

|  |
| --- |
| Project Final Report  SYRACUSE ONLINE HOUSING SERVICES |
|  |
| IST 659  Authored by:  *SATYEN AMONKAR* |

**

# Section II

PROJECT SUMMARY

Most individuals consider safety as priority over cost of rent and amenities provided. . In Syracuse, the online housing websites provide only information such as price, amenities provided, number of bedrooms available and the location. Presently, there is no database in the housing industry where safety of a home is measured. Therefore, when a person rents a home, he is not sure how safe the area is. Syracuse Online Housing services acts as the mediator between: a) The Landlords who are looking to lease their owned property in Syracuse and b) The Tenants who are looking to rent property in Syracuse. It focuses on providing the Tenants home in a safe vicinity and also to provide Landlords more information about the Tenants looking for these properties on rent.

The general business functions of the system will be to maintain databases about the Landlords (Renter), the Tenants, the Properties, Location of the houses, Incidents reported within the vicinity, Landlord Feedback and Tenant Feedback. The organization will first validate the Tenants, Landlords and Property information by doing a background check on each manually. The Property will then be mapped to the informed location. A safety index will be assigned to the Property based on the crimes recorded in its vicinity. The safety index will help the tenants to take a rental house in a safe location in Syracuse. The safety index will be scaled from 1 to 10 with 10 considered as the safest. The System will then take feedback from tenants that reside in a house. The System will also take feedback from the Landlords about their previous Tenants. The System will also maintain information of the lease document between the Landlord and Tenants for their properties.

The Tenant can look for properties and its safety index and choose a property. He can also view the previous Tenant Feedbacks to know more about the property.

The Landlord can view the Tenants looking for their properties for rent and choose one among them by looking at their previous Landlord feedbacks. Both Tenant and Landlord can view their validity of lease using the Lease Document information.

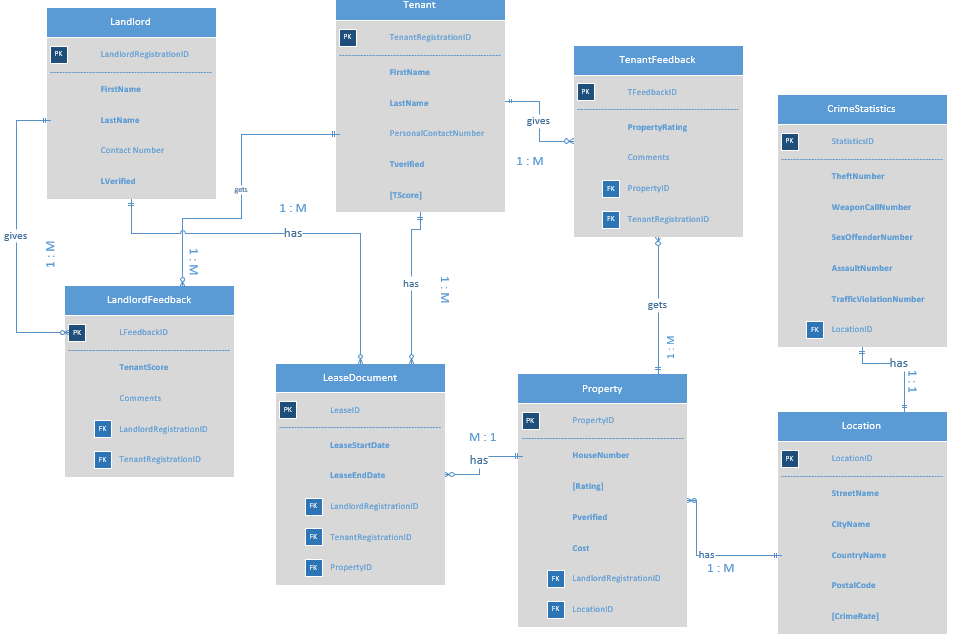
Syracuse Online Housing Services thereby improves Tenant and Landlord satisfaction by targeting the safety of the property.

# Section III

TABLES AND ATTRIBUTES

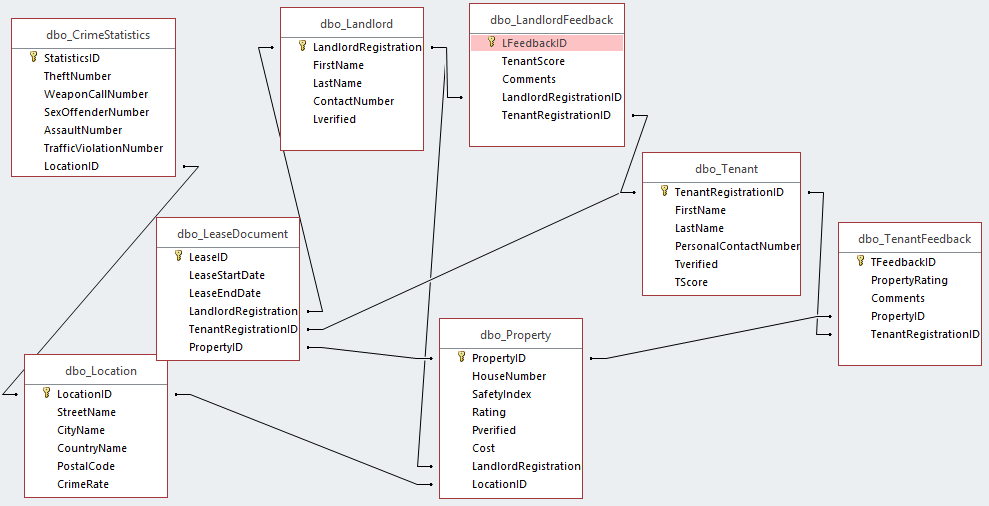
|  |  |
| --- | --- |
| **Data object:**  **OnlHousing\_Syracuse** | **This database contains all the tables and relations that together build the Online Housing System for Syracuse.** |
| **1. Landlord** | **Stores information about the Landlord (Renter) who lease their properties which are in Syracuse.** |
| Primary key: **LandlordRegistrationID** | One Landlord can be uniquely identified by its Registration ID. |
| FirstName | First Name of the Landlord. |
| LastName | Last Name of the Landlord. |
| ContactNumber | Primary Contact Number to contact the Landlord. |
| Lverified | A Boolean value which will be true if the Landlord is Verified. |
| **2. Tenant** | **Stores information about the Tenants who rent a property in Syracuse** |
| Primary key: **TenantRegistrationID** | One Tenant can be uniquely identified by its Registration ID. |
| FirstName | First Name of the Tenant. |
| LastName | Last Name of the Tenant. |
| PersonalContactNumber | Primary Contact Number to contact the Tenant. |
| Tverified | A Boolean value which will be true if the Landlord is Verified. |
| [**TScore**] | It is the average rating (from a scale 1:5, with 5 as the best). It is derived from the Landlord feedback. |
| **3. Property** | **Stores information about the properties available to rent in Syracuse.** |
| Primary Key:  **PropertyID** | One Property can be uniquely identified by its Property ID. |
| HouseNumber | It is the house number within the vicinity. |
| **[SafetyIndex]** | It is the index which tells how safe the property is. It is derived using the crime rate for its location. It is scaled from 1 to 10, with 10 as the safest. |
| **[Rating]** | It is the average rating of the property. It is derived from Tenant Feedback. It is scaled from 1 to 5, with 5 as the best property. |
| Pverified | A Boolean value which will be true if the Property is Verified. |
| Cost | Cost to rent the complete property per month. (In Dollars) |
| Foreign key:  **LandlordRegistrationID** | It is in relationship with the primary key of Landlord. The foreign key establishes one to many relationship, i.e. one Landlord can own zero or many Properties. One Property can be owned by only one Landlord. |
|  |  |
| Foreign key:  **LocationID** | It is in relationship with the primary key of Location. The foreign key establishes one to many relationship. i.e. One Location can have many properties. One Property can be present at only one Location. |
| **4. Location** | **Stores information about all the locations in Syracuse.** |
| Primary Key:  **LocationID** | One Location can be uniquely identified by its Location ID. |
| StreetName | Name of the street. |
| CityName | Name of the city. |
| CountryName | Name of the country. |
| PostalCode | Postal code of the location. |
| **[CrimeRate]** | It is derived using the Crime attributes TheftNumber, WeaponCallNumber, SexOffenderNumber, AssaultNumber, TrafficViolationNumber. for its location. |
| **5. CrimeStatistics** | **Stores information about the crime statistics for different locations in Syracuse.** |
| Primary Key:  **StatisticsID** | Crime Statistics for a location can be uniquely identified by Statistics ID. |
| TheftNumber | Total Number of thefts reported at the location |
| WeaponCallNumber | Total Number of weapon calls reported at the location |
| SexOffenderNumber | Total Number of sex offenders reported at the location |
| AssaultNumber | Total Number of Assaults reported at the location |
| TrafficViolationNumber | Total Number of traffic violations reported at the location |
| Foreign key:  **LocationID** | It is in relationship with the primary key of Location. The foreign key establishes one to one relationship, i.e. One Location can have only one Crime Statistic. One Crime Statistic can be present for only one Location. |
| **6. LeaseDocument** | **Stores information about all the Tenants renting a Property from a Landlord.** |
| Primary Key:  **LeaseID** | A lease document for a particular property can be uniquely identified by Lease ID. |
| LeaseStartDate | Start date of the lease. |
| LeaseEndDate | End date of the lease. |
| Foreign key:  **LandlordRegistrationID** | It is in a relationship with the primary key of Landlord. The foreign key establishes one to many relationship, i.e. One Lease Document can be present for only one Landlord. One Landlord can have zero or many Lease Documents. |
| Foreign key:  **TenantRegistrationID** | It is in a relationship with the primary key of Tenant. The foreign key establishes one to many relationship, i.e. One Lease Document can be present for only one Tenant. One Tenant can have zero or many Lease Documents. |
| Foreign key:  **PropertyID** | It is in a relationship with the primary key of Property. The foreign key establishes one to many relationship, i.e. One Lease Document can be present for only one Property. One Property can have zero or many Lease Documents. |
| **7. TenantFeedback** | **Stores feedback about the property from the Tenants.** |
| Primary Key:  **TFeedbackID** | A tenant feedback can be uniquely identified by its Feedback ID. |
| PropertyRating | Total Rating in the scale from 1 to 5, with 5 as the best. |
| Comments | Stores any comments given by the tenant about the house. |
| Foreign key:  **PropertyID** | It is in a relationship with the primary key of Property. The foreign key establishes one to many relationship, i.e. One Feedback can be given to only one Property. One Property can get zero or many feedbacks. |
| Foreign key:  **TenantRegistrationID** | It is in a relationship with the primary key of Tenant. The foreign key establishes one to many relationship, i.e. One Feedback can be given by only one Tenant. One Tenant can give zero or many feedbacks. |
| **8. LandlordFeedback** | **Stores feedback about the Tenants from the Landlords.** |
| Primary Key:  **LFeedbackID** | A landlord feedback can be uniquely identified by its Feedback ID. |
| TenantScore | Total Rating in the scale from 1 to 5, with 5 as the best. |
| Comments | Stores any comments given by the Landlord about the Tenant. |
| Foreign key:  **LandlordRegistrationID** | It is in a relationship with the primary key of Landlord. The foreign key establishes one to many relationship, i.e. One Feedback can be given by only one Landlord. One Landlord can give zero or many feedbacks. |
| Foreign key:  **TenantRegistrationID** | It is in a relationship with the primary key of Tenant. The foreign key establishes one to many relationship, i.e. One Feedback can be given to only one Tenant. One Tenant can be given zero or many feedbacks. |
|  |  |

RELATIONAL DATA MODEL



Since all are Strong entities, Associations (Identifying and non-identifying relationships) do not exist.

ACCESS RELATIONSHIP DIAGRAM



ADDITIONAL BUSINESS RULES

1. Only current year crime data will be available in CrimeStatistics.
2. Only verified Landlords will be allowed to lease a property.
3. Before signing a lease, the Tenant should be verified.
4. Only verified properties will be available on the online housing portal.
5. Only CrimeStatistics for the locations in Syracuse will be considered for calculating safety index.
6. TScore is scaled from 1 to 5, with 5 as the best score for a tenant.
7. SafetyIndex is scaled from 1 to 10, with 10 as the safest property.
8. PropertyRating is scaled from 1 to 5, with 5 as the best property.

# Section IV

SQL Creation Scripts

CREATE TABLE Landlord

(

[LandlordRegistrationID] varchar(5) NOT NULL,

[FirstName] varchar(30) NOT NULL,

[LastName] varchar(30) NOT NULL,

[ContactNumber] varchar(30),

[Lverified] varchar(3) NOT NULL,

CONSTRAINT Landlord\_pk PRIMARY KEY(LandlordRegistrationID)

)

CREATE TABLE Tenant

(

[TenantRegistrationID] varchar(5) NOT NULL,

[FirstName] varchar(30) NOT NULL,

[LastName] varchar(30) NOT NULL,

[PersonalContactNumber] varchar(30),

[Tverified] varchar (3) NOT NULL,

[TScore] float,

CONSTRAINT Tenant\_pk PRIMARY KEY(TenantRegistrationID)

)

CREATE TABLE Location

(

[LocationID] varchar(5) NOT NULL,

[StreetName] varchar(30) NOT NULL,

[CityName] varchar(30) NOT NULL,

[CountryName] varchar(30) NOT NULL,

[PostalCode] varchar(10) NOT NULL,

[CrimeRate] float,

CONSTRAINT Location\_pk PRIMARY KEY(LocationID)

)

CREATE TABLE Property

(

[PropertyID] varchar(5) NOT NULL,

[HouseNumber] varchar(40) NOT NULL,

[SafetyIndex] float ,

[Rating] float NOT NULL,

[Pverified] varchar(3) NOT NULL,

[Cost] float NOT NULL,

[LandlordRegistrationID] varchar(5) NOT NULL,

[LocationID] varchar(5) NOT NULL,

CONSTRAINT Property\_pk PRIMARY KEY(PropertyID),

CONSTRAINT Landlord\_fk1 FOREIGN KEY(LandlordRegistrationID) REFERENCES Landlord(LandlordRegistrationID),

CONSTRAINT Location\_fk1 FOREIGN KEY(LocationID) REFERENCES Location(LocationID)

)

CREATE TABLE CrimeStatistics

(

[StatisticsID] varchar(5) NOT NULL,

[TheftNumber] int DEFAULT 0,

[WeaponCallNumber] int DEFAULT 0,

[SexOffenderNumber] int DEFAULT 0,

[AssaultNumber] int DEFAULT 0,

[TrafficViolationNumber] int DEFAULT 0,

[LocationID] varchar(5) NOT NULL,

CONSTRAINT CrimeStatistics\_pk PRIMARY KEY(StatisticsID),

CONSTRAINT Location\_fk2 FOREIGN KEY(LocationID) REFERENCES Location(LocationID),

)

CREATE TABLE LeaseDocument

(

[LeaseID] varchar(5) NOT NULL,

[LeaseStartDate] date NOT NULL,

[LeaseEndDate] date NOT NULL,

[LandlordRegistrationID] varchar(5) NOT NULL,

[TenantRegistrationID] varchar(5) NOT NULL,

[PropertyID] varchar(5) NOT NULL,

CONSTRAINT LeaseDocument\_pk PRIMARY KEY(LeaseID),

CONSTRAINT Landlord\_fk2 FOREIGN KEY(LandlordRegistrationID) REFERENCES Landlord(LandlordRegistrationID),

CONSTRAINT Tenant\_fk2 FOREIGN KEY(TenantRegistrationID) REFERENCES Tenant(TenantRegistrationID),

CONSTRAINT Property\_fk1 FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID)

)

CREATE TABLE TenantFeedback

(

[TFeedbackID] varchar(5) NOT NULL,

[PropertyRating] float NOT NULL,

[Comments] varchar(30),

[PropertyID] varchar(5) NOT NULL,

[TenantRegistrationID] varchar(5) NOT NULL,

CONSTRAINT TenantFeedback\_pk PRIMARY KEY(TFeedbackID),

CONSTRAINT Tenant\_fk3 FOREIGN KEY(TenantRegistrationID) REFERENCES Tenant(TenantRegistrationID),

CONSTRAINT Property\_fk2 FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID)

)

CREATE TABLE LandlordFeedback

(

[LFeedbackID] varchar(5) NOT NULL,

[TenantScore] float NOT NULL,

[Comments] varchar(30),

[LandlordRegistrationID] varchar(5) NOT NULL,

[TenantRegistrationID] varchar(5) NOT NULL,

CONSTRAINT LandlordFeedback\_pk PRIMARY KEY(LFeedbackID),

CONSTRAINT Landlord\_fk3 FOREIGN KEY(LandlordRegistrationID) REFERENCES Landlord(LandlordRegistrationID),

CONSTRAINT Tenant\_fk4 FOREIGN KEY(TenantRegistrationID) REFERENCES Tenant(TenantRegistrationID)

)

SQL Insertion Scripts

Inserting sample data into Landlord table.

INSERT INTO Landlord (LandlordRegistrationID, FirstName, LastName,ContactNumber, Lverified)

VALUES ('L100','Kevin','Mahoney','3154893587','YES'),

('L101','Erica','Mayer','3157896547','YES'),

('L102','John','Sawyer','3154574896','YES'),

('L103','Jack','Johnson','3469875635','YES'),

('L104','Norman','Jamal','3156987568','YES');



Inserting sample data into Tenant table.

INSERT INTO Tenant (TenantRegistrationID, FirstName, LastName,PersonalContactNumber, Tverified)

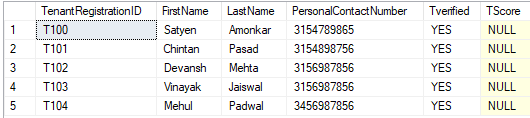
VALUES ('T100','Satyen', 'Amonkar', '3154789865','YES'),

('T101','Chintan','Pasad','3154898756','YES'),

('T102','Devansh','Mehta','3156987856','YES'),

('T103','Vinayak','Jaiswal','3156987856','YES'),

('T104','Mehul','Padwal','3456987856','YES');



Inserting sample data into Location table.

INSERT INTO Location(LocationID,StreetName,CityName,CountryName, PostalCode)

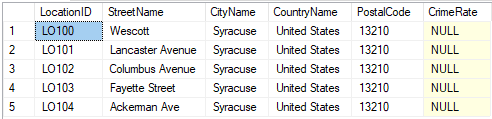
VALUES ('LO100','Wescott','Syracuse','United States','13210'),

('LO101','Lancaster Avenue','Syracuse','United States','13210'),

('LO102','Columbus Avenue','Syracuse','United States','13210'),

('LO103','Fayette Street','Syracuse','United States','13210'),

('LO104','Ackerman Ave','Syracuse','United States','13210');



Inserting sample data into Property table.

INSERT INTO Property (PropertyID, HouseNumber, Rating, Pverified, Cost, LandlordRegistrationID, LocationID)

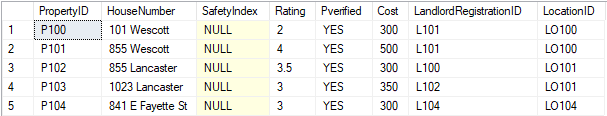
VALUES ('P100','101 Wescott',2,'YES',300,'L101','LO100'),

('P101','855 Wescott',4,'YES',500,'L101','LO100'),

('P102','855 Lancaster',3.5,'YES',300,'L100','LO101'),

('P103','1023 Lancaster',3,'YES',350,'L102','LO101'),

('P104','841 E Fayette St',3,'YES',300,'L104','LO104');



Inserting sample data into Crime Statistics table.

INSERT INTO CrimeStatistics (StatisticsID, TheftNumber, WeaponCallNumber, SexOffenderNumber, AssaultNumber, TrafficViolationNumber, LocationID)

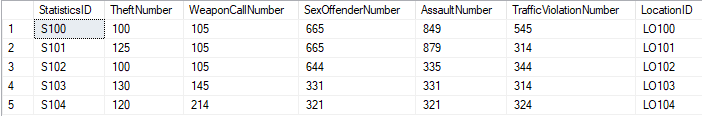
VALUES ('S100',100,105,665,849,545,'LO100'),

('S101',125,105,665,879,314,'LO101'),

('S102',100,105,644,335,344,'LO102'),

('S103',130,145,331,331,314,'LO103'),

('S104',120,214,321,321,324,'LO104');



Inserting sample data into Lease Document table.

INSERT INTO LeaseDocument(LeaseID, LeaseStartDate, LeaseEndDate, LandlordRegistrationID, TenantRegistrationID, PropertyID)

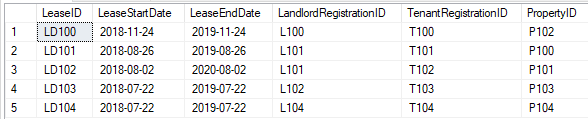
VALUES ('LD100','11-24-2018','11-24-2019','L100','T100','P102'),

('LD101','08-26-2018','08-26-2019','L101','T101','P100'),

('LD102','08-02-2018','08-02-2020','L101','T102','P101'),

('LD103','07-22-2018','07-22-2019','L102','T103','P103'),

('LD104','07-22-2018','07-22-2019','L104','T104','P104');



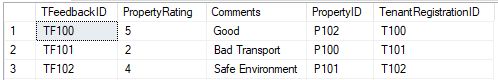
Inserting sample data into Tenant Feedback table.

INSERT INTO TenantFeedback(TFeedbackID,PropertyRating, Comments,PropertyID, TenantRegistrationID)

VALUES ('TF100','5','Good','P102','T100'),

('TF101','2','Bad Transport','P100','T101'),

('TF102','4','Safe Environment','P101','T102');



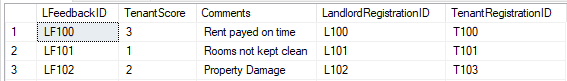
Inserting sample data into Landlord Feedback table.

INSERT INTO LandlordFeedback(LFeedbackID,TenantScore,Comments,LandlordRegistrationID, TenantRegistrationID)

VALUES ('LF100','3','Rent payed on time','L100', 'T100'),

('LF101','1','Rooms not kept clean','L101', 'T101'),

('LF102','2','Property Damage','L102','T103');



# Section V

MAJOR DATA QUESTIONS

* Which properties in Syracuse have Safety Index greater than 5?
* What is the tenure of stay in a property for which the Tenant has given feedback?
* How many properties does each landlord own?

The first question is important as it gives the tenant a fair idea of which properties are safe to reside in Syracuse.

The second question is important as it provides the administrator and Landlords an insight to how long the tenant has stayed, based on which the tenant has given the score to the Landlord.

The third question is important as Syracuse Online Housing Services can use this information to attract Landlords in Syracuse with more properties.

SQL Statements

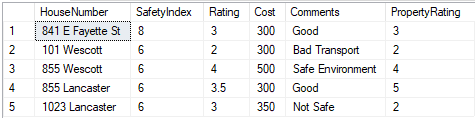
1. Which properties in Syracuse have Safety Index greater than 5?

SELECT p.HouseNumber, p.SafetyIndex, p.Rating, p.Cost, tf.Comments, tf.PropertyRating

FROM Property p INNER JOIN TenantFeedback tf ON p.PropertyID = tf.PropertyID

WHERE p.SafetyIndex > 5

ORDER BY p.SafetyIndex DESC;



Here Rating is the static Rating given to the property, while Property Rating is calculated based on the feedback provided by the tenants.

1. What is the tenure of stay in a property for which the Tenant has given feedback?

SELECT l.LandlordRegistrationID, l.FirstName AS Landlord\_FirstName, ld.LeaseStartDate, ld.LeaseEndDate, p.HouseNumber, p.Cost, t.TenantRegistrationID, t.FirstName AS Tenant\_FirstName, TScore

FROM Landlord l INNER JOIN Property p ON l.LandlordRegistrationID = p.LandlordRegistrationID

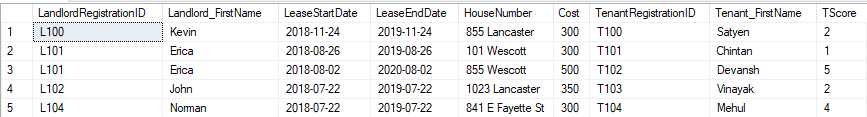
INNER JOIN LeaseDocument ld ON l.LandlordRegistrationID = ld.LandlordRegistrationID

INNER JOIN Tenant t

ON t.TenantRegistrationID = ld.TenantRegistrationID

WHERE p.LandlordRegistrationID=ld.LandlordRegistrationID AND

p.PropertyID=ld.PropertyID;



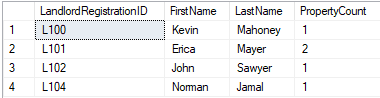
1. How many properties does each landlord own?

SELECT l.LandlordRegistrationID, l.FirstName, l.LastName, COUNT(p.PropertyID) AS PropertyCount

FROM Landlord l INNER JOIN Property p

ON l.LandlordRegistrationID = p.LandlordRegistrationID

GROUP BY l.LandlordRegistrationID, l.FirstName, l.LastName;



PROCEDURES

1. Procedure to calculate Crime Rate for each location.

CREATE PROCEDURE updateCrimeRate

AS

BEGIN

UPDATE Location

SET CrimeRate = CrimeCount.Rate

FROM

(SELECT ((c.TheftNumber + c.WeaponCallNumber + c.SexOffenderNumber + c.AssaultNumber + c.TrafficViolationNumber)/5.0) AS Rate, l.LocationID

FROM CrimeStatistics c

INNER JOIN Location l

ON c.LocationID = l.LocationID

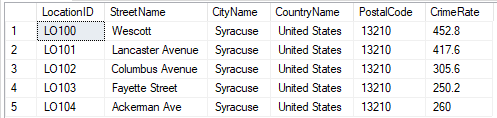
) AS CrimeCount

WHERE Location.LocationID = CrimeCount.LocationID

END;

EXECUTE updateCrimeRate;

Updated Location table



1. Procedure to generate Safety Index for each property.

CREATE PROCEDURE updateProperty

AS

BEGIN

UPDATE Property

SET SafetyIndex = CrimeIndex.SafetyIndex

FROM

(SELECT (10-FLOOR((l.CrimeRate)/100)) AS SafetyIndex, l.LocationID

FROM LOCATION l

INNER JOIN Property p

ON l.LocationID = p.LocationID

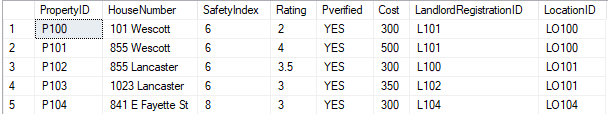
) AS CrimeIndex

WHERE Property.LocationID = CrimeIndex.LocationID

END;

EXECUTE updateProperty;

Updated Property table



Here, Procedures are used since the data updating process is static for crime statistics. Meaning, this data is updated on yearly basis.

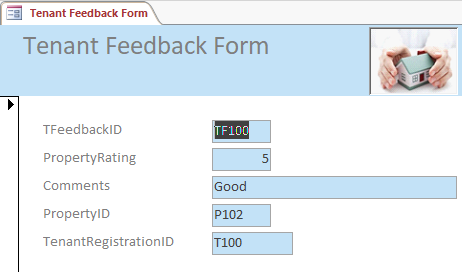
# Section VI

INTERFACES:

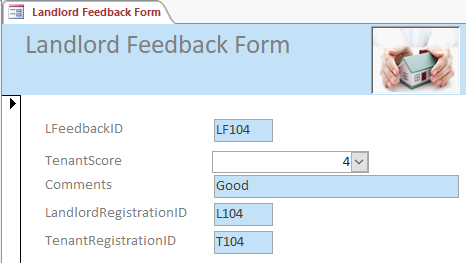
1. FORMS

There are three forms in the system to input feedback of Landlords and Tenants and to add properties.

1. Tenant Feedback Form: Allows you to add and view Tenant Feedback.

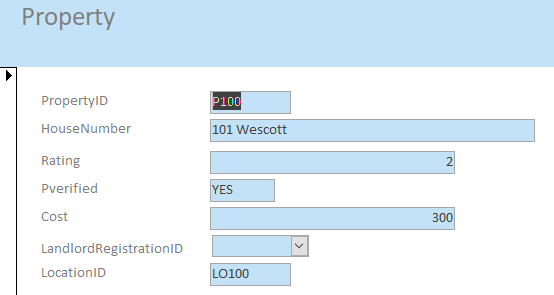
****

1. Landlord Feedback Form: Allows you to add and view Landlord Feedback.

****

The Combo box, TenantScore restricts the input to values between 1 and 5.

1. Property Form: allows you to add and view Properties

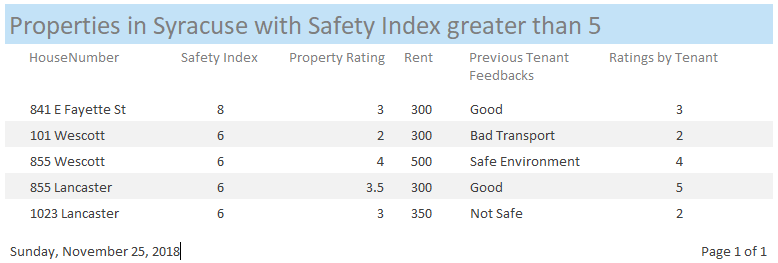


The Combo box, LandlordRegistrationID restricts the input to preregistered Landlords.

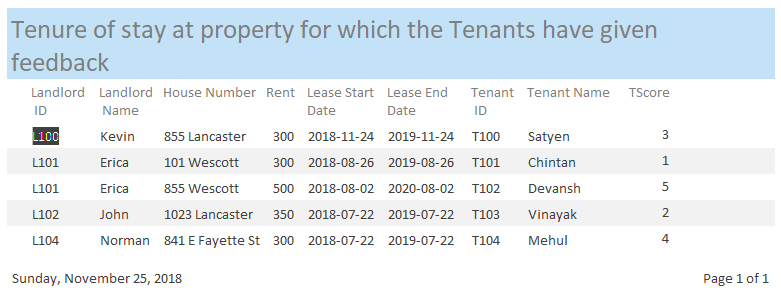
1. REPORTS

There are three reports in the system based on the major data questions.

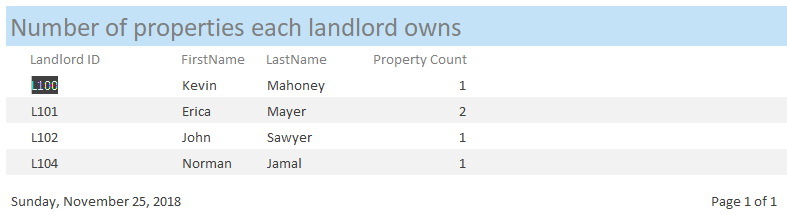
1. Which properties in Syracuse have Safety Index greater than 5?

****

1. What is the tenure of stay in a property for which the Tenant has given feedback?

****

1. How many properties does each landlord own?

****

# Section VII

Triggers

CREATE TRIGGER updateTentantScore

ON LandlordFeedback

FOR INSERT, UPDATE, DELETE

AS

IF @@ROWCOUNT >= 1

BEGIN

UPDATE Tenant

SET TScore = Feedback.Score

FROM

( SELECT AVG(l.TenantScore) AS Score, l.TenantRegistrationID

FROM LandlordFeedback l

INNER JOIN Tenant t

ON l.TenantRegistrationID = t.TenantRegistrationID

GROUP BY l.TenantRegistrationID

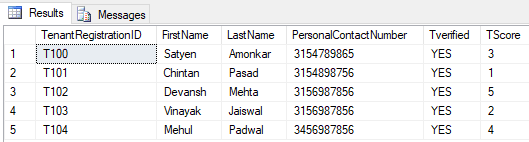
) AS Feedback

WHERE Tenant.TenantRegistrationID = Feedback.TenantRegistrationID

END;

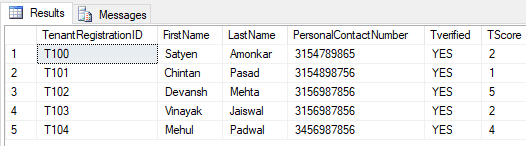
Explanation: Whenever there is insertion, updation or deletion in the Landlord Feedback, the code calculates the updated average tenant score given to a tenant by landlords, using landlord feedback. It then assigns the updated TScore to the tenants. I have used a trigger here as the Tscore of tenant needs to dynamically changed based on new Landlord Feedbacks.

Tenant table with TScore before data is updated.

****

Tenant table with TScore after data is updated.

INSERT INTO LandlordFeedback VALUES('LF105',1,'Noisy','L104','T100');

****

Transactions

Transaction to add new landlord, property and tenant to a lease document.

SET XACT\_ABORT ON;

BEGIN TRANSACTION leaseSign

INSERT INTO Landlord (LandlordRegistrationID, FirstName, LastName,ContactNumber, Lverified)

VALUES ('L105','Vincent','Johnson','3154894487','YES');

INSERT INTO Property (PropertyID, HouseNumber, Rating, Pverified, Cost, LandlordRegistrationID, LocationID)

VALUES ('P105','800 Ackerman',4,'YES',600,'L105','LO104');

INSERT INTO Tenant (TenantRegistrationID, FirstName, LastName,PersonalContactNumber, Tverified)

VALUES ('T105','Yash', 'Amonkar', '3154559865','YES');

EXECUTE updateProperty;

INSERT INTO LeaseDocument(LeaseID, LeaseStartDate, LeaseEndDate, LandlordRegistrationID, TenantRegistrationID, PropertyID)

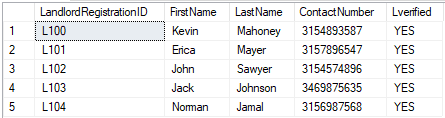
VALUES ('LD105','12-04-2018','11-24-2019','L105','T105','P105');

COMMIT TRANSACTION leaseSign;

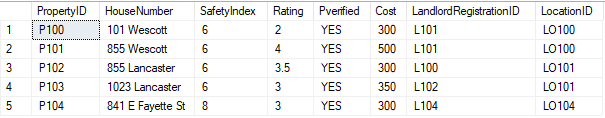
Note: The Execute command above runs the updateProperty procedure and updates the Safety Index of the newly added property.

Before Transaction:

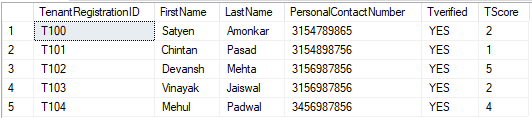
select \* from Landlord;



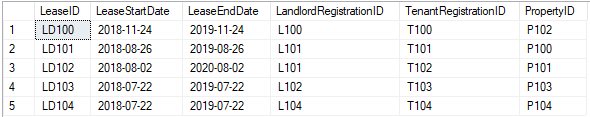
select \* from Property;



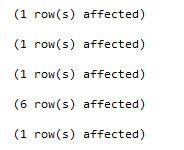
select \* from Tenant;



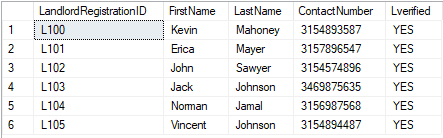
select \* from LeaseDocument;



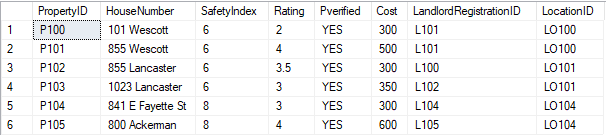
After Transaction:



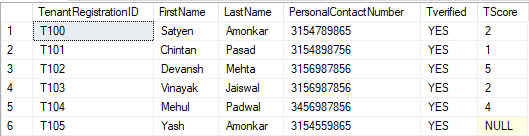
select \* from Landlord;



select \* from Property;



select \* from Tenant;



Note: Since any landlord hasn’t given Yash a feedback, his Tscore is NULL.

select \* from LeaseDocument;

