



ONE CENTER

PROPERTY SOLUTIONS

Building space rental
Database Management System

CIS 9340 FINAL PROJECT

SUBMITTED BY:
NOYONIKA ROY
RIDWIK MODI
JAYANT BISHNOI
YOOEUN PARK
AYUSH VYAS

TABLE OF CONTENTS

| | |
|--|-----------|
| Business Scenario | 2 |
| ER Model..... | 3 |
| Conversion to Relational Model..... | 5 |
| Normalization..... | 6 |
| Creating the Database..... | 14 |
| Database Application..... | 17 |
| Conclusions..... | 26 |

Business Scenario

One Centre Property Solutions owns and provides rented spaces for commercial uses such as Restaurants, offices (private, public), BPO's, shops, cafes, movie theatre, gym, banks, and ATMs etc.

Till now we have been using paper logbooks, spreadsheets, notes and other ways to keep track of all records that are required in this business. Now, we are building a database system for our business to have an organized set of data and to remove manual tracking systems. This will help in removing manual error, improving accuracy and will also help us in organizing a big set of data.

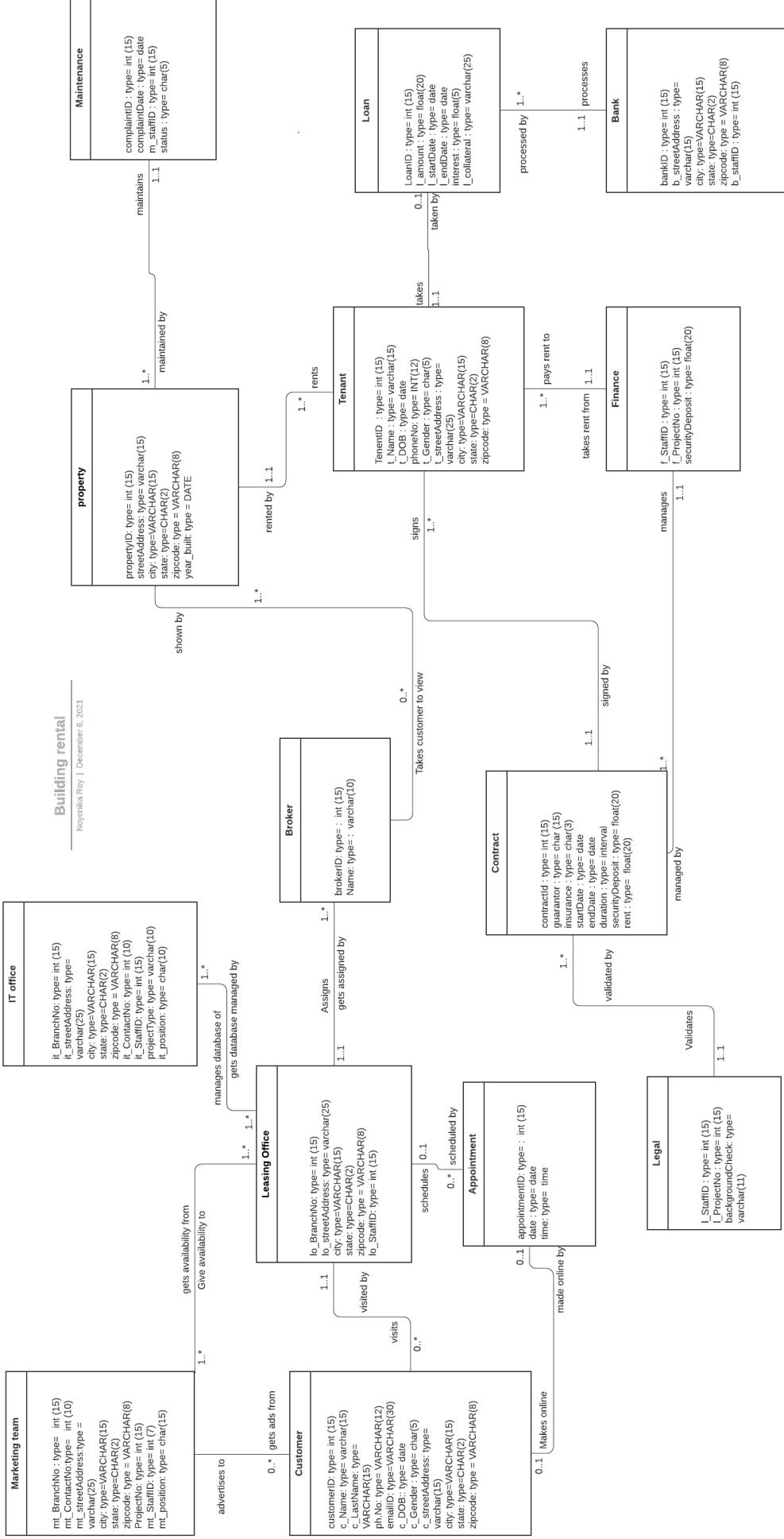
In our business, customers can rent one of our properties according to their needs. Customers can visit our leasing office or book an appointment online through which they get assigned a broker who will help them look for a property with the specifications as per their requirements. We need to keep track of customer records and their information along with their requirements. Once the customer selects the desired property. Our dedicated teams in each department will help in completing all the formalities regarding this matter. A contract is drawn up where tenant information is updated along with other renting details. We also keep track of all the employees and data coming across the finance team, marketing team, IT team, building maintenance team, leasing office team and legal team.

Once the customer rents a property, we need to track the finances (rent and loans) which is handled by our Finance Team. Each property is up to code and is maintained by our dedicated Maintenance Team. Our in-house Legal team manages contracts and services, and we also have a Marketing team that advertises our properties to potential new customers.

Challenges during setting up the company concept:

The purpose of our company is to render a rental service. The main challenge here was to imitate the working of a real company. After a fair amount of research on rental companies and through brainstorming we came up with a plan that emulates the working of an actual company and what data would be required to be kept track of in such an institution. We even went a little further to include the marketing team as we figured that would also be integral to a business.

ER Model using UML Notation



Relationship sentences:

1. Marketing team must get availability from one or many Leasing offices
2. Leasing office must give availability to one or many marketing teams
3. Marketing team may advertise to none to many customers
4. Customer may get ads from none to many Marketing teams
5. Customer may visit one and only one Leasing office
6. Leasing office may be visited by none to many customers
7. IT office must manage database of one or more leasing offices
8. Leasing office must get their database managed by one or more IT offices
9. Customer may make zero or 1 appointment online
10. Appointment may be made online by zero or 1 customer
11. Leasing office may schedule zero to many appointments
12. Appointment may be scheduled by zero to 1 leasing office
13. Leasing office must assign one to many brokers
14. Broker must be assigned by one and only 1 leasing office
15. Broker must take customer to view one or more properties
16. Property may be shown by zero to many brokers
17. Property must be maintained by 1 and only 1 maintenance team
18. Maintenance team must maintain one or more properties
19. Property must be rented by one or more tenants
20. Tenant must rent one and only one property
21. Tenant must sign one and only one contract
22. Contract must be signed by one or more tenants
23. Legal must validate one or more contracts
24. Contract must be validated by one and only one Legal team
25. Contract must be managed by one and only one finance team
26. Finance team must manage one or more contracts
27. Tenant must pay rent to one and only one finance team
28. Finance team must take rent from one or many tenants
29. Tenant may take zero to one loan
30. Loan may be taken by one and only one tenant
31. Loan must be processed by one and only one bank
32. Bank must process one to many loans

Challenges faced while making the ER diagram:

The ER diagram is a representative tool of conceptual modeling and easily represents the relationship with data existing in the real world. We had the theme of building rental, and there was no difficulty in identifying entities after we wrote out our business scenario. In addition, in the process of showing the relationship between entities we established the multiplicities. However, as the number of entities increased, we needed time in identifying the correlations between them and keeping track of the cardinality.

Conversion to Relational Model

Marketing Team(mt_BranchNo, PropertyID(fk), mt_ContactNo, mt_streetAddress, city, state, zipcode, ProjectNo, mt_StaffID, mt_position)
Customer(customerID, c_Name, c_DOB, c_Gender, c_streetAddress, city, state, zipcode)
Leasing Office(lo_BranchNo, lo_StaffID ,lo_streetAddress, city, state, zipcode, contractID(fk))
IT Office(it_BranchNo, , it_StaffID, it_streetAddress, city, state, zipcode, it_ContactNo, projectType, it_position)
Broker(brokerID, Name, PropertyID(fk))
Appointment(appointmentID, Date, Time, customerID(fk))
Property(propertyID, streetAddress, city, state, zipcode, brokerID(fk),year_built)
Maintenance(complaintID, complaintDate, m_staffID, status, propertyID(fk))
Legal(l_staffID, l_ProjectNo, contactID(fk),backgroundCheck)
Contract(contractID, guarantor, insurance, stateDate, endDate, duration, tenantID(fk), propertyID(fk), l_Staffid(fk), f_StaffID(fk), securityDeposit, rent)
Tenant(TenantID, t_Name, t_DOB, t_Gendor, t_Address, propertyID(fk))
Loan(LoanID, contractID(fk), l_amount, l_startDate,l_endDate, interest, l_collateral,)
Finance(f_StaffID,f_ProjectNo, loanID(fk),contractID(fk),securityDeposit)
Bank(bankID, bankID(fk), b_Address, b_staffID, loanID(fk))

Challenges while converting to relational model:

Our team has faced certain challenges while converting the conceptual ER model into the logical relational model. The relation sentences get complex as it involves the situation that exists between two database tables when one table has a foreign key that references the primary key of the other table. Every department in the company had a unique identification with an independent existence to be differentiated with, it comes with a challenge to relate the unique keys to each other.

We needed to keep track of employees in each department, and there were multiple departments in our company. Most of the department has entities and attributes that are dependent on the other departments. Customer, broker, leasing office, and other departments can co-exist in multiple departments, so to arrange all the relations in a meaningful manner is defined in the relationship sentences. Each relation is defined with minimum or maximum cardinality which should make sense.

Normalization

- 1) Marketing Team(mt_BranchNo, PropertyID(fk), mt_contactNo, mt_buildingNo, mt_street, mt_city, mt_state, mt_zipcode, mt_StaffID, mt_position)

| mt_StaffID | mt_position | mt_BranchNo | mt_ContactNo | mt_streetAddress | state | city | zipcode | mt_BranchNo | propertyID | mt_StaffID |
|------------|----------------------|-------------|--------------|------------------|----------|----------|---------|-------------|------------|------------|
| 1230 | Manager | 123 | 3456789276 | 345 50th street | New York | New York | 10010 | 123 | 456 | 1230 |
| 1231 | Sales representative | 123 | 3456789276 | 346 50th street | New York | New York | 10010 | 123 | 457 | 1231 |
| 1232 | Sales representative | 123 | 3456789276 | 347 50th street | New York | New York | 10010 | 123 | 458 | 1232 |
| 1233 | Sales representative | 123 | 3456789276 | 348 50th street | New York | New York | 10010 | 123 | 459 | 1233 |
| 1234 | Sales representative | 123 | 3456789276 | 349 50th street | New York | New York | 10010 | 123 | 460 | 1234 |

Key: mt_BranchNo, propertyID

FD: (Functional Dependency)

FD1: mt_BranchNo → mt_contactNo, mt_streetAddress, mt_city, mt_state, mt_zipcode

FD2: mt_BranchNo, propertyID → mt_StaffID

FD3: mt_StaffID → mt_position

1NF: Since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF: Partial dependency FD1

Making new entities: MKT Branch Loc and Marketing Team

Entities in 2NF :

- a) Marketing Team(mt_BranchNo, PropertyID(fk), mt_StaffID, mt_position)

FD4: mt_BranchNo, propertyID → mt_StaffID

FD5: mt_StaffID → mt_position

| mt_BranchNo | propertyID | mt_StaffID |
|-------------|------------|------------|
| 123 | 456 | 1230 |
| 123 | 457 | 1231 |
| 123 | 458 | 1232 |
| 123 | 459 | 1233 |

- b) MKT Branch Loc(mt_BranchNo, mt_contactNo, mt_streetAddress, mt_city, mt_state, mt_zipcode)

FD6: mt_BranchNo → mt_contactNo, mt_streetAddress, mt_city, mt_state, mt_zipcode

| mt_BranchNo | mt>ContactNo | mt_streetAddress | state | city | zipcode |
|-------------|--------------|------------------|----------|----------|---------|
| 123 | 3456789276 | 345 50th street | New York | New York | 10010 |
| 123 | 3456789276 | 346 50th street | New York | New York | 10010 |
| 123 | 3456789276 | 347 50th street | New York | New York | 10010 |
| 123 | 3456789276 | 348 50th street | New York | New York | 10010 |

3NF: transitive dependency in FD5

New entity: Mkt_Staff

| mt_StaffID | mt_position |
|------------|----------------------|
| 1230 | Manager |
| 1231 | Sales representative |
| 1232 | Sales representative |
| 1233 | Sales representative |

FD7: mt_StaffID → mt_position

Entities in 3NF:

- a) Marketing Team(mt_BranchNo, PropertyID(fk), mt_StaffID)
- FD7: mt_BranchNo, propertyID → mt_StaffID**
- b) Mkt_Staff(mt_StaffID, mt_position)
- c) **FD8: mt_StaffID → mt_position**

- 2) IT Office(it_BranchNo, it_streetAddress, City, State, Zip code, it_ContactNo, it_StaffID, projectType, it_position)

| it_BranchNo | it_streetAddress | city | state | zipcode | it_ContactNo | it_StaffID | projectType | it_Position | it_BranchNo |
|-------------|-------------------|-------------|----------|---------|--------------|------------|---------------------|-------------|-------------|
| 101 | 55 Wright Ave | Westchester | New York | 17008 | 8989762534 | 10111 | Database management | Manager | 101 |
| 201 | 785 Dacsun Street | Brooklyn | New York | 15008 | 2332456567 | 10122 | Support | Manager | 101 |

Key: it_BranchNo, it_StaffID

FD1: $it_branchNo \rightarrow it_streetAddress, City, State, Zip\ code, it_ContactNo$

FD2: $it_StaffID \rightarrow projectType, it_position$

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF : Partial dependency in FD1 and FD2

Making new entity: IT Branch Loc and IT Staff

Entities in 2NF :

a) IT Staff(it_staffID, projectType, it_position, it_BranchNo(fk))

| it_StaffID | projectType | it_Position | it_BranchNo |
|------------|---------------------|-------------|-------------|
| 10111 | Database management | Manager | 101 |
| 10122 | Support | Manager | 101 |
| 10133 | Marketing IT | Manager | 101 |

FD3: $it_StaffID \rightarrow projectType, it_position, it_BranchNo$

b) IT Branch Loc(it_branchNo, it_streetAddress, City, State, Zip code, it_ContactNo)

| A | B | C | D | E | F |
|-------------|-------------------|-------------|----------|---------|--------------|
| it_BranchNo | it_streetAddress | city | state | zipcode | it_ContactNo |
| 101 | 55 Wright Ave | Westchester | New York | 17008 | 8989762534 |
| 201 | 785 Dacsun Street | Brooklyn | New York | 15008 | 2332456567 |

FD4 : $it_branchNo \rightarrow it_streetAddress, City, State, Zip\ code, it_ContactNo$

3NF:

No transitive dependency so they are already in 3NF

3) Customer(CustomerID, c_Name, c_DOB, c_Gender, c_streetAddress, city, state, pincode)

Key : CustomerID

FD1: CustomerID → *c_Name, c_DOB, c_Gender, c_Address*

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF and 3NF:

No partial dependencies and no transitive dependencies

- 4) Broker(BrokerID, Name)

Key: BrokerID

FD1: BrokerID → *Name*

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF and 3NF:

No partial dependencies and no transitive dependencies

- 5) Appointment(AppointmentID, Date, Time, CustomerID(fk))

Key: AppointmentID

FD1: AppointmentID → *Date, Time, CustomerID*

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF and 3NF:

No partial dependencies and no transitive dependencies

- 6) Property(PropertyID, streetAddress, City, State, Pincode, brokerID, year_built)

Key: PropertyID

FD1: PropertyID → *streetAddress, City, State, Pincode, brokerID, year_built*

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF and 3NF:

no partial dependencies or transitive dependencies

- 7) Legal Team (I_staffID , I_ProjectNo ,contractID (fk) , backgroundCheck)

Key: I_staffID

FD1: I_staffID → I_ProjectNo

FD2: contractID (fk) → backgroundCheck

1NF : all column values are atomic and consistent in 1NF

2NF and 3NF : no partial or transitive dependencies.

- 8) Contract(contractID, Guarantor, insurance, startDate, endDate, duration, tenantID(fk), propertyID(fk), securityDeposit, rent)

Key: contractID

FD1: contractID → startDate, endDate, duration, tenantID(fk)

FD2: propertyID → securityDeposit, rent

1NF: since the columns have atomic values and have consistent data types we can say it's in 1NF

2NF and 3NF: no partial dependencies or transitive dependencies

- 9) Tenant(TenantID, t_Name, t_DOB, phoneNo, t_Gender, t_streetAddress, city, state, zipcode, propertyID(fk))

Key: TenantID

FD1: TenantID → t_Name, t_DOB, phoneNo, t_Gender, t_streetAddress, city, state, zipcode, propertyID(fk)

- 10) Finance(f_StaffID,f_ProjectNo, loanID(fk),contractID(fk),securityDeposit, rent)

Key: f_StaffID

FD1: f_StaffID → f_ProjectNo, loanID, l_amount, contractID, securityDeposit

1NF: Meets the definition of a relation

2NF and 3NF: No partial or transitive dependencies

11) Bank(bankID, b_streetAddress, city, state, zipcode b_staffID)

| bankId | b_staffID | bankId | b_address | city | state | zipcode |
|--------|-----------|--------|----------------------|----------|-------|---------|
| 4567 | 4607 | 4567 | 6 Washington Str | New York | NY | 10010 |
| 4568 | 4608 | 4568 | 77 Flamingo Street | New York | NY | 10006 |
| 4569 | 4609 | 4569 | 90 west 3rd street | New York | NY | 10089 |
| 4570 | 4610 | 4570 | 1010 hartfelt street | Brooklyn | NY | 11220 |

Key: bankID, bstaffID

FD1: bankID → b_streetAddress, city, state, zipcode

FD2: bankID → bstaffID

1NF: Meets the definition of a relation

2NF: Partial dependency exists

2 new entities created

a) Bank_staff(bankID, bstaffID)

| bankId | b_staffID |
|--------|-----------|
| 4567 | 4607 |
| 4568 | 4608 |
| 4569 | 4609 |
| 4570 | 4610 |

FD3: bankID → bstaffID

b) Bank_branch(bankID, b_streetAddress, city, state, zipcode)

| bankID | b_address | city | state | zipcode |
|---------------|-----------------------|-------------|--------------|----------------|
| 4567 | 166 Washington Street | New York | NY | 10010 |
| 4568 | 377 Flamingo Street | New York | NY | 10006 |

FD4: bankID → b_streetAddress, city, state, zipcode

- 12) Loan(LoanID, contractID, l_amount, l_startDate, l_endDate, interest, l_collateral, bankID(fk))

Key: LoanID, bankID

FD1: LoanID, bankID → contractID, l_amount, l_startDate, l_endDate, interest, l_collateral

The entity is in 3NF

- 13) Maintenance (complaintID, complaintdate, m_staffID, status)

FD1 : complaintID, complaintdate, m_staffID, status

1NF: Meets the definition of a relation

2NF: No partial dependencies

3NF: No transitive dependencies

- 14) Leasing office (lo_BranchNo, lo_Address, City, state, Pincode, lo_StaffID, contractID)

| lo_StaffID | contractID | lo_BranchNo | lo_BranchNo | lo_Address | city | state | zipcode |
|-------------------|-------------------|--------------------|--------------------|-------------------|-------------|--------------|----------------|
| 281919 | OJ4714 | 8543 | 8543 | 9000 high street | New York | NY | 10090 |
| 283866 | KO7010 | 8602 | 8602 | 44 Parkway ave | New York | NY | 10005 |

Key: lo_BranchNo, lo_StaffID

FD1: staffID → contractID

FD2: lo_BranchNo → lo_Address, City, State, Pincode

1NF: Meets the definition of a relation

2NF: partial dependencies exist

2 new entities Lo_Staff and Lo_Branch

a) Lo_Staff(staffID, contractID)

| lo_StaffID | contractID | lo_BranchNo |
|------------|------------|-------------|
| 281919 | OJ4714 | 8543 |
| 283866 | KO7010 | 8602 |
| 281920 | WH2020 | 8543 |

b) Lo_Branch(lo_BranchNo, lo_Address,City,State, Pincode)

| lo_BranchNo | lo_Address | city | state | zipcode |
|-------------|------------------|----------|-------|---------|
| 8543 | 9000 high street | New York | NY | 10090 |
| 8602 | 44 Parkway ave | New York | NY | 10005 |

They are now in 3NF

Final relations:

MKT Branch Loc(mt_BranchNo, mt_contactNo, mt_streetAdress, mt_city, mt_state, mt_zipcode)

Marketing Team(mt_BranchNo, PropertyID(fk), mt_StaffID)

Mkt_Staff(mt_StaffID, mt_position)

IT Staff(it_staffID, projectType, it_position)

IT Branch Loc(it_branchNo, it_streetAddress, City, State, Zip code, it_ContactNo)

Customer(CustomerID, c_Name,c_DOB, c_Gender, c_streetAddress, city, state, pincode)

Broker(BrokerID, Name)

Appointment(AppointmentID, Date, Time, CustomerID(fk))

Property(PropertyID, streetAddress, City, State, Pincode, brokerID, year_built)

Legal Team (l_staffID , l_ProjectNo ,contractID (fk) , backgroundCheck)

Contract(contractID, Guarantor, insurance, startDate, endDate, duration, tenantID(fk), propertyID(fk), securityDeposit, rent)

Finance(loanID, l_amount, contractID, securityDeposit,f_projectNo,f_StaffID)

Bank_branch(bankID, b_streetAddress, city, state, zipcode)

Bank_staff(bankID, bstaffID)

Loan(LoanID, contractID, l_amount, l_startDate,l_endDate, interest, l_collateral, bankID(fk))

Maintenance (complaintID, complaintdate , m_staffID, status)

Lo_Staff(staffID, contractID)

Lo_Branch(lo_BranchNo, lo_Address,City,State, Pincode)

Challenges during normalization:

One of the major challenges during normalization was trying to figure out whether to import data to access and then normalize or normalize in excel and then export to Access. We ended up doing the latter. It was time consuming to look at all the separate entities and to apply rules of normalization one-by-one. To minimize redundancy and ensure absolute primary key dependency of the elements of a table.

Moreover we realized that some attributes that were earlier not present needed to be added, so we appended them to the table and had to modify our ER diagram accordingly. The presence of elements that are related to one another but not partially or transitively dependent on any part of the primary key presented a dilemma. So, we sought help from similar questions from the Homework and were able to come up with appropriate solutions.

Creating the Database

The entire database was created on excel first before importing it to Microsoft Access. We made sure the data we entered was sufficient for each attribute as well as entity. The motive was to keep the data precise and to the point. For example, in the entity Broker, we just added two attributes, BrokerID and name as that was sufficient for the requirement of the dataset. Whereas, in the Tenant entity we added more than 5 attributes. It is necessary for the entity to have details like, name, dob, phone number and address to create valuable and resourceful information. After we added data in excel and then we eventually imported it to Access.

Creating tables using SQL:

```
CREATE TABLE Customer
```

```

CustomerID  VARCHAR(10) NOT NULL,
c_Name      VARCHAR(35),
c_LastName   VARCHAR(35),
phNo        VARCHAR(15),
emailID     VARCHAR(35),
c_DOB        DATE,
c_streetAddress  VARCHAR(35),
city         CHAR(15),
state        CHAR(2),
zipCode      VARCHAR(8),
c_Gender     VARCHAR(2),
CONSTRAINT pk_customer
PRIMARY KEY (CustomerID)
```

```
CREATE TABLE Appointment
```

```

(
AppointmentID  VARCHAR(10) NOT NULL,
"date"        DATE,
"time"        DATETIME,
CustomerID    VARCHAR(10) NOT NULL,
CONSTRAINT pk_appointment
PRIMARY KEY (AppointmentID)
)
```

Our actual database in MS Access:

| customerID | c_Name | c_lastName | phNo | emailID | c_DOB | c_Gender | c_streetAddress | city | state | zipcode |
|------------|-------------|------------|------------|-------------------|--------------|--------------------------|------------------|------|-------|---------|
| 24001 | Kole | Jimenez | 974977843 | Kole@gmail.cor | 9/13/1964 F | 334 Fifth Dr. | Brooklyn | NY | | 11213 |
| 24002 | Lilianna | Pierce | 9426712539 | Lilianna@gmail.co | 7/4/1948 M | 7830 Lilac Court Meadows | | NY | | 11365 |
| 24003 | Gina | Chavez | 9668554394 | Gina@gmail.co | 1/3/1988 F | 8830 Grove Driv | Bronx | NY | | 10456 |
| 24004 | Mckenna | Collier | 9360349686 | Mckenna@gma | 10/28/1954 M | 9908 Newbridge | Brooklyn | NY | | 10034 |
| 24005 | Vanessa | Patel | 9884241483 | Vanessa@gmail | 5/7/1962 F | 8826 Purple Finc | East Elmhurst | NY | | 11370 |
| 24006 | Sebastian | Marks | 9793208577 | Sebastian@gma | 5/16/1988 M | 9694 North You | Spring Valley | NY | | 10977 |
| 24007 | Kayleigh | Houston | 9844772550 | Kayleigh@gmail | 3/21/1942 M | 8512 NE. Mill Di | Fairport | NY | | 4450 |
| 24008 | Javier | Gates | 9387573310 | Javier@gmail.cc | 10/3/1963 M | 7329 Bayport Av | Brooklyn | NY | | 11230 |
| 24009 | Allie | Logan | 9757571418 | Allie@gmail.cor | 2/15/1972 F | 673 Longfellow | New York | NY | | 10033 |
| 24010 | Jaxson | Porter | 9168646988 | Jaxson@gmail.c | 9/22/1997 M | 9407 Beach Av | Brooklyn | NY | | 11207 |
| 24011 | Ahung | Choi | 9044185078 | Ahung@gmail.c | 9/29/1975 F | 555 Amber road | Long Island City | NY | | 11009 |
| 24012 | Allessandra | Slovensky | 9397171597 | Allessandra@gn | 5/16/1965 F | 71 riverside driv | New York | NY | | 10027 |
| 24013 | Preeti | Chauhan | 9702244723 | Preeti@gmail.cc | 4/23/1982 F | 8888 pickering s | Brooklyn | NY | | 13008 |
| 24014 | Sylvia | Mathews | 9198302009 | Sylvia@gmail.cc | 4/5/1971 F | 898 Walters roa | Queens | NY | | 11008 |
| 24015 | Peter | Mallark | 9656335964 | Peter@gmail.cc | 6/11/1994 M | 9090 Semfrod S | New York | NY | | 110019 |
| 24016 | Aria | Stark | 9369563856 | Aria@gmail.con | 5/31/1986 F | 30 Newport Par | Jersey City | NJ | | 7310 |
| 24017 | Jeevansh | Jotish | 9103074554 | Jeevansh@gmai | 9/15/1956 M | 20 River drive Sc | Jersey City | NJ | | 7310 |
| 24018 | Terry | Crews | 9538530763 | Terry@gmail.co | 10/14/1979 M | 565 Grove Street | Jersey City | NJ | | 7311 |
| 24019 | Alba | Kronks | 9811033107 | Alba@gmail.cor | 3/5/1998 F | 456 Turnpike | Edison | NJ | | 17890 |
| 24020 | Sylvester | Troy | 9688075074 | Sylvester@gmai | 8/9/1998 M | 566 Pleet Street | Wycoff | NJ | | 17556 |
| 24021 | Winona | Mattox | 9299872067 | Winona@gmail. | 12/19/1998 M | 421 W. Bay Dr | San Antonio | TX | | 78245 |
| 24022 | Jaqeline | Akers | 9341548559 | Jaqeline@gma | 9/17/1951 F | 94 North Franklin | Arlington | TX | | 76014 |
| 24023 | Daria | Randle | 9602618465 | Daria@gmail.cc | 12/29/1995 F | 972 Ann Drive | Houston | TX | | 77008 |
| 24024 | Rosalva | Bertram | 9859693847 | Rosalva@gmail. | 5/29/1978 F | 10 Carriage Stre | Plano | TX | | 75025 |
| 24025 | Robbyn | Vu | 9651637635 | Robbyn@gmail. | 8/14/1979 M | 1 Alderwood Dr | Baytown | TX | | 77520 |
| 24026 | William | Hawk | 9465484659 | William@gmail. | 1/14/1962 M | 864 Littleton St | Mission | TX | | 78572 |
| 24027 | Randy | Danner | 9243873016 | Randy@gmail.c | 2/3/1976 M | 9158 Dogwood | Anaheim | CA | | 92804 |
| 24028 | Keren | Sandlin | 9284299923 | Keren@gmail.cc | 1/23/1972 F | 56 River Court | Chula Vista | CA | | 91911 |
| 24029 | Elenor | Hidalgo | 9436649827 | Elenor@gmail.c | 7/28/1982 F | 8602 Pheasant I | La Puente | CA | | 91744 |
| 24030 | Pa | Worth | 9798766989 | Pa@gmail.com | 7/19/1984 M | 79 Griffin St | Oxnard | CA | | 93033 |

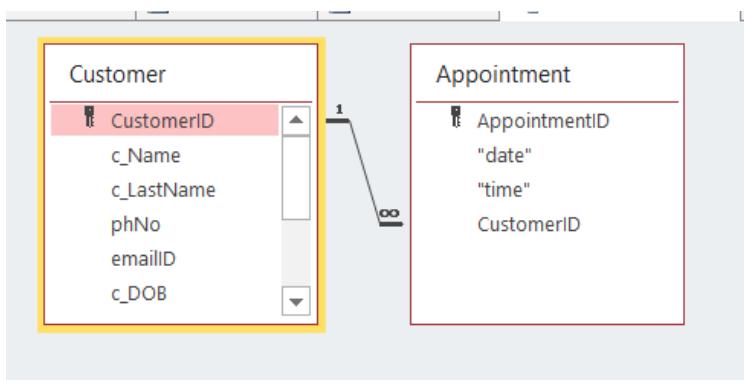
Datatypes:

- In columns such as Address, Zip code and city, “Varchar” Datatype is used to store character strings of variable length but maximum of set length is specified.
- For the State column “Char(2)” datatype is used to store single characters such as letters, digits and symbols etc.
- Unique columns are assigned as Primary key in each table and Foreign key Constraints are defined with meaningful names such as “PropertyID(fk)”.
- Primary keys and Foreign keys have the same datatype “ int(15) ” in all the tables. It is used to store a unique ID which is in Integer type.
- Date (date : type = date)” datatype is used to declare variables for storing date tokens in fields like DOB and Date.
- Time (time : type = time)” datatype is used to declare variables for time tokens in hours, minutes, seconds, optional fractions of seconds and time zones.

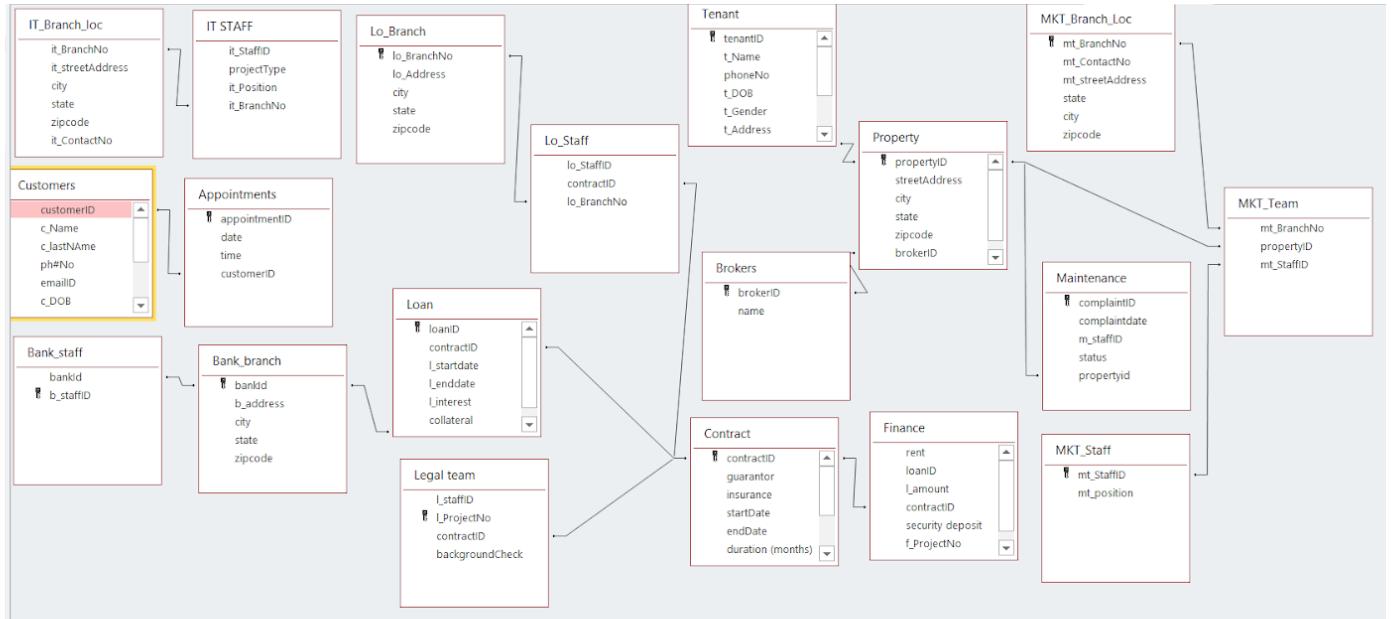
Foreign keys assignment using SQL code:

```
Query1  X  Customer  X  Query
ALTER TABLE Appointment
ADD CONSTRAINT fk_customer_appointment
FOREIGN KEY (CustomerID)
REFERENCES Customer (CustomerID)
```

This code results in a relationship as shown below:



RELATIONSHIPS



Database Application

QUERIES:

1)

```
SELECT *
FROM Customers INNER JOIN Appointments ON Customers.customerID = Appointments.customerID;
```

File Building rentals : Database - C:\Users\awalk\Documents\Final Building rentals.accdb (Access 2016) - Access NOONIKA.ROY@baruchmail.cuny.edu

Home Create External Data Database Tools Help Tell me what you want to do

Views Clipboard F4 Filter Ascending Descending Advanced Remove Sort Toggle Filter

Sort & Filter Records Find Text Formatting

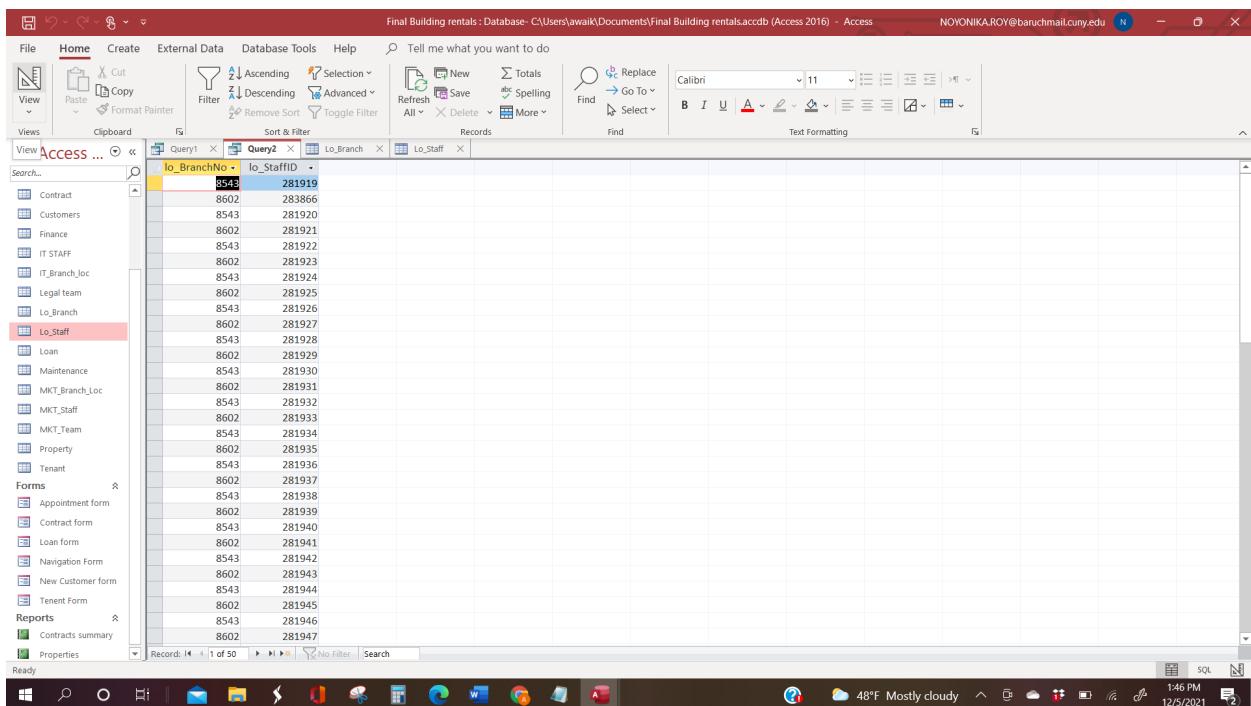
All Access ... Search... Tables Appointments Bank_branch Bank_staff Brokers Contract Customers Finance IT STAFF IT_branch_loc Legal team Lo_branch Lo_staff Loan Maintenance MKT_branch_loc MKT_staff MKT_team Property Tenant Forms Appointment form Contract form Loan form Navigation Form

Query1

| c_Name | c_LastName | phNo | emailID | c_DOB | c_Gender | c_streetAddr | city | state | zipcode | appointment | date | time | Appointment |
|-------------|------------|------------|-------------------|--------------|----------|-----------------------|------------------|-------|---------|-------------|------------------|------|-------------|
| Kole | Jimenez | 9747977843 | Kole@gmail.com | 9/13/1964 F | | 334 Fifth Dr. | Brooklyn | NY | 11213 | 20211127 | 11/19/2021 10:26 | 24C | |
| Lilanna | Pierce | 9426712539 | Lilanna@gmail.com | 7/4/1948 M | | 7830 Lilac Court | Meadows | NY | 11365 | 20211128 | 11/22/2021 16:19 | 24C | |
| Gina | Chavez | 9668554394 | Gina@gmail.com | 1/3/1988 F | | 8830 Grove Driv. | Bronx | NY | 10456 | 20211129 | 12/3/2021 14:57 | 24C | |
| Mckenna | Collier | 9360349686 | Mckenna@gmail | 10/28/1954 M | | 9908 Newbridge | Brooklyn | NY | 10034 | 20211130 | 11/10/2021 05:02 | 24C | |
| Vanessa | Patel | 988421483 | Vanessa@gmail | 5/7/1962 F | | 8826 Purple Fin | East Elmhurst | NY | 11370 | 20211131 | 11/24/2021 06:23 | 24C | |
| Sebastian | Marks | 9793208577 | Sebastian@gmail | 5/16/1988 M | | 9694 North You Spring | Valley | NY | 10977 | 20211132 | 1/6/2022 01:10 | 24C | |
| Kayleigh | Houston | 9844772550 | Kayleigh@gmail | 3/21/1942 M | | 8512 NE. Mill Dr | Fairport | NY | 4450 | 20211133 | 1/17/2022 19:19 | 24C | |
| Javier | Gates | 9387573310 | Javier@gmail.com | 10/3/1963 M | | 7329 Bayport Av | Brooklyn | NY | 11230 | 20211134 | 11/16/2021 11:19 | 24C | |
| Allie | Logan | 975751418 | Allie@gmail.cor | 2/15/1972 F | | 673 Longfellow | New York | NY | 10033 | 20211135 | 11/18/2021 11:26 | 24C | |
| Jaxson | Porter | 9168646988 | Jaxson@gmail.c | 9/22/1997 M | | 9407 Beach Ave | Bronx | NY | 11207 | 20211136 | 11/20/2021 12:55 | 24C | |
| Ahung | Choi | 9044185078 | Ahung@gmail.c | 9/29/1975 F | | 555 Amber road | Long Island City | NY | 11009 | 20211137 | 1/2/2022 07:58 | 24C | |
| Allessandra | Slovensky | 9397171597 | Allessandra@gm | 5/16/1965 F | | 71 riverside dr | New York | NY | 10027 | 20211138 | 1/21/2022 06:32 | 24C | |
| Preeti | Chauhan | 9702244723 | Preeti@gmail.c | 4/23/1982 M | | 8888 pickering s | Brooklyn | NY | 13008 | 20211139 | 11/22/2021 08:30 | 24C | |
| Sylvia | Mathews | 9198302009 | Sylvia@gmail.cc | 4/5/1971 F | | 898 Walters roa | Queens | NY | 11008 | 20211140 | 12/6/2021 02:03 | 24C | |
| Peter | Mallark | 9656335964 | Peter@gmail.cc | 6/11/1994 M | | 9090 Semiford S | New York | NY | 11009 | 20211141 | 11/15/2021 17:17 | 24C | |
| Aria | Stark | 9369563856 | Aria@gmail.com | 5/31/1986 F | | 30 Newport Par | Jersey City | NJ | 7310 | 20211142 | 1/21/2022 19:21 | 24C | |
| Jeevansh | Jotish | 9103074554 | Jeevansh@gmail | 9/15/1956 M | | 20 River drive St. | Jersey City | NJ | 7310 | 20211143 | 1/19/2022 16:37 | 24C | |
| Terry | Crews | 9538530763 | Terry@gmail.co | 10/14/1979 M | | 565 Grove Street | Jersey City | NJ | 7311 | 20211144 | 1/2/2022 09:37 | 24C | |
| Alba | Kronks | 9811033107 | Alba@gmail.co | 3/5/1998 F | | 456 Turnpike | Edison | NJ | 17890 | 20211145 | 1/17/2021 13:22 | 24C | |
| Sylvester | Troy | 9688075074 | Sylvester@gmail | 8/9/1998 M | | 566 Fleet Street | Wycoff | NJ | 17556 | 20211146 | 1/16/2022 15:04 | 24C | |
| Winona | Mattox | 9299871067 | Winona@gmail. | 12/19/1998 M | | 421 W. Bay Dr | San Antonio | TX | 78245 | 20211147 | 12/29/2021 13:38 | 24C | |
| Jaqueline | Akers | 9341548559 | Jaqueline@gmail | 9/17/1951 F | | 94 North Franklin | Arlington | TX | 76014 | 20211148 | 11/6/2021 23:50 | 24C | |
| Daria | Randle | 9602618465 | Daria@gmail.cc | 12/29/1995 F | | 972 Ann Drive | Houston | TX | 77008 | 20211149 | 1/16/2022 05:14 | 24C | |
| Rosalva | Berttram | 9859693847 | Rosalva@gmail. | 5/29/1978 F | | 10 Carridge Stree | Plane | TX | 75025 | 20211150 | 1/6/2022 23:25 | 24C | |
| Robbyn | Vu | 9651637635 | Robbyn@gmail. | 8/14/1979 M | | 1 Alderwood Dr | Baytown | TX | 77520 | 20211151 | 11/4/2021 23:40 | 24C | |
| William | Hawk | 9465484659 | William@gmail. | 1/14/1962 M | | 864 Littleton St | Mission | TX | 78572 | 20211152 | 12/23/2021 20:51 | 24C | |
| Randy | Danner | 9243873016 | Randy@gmail.c | 2/3/1976 M | | 9158 Dogwood | Anaheim | CA | 92804 | 20211153 | 11/22/2021 05:12 | 24C | |
| Keren | Sandlin | 9242499293 | Keren@gmail.cc | 1/23/1972 F | | 56 River Court | Chula Vista | CA | 91911 | 20211154 | 12/29/2021 11:53 | 24C | |
| Elenor | Hidalgo | 9436649827 | Elenor@gmail.c | 7/28/1982 F | | 8602 Pheasant I La | Puente | CA | 91744 | 20211155 | 12/3/2021 04:06 | 24C | |
| Pa | Worth | 9798766989 | Pa@gmail.com | 7/19/1984 M | | 79 Griffin St | Oxnard | CA | 93033 | 20211156 | 12/6/2021 14:57 | 24C | |

2)

```
SELECT Lo_Branch.lo_BranchNo, Lo_Staff.lo_StaffID
FROM Lo_Branch INNER JOIN Lo_Staff
ON Lo_Branch.lo_BranchNo=Lo_Staff.lo_BranchNo;
```



3)

```
UPDATE Customers
SET c_Name = "Cami", c_Gender = "F"
WHERE customerID=24001;
```

| customerID | c_Name | c_lastNAme | ph#No | emailID | c_DOB | c_Gender | c_streetAddr | city | state | zipcode |
|------------|----------|------------|------------|-----------------|------------|----------|------------------------------|----------|-------|---------|
| 24001 | Cami | Jimenez | 9747977843 | Kole@gmail.cor | 9/13/1964 | F | 334 Fifth Dr. | Brooklyn | NY | 11213 |
| 24002 | Lilianna | Pierce | 9426712539 | Lilianna@gmail. | 7/4/1948 | M | 7830 Lilac Court Meadows | Brooklyn | NY | 11365 |
| 24003 | Gina | Chavez | 9668554394 | Gina@gmail.cor | 1/3/1988 | F | 8830 Grove Driv Bronx | Bronx | NY | 10456 |
| 24004 | Mckenna | Collier | 9360349686 | Mckenna@gma | 10/28/1954 | M | 9908 Newbridge Brooklyn | Brooklyn | NY | 10034 |
| 24005 | Vanessa | Patel | 0901211102 | Vanessa@gmail | 5/7/1062 | F | 9976 Purple Dr East Elmhurst | Elmhurst | NY | 11270 |

4)

```
SELECT mt_BranchNo
FROM MKT_Branch_Loc
WHERE city='New York';
```

| mt_BranchNo |
|-------------|
| 123 |
| 133 |
| * |

5) Design view of a query:

Screenshot of Microsoft Access 2016 showing the Query Design view for a query named "Query5".

The "Tables" pane on the left lists various tables: Appointments, Bank_branch, Bank_staff, Brokers, Contract, Customers (selected), Finance, IT_STAFF, IT_Branch_loc, Legal_team, Lo_Branch, Lo_Staff, Loan, Maintenance, MKT_Branch_loc, MKT_Staff, MKT_Team, Property, Tenant, Appointment form, Contract form, Loan form, and Navigation Form.

The "Property Sheet" pane on the right shows properties for the selected table "Customers".

The main area displays the query design grid for "Query5". The query is defined as follows:

```

SELECT *
FROM Customers
WHERE I_interest > "4"
    
```

The "Criteria" row shows the condition `I_interest > "4"`.

The status bar at the bottom indicates the date and time: 12/5/2021 7:07 PM.

The screenshot shows the Microsoft Access 2016 interface. The title bar reads "Final Building rentals : Database- C:\Users\jawaik\Documents\Final Building rentals.accdb (Access 2016) - Access". The status bar at the bottom right shows "NOYONIKA.ROY@baruchmail.cuny.edu", the date "12/5/2021", and the time "7:07 PM".

The main window displays a query results grid titled "All Access ...". The grid has columns: "L_interest", "loanID", "contractID", and "collateral". The data shows 27 records, all of which have the value "recieved" in the "collateral" column. The "loanID" column contains values such as 2030405, 2030406, 2030407, etc.

The left sidebar lists tables and forms. Under "Tables", there are 18 entries including Appointments, Bank_branch, Bank_staff, Brokers, Contract, Customers, Finance, IT_STAFF, IT_Branch_loc, Legal team, Lo_Branch, Lo_Staff, Loan, Maintenance, MKT_Branch_Loc, MKT_Staff, MKT_Team, Property, and Tenant. Under "Forms", there are 5 entries: Appointment form, Contract form, Loan form, and Navigation Form.

The top ribbon menu includes File, Home, Create, External Data, Database Tools, Help, and a search bar "Tell me what you want to do". The Home tab is selected. The ribbon also includes sections for Views, Clipboard, Sort & Filter, Records, Find, and Text Formatting.

FORMS:

1) Navigation form allows one to navigate between forms and is the main menu when the database opens up

Relationships Navigation Form

One Center Main Menu

ONE CENTER PROPERTY SOLUTIONS

NEW CUSTOMER FORM

Appointment Form
New Customer Form
Loan form
Tenant form
Contract form

| | |
|----------------|----------------------|
| First Name | <input type="text"/> |
| Last Name | <input type="text"/> |
| emailID | <input type="text"/> |
| PH.No. | <input type="text"/> |
| Gender | <input type="text"/> |
| Street Address | <input type="text"/> |
| City | <input type="text"/> |
| State | <input type="text"/> |
| Zipcode | <input type="text"/> |
| CustomerID | <input type="text"/> |

Buttons:
Next Record
Save Record
Add Record
Close Form
Make an appointment online

We created a few examples pf forms that allow one to add new records into tables.

Relationships Appointment form Contract New Customer form

NEW CUSTOMER FORM

ONE CENTER PROPERTY SOLUTIONS

| | |
|----------------|----------------------|
| First Name | <input type="text"/> |
| Last Name | <input type="text"/> |
| emailID | <input type="text"/> |
| PH.No. | <input type="text"/> |
| Gender | <input type="text"/> |
| Street Address | <input type="text"/> |
| City | <input type="text"/> |
| State | <input type="text"/> |
| Zipcode | <input type="text"/> |
| CustomerID | <input type="text"/> |

Buttons:
Next Record
Save Record
Add Record
Close Form
Make an appointment online

Relationships Appointment form

Appointment form

| | |
|---------------|----------------------|
| date | <input type="text"/> |
| time | <input type="text"/> |
| appointmentID | <input type="text"/> |

Buttons:
Save Record Add Record Close Form

Loan

| | | |
|------------|----------------------|--|
| Start date | <input type="text"/> | <input type="button" value="Save Record"/> |
| End date | <input type="text"/> | <input type="button" value="Add Record"/> |
| Interest | <input type="text"/> | |
| collateral | <input type="text"/> | |
| bankID | <input type="text"/> | |

Contract

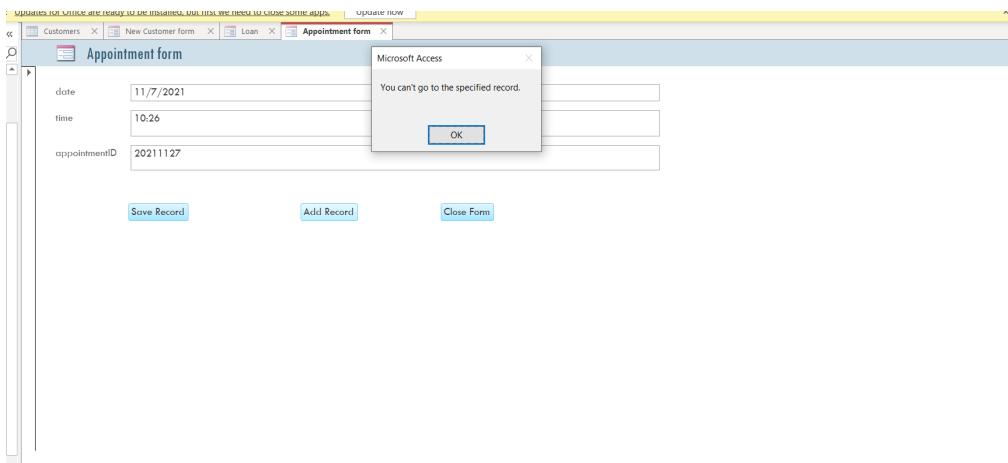
| | | |
|-------------------|----------------------|--|
| guarantor | <input type="text"/> | <input type="button" value="Add Record"/> |
| tenantID | <input type="text"/> | <input type="button" value="Save Record"/> |
| contractID | <input type="text"/> | |
| insurance | <input type="text"/> | |
| startDate | <input type="text"/> | |
| endDate | <input type="text"/> | |
| duration (months) | <input type="text"/> | |
| propertyid | <input type="text"/> | |
| securityDeposit | <input type="text"/> | |
| rent | <input type="text"/> | |

Tenant Form

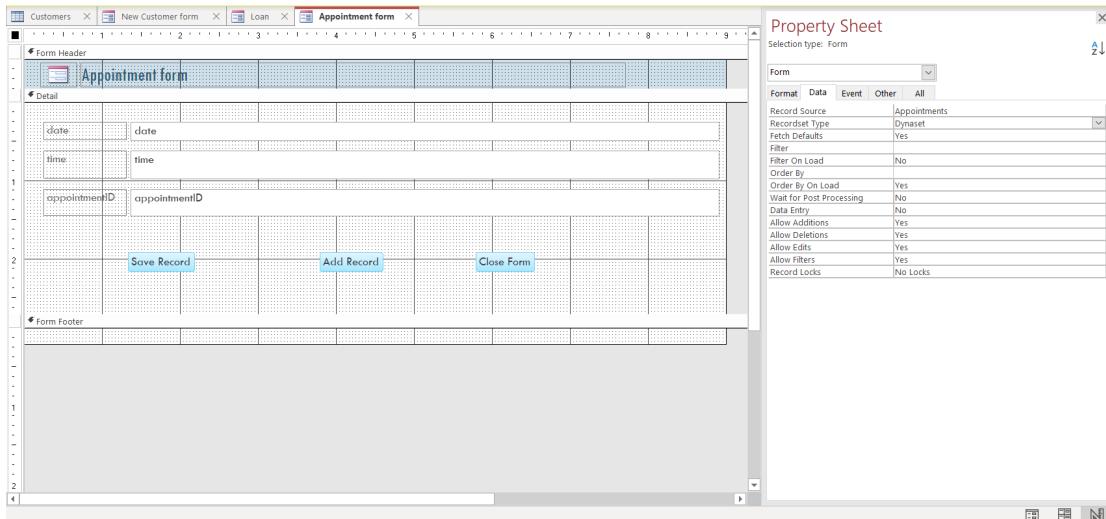
| | | |
|----------------------------------|----------------------|--|
| ONE CENTER PROPERTY SOLUTIONS | <input type="text"/> | <input type="button" value="Save Record"/> |
| Name | <input type="text"/> | <input type="button" value="Add Record"/> |
| phoneNo | <input type="text"/> | |
| DOB | <input type="text"/> | |
| Gender | <input type="text"/> | |
| Address | <input type="text"/> | |
| City | <input type="text"/> | |
| State | <input type="text"/> | |
| zipcode | <input type="text"/> | |

Challenges while making forms:

One of the main challenges we faced while making forms was that the form did not allow new records to be saved and the following error message kept showing up:



To resolve this, we looked at the properties of the form and learnt that the record source has to be a table not a query to allow record addition. Further the tabs “Allow addition”, “Allow edits” should be set to “yes” and the locks should not exist.



REPORTS:

Reports can be generated on MS Access through the ‘create’ tab. This allows the user to attain summaries of counts, sums, average etc and to make visual graphs from the data attained. One can also group the data for more structured view with this feature.

Relationships X Navigation Form X Properties X

Properties

| city | propertyID | streetAddress | No. of Properties |
|-------|------------|-------------------------|-------------------|
| Bronx | 478 | 1 Rock Creek St | 6 |
| | 481 | 835 10th St | |
| | 482 | 166B East Williams Lane | |
| | 490 | 30 Devon Ave | |
| | 495 | 64 Park Road | |
| | 505 | 688 Trout Street | |

| city | propertyID | streetAddress | No. of Properties |
|----------|------------|-----------------------|-------------------|
| Brooklyn | 477 | 893 Marvon Court | 9 |
| | 486 | 624 Wild Horse Road | |
| | 491 | 91 School Lane | |
| | 492 | 442 Lower River Court | |
| | 497 | 737 2nd Rd | |
| | 499 | 8826 Sunnyslope Lane | |
| | 501 | 188 Prospect Lane | |

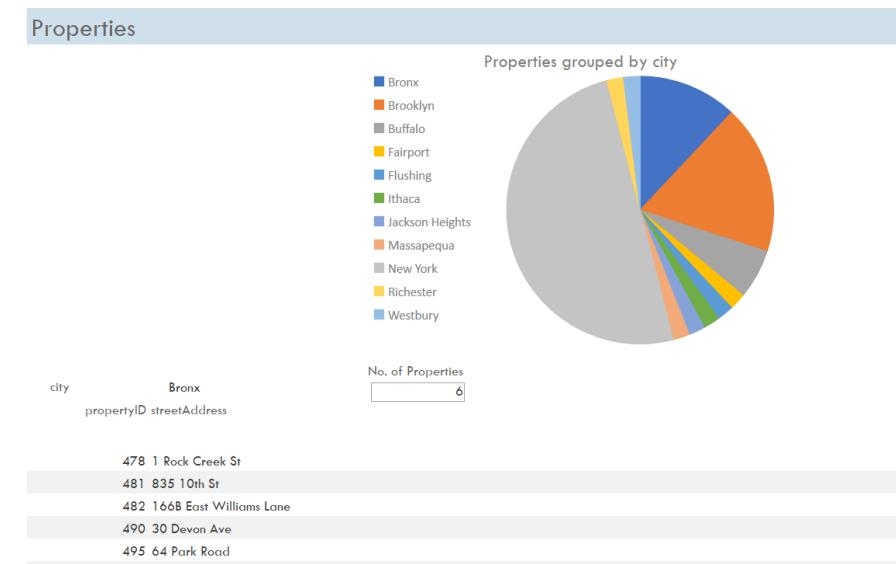
Contracts summary X

Contracts summary

| Total number of contracts | Avg security Deposit | Avg monthly rent |
|---------------------------|----------------------|------------------|
| 50 | \$ 10506.96 | \$ 7,004.64 |

Sunday, December 5, 2021 Page 1 of 1

An example of a visual graphic in a report:



Conclusions

In conclusion, we have managed to develop a database that can be used as an inventory system to provide a framework that enables the end users to manage high bulk of data in the building rental industry.

We have used Microsoft Access 2007 which is the back end. This project was a good teamwork assignment that helped us reinforce the material, practice the SQL coding, and understand a business model. We have gained new knowledge and experience in system analysis, modeling, and implementation.

Some parts of the project have been challenging. One of them is data entry because it was time-consuming to add every set of records individually. The optimal solution is to perform a bulk data insertion that was not feasible in the Microsoft Office Access. Creation of the Entity Relationship Diagram (ERD) happened to be another rather challenging activity, as there were multiple revisions. Initially, we created a diagram with many relationship links between the tables that added more complexity to our database design but after brainstorming, we had to find a solution

to simplify the tables' relationships by revising the ERD. Once the ERD was finalized and all relationships were revised, converting the ERD to a relational model was smooth. Normalization was a different challenge as we had not factored in normalization at the time, we were modelling our ERD hence we had to make changes in successful iterations to be able to reduce redundancy amongst the data and we went step by step checking the conditions of various normal forms and ways to satisfy them. The form design posed a bit of a challenge as we made a form consisting of an aggregation of two entities and we couldn't update any records to it as we later realized that a join cannot be appended.

Our project is meant to satisfy the needs of rental space owners. It is user-friendly, this package shall prove to be powerful in satisfying all the requirements of the users. Our project is just a simple and humble venture that is easy to understand. However, it can be improved upon to explore higher capabilities and achieve better functionality.