Course Project

Phase-1

Requirements Collection and Analysis

MINI-WORLD

Advanced Real Estate Management System

Mini-World Description and Purpose of the Database

This database represents a real estate investment and management platform designed to support a diverse range of stakeholders, including investors, tenants, real estate firms, construction companies, and marketing agencies. The platform aims to streamline processes related to property investment, leasing, tenant management, and project development by organizing and storing critical information. It also serves as a comprehensive repository for tracking property transactions, managing leases, and maintaining records of real estate investments and construction projects.

Key entities like investors, tenants, and lessees are linked with properties and projects, allowing detailed analysis of investment returns, risk levels, and lease terms. Furthermore, real estate properties, construction projects, and marketing firms are tracked to facilitate decision-making based on property status, investment performance, and market trends. The database includes entities like PropertyDealer, ConstructionFirm, and MarketingFirm, each with attributes and relationships that support industry-specific needs, such as licensing, client histories, and project portfolios.

Users and Their Interactions with the Database

The primary users of the database are:

1. **Investors**: They will use the platform to view potential investment opportunities, assess risks, track profit margins, monitor projected returns (ROI), and manage their investment portfolios.
2. **Tenants and Lessees**: Tenants can access their lease details, payment histories, and lease terms. Lessees provide information about their lease purposes (residential or commercial) and financial credentials, which assist investors and property managers in making leasing decisions.
3. **Property Dealers and Construction Firms**: Dealers and construction firms can access information on properties, including past sales, current market value, and associated projects. Construction firms manage project portfolios, track project histories, and evaluate feedback and ratings from investors and lessees.
4. **Marketing Firms**: Marketing agencies use the database to track clients, assess reputation ratings, and analyze revenues, enabling them to tailor marketing strategies for available and upcoming properties.

5. **Property Managers and Platform Administrators**: They will use the database for an overview of all leases, investments, and property statuses to maintain smooth operations, ensure compliance, and generate performance reports for stakeholders.

Overall, the database serves as a centralized, structured platform supporting property investment decisions, leasing processes, and real estate project development, allowing users to make informed, data-driven decisions. With a robust relational design, the database ensures that data is easily accessible and maintains consistency across interconnected entities such as investors, tenants, projects, and properties. By consolidating data on project histories, investment returns, tenant preferences, and market trends, the platform enables users to analyze performance metrics and market dynamics with precision.

Furthermore, the database supports scalability and adaptability as new properties, projects, and user types can be added without disrupting existing records. This flexibility allows real estate firms, investors, and purchasers to respond quickly to changing market conditions, assess investment risks, and implement targeted marketing and leasing strategies. Ultimately, the platform not only optimizes property management and investment operations but also fosters stronger collaboration among industry professionals, enhancing transparency, efficiency, and profitability across the real estate ecosystem.

Data Requirements

Listed below are the entities along with relevant attributes:

Entities:

1. **Investor**

a.	InvestorID	INT, PRIMARY KEY
b.	Name	VARCHAR (255)
c.	RiskAssessment	INT (RANGE [0,10])
d.	Preferences	LIST <VARCHAR (255)>
e.	InvestorType	{Contractor, RealEstateInvestor, Both}
f.	Mentor	INT, FOREIGN KEY (Investor.InvestorID)

2. **Lease (WEAK: Dependent on Lesse and property)**

a.	LeaseID	INT, PARTIAL KEY
b.	LeaseStartDate	DATE
c.	LeaseEndDate	DATE
d.	LeaseTerms	LIST <VARCHAR (255)>
e.	Lesse	INT, FOREIGN KEY (LesseDetails.LesseID)
f.	Property	INT, FOREIGN KEY (Property.PropertyID)

3. **ProjectInvestments (WEAK: Dependent on Investor and Project)**

a.	InvestmentID	INT, PARTIAL KEY
b.	Contractor	INT, FOREIGN KEY (Investor.InvestorID)
c.	InvestedAmount	DECIMAL (12, 2)
d.	ProjectConcerned	INT, FOREIGN KEY (Project.ProjectID)
e.	InvestmentDate	INT

f. ProfitMargin	DECIMAL (3,3) (percentage)
g. ProjectedROI	DECIMAL (12, 2)
h. RiskLevel	INT (RANGE (0, 10))

4. **Tenant**

a. TenantID	INT, PRIMARY KEY
b. Name	VARCHAR (255)
c. Preferences	LIST <VARCHAR (255)>
d. PaymentHistory	MAP <DATE, VARCHAR (255)>
e. DependsOn	INT, FOREIGN KEY (Tenant.TenantID)

5. **LesseDetails**

a. LesseID	INT, PRIMARY KEY
b. Name	VARCHAR (255)
c. Occupation	VARCHAR (255)
d. AnnualIncome	DECIMAL (12, 2)
e. LeasePurpose	{RESIDENTIAL, COMMERCIAL}
f. CreditScore	DECIMAL (5, 2)
g. GuarantorName	VARCHAR (255)
h. GuarantorContact	INT [MOBILE NUMBER]

6. **Marketing Firm**

a. MarketFirmID	INT, PRIMARY KEY
b. Name	VARCHAR (255)
c. Location	TEXT (GEOLOCATION)
d. FoundedYear	DATE (YYYY)
e. ContactInfo	INT [MOBILE NUMBER]
f. Specialization	LIST <VARCHAR (255)>
g. ReputationRating	INT (RANGE (0,10))
h. NumberOfClients	INT
i. AnnualRevenue	DECIMAL (12, 2)
j. ServicesOffered	LIST <VARCHAR (255)>

7. **PropertyDealer**

a. DealerID	INT, PRIMARY_KEY
b. Name	Varchar (255)
c. ContactInfo	INT [MOBILE NUMBER]
d. ExperienceYears	INT
e. PropertiesSold	MAP <DATE, INT (FOREIGN KEY (Property.PropertyID))>
f. Rating	Decimal (5,2)
g. License	TEXT

8. **ConstructionFirm**

a. FirmID	INT, PRIMARY KEY
b. FirmName	VARCHAR (255)

c.	Portfolio	TEXT
d.	ProjectHistory	LIST<INT (FOREIGN KEY (Project.ProjectID)) >
e.	Ratings	DECIMAL (3,2)
f.	Reviews	LIST <VARCHAR (255)>
g.	License	TEXT

G. RealProperty

a.	PropertyID	INT, PRIMARY KEY
b.	Name	VARCHAR (255)
c.	Description	TEXT
d.	Price	DECIMAL (18,2)
e.	Amenities	(Multi-valued Attribute) LIST <VARCHAR (255)>
f.	Status	{Available, Sold, Rented, Marked for Demolition}
g.	PropertyType	{Residential, Commercial, Intellectual}
h.	MaintenanceHistory	TEXT
i.	Image	BLOB
j.	DeveloperFirmID	INT, FOREIGN KEY (ConstructionFirm.FirmID)
k.	CurrentOwner	List<INT>, FOREIGN KEY (Purchaser.PurchaseID)
l.	Address	('StreetNo' VARCHAR (255), 'StreetName' VARCHAR(255), 'Building No' INT, 'CITY' VARCHAR(255), 'ZIPCODE' INT)

10. Project (Dependent on ConstructionFirm)

a.	ProjectID	INT, Partial key
b.	ProjectName	VARCHAR (255)
c.	Status	{Completed, Ongoing}
d.	EstimatedCost	DECIMAL (18,2)
e.	FirmID	INT, FOREIGN KEY (ConstructionFirm.FirmID)
f.	ParentProject	INT, FOREIGN KEY (Project.ProjectD)

11. Purchaser

a.	PurchaserID	INT
b.	Name	VARCHAR (255)
c.	ContactInfo	INT [MOBILE NUMBER]
d.	Address	('StreetNo' Varchar (255), 'StreetName' VARCHAR(255), 'BuildingNo' INT, 'CITY' VARCHAR(255), 'ZIPCODE' INT)
e.	DateOfPurchase	DATE(YYYY-DD-MM)
f.	PaymentMethod	VARCHAR (255)
g.	AmountPayed	DECIMAL (12,2)
h.	PropertyId	INT, FOREIGN KEY (RealProperty.PropertyID)
i.	Purpose	VARCHAR (255)
j.	Equity	DECIMAL (3,3) (PERCENTAGE)
k.	DateOfSell	DATE(YYYY-DD-MM)

12. **RealEstateInvestment**

a.	InvestmentID	INT, PARTIAL KEY
b.	Investor	INT, FOREIGN KEY (Investor.InvestorID)
c.	InvestedAmount	INT
d.	ProjectConcerned	INT, FOREIGN KEY (RealEstate.RealEstateID)
e.	InvestmentDate	DATE (YYYY-MM-DD)
f.	Equity	DECIMAL (3, 3) (percentage)
g.	ProjectedROI	DECIMAL (18, 2)
h.	RiskLevel	INT (RANGE (0, 10))

13. **RealEstate (Subclass of RealProperty)**

a.	RealEstateID	INT, PRIMARY KEY
b.	CurrentValue	DECIMAL (18,2)
c.	PastValueTrends	TimeSeriesData
d.	RelatedProperty	INT, FOREIGN KEY (Property.PropertyID)

Assumptions:

1. One investor can make a single investment on a single project and each of these investments will be distinguished by (project, investor) pair.
2. A single Lesse can have multiple leases for the same property spanning over different time periods, each of them with different LesseID (partial key).
3. A single property can have multiple leases spanning over different time periods, each of them with different LesseID (partial key).
4. There can be multiple joint purchasers of the same property.
5. Assumption for the (min, max) constraint of the degree 4 relationship below:
   (A real estate (essentially it is a property at the time of its construction) is initially invested in by multiple investors each holding a stake in it in accordance with their respective investments. However, if an individual or company buys the entire property, they become the sole owner (having the legal right to claim it in addition to having other legal privileges), and the investors are bought out. Any stake held by the investor now becomes defunct and the buyer becomes the sole and ultimate owner. That is why property has (0, N) constraint as the newly made property has no purchaser. A purchaser can be an owner of a property at a time but not after it sells that property to someone else, so constraint is (0, N) for that. (Note that an investor doesn’t hold the legal right to own a property.))
6. It is assumed that an investor has invested in at least one project.
7. It is assumed that firm has at least took up a project and a project is made by one and only one firm.
8. One investor can be mentored by multiple mentors.
9. Assume that there is at max 1 provider who is accountable for a dependent.

Relationships:

1. **InvestmentMadeInProject (Identifying relationship for ProjectInvestment)**
   i. Degree: 2
   ii. Participating entity types: Investor and ProjectInvestment
   iii. Min/Max: Investor: (1, N), ProjectInvestment: (1, N)

2. **InvestmentMadeInRealEstate**
   i. Degree: 2
   ii. Participating entity types: Investor and RealEstateInvestment
   iii. Min/Max: Investor: (1, N), RealEstateInvestment: (1, N)

3. **Given on (Identifying relationship for Lease)**
   i. Degree: 2
   ii. Participating entity types: Lease and RealProperty
   iii. Min/Max: Lease: (1, 1), RealProperty: (0, N)

4. **Undertakes**
   i. Degree: 2
   ii. Participating entity types: Project and ConstructionFirm
   iii. Min/Max: Project: (1, 1), ConstructionFirm: (1, N)

5. **DealtBy**
   i. Degree: 4
   ii. Participating entity types: ConstructionFirm, RealProperty, PropertyDealer and Purchaser
   iii. Min/Max:
      a. ConstructionFirm: (1, N)
      b. Property: (1, N)
      c. PropertyDealer: (0, N)
      d. Purchaser: (0, N)

6. **LeaseLesseConnection (Identifying relationship for Lease)**

   i. Degree: 2
   ii. Participating entity types: Lease and LesseDetails
   iii. Min/Max: Lease: (1, 1) and LesseDetails: (1, N)

7. **MadeOn (Identifying relationship for ProjectInvestment)**
   i. Degree: 2
   ii. Participating entity types: Project and ProjectInvestment
   iii. Min/Max: Project: (1, N) and ProjectInvestment: (0, 1)

8. **RelatedTo**
   i. Degree: 2
   ii. Participating entity types: RealProperty and RealEstate

iii. Min/Max: RealProperty: (0,1) and RealEstate: (1, 1)

Recursive relationships:

1) **MentoredBy**
   i. Degree: 2
   ii. Participating entity types: Investor (Roles: Mentor and Mentee)
   iii. Min/Max: For mentor role: (0, N) and mentee role: (0, N)

2) **DependsOn**
   i. Degree: 2
   ii. Participating entity types: Tenant (Roles: Provider and Dependent)
   iii. Min/Max: For Provider role: (0,N) and Dependent role: (0 ,1)

3) **SubProject**
   i. Degree: 2
   ii. Participating entity types: Project (Roles: Parent and Child)
   iii. Min/Max: For Parent role: (0,N) and Child role: (0 ,1)

Functional Requirements:

1. Retrieval

a. Query

i. Selection

Retrieve all information about investors who are classified as "RealEstateInvestor" with a RiskAssessment score of 8 or above.

```
SELECT * FROM Investor WHERE InvestorType = 'RealEstateInvestor' AND RiskAssessment >=
8;
```

*Retrieve complete data for "high-risk" real estate investors involved in commercial property projects. (**Context**: This query aims to identify investors who have a higher tolerance for risk and are involved in commercial properties)*

```
SELECT Investor.*, Project.*
FROM Investor
JOIN ProjectInvestments ON Investor.InvestorID = ProjectInvestments.Contractor
JOIN Project ON Project.ProjectID = ProjectInvestments.ProjectConcerned
JOIN RealProperty ON RealProperty.PropertyID = Project.ProjectConcerned
WHERE Investor.RiskAssessment >= 8
AND RealProperty.PropertyType = 'Commercial';
```

*Retrieve complete data for **all tenants with a specific preference**, such as "Near Metro Access".*

```
SELECT * FROM Tenant WHERE 'Near Metro Access' = ANY (Preferences);
```

*Retrieve complete data for **all properties currently available for sale**.*

```
SELECT * FROM RealProperty WHERE Status = 'Available';
```

ii. Projection

***Investors with Multi-City Investments**: Retrieve investors who have invested in projects across multiple cities, displaying their investment totals and involved cities.*

```
SELECT Investor.InvestorID, Investor.Name, SUM(ProjectInvestments.InvestedAmount) AS
TotalInvestment, ARRAY_AGG(DISTINCT RealProperty.Address->'CITY') AS CitiesInvolved
FROM Investor
JOIN ProjectInvestments ON Investor.InvestorID = ProjectInvestments.Contractor
JOIN Project ON ProjectInvestments.ProjectConcerned = Project.ProjectID
JOIN RealProperty ON Project.ProjectConcerned = RealProperty.PropertyID
GROUP BY Investor.InvestorID, Investor.Name
HAVING COUNT(DISTINCT RealProperty.Address->'CITY') > 1;
```

***High-Credit Tenants for Luxury Properties:** List tenants with credit scores above 750 leasing properties labeled as "Luxury" or "Prime."*

```
SELECT Tenant.TenantID, Tenant.Name, LesseDetails.CreditScore, RealProperty.Name AS
PropertyName
FROM Tenant
JOIN LesseDetails ON Tenant.TenantID = LesseDetails.LesseID
JOIN Lease ON LesseDetails.LesseID = Lease.Lesse
JOIN RealProperty ON Lease.Property = RealProperty.PropertyID
WHERE LesseDetails.CreditScore > 750
AND (RealProperty.Name LIKE '%Luxury%' OR RealProperty.Name LIKE '%Prime%');
```

__Display the names and estimated costs of all ongoing projects__ managed by a particular construction firm.

```
SELECT ProjectName, EstimatedCost FROM Project WHERE Status = 'Ongoing' AND FirmID =
[ConstructionFirmID];
```

Get __all project names__ handled by a specific construction firm, showing only project names and firm IDs.

```
SELECT ProjectName, FirmID FROM Project WHERE FirmID = [SPECIFIC_FIRM_ID];
```

Get **contact information of all property dealers**.

```
SELECT DealerID, Name, ContactInfo FROM PropertyDealer;
```

List **names of tenants** who have a specific lease purpose, such as "Residential".

```
SELECT TenantID, Name
FROM LesseDetails
WHERE LeasePurpose = 'Residential';
```

iii. Aggregation

Calculate the average annual revenue of all marketing firms with a ReputationRating of 8 or above.

```
SELECT AVG(AnnualRevenue) AS AvgRevenue FROM MarketingFirm WHERE
ReputationRating >= 8;
```

*Calculate the total invested amount and average projected ROI for investments made by real estate investors in the last year. (**Context**: This function is helpful for assessing total investments and average returns by a specific type of investor in the previous year.)*

```
SELECT SUM(ProjectInvestments.InvestedAmount) AS TotalInvestment,
AVG(ProjectInvestments.ProjectedROI) AS AvgROI FROM ProjectInvestments JOIN Investor
ON ProjectInvestments.Contractor = Investor.InvestorID WHERE Investor.InvestorType =
'RealEstateInvestor' AND ProjectInvestments.InvestmentDate >= '2023-01-01';
```

***Project ROIs by Contractor Type**: Summarize each project’s average projected ROI and risk level by contractor type.*

```
SELECT Project.ProjectName, Investor.InvestorType,
    AVG(ProjectInvestments.RiskLevel) AS AvgRiskLevel,
    AVG(ProjectInvestments.ProjectedROI) AS AvgProjectedROI
FROM Project
JOIN ProjectInvestments ON Project.ProjectID = ProjectInvestments.ProjectConcerned
JOIN Investor ON ProjectInvestments.Contractor = Investor.InvestorID
GROUP BY Project.ProjectName, Investor.InvestorType;
```

*Find the **average profit margin** across all project investments.*

```
SELECT AVG(ProfitMargin) AS AvgProfitMargin
FROM ProjectInvestments;
```

*Retrieve the **highest-rated marketing firm** based on the reputation rating.*

```
SELECT Name, ReputationRating
FROM MarketingFirm
ORDER BY ReputationRating DESC
LIMIT 1;
```

*Calculate the **total amount invested by all investors** in a specific project.*

```
SELECT ProjectConcerned, SUM(InvestedAmount) AS TotalInvestment
FROM ProjectInvestments
WHERE ProjectConcerned = [SPECIFIC_PROJECT_ID]
GROUP BY ProjectConcerned;
```

iv. Search

Search for property names that contain the keyword "Park" (e.g., "Parkview Apartments").

```
SELECT * FROM RealProperty WHERE Name LIKE '%Park%';
```

*Find all properties with names containing a specific keyword (e.g., "Green") and amenities matching "Gym" or "Pool." (**Context**: Useful for clients searching for properties with specific keywords and desired amenities.)*

```
SELECT *
FROM RealProperty
WHERE Name LIKE '%Green%'
AND Amenities && ARRAY['Gym', 'Pool'];
```

Retrieve properties named "Luxury" or "Prime" with specific amenities and within a set price range.

```
SELECT RealProperty.*
FROM RealProperty
WHERE (Name LIKE '%Luxury%' OR Name LIKE '%Prime%')
AND Price BETWEEN 500000 AND 1500000
AND Amenities && ARRAY['Pool', 'Gym', 'Parking'];
```

Search for all marketing firms that specialize in "Digital Marketing".

```
SELECT *
FROM MarketingFirm
WHERE 'Digital Marketing' = ANY (Specialization);
```

Search for projects with names containing the word "Park".

```
SELECT *
FROM Project
WHERE ProjectName LIKE '%Park%';
```

Search for tenants with partial name match "Sam".

```
SELECT * FROM Tenant WHERE Name LIKE '%Sam%';
```

b. Analysis

***Investors' Returns Analysis**: Determine the number of investors with investments that have a projected ROI above 10% in projects managed by the top-rated construction firms.*

```
SELECT COUNT(DISTINCT Investor.InvestorID) AS NumHighROIInvestors
FROM Investor
JOIN ProjectInvestments ON Investor.InvestorID = ProjectInvestments.Contractor
JOIN Project ON ProjectInvestments.ProjectConcerned = Project.ProjectID
JOIN ConstructionFirm ON Project.FirmID = ConstructionFirm.FirmID
```

```
WHERE ProjectedROI > 10 AND ConstructionFirm.Ratings >= 9;
```

Tenant Renting Preferences Report: *Calculate the number of tenants who have preferences that match the amenities offered by available properties.*

```
SELECT COUNT(DISTINCT Tenant.TenantID) AS NumMatchingTenants
FROM Tenant
JOIN Lease ON Tenant.TenantID = Lease.Lesse
JOIN RealProperty ON Lease.Property = RealProperty.PropertyID
WHERE RealProperty.Status = 'Available' AND Tenant.Preferences && RealProperty.Amenities;
```

Report on Tenant Renting Preferences vs. Property Availability. *(**Context:** Determine how well tenant preferences align with available properties and assess tenant satisfaction based on match rates.)*

```
SELECT Tenant.TenantID, Tenant.Name, COUNT(RealProperty.PropertyID) AS MatchedProperties
FROM Tenant
JOIN Lease ON Tenant.TenantID = Lease.Lesse
JOIN RealProperty ON Lease.Property = RealProperty.PropertyID
WHERE RealProperty.Status = 'Available'
AND Tenant.Preferences && RealProperty.Amenities
GROUP BY Tenant.TenantID, Tenant.Name
ORDER BY MatchedProperties DESC;
```

Analysis of Marketing Firm Client Growth Relative to Reputation Rating. *(**Context:** Identify how client numbers and annual revenue correlate with the firm's reputation rating, offering insights into market positioning.)*

```
SELECT MarketFirmID, Name, ReputationRating, NumberOfClients, AnnualRevenue
FROM MarketingFirm
ORDER BY ReputationRating DESC, NumberOfClients DESC;
```

2. Modification

Insertion: Add a new project investment record while verifying investment limits based on investor risk levels. (**Context:** Adds an investment record while checking that the invested amount does not exceed a limit set by the investor’s risk assessment.)

```
INSERT INTO ProjectInvestments (InvestmentID, Contractor, InvestedAmount, ProjectConcerned,
InvestmentDate, ProfitMargin, ProjectedROI, RiskLevel)
VALUES ([NewID], [InvestorID], [Amount], [ProjectID], [Date], [Margin], [ROI], [RiskLevel])
WHERE (SELECT RiskAssessment FROM Investor WHERE InvestorID = [InvestorID]) >= 5
AND [Amount] <= CASE WHEN (SELECT RiskAssessment FROM Investor WHERE InvestorID = [InvestorID]) >=
7 THEN 100000 ELSE 50000 END;
```

Insertion: Insert a **new tenant record** with all necessary details.

```
INSERT INTO Tenant (TenantID, Name, Preferences, PaymentHistory)
VALUES (1001, 'New Tenant', ARRAY['Near School', 'Pool Access'], '{2024-05-10: "Payment received"}');
```

Insertion: I*nsert* ***new property data*** *with owner information.*

```
INSERT INTO RealProperty (PropertyID, Name, Description, Price, Status, PropertyType, Amenities,
DeveloperFirmID)
VALUES (2005, 'Green Acres', 'A spacious residential plot with scenic views.', 450000.00, 'Available',
'Residential', ARRAY['Gym', 'Park'], 103);
```

Insertion: *Insert a* ***new marketing firm*** *with specialization and reputation rating.*

```
INSERT INTO MarketingFirm (MarketFirmID, Name, Location, FoundedYear, ContactInfo, Specialization,
ReputationRating, NumberOfClients, AnnualRevenue, ServicesOffered)
VALUES (301, 'Peak Realty Marketing', '37.7749° N, 122.4194° W', '2015-01-01', 1234567890, ARRAY['Social
Media', 'Event Marketing'], 9, 150, 1000000.00, ARRAY['Branding', 'SEO'])
```

Update: *Increase the projected ROI by 5% for ongoing projects with investments from investors marked as "Contractor."*

```
UPDATE ProjectInvestments
SET ProjectedROI = ProjectedROI * 1.05
WHERE ProjectConcerned IN (SELECT ProjectID FROM Project WHERE Status = 'Ongoing')
AND Contractor IN (SELECT InvestorID FROM Investor WHERE InvestorType = 'Contractor');
```

Update: *Increase reputation ratings by 1 for marketing firms that have gained more than 100 clients in the last year. (****Context:*** *Rewards marketing firms for client growth by increasing their reputation rating.)*

```
UPDATE MarketingFirm
SET ReputationRating = ReputationRating + 1
WHERE NumberOfClients > 100
AND FoundedYear >= '2023-01-01';
```

Update: *Update the* ***reputation rating*** *of a marketing firm based on new client reviews.*

```
UPDATE MarketingFirm
SET ReputationRating = 9
WHERE MarketFirmID = [SPECIFIC_FIRM_ID];
```

Update: ***Increase the credit score of all tenants*** *whose lease purpose is residential and who have consistent payment records.*

```
UPDATE LesseDetails
SET CreditScore = CreditScore + 5
WHERE LeasePurpose = 'Residential'
```

```
AND (SELECT COUNT(*) FROM Tenant WHERE Tenant.PaymentHistory IS NOT NULL) > 12;
```

Update: ***Update the status of a project*** *from "Ongoing" to "Completed".*

```
UPDATE Project
SET Status = 'Completed'
WHERE ProjectID = [SPECIFIC_PROJECT_ID];
```

Delete*: Remove tenant records with expired leases and poor payment history (more than 5 missed payments). (****Context:*** *Clears outdated tenant records while removing tenants with a history of payment defaults.)*

```
DELETE FROM Tenant
WHERE TenantID IN (
   SELECT Lesse FROM Lease WHERE LeaseEndDate < CURRENT_DATE
)
AND TenantID IN (
   SELECT TenantID FROM PaymentHistory
   WHERE COUNT(Payment) < [ThresholdAmount]
   GROUP BY TenantID
   HAVING COUNT(Payment) < 5
);
```

Delete: *Remove tenant records of those whose leases have expired and who have a payment history with multiple defaults (for instance, payment below a threshold).*

```
DELETE FROM Tenant
WHERE TenantID IN (
   SELECT Lesse FROM Lease WHERE LeaseEndDate < CURRENT_DATE
)
AND TenantID IN (
   SELECT TenantID FROM PaymentHistory WHERE PaymentHistory.Payment < [ThresholdAmount]
);
```

Delete: *Delete tenant records who have defaulted on payments (assuming they have no payment records).*

```
DELETE FROM Tenant
WHERE PaymentHistory IS NULL;
```

Delete: *Delete property records marked for demolition.*

```
DELETE FROM RealProperty
WHERE Status = 'Marked for Demolition';
```

Delete: *Remove project investments with a risk level above S.*

```
DELETE FROM ProjectInvestments
WHERE RiskLevel > 9;
```