold, Underlined **Foreign Key**: Bold, Italicized

Building(**BuildingId**, BuildingName, BuildingNo, StreetName, City, ZipCode, Location, Status)

Building_Floor(***BuildingId***, ***FloorNumber***)

Primary Key and Foreign key are both BuildingId and FloorNumber; NULL NOT ALLOWED

Space(**SpaceId**, SpaceType, Max_capacity, Space_description, SpaceName, Per_seat_cost, Availability, ***BuildingId***, ***FloorNumber***)

Foreign key is both BuildingId and FloorNumber; NULL NOT ALLOWED

User(UserId, FirstName, LastName, ContactNo)

Primary and Foreign key is UserId ; NULL NOT ALLOWED

Freelancer(***UserId***, Email, Identification_number, Identification_type, HouseNo, StreetName, City, Zipcode)

Primary key and Foreign key is UserId in Users table; NULL NOT ALLOWED

CorporatePoc (***UserId***, Corporate_email, SSN)

Primary key and Foreign key is UserId in Users table; NULL NOT ALLOWED

SpaceBooking(SpaceBookingId, Num_seats_booked, BookingTimeStamp, Start_Timestamp, End_Timestamp, ***SpaceId***, ***UserId***)

Foreign key is SpaceId in Spaces table, NULL NOT ALLOWED, Foreign key is UserId in Users table; NULL NOT ALLOWED

Services (**ServiceId**, ServiceType, Cost_per_serivce)

ServiceBooking(***SpaceBookingId***, ***ServiceId***, quantity_requested, service_requested_timestamp)

Foreign key is SpaceBookingId in SpaceBooking table, NULL NOT ALLOWED, Foreign key is ServiceId in Services table; NULL NOT ALLOWED

Invoice(InvoiceNumber, ***SpaceBookingId***, BillingDate, SpaceCost, ServiceCost, TotalBillingCost)

Foreign key is SpaceBookingId in SpaceBooking table, NULL NOT ALLOWED

Corporate(**CorporateId**, CorporateName, Vat_number , UnitNum, BuildingName, StreetName, City, ZipCode, website, ***UserId***)

Foreign key is UserId in CorporatePoc table; NULL NOT ALLOWED

Employees(EmployeeId, FirstName, LastName, CorporateEmail, ***CorporateId***)

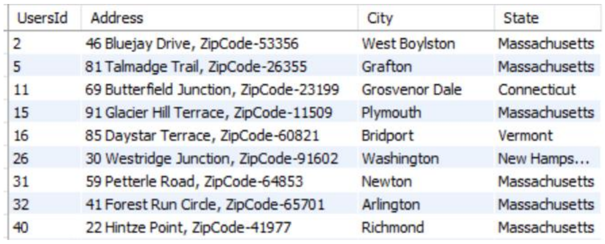
Foreign key is CorporateId in Corporate table; NULL NOT ALLOWED

V. Implementation in MySQL and NoSQL

MySQL Implementation

Query 1: The company wants to send discount offers to all Freelancers who stay in New England Region. Find their Address, City and State details

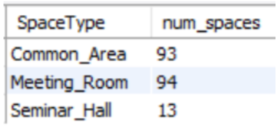
```
SELECT f.UsersId, concat(f.HouseNumber, " ",
f.StreetName, " ", f.StreetSuffix, ", ",
"ZipCode", "-", f.ZipCode) Address,
                 f.City, f.State
FROM freelancer f
WHERE f.state IN ('New Hampshire',
'Maine','Vermont','Massachusetts',
'Rhode Island','Connecticut');
```



UsersId	Address	City	State
2	46 Bluejay Drive, ZipCode-53356	West Boylston	Massachusetts
5	81 Talmadge Trail, ZipCode-26355	Grafton	Massachusetts
11	69 Butterfield Junction, ZipCode-23199	Grosvenor Dale	Connecticut
15	91 Glacier Hill Terrace, ZipCode-11509	Plymouth	Massachusetts
16	85 Daystar Terrace, ZipCode-60821	Bridport	Vermont
26	30 Westridge Junction, ZipCode-91602	Washington	New Hamps...
31	59 Petterle Road, ZipCode-64853	Newton	Massachusetts
32	41 Forest Run Circle, ZipCode-65701	Arlington	Massachusetts
40	22 Hintze Point, ZipCode-41977	Richmond	Massachusetts

Query 2: Number of Spaces in each SpaceType

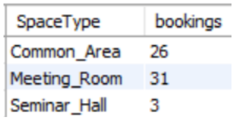
```
SELECT s.SpaceType, COUNT(s.Id) num_spaces
FROM spaces s GROUP BY 1;
```



SpaceType	num_spaces
Common_Area	93
Meeting_Room	94
Seminar_Hall	13

Query 3: Number of Bookings in each Space Type between 1st February 2022 to 30th June 2022

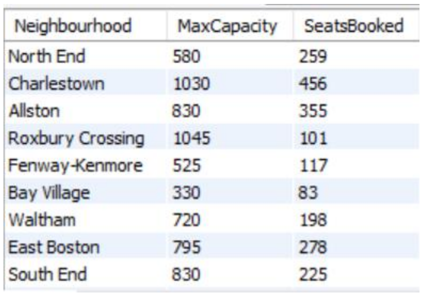
```
SELECT s.SpaceType, COUNT(DISTINCT sb.Id) bookings
FROM spacebooking sb
INNER JOIN spaces s ON s.Id = sb.SpaceId
WHERE DATE(sb.bookingtime) >='2022-02-01'
AND DATE(sb.bookingtime)<'2022-07-01'
GROUP BY s.SpaceType;
```



SpaceType	bookings
Common_Area	26
Meeting_Room	31
Seminar_Hall	3

Query 4: Neighbourhoods with MaxCapacity and Seats Booked

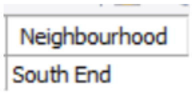
```
SELECT  b2.Neighbourhood,  SUM(n.MaxCapacity)  MaxCapacity,
SUM(n.SeatsBooked) SeatsBooked
FROM
(SELECT   c.BuildingId,   SUM(c.MaxCap)   MaxCapacity,
SUM(c.SeatBooked) SeatsBooked
FROM
(SELECT  b.Id  BuildingId  ,  s.Id  SpaceId,  SUM(s.MaxCapacity)
MaxCap, SUM(sb.NumSeatsBooked) SeatBooked
FROM spaces s
JOIN building b ON b.Id = s.BuildingId
LEFT JOIN spacebooking sb ON sb.SpaceId = s.Id
GROUP BY BuildingId, SpaceId) c
GROUP BY BuildingId) n
JOIN building b2 ON b2.Id = n.BuildingId
 GROUP BY Neighbourhood;
```



Neighbourhood	MaxCapacity	SeatsBooked
North End	580	259
Charlestown	1030	456
Allston	830	355
Roxbury Crossing	1045	101
Fenway-Kenmore	525	117
Bay Village	330	83
Waltham	720	198
East Boston	795	278
South End	830	225

Query 5. In what Neighbourhoods of Boston, does 'Photobug' use the SmartSpaces ?

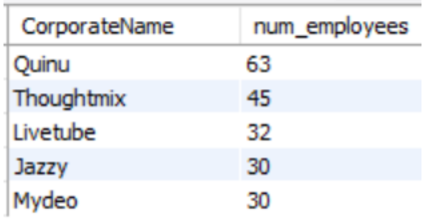
```
SELECT b.Neighbourhood
FROM building b
WHERE b.Id IN
        (SELECT s.BuildingId
        FROM spaces s
        WHERE s.Id IN
                (SELECT sb.SpaceId
                FROM spacebooking sb
                         WHERE sb.UsersId IN (SELECT c.CorporatePocUsersId
                         FROM corporate c
                         WHERE c.CorporateName IN ('Photobug'))));
```



Neighbourhood
South End

Query 6: Find top 5 corporates with maximum number of employees

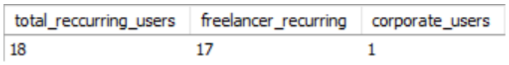
```
SELECT c2.* FROM
(SELECT c.CorporateName, COUNT(e.Id) num_employees
FROM corporate c
JOIN employees e ON e.CorporateId = c.Id
GROUP BY c.CorporateName) c2
WHERE 5>
(SELECT COUNT(*) FROM
(SELECT c.CorporateName, COUNT(e.Id) num_employees
FROM corporate c
JOIN employees e ON e.CorporateId = c.Id
GROUP BY c.CorporateName) c1
WHERE c1.num_employees>c2.num_employees)
ORDER BY c2.num_employees DESC;
```



CorporateName	num_employees
Quinu	63
Thoughtmix	45
Livetube	32
Jazzy	30
Mydeo	30

Query 7: Distribution of Recurring users amongst Freelancers and Corporates

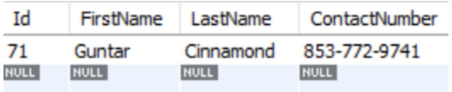
```
SELECT COUNT(DISTINCT sb1.UsersId)
total_reccurring_users,
COUNT(DISTINCT f.UsersId) freelancer_recurring,
             COUNT(DISTINCT cp.UsersId) corporate_users
FROM spacebooking sb1
LEFT JOIN freelancer f ON f.UsersId = sb1.UsersId
LEFT JOIN corporatepoc cp ON cp.UsersId = sb1.UsersId
WHERE EXISTS (SELECT MIN(sb2.BookingTime) min_bookingtime
                        FROM spacebooking sb2
                        WHERE sb1.UsersId = sb2.UsersId
                        HAVING sb1.BookingTime> min_bookingtime)
```



total_reccurring_users	freelancer_recurring	corporate_users
18	17	1

Query 8: Find Users who have booked all 3 types of Spaces

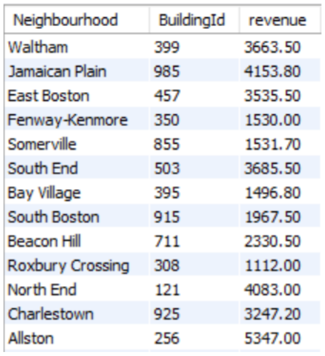
```
SELECT u2.*
FROM users u2
WHERE NOT EXISTS
        (SELECT *
        FROM spaces s2
        WHERE NOT EXISTS
                (SELECT u.Id UserId, s.SpaceType
                FROM spacebooking sb
                JOIN spaces s ON s.Id = sb.SpaceId
                JOIN users u ON u.Id = sb.UsersId
                WHERE s2.SpaceType = s.SpaceType
                AND u.Id = u2.Id));
```



Id	FirstName	LastName	ContactNumber
71	Guntar	Cinnamond	853-772-9741
NULL	NULL	NULL	NULL

Query 9: What are the buildings in the neighborhoods with maximum revenue

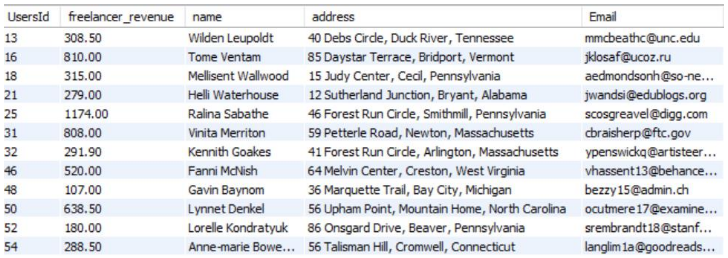
```
SELECT b.Neighbourhood, s.BuildingId, SUM(i.TotalBillingCost)
revenue
FROM invoice i
JOIN spacebooking sb ON sb.Id = i.SpaceBookingId
JOIN spaces s ON s.Id = sb.SpaceId
JOIN building b On b.Id = s.BuildingId
GROUP BY b.Neighbourhood, s.BuildingId
HAVING revenue>= ALL ( SELECT t.building_rev FROM
        (SELECT b.Neighbourhood, s.BuildingId,
         SUM(i.TotalBillingCost) building_rev FROM invoice i
        JOIN spacebooking sb ON sb.Id = i.SpaceBookingId
        JOIN spaces s ON s.Id = sb.SpaceId
        JOIN building b On b.Id = s.BuildingId
        GROUP BY b.Neighbourhood, s.BuildingId) t
                        WHERE t.Neighbourhood = b.Neighbourhood )
```



Neighbourhood	BuildingId	revenue
Waltham	399	3663.50
Jamaican Plain	985	4153.80
East Boston	457	3535.50
Fenway-Kenmore	350	1530.00
Somerville	855	1531.70
South End	503	3685.50
Bay Village	395	1496.80
South Boston	915	1967.50
Beacon Hill	711	2330.50
Roxbury Crossing	308	1112.00
North End	121	4083.00
Charlestown	925	3247.20
Allston	256	5347.00

Query 10: Find all freelancers that have generated more revenue than any of the corporates

```
SELECT f.UsersId, SUM(i.TotalBillingCost)
freelancer_revenue,
CONCAT(u.FirstName, " ",u.LastName) name,
CONCAT(f.HouseNumber, " ", f.Streetname, "
",
f.StreetSuffix, ", ", f.City, ", ", f.state)
address, f.Email, u.ContactNumber
FROM users u
JOIN freelancer f ON f.UsersId = u.Id
JOIN spacebooking sb ON sb.UsersId = u.Id
JOIN invoice i ON i.SpaceBookingId = sb.Id
GROUP BY 1
HAVING freelancer_revenue > ANY
        (SELECT SUM(i.TotalBillingCost) rev
        FROM users u
        JOIN corporatepoc cp ON cp.UsersId = u.Id
        JOIN spacebooking sb ON sb.UsersId = u.Id
        JOIN invoice i ON i.SpaceBookingId = sb.Id
        GROUP BY u.Id);
```



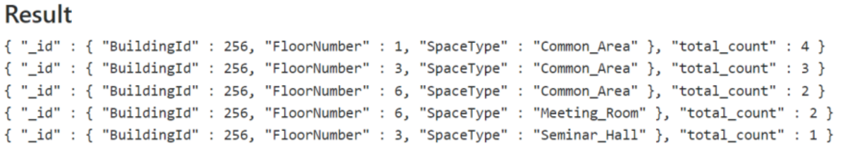
UsersId	freelancer_revenue	name	address	Email
13	308.50	Wilden Leupoldt	40 Debs Circle, Duck River, Tennessee	mmcbeathc@unc.edu
16	810.00	Tome Ventam	85 Daystar Terrace, Bridport, Vermont	jklosaf@ucoz.ru
18	315.00	Mellisent Wallwood	15 Judy Center, Cecil, Pennsylvania	aedmondsonh@so-ne...
21	279.00	Helli Waterhouse	12 Sutherland Junction, Bryant, Alabama	jwandsi@edublogs.org
25	1174.00	Ralina Sabathe	46 Forest Run Circle, Smithmill, Pennsylvania	scosgreavel@digg.com
31	808.00	Vinita Merriton	59 Petterle Road, Newton, Massachusetts	cbraisherp@ftc.gov
32	291.90	Kennith Goakes	41 Forest Run Circle, Arlington, Massachusetts	ypenswickq@artisteer...
46	520.00	Fanni McNish	64 Melvin Center, Creston, West Virginia	vhassent13@behance...
48	107.00	Gavin Baynom	36 Marquette Trail, Bay City, Michigan	bezzy15@admin.ch
50	638.50	Lynnet Denkel	56 Upham Point, Mountain Home, North Carolina	ocutmere17@examine...
52	180.00	Lorelle Kondratyuk	86 Onsgard Drive, Beaver, Pennsylvania	srembrandt18@stanf...
54	288.50	Anne-marie Bowe...	56 Talisman Hill, Cromwell, Connecticut	langlim1a@goodreads...

NoSQL Implementation

Spaces, Freelancer and Invoice Collections were made on MongoDb

Q1: Total number of spaces in BuildingId 256 w.r.t Floor Number and SpaceType.

```
db.spaces.aggregate([
    {$project : {"BuildingId":1, "FloorNumber":1, "SpaceType":1, "_id":0}},
    {$match : {'BuildingId':{$in:[256]}}},
    {$group : {"_id":{"BuildingId":"$BuildingId","FloorNumber": "$FloorNumber","SpaceType":"$SpaceType"},
    "total_count":{$sum:1}}},
    {$sort:{"total_count":-1}}
    ])
```

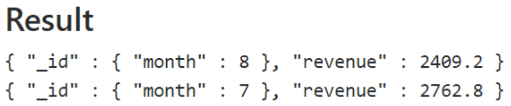


Result

```
{ "_id" : { "BuildingId" : 256, "FloorNumber" : 1, "SpaceType" : "Common_Area" }, "total_count" : 4 }
{ "_id" : { "BuildingId" : 256, "FloorNumber" : 3, "SpaceType" : "Common_Area" }, "total_count" : 3 }
{ "_id" : { "BuildingId" : 256, "FloorNumber" : 6, "SpaceType" : "Common_Area" }, "total_count" : 2 }
{ "_id" : { "BuildingId" : 256, "FloorNumber" : 6, "SpaceType" : "Meeting_Room" }, "total_count" : 2 }
{ "_id" : { "BuildingId" : 256, "FloorNumber" : 3, "SpaceType" : "Seminar_Hall" }, "total_count" : 1 }
```

Q2: What is the revenue generated in the months of July and August ?

```
db.invoice.aggregate(
    {$project:{'month': { $month: "$iso_date" }, 'TotalBillingCost':1, '_id':0}},
     {$group : {'_id':{'month':'$month'}, 'revenue':{$sum:"$TotalBillingCost"}}})
```

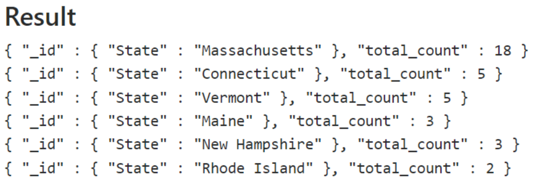


Result

```
{ "_id" : { "month" : 8 }, "revenue" : 2409.2 }
{ "_id" : { "month" : 7 }, "revenue" : 2762.8 }
```

Q3: How many Freelancers are originally from New England?

```
db.freelancer.aggregate
({$project : {"State":1,"_id":0}},
 {$match : {"State":{$in:['New Hampshire',
'Maine','Vermont',
 'Massachusetts','Rhode Island','Connecticut']}}},
  {$group : {"_id":{"State":"$State"},
"total_count":{$sum:1}}},
  {$sort : {"total_count":-1}})
```



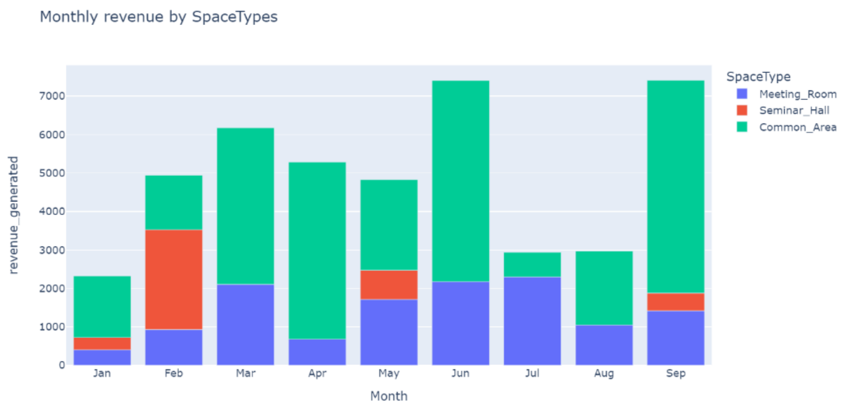
Result

```
{ "_id" : { "State" : "Massachusetts" }, "total_count" : 18 }
{ "_id" : { "State" : "Connecticut" }, "total_count" : 5 }
{ "_id" : { "State" : "Vermont" }, "total_count" : 5 }
{ "_id" : { "State" : "Maine" }, "total_count" : 3 }
{ "_id" : { "State" : "New Hampshire" }, "total_count" : 3 }
{ "_id" : { "State" : "Rhode Island" }, "total_count" : 2 }
```

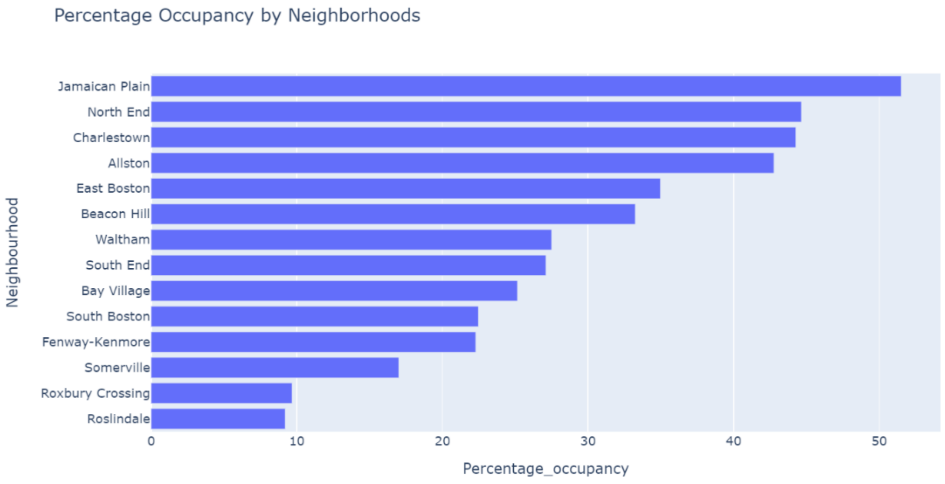
VI. Application & Analyics (Python)

The database was accessed using Python and connected through a function made using PyMySQL library. The fetched data was in dataframe format and after performing some data wrangling, visualization was created using Plotly express and Geopandas library for Stacked, Horizontal charts and Boston map respectively.

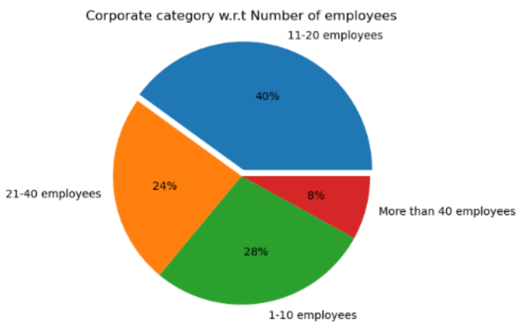
1. Monthly revenue generated w.r.t SpaceTypes



2. Percentage Occupancy by Neighborhoods

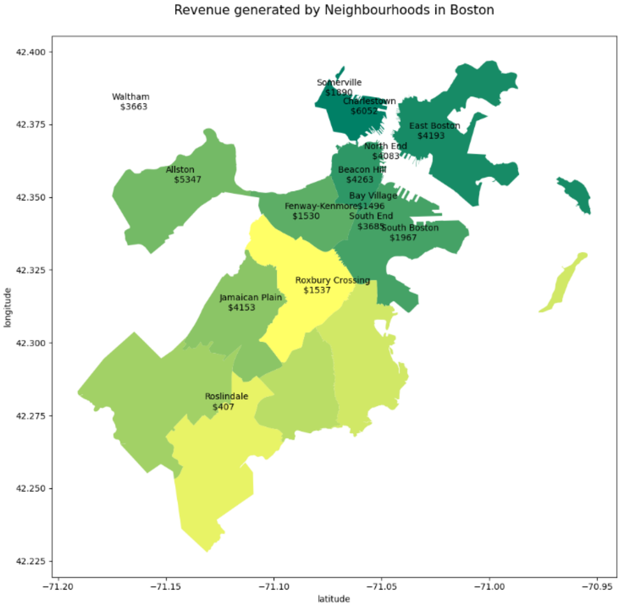


3. Corporate Category w.r.t Number of employees



[2]

4. Revenue generated by Neighborhoods in Boston



VII. Summary and Recommendation

The ‘SmartSpaces’ database allows the organization to understand the areas across Boston which have the capacity to expand as well as generate revenue and become more profitable. The coworking management system also helped in identifying the types of recurring users and how can we target them using targeted marketing activities. We also recognized that ‘Common Area’ are the most preferred type of Space, and through previous queries and analysis submitted, we gathered that ‘8-hour parking’ is one of the most sought-after services. The enterprise is recommended to expand their business in areas which have ample space for long hours parking as well as build enough Common Spaces.

While many requirements of the ‘SmartSpaces’ have been covered in this Coworking Space Management system, we can further improve this system by integrating triggers to automatically calculating the total billing amount using PerSeatCost and SeatsBooked. Stored Procedures can be used to update the PerSeatCost of the spaces as well as adding discounts. Adding new Corporates and updating CorporatePoc could be further automated using stored procedures.

While MongoDb helped in efficiently capturing straight forward problem statements, a lot of research has to be done in understanding how to join collections and solve complex questions. Since this database can easily expand, NoSQL can be beneficial the data as Map Reduce can assist in concurrent processing of the data. However, given the present problem statement, we can continue with the relational model and further improve its processes and add more complex functionalities.

VIII. References

[1] G. W. E. J. M. McCoy, "The Potential Role of the Physical Environment in Fostering Creativity," *Creativity Research Journal*, vol. 14, p. 19, 2002.

[2] N. C. a. G. L. A. Fuzi, "New in-house organizational spaces that support creativity and innovation: coworking pace," in *R & D Management Conference*, Stuttgart, 2014.