

# Triangle Property Management System



COURSE:SQL AND RELATIONAL  
DATABASE



NAME :MD AZMIR BHUIYAN



ID: 19985



PROFESSOR: Dr. Bhaskar,  
Vidhyacharan

# PROJECT OVERVIEW

## ❖INTRODUCTION

### ❖Entity Relationship model

- ER Diagram

- ER Diagram description

### ❖TRIANGLE relational database schemas.

### ❖Queries

- Basic

- Intermediate Query (Inner, Right, Left, Full Outer and Cross join commands.)

- Advanced

### ❖Conclusion



# INTRODUCTION

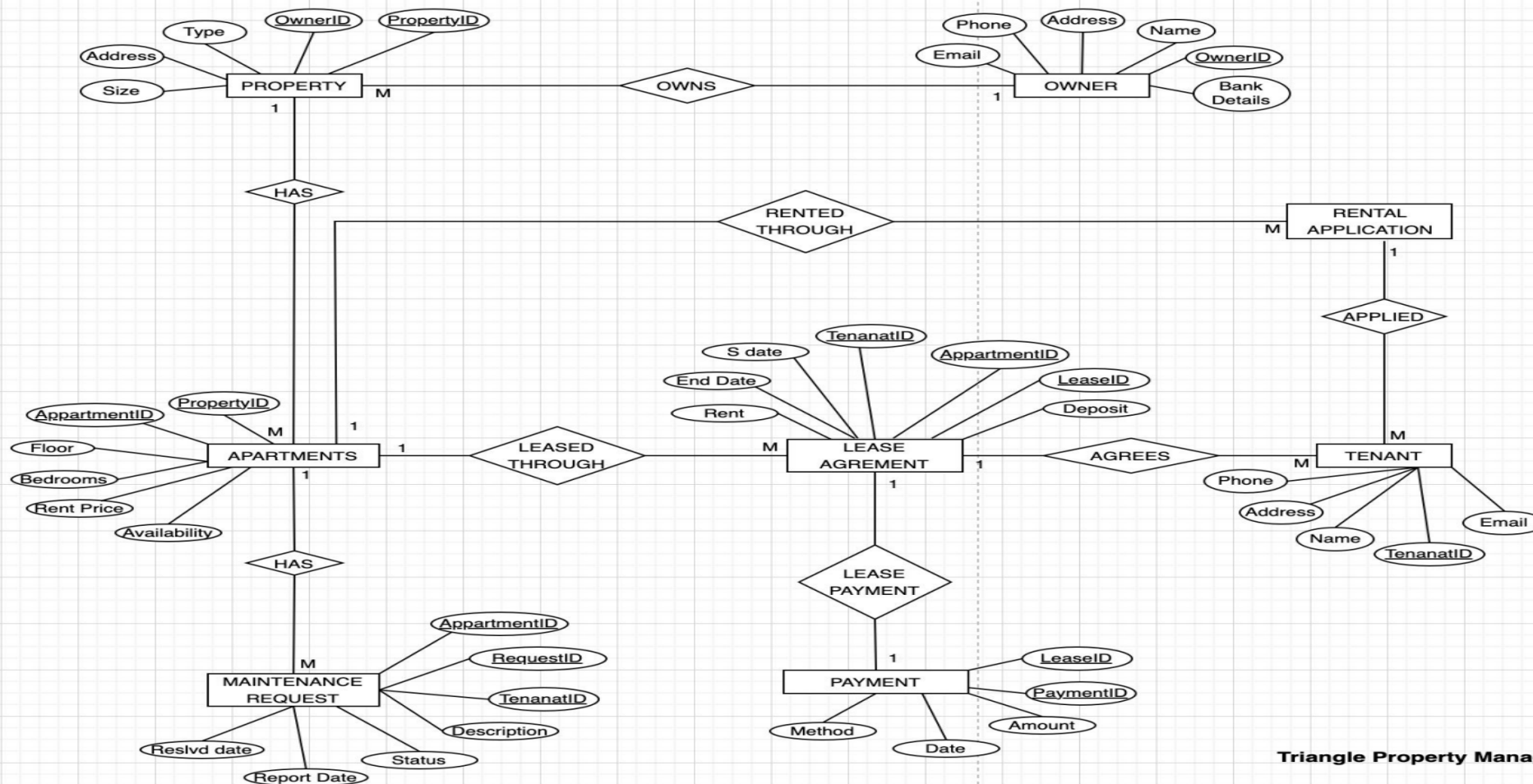
The TRIANGLE property coordination framework is an all- encompassing software solution meticulously crafted to streamline property management operations. It boasts an array of features and functionalities tailored to assist property managers and owners in effectively overseeing their properties with efficiency.

- ❑ TRIANGLE Property Management System is a digital ecosystem that simplifies the rental process.
- ❑ It serves as a centralized platform for managing properties, tenants, leases, and maintenance requests.
- ❑ The ER model is the backbone of this system, defining how data is organized and interrelated.

# What is Entity Relationship model?

- **Entity-Relationship Diagram (ERD):** A visual representation of the data structure for a system's database.
- **Entities:** These are objects or concepts that can have data stored about them.
- **Relationships:** They define how entities relate to each other and the nature of the connection.
- **Attributes:** Characteristics or properties of an entity.
- **ERD Symbols:** Include rectangles for entities, diamonds for relationships, and ovals for attributes.
- **Cardinality:** Specifies the numerical relationships between entities (one-to-one, one-to-many, many-to-many).
- **Use Cases:** ERDs are used for database design, to clarify system requirements, and to improve communication among stakeholders.

# ER DIAGRAM:



Triangle Property Management ER model

# RELATIONSHIPS

- **Property and Owner:** Each property is owned by one owner (1:M relationship), meaning a single owner can own multiple properties but a property cannot have multiple owners.
- **Property and Apartment:** A property can have multiple apartments (1:M relationship), but an apartment belongs to only one property.
- **Apartment and Maintenance Request:** An apartment can have multiple maintenance requests (1:M relationship), while a maintenance request is specific to one apartment.
- **Owner and Rental Application:** An owner can have multiple rental applications made to their properties (1:M relationship).
- **Rental Application and Tenant:** A rental application is made by a tenant (1:M relationship), implying that a tenant can apply for multiple rentals.
- **Lease Agreement:** This is a central entity that ties the Tenant, Apartment, and Lease Payment together. An apartment can have multiple lease agreements over time (1:M relationship), and a tenant can have multiple lease agreements if they rent multiple apartments.
- **Lease Payment:** A lease agreement will have multiple lease payments (1:M relationship), one for each payment period.

# Create table commands:

- CREATE TABLE Owner (
  - ownerid INT,
  - name VARCHAR(255),
  - address VARCHAR(255),
  - phone VARCHAR(255),
  - email VARCHAR(255),
  - bank\_details VARCHAR(255)
- );
  
- CREATE TABLE tenant (
  - tenantid INT,
  - name VARCHAR(255),
  - address VARCHAR(255),
  - phone VARCHAR(255),
  - email VARCHAR(255)
- );

# CREATE TABLE COMMANDS

- CREATE TABLE apartments (
  - apartmentid INT,
  - propertyid INT,
  - floor INT,
  - bedrooms INT,
  - rent\_price DECIMAL(10, 2),
  - availability TINYINT
- );
  
- CREATE TABLE rental\_application (
  - applicationid INT,
  - tenantid INT,
  - propertyid INT,
  - status VARCHAR(255)
- );



# CREATE TABLE COMMANDS

- CREATE TABLE Property (
  - propertyid INT PRIMARY KEY,
  - ownerid INT,
  - address VARCHAR(255),
  - size INT,
  - type VARCHAR(255)
- );
- CREATE TABLE Maintenance\_Request (
  - requestid INT PRIMARY KEY,
  - apartmentid INT,
  - tenantid INT,
  - description VARCHAR(255),
  - report\_date DATE,
  - resolved\_date DATE,
  - status VARCHAR(255)
- );



# CREATE TABLE COMMANDS

- CREATE TABLE Payment (
  - paymentid INT PRIMARY KEY,
  - leaseid INT,
  - date DATE,
  - amount DECIMAL(10, 2),
  - method VARCHAR(255)
- );
- CREATE TABLE Lease (
  - leaseid INT PRIMARY KEY,
  - apartmentid INT,
  - tenantid INT,
  - start\_date DATE,
  - end\_date DATE,
  - rent DECIMAL(10, 2),
  - deposit DECIMAL(10, 2)
- );

# SCHEMAS (Owner, apartments, tenant, rental\_application)

MariaDB [19985bm]> SELECT \* FROM Owner;

ownerid	name	address	phone	email	bank_details
1	Ahmad Khan	321 Elm St, Boston, 12321, MA	123-456-7890	ahmadk@email.com	Bank A
2	Fatima Ali	654 Cedar Blvd, Austin, 45632, TX	234-567-8901	fatimaa@email.com	Bank B
3	Yusuf Ahmed	987 Willow Way, Seattle, 78901, WA	345-678-9012	yusufa@email.com	Bank C
4	Aisha Rahman	432 Oak Lane, Miami, 89012, FL	456-789-0123	aishar@email.com	Bank D
5	Omar Farooq	876 Pine Ct, Portland, 90123, OR	567-890-1234	omarf@email.com	Bank E
6	Sara Malik	543 Maple Dr, Chicago, 01234, IL	678-901-2345	saram@email.com	Bank F
7	Mohammed Hussain	210 Birch St, San Diego, 12345, CA	789-012-3456	mohammedh@email.com	Bank G
8	Hassan Iqbal	321 Oak St, Philadelphia, 23456, PA	890-123-4567	hassani@email.com	Bank H
9	Noor Khan	654 Maple Ave, Orlando, 34567, FL	901-234-5678	noork@email.com	Bank I
10	Ali Zafar	987 Elm Rd, New York, 45678, NY	012-345-6789	aliz@email.com	Bank J

MariaDB [19985bm]> SELECT \* FROM apartments;

apartmentid	propertyid	floor	bedrooms	rent_price	availability
1	1	1	2	1300.00	1
2	1	2	3	1600.00	0
3	2	1	1	1100.00	1
4	2	3	2	1400.00	0
5	3	1	3	1500.00	1
6	3	2	1	1200.00	0
7	4	1	2	1350.00	1
8	4	2	3	1650.00	0
9	5	1	1	1150.00	1
10	5	3	2	1450.00	0

MariaDB [19985bm]> SELECT \* FROM tenant;

tenantid	name	address	phone	email
1	Ahmad Khan	321 Elm St., Boston, 12321, MA	123-456-7890	ahmadk@email.com
2	Fatima Ali	654 Cedar Blvd, Austin, 45632, TX	234-567-8901	fatimaa@email.com
3	Yusuf Ahmed	987 Willow Way, Seattle, 78901, WA	345-678-9012	yusufa@email.com
4	Aisha Rahman	432 Oak Lane, Miami, 89012, FL	456-789-0123	aishar@email.com
5	Omar Farooq	876 Pine Ct, Portland, 90123, OR	567-890-1234	omarf@email.com
6	Sara Malik	543 Maple Dr, Chicago, 01234, IL	678-901-2345	saram@email.com
7	Mohammed Hussain	210 Birch St, San Diego, 12345, CA	789-012-3456	mohammedh@email.com
8	Hassan Iqbal	321 Oak St, Philadelphia, 23456, PA	890-123-4567	hassani@email.com
9	Noor Khan	654 Maple Ave, Orlando, 34567, FL	901-234-5678	noork@email.com
10	Ali Zafar	987 Elm Rd, New York, 45678, NY	012-345-6789	aliz@email.com

MariaDB [19985bm]> SELECT \* FROM rental\_application;

applicationid	tenantid	propertyid	status
1	1	1	Pending
2	2	2	Accepted
3	3	3	Rejected
4	4	4	Pending
5	5	5	Accepted
6	6	6	Rejected
7	7	7	Pending
8	8	8	Accepted
9	9	9	Rejected
10	10	10	Pending

# SCHEMAS (Property, maintenance request, payments, lease \_ agreement )

[MariaDB [19985bm]> SELECT \* FROM Property;

propertyid	ownerid	address	size	type
1	1	123 Oak St, Springfield, 12345, IL	1200	Residential
2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial
3	3	789 Pine Rd, Salem, 11223, OR	900	Residential
4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential
5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial
6	6	303 Peach St, Denver, 23456, CO	1300	Residential
7	7	404 Berry Ave, Tampa, 34567, FL	1400	Commercial
8	8	505 Grape Rd, Reno, 45678, NV	1200	Residential
9	9	606 Lemon Ln, Dallas, 56789, TX	1500	Commercial
10	10	707 Lime Blvd, Charlotte, 67890, ...	1000	Residential

[MariaDB [19985bm]> SELECT \* FROM Maintenance\_Request;

requestid	apartmentid	tenantid	description	report_date	resolved_date	status
1	1	1	Leaky faucet	2023-01-10	NULL	Pending
2	2	2	Broken window	2023-02-15	NULL	Pending
3	3	3	Heating not working	2023-03-10	2023-03-15	Resolved
4	4	4	Clogged drain	2023-04-12	NULL	Pending
5	5	5	Electrical issue	2023-05-14	2023-05-20	Resolved
6	6	6	Damaged door	2023-06-18	NULL	Pending
7	7	7	Paint peeling	2023-07-22	2023-07-30	Resolved
8	8	8	Broken light fixture	2023-08-25	NULL	Pending
9	9	9	Leaky ceiling	2023-09-28	2023-10-03	Resolved
10	10	10	Window seal broken	2023-10-31	NULL	Pending

[MariaDB [19985bm]> SELECT \* FROM Payment;

paymentid	leaseid	date	amount	method
1	1	2023-01-05	1300.00	Credit Card
2	2	2023-02-05	1600.00	Bank Transfer
3	3	2023-03-05	1100.00	Cash
4	4	2023-04-05	1400.00	Check
5	5	2023-05-05	1500.00	Credit Card
6	6	2023-06-05	1200.00	Bank Transfer
7	7	2023-07-05	1350.00	Cash
8	8	2023-08-05	1650.00	Check
9	9	2023-09-05	1150.00	Credit Card
10	10	2023-10-05	1450.00	Bank Transfer

[MariaDB [19985bm]> SELECT \* FROM Lease;

leaseid	apartmentid	tenantid	start_date	end_date	rent	deposit
1	1	1	2023-01-01	2024-01-01	1300.00	600.00
2	2	2	2023-02-01	2024-02-01	1600.00	750.00
3	3	3	2023-03-01	2024-03-01	1100.00	500.00
4	4	4	2023-04-01	2024-04-01	1400.00	650.00
5	5	5	2023-05-01	2024-05-01	1500.00	700.00
6	6	6	2023-06-01	2024-06-01	1200.00	550.00
7	7	7	2023-07-01	2024-07-01	1350.00	625.00
8	8	8	2023-08-01	2024-08-01	1650.00	775.00
9	9	9	2023-09-01	2024-09-01	1150.00	525.00
10	10	10	2023-10-01	2024-10-01	1450.00	675.00

# Basic Database queries

## SELECT Query:

- Retrieve propertyid, address and type from PROPERTY table where property types are 'Apartment' or 'Condo'
- **SELECT** propertyid, address, type
- **FROM** Property
- **WHERE** type='Residential';

```
MariaDB [19985bm]> SELECT propertyid, address, type  
-> FROM Property  
-> WHERE type='Residential';
```

propertyid	address	type
1	123 Oak St, Springfield, 12345, IL	Residential
3	789 Pine Rd, Salem, 11223, OR	Residential
4	101 Apple Blvd, Madison, 45678, ...	Residential
6	303 Peach St, Denver, 23456, CO	Residential
8	505 Grape Rd, Reno, 45678, NV	Residential
10	707 Lime Blvd, Charlotte, 67890, ...	Residential

# ALTER QUERY :

- Add a new column  
Description to the Property  
table to store a textual  
description of the properties.
- ALTER TABLE Property
- ADD Description TEXT;
- SELECT \* FROM Property;

```
MariaDB [19985bm]> SELECT * FROM Property;
```

propertyid	ownerid	address	size	type	Description
1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL
2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL
3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL
4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL
5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL
6	6	303 Peach St, Denver, 23456, CO	1300	Residential	NULL
7	7	404 Berry Ave, Tampa, 34567, FL	1400	Commercial	NULL
8	8	505 Grape Rd, Reno, 45678, NV	1200	Residential	NULL
9	9	606 Lemon Ln, Dallas, 56789, TX	1500	Commercial	NULL
10	10	707 Lime Blvd, Charlotte, 67890, ...	1000	Residential	NULL

# UPDATE Query:

- Update the rent price for all 1-bedroom apartments to \$1300.00
- **UPDATE apartments**
- **SET Rent\_Price = 1300.00**
- **WHERE Bedrooms = 1;**
- **SELECT \* FROM apartments;**

```
MariaDB [19985bm]> SELECT * FROM apartments;
```

apartmentid	propertyid	floor	bedrooms	rent_price	availability
1	1	1	2	1300.00	1
2	1	2	3	1600.00	0
3	2	1	1	1300.00	1
4	2	3	2	1400.00	0
5	3	1	3	1500.00	1
6	3	2	1	1300.00	0
7	4	1	2	1350.00	1
8	4	2	3	1650.00	0
9	5	1	1	1300.00	1
10	5	3	2	1450.00	0

# ORDER BY and GROUP BY Query:

\*Get a list of tenants ordered by their name in ascending order

**SELECT TenantID,Name**

**FROM tenant**

**ORDER BY Name ASC ;**

\*Calculate the average rent price grouped by the number of bedrooms in apartments

**SELECT Bedrooms, ROUND(AVG(Rent\_Price), 2) as  
AverageRent**

**FROM apartments**

**GROUP BY Bedrooms;**

```
MariaDB [19985bm]> SELECT Bedrooms, ROUND(AVG(Rent_Price), 2) as AverageRent  
-> FROM apartments  
-> GROUP BY Bedrooms;
```

Bedrooms	AverageRent
1	1300.00
2	1375.00
3	1583.33

```
MariaDB [19985bm]> SELECT TenantID, Name  
-> FROM tenant  
-> ORDER BY Name ASC;
```

TenantID	Name
1	Ahmad Khan
4	Aisha Rahman
10	Ali Zafar
2	Fatima Ali
8	Hassan Iqbal
7	Mohammed Hussain
9	Noor Khan
5	Omar Farooq
6	Sara Malik
3	Yusuf Ahmed



- Returns rows when there is a match in both tables
- **SELECT \***
- **FROM Property**
- **INNER JOIN Owner ON Property.OwnerID = Owner.OwnerID;**

-----+												
propertyid   ownerid   address   size   type   Description   ownerid   name   address   phone   email   bank_												
details												
+-----+												
[A   B   C   D   E   F   G   H   I   J	1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL	1	Ahmad Khan	321 Elm St, Boston, 12321, MA	123-456-7890	ahmadk@email.com	Bank
	2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL	2	Fatima Ali	654 Cedar Blvd, Austin, 45632, TX	234-567-8901	fatimaa@email.com	Bank
	3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL	3	Yusuf Ahmed	987 Willow Way, Seattle, 78901, WA	345-678-9012	yusufa@email.com	Bank
	4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL	4	Aisha Rahman	432 Oak Lane, Miami, 89012, FL	456-789-0123	aishar@email.com	Bank
	5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL	5	Omar Farooq	876 Pine Ct, Portland, 90123, OR	567-890-1234	omarf@email.com	Bank
	6	6	303 Peach St, Denver, 23456, CO	1300	Residential	NULL	6	Sara Malik	543 Maple Dr, Chicago, 01234, IL	678-901-2345	saram@email.com	Bank
	7	7	404 Berry Ave, Tampa, 34567, FL	1400	Commercial	NULL	7	Mohammed Hussain	210 Birch St, San Diego, 12345, CA	789-012-3456	mohammedh@email.com	Bank
	8	8	505 Grape Rd, Reno, 45678, NV	1200	Residential	NULL	8	Hassan Iqbal	321 Oak St, Philadelphia, 23456, PA	890-123-4567	hassani@email.com	Bank
	9	9	606 Lemon Ln, Dallas, 56789, TX	1500	Commercial	NULL	9	Noor Khan	654 Maple Ave, Orlando, 34567, FL	901-234-5678	noork@email.com	Bank
	10	10	707 Lime Blvd, Charlotte, 67890, ...	1000	Residential	NULL	10	Ali Zafar	987 Elm Rd, New York, 45678, NY	012-345-6789	aliz@email.com	Bank
+-----+												

# LEFT JOIN Query:

- Returns all rows from the left table, and the matched rows from the right table.
- **SELECT \***
- **FROM Property**
- **LEFT JOIN apartments ON Property.PropertyID = apartments.PropertyID;**

```
MariaDB [19985bm]> SELECT *  
-> FROM Property  
-> LEFT JOIN apartments ON Property.PropertyID = apartments.PropertyID;
```

propertyid	ownerid	address	size	type	Description	apartmentid	propertyid	floor	bedrooms
1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL	1	1	1	2
1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL	2	1	2	3
2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL	3	2	1	1
2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL	4	2	3	2
3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL	5	3	1	3
3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL	6	3	2	1
4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL	7	4	1	2
4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL	8	4	2	3
5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL	9	5	1	1
5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL	10	5	3	2
6	6	303 Peach St, Denver, 23456, CO	1300	Residential	NULL	NULL	NULL	NULL	NULL
7	7	404 Berry Ave, Tampa, 34567, FL	1400	Commercial	NULL	NULL	NULL	NULL	NULL
8	8	505 Grape Rd, Reno, 45678, NV	1200	Residential	NULL	NULL	NULL	NULL	NULL
9	9	606 Lemon Ln, Dallas, 56789, TX	1500	Commercial	NULL	NULL	NULL	NULL	NULL
10	10	707 Lime Blvd, Charlotte, 67890, ...	1000	Residential	NULL	NULL	NULL	NULL	NULL

# RIGHT JOIN

## Query:

- Returns all rows from the right table, and the matched rows from the left table.
- **SELECT \***
- **FROM apartments**
- **RIGHT JOIN Property ON apartments.PropertyID = Property.PropertyID;**

```
ariaDB [19985bm]> SELECT *  
-> FROM apartments  
-> RIGHT JOIN Property ON apartments.PropertyID = Property.PropertyID;
```

apartmentid	propertyid	floor	bedrooms	rent_price	availability	propertyid	ownerid	address	size	type	Description
1	1	1	2	1300.00	1	1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL
2	1	2	3	1600.00	0	1	1	123 Oak St, Springfield, 12345, IL	1200	Residential	NULL
3	2	1	1	1300.00	1	2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL
4	2	3	2	1400.00	0	2	2	456 Maple Ave, Dayton, 67890, OH	1500	Commercial	NULL
5	3	1	3	1500.00	1	3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL
6	3	2	1	1300.00	0	3	3	789 Pine Rd, Salem, 11223, OR	900	Residential	NULL
7	4	1	2	1350.00	1	4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL
8	4	2	3	1650.00	0	4	4	101 Apple Blvd, Madison, 45678, ...	1100	Residential	NULL
9	5	1	1	1300.00	1	5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL
10	5	3	2	1450.00	0	5	5	202 Cherry Ln, Phoenix, 78901, AZ	1600	Commercial	NULL
NULL	NULL	NULL	NULL	NULL	NULL	6	6	303 Peach St, Denver, 23456, CO	1300	Residential	NULL
NULL	NULL	NULL	NULL	NULL	NULL	7	7	404 Berry Ave, Tampa, 34567, FL	1400	Commercial	NULL
NULL	NULL	NULL	NULL	NULL	NULL	8	8	505 Grape Rd, Reno, 45678, NV	1200	Residential	NULL
NULL	NULL	NULL	NULL	NULL	NULL	9	9	606 Lemon Ln, Dallas, 56789, TX	1500	Commercial	NULL
NULL	NULL	NULL	NULL	NULL	NULL	10	10	707 Lime Blvd, Charlotte, 67890, ...	1000	Residential	NULL

# Advanced Query:

- **Nested Inner Join (Two-way)**
- SELECT A.\*, PO.PropertyAddress, PO.OwnerName
- FROM apartments A
- INNER JOIN (
  - SELECT P.PropertyID, P.Address AS PropertyAddress, O.Name AS OwnerName
  - FROM Property P
  - INNER JOIN Owner O ON P.OwnerID = O.OwnerID) AS PO ON A.PropertyID = PO.PropertyID;

```
MariaDB [19985bm]> SELECT A.*, PO.PropertyAddress, PO.OwnerName
-> FROM apartments A
-> INNER JOIN (
->     SELECT P.PropertyID, P.Address AS PropertyAddress, O.Name AS OwnerName
->     FROM Property P
->     INNER JOIN Owner O ON P.OwnerID = O.OwnerID
-> ) AS PO ON A.PropertyID = PO.PropertyID;
```

apartmentid	propertyid	floor	bedrooms	rent_price	availability	PropertyAddress	OwnerName
1	1	1	2	1300.00	1	123 Oak St, Springfield, 12345, IL	Ahmad Khan
2	1	2	3	1600.00	0	123 Oak St, Springfield, 12345, IL	Ahmad Khan
3	2	1	1	1300.00	1	456 Maple Ave, Dayton, 67890, OH	Fatima Ali
4	2	3	2	1400.00	0	456 Maple Ave, Dayton, 67890, OH	Fatima Ali
5	3	1	3	1500.00	1	789 Pine Rd, Salem, 11223, OR	Yusuf Ahmed
6	3	2	1	1300.00	0	789 Pine Rd, Salem, 11223, OR	Yusuf Ahmed
7	4	1	2	1350.00	1	101 Apple Blvd, Madison, 45678, ...	Aisha Rahman
8	4	2	3	1650.00	0	101 Apple Blvd, Madison, 45678, ...	Aisha Rahman
9	5	1	1	1300.00	1	202 Cherry Ln, Phoenix, 78901, AZ	Omar Farooq
10	5	3	2	1450.00	0	202 Cherry Ln, Phoenix, 78901, AZ	Omar Farooq

# Nested Right Join (Two-way)

- SELECT L.\*, PayInfo.PaymentAmount, PayInfo.PaymentMethod
- FROM Lease L
- RIGHT JOIN (
  - SELECT P.leaseid, P.amount AS PaymentAmount, P.method AS PaymentMethod
  - FROM Payment P) AS PayInfo ON L.leaseid = PayInfo.leaseid;

```
-> RIGHT JOIN (  
->     SELECT P.leaseid, P.amount AS PaymentAmount, P.method AS PaymentMethod  
->     FROM Payment P  
-> ) AS PayInfo ON L.leaseid = PayInfo.leaseid;
```

leaseid	apartmentid	tenantid	start_date	end_date	rent	deposit	PaymentAmount	PaymentMethod
1	1	1	2023-01-01	2024-01-01	1300.00	600.00	1300.00	Credit Card
2	2	2	2023-02-01	2024-02-01	1600.00	750.00	1600.00	Bank Transfer
3	3	3	2023-03-01	2024-03-01	1100.00	500.00	1100.00	Cash
4	4	4	2023-04-01	2024-04-01	1400.00	650.00	1400.00	Check
5	5	5	2023-05-01	2024-05-01	1500.00	700.00	1500.00	Credit Card
6	6	6	2023-06-01	2024-06-01	1200.00	550.00	1200.00	Bank Transfer
7	7	7	2023-07-01	2024-07-01	1350.00	625.00	1350.00	Cash
8	8	8	2023-08-01	2024-08-01	1650.00	775.00	1650.00	Check
9	9	9	2023-09-01	2024-09-01	1150.00	525.00	1150.00	Credit Card
10	10	10	2023-10-01	2024-10-01	1450.00	675.00	1450.00	Bank Transfer

# CONCLUSION

- Studying any management system requires to understand the Entity Relationship Model
- ❖ Draw Entity Relationship Diagram
  - ➤ Understanding tuples, attributes, relationships between entities
  - ➤ Defining key relationships
- ❖ Understand one management very well

# REFERENCES

- "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe.
- <https://www.amazon.com/Database-System-Concepts-Abraham-Silberschatz/dp/0078022150>