

Database Concepts -ISYS2390

1. Choose entities in your database, define their attributes, and define primary key for each entity. (5 points)

The entities I have chosen for A1's Database are:

- (1) **Office**
- (2) **Landlord**
- (3) **Property**
- (4) **Tenant**

I defined some attributes that are indicated by the business rules; as well as assumed some common, and likely attributes to be present in the A1 Database.

(1) **Office**

Attributes: Office_ID (Primary Key), Office_Name, Office_Location

(2) **Landlord**

Attributes: Landlord_ID (Primary Key), Office_ID, Landlord_Name, Contact_Details

(3) **Property**

Attributes: Property_ID (Primary Key), Landlord_ID, Property_Location, Property_Type

(4) **Tenant**

Attributes: Tenant_ID (Primary Key), Office_ID, Contact_Details, Desired_Location, Desired_Property_Type

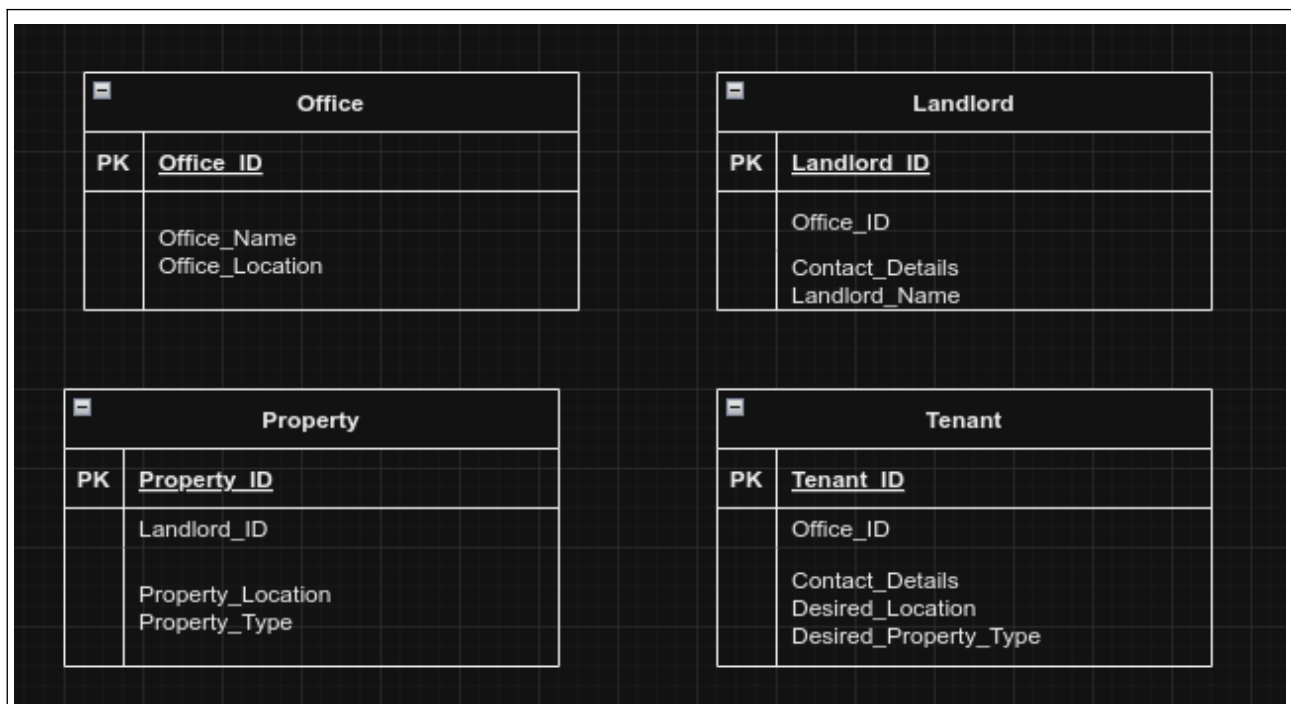


Figure 1: Shows an all of the Entity Tables; including their attributes, and their primary keys. This covers some of **Steps 2 – 4** of Mapping an ERD into a Final Schema such as the choosing of the primary keys.

This part of the task coincides with **Step 1** of Mapping an ERD into a Final Schema:

Step 1: – Regular Entities

- For each entity (table) create a relation schema consisting of attributes.

Office(Office_ID (Primary Key), Office_Name, Office_Location)

Landlord(Landlord_ID (Primary Key), Office_ID (Foreign Key), Landlord_Name, Contact_Details)

Property(Property_ID (Primary Key), Landlord_ID (Foreign Key), Property_Location, Property_Type)

Tenant(Tenant_ID (Primary Key), Office_ID (Foreign Key), Contact_Details, Desired_Location, Desired_Property_Type)

Figure 2: Relation schema consisting for each entity consisting of their different attributes. (added Foreign Keys later).

2. Create an Entity Relationship Diagram (ERD). Show relationships between entities (1:1, 1:M, M:N). (5 points)

NOTE:

ERDs do not show foreign keys.

ERDs do not show middle (junction) tables.

This task continues Steps 2 – 4; choosing the type of relationships between entities (1:1, 1:M, M:N), and shows a visual representation of the model in the form of an ERD.

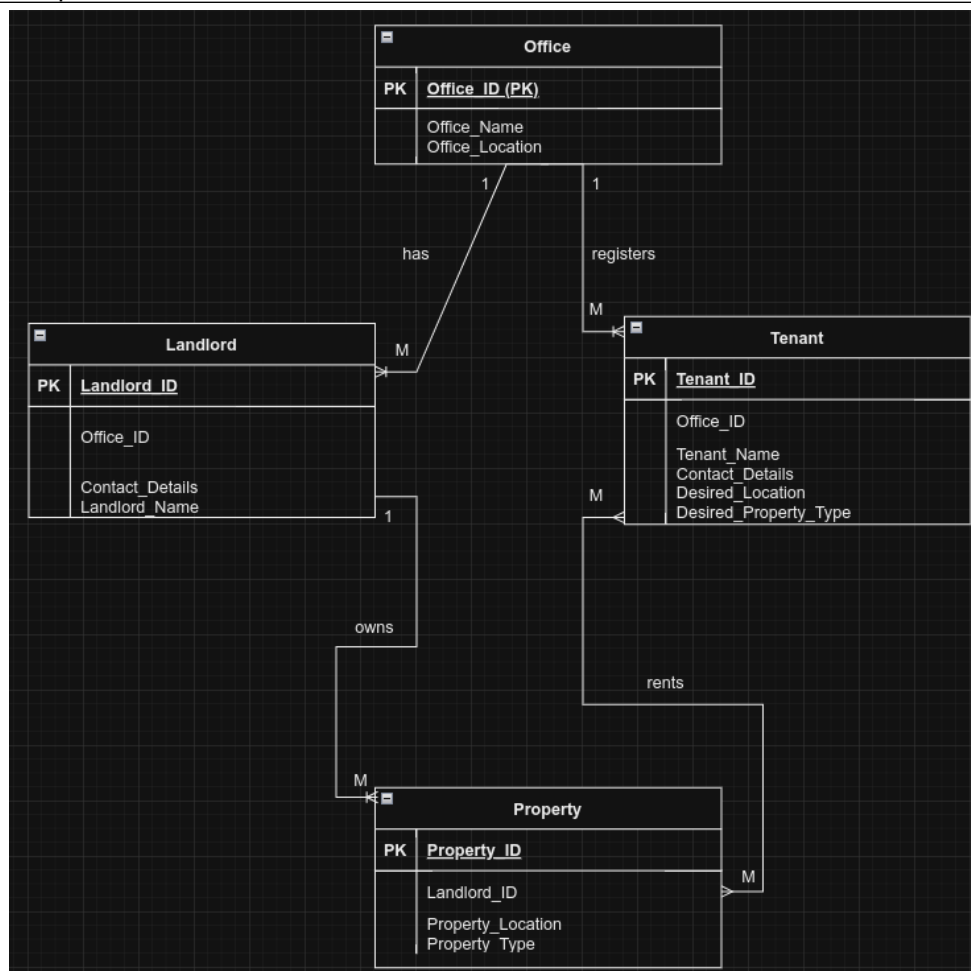


Figure 3: Shows the ERD of A1's Database Model. This ERD each entity's attributes, shows primary keys, and relationship types between the entities.

3. Map your ERD into schema using necessary steps. Write the final schema, showing primary and foreign keys. (7 points)
Note: In the mapping process follow the mapping steps. For each missing step you will lose one point even if your final schema is correct.

As covered in parts 1, and 2 of this Assignment we have completed most of the work for Steps 1 – 4 to Map an ERD into a Final Schema. However, one large part that we are missing is our foreign keys, and junction table. Below I have worked them out:

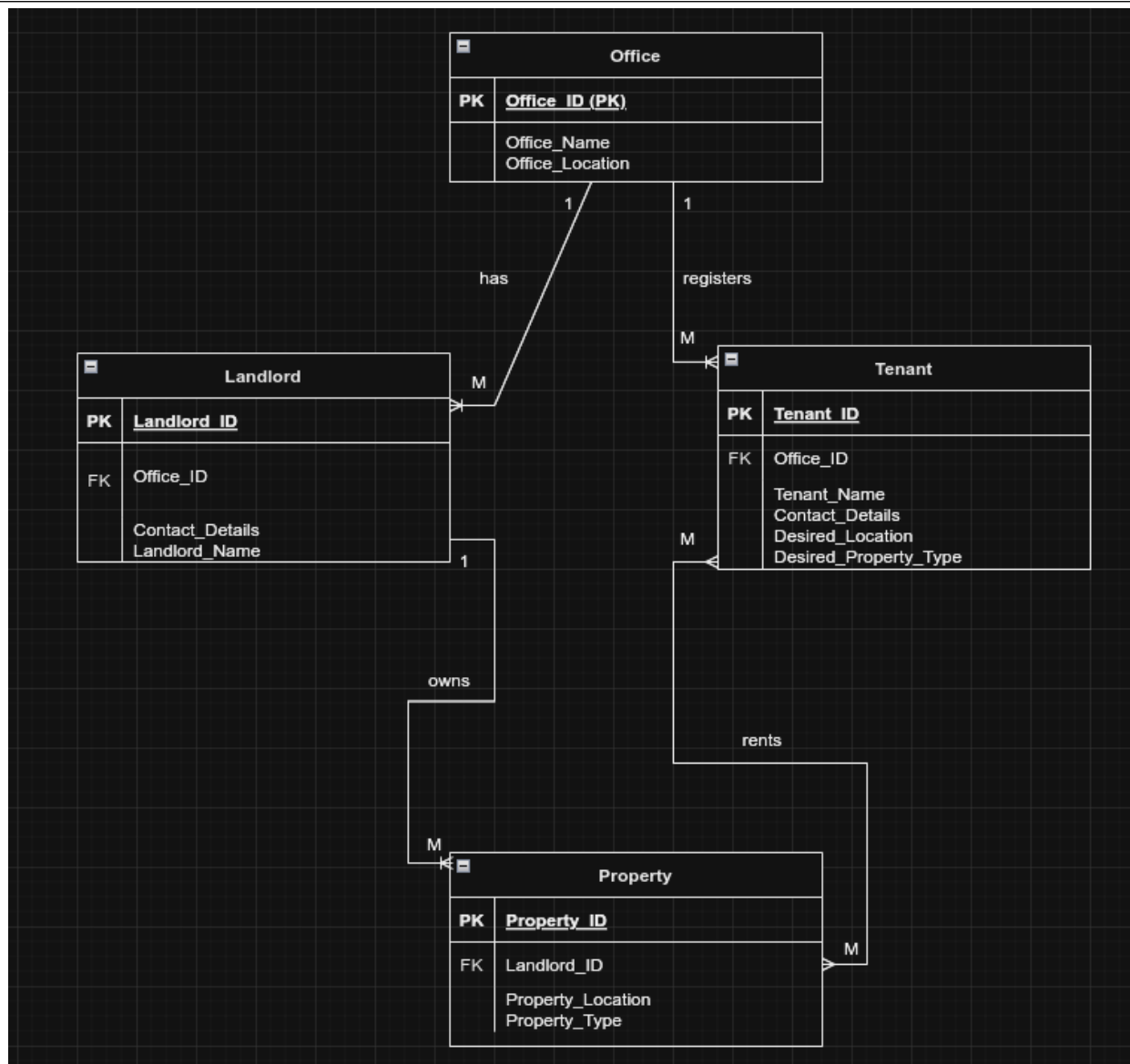


Figure 4: This is after I have added Foreign Keys. (Completing step 3)

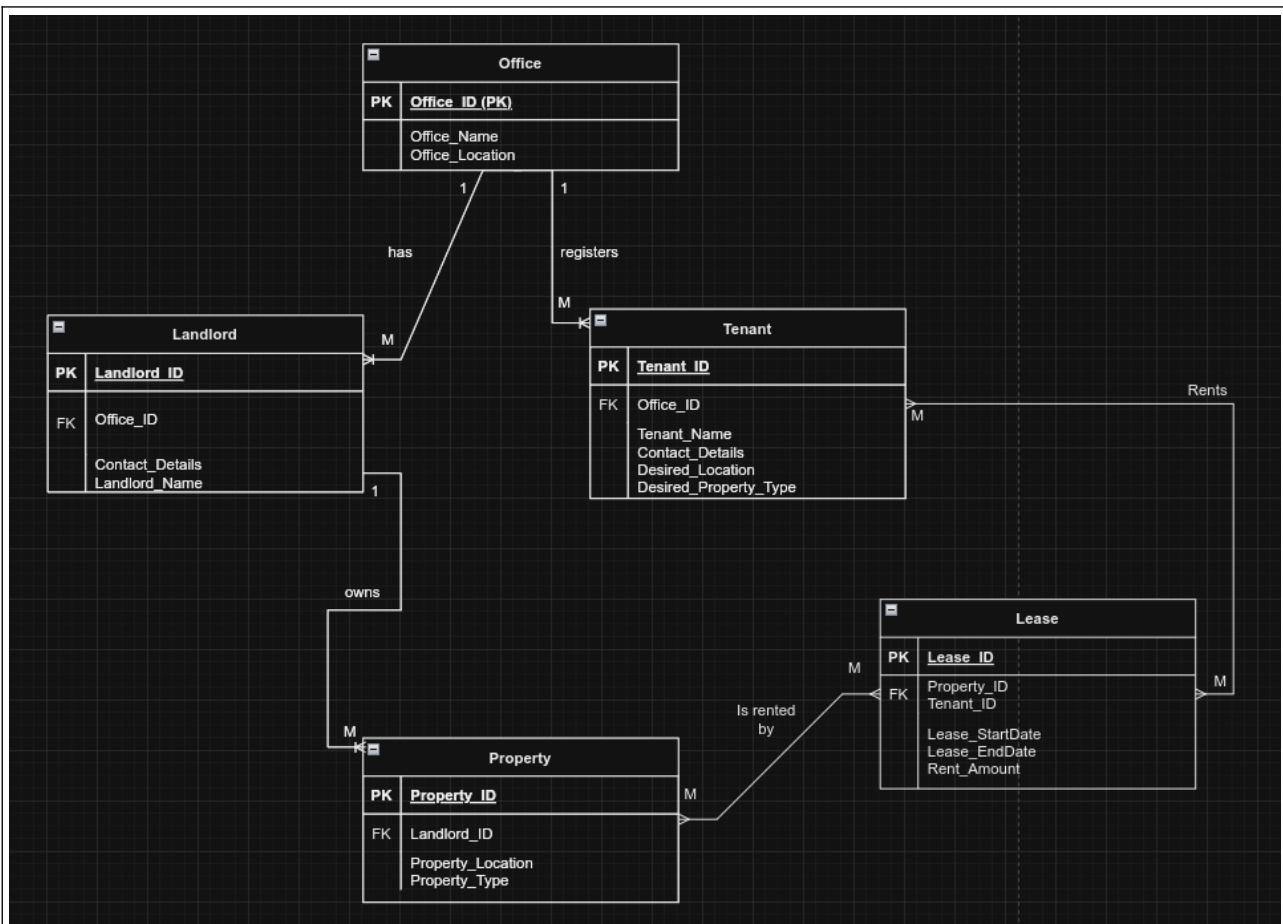


Figure 5: This is after I added a middle (junction) table of 'Lease' to get rid of the M:M relationship between Property, and Tenant. (Completing step 4)

Step 5: Final Schema

- Relational schemas of all relations (tables).

```

OFFICE (Office_ID, Office_Name, Office_Location)
LANDLORD (Landlord_ID, Landlord_Name, Contact_Details, Office_ID*)
PROPERTY (Property_ID, Property_Location, Property_Type, Landlord_ID*)
TENANT (Tenant_ID, Tenant_Name, Contact_Details, Desired_Location,
Desired_Property_Type, Office_ID*)
LEASE (Lease_ID, Lease_Start_Date, Lease_End_Date, Rent_Amount, Tenant_ID*,
Property_ID*)
  
```

Figure 6: This shows the final schema that A1's Database will form after my ERD Mapping. This will work to create a real SQL Database!

However, I don't particularly like this format so I opted to create a more complex, and in-depth version that shows more detail.

```

Office(
    Office_ID: INT PRIMARY KEY,
    Office_Name: VARCHAR,
    Office_Location: VARCHAR
)

Landlord(
    Landlord_ID: INT PRIMARY KEY,
  
```

```

Landlord_Name: VARCHAR,
Contact_Details: VARCHAR,
Office_ID: INT,
FOREIGN KEY (Office_ID) REFERENCES Office(Office_ID)
)

Property(
Property_ID: INT PRIMARY KEY,
Property_Location: VARCHAR,
Property_Type: VARCHAR,
Landlord_ID: INT,
FOREIGN KEY (Landlord_ID) REFERENCES Landlord(Landlord_ID)
)

Tenant(
Tenant_ID: INT PRIMARY KEY,
Tenant_Name: VARCHAR,
Contact_Details: VARCHAR,
Desired_Location: VARCHAR,
Desired_Property_Type: VARCHAR,
Office_ID: INT,
FOREIGN KEY (Office_ID) REFERENCES Office(Office_ID)
)

Lease(
Lease_ID: INT PRIMARY KEY,
Lease_Start_Date: DATE,
Lease_End_Date: DATE,
Rent_Amount: DECIMAL,
Tenant_ID: INT,
Property_ID: INT,
FOREIGN KEY (Tenant_ID) REFERENCES Tenant(Tenant_ID),
FOREIGN KEY (Property_ID) REFERENCES Property(Property_ID)
)

```

Figure 7: This shows a much more in-depth, and precise Final Schema that shows the Foreign Key references, as well as the types each attribute would take.

Test:

Physical Schemas

mydb MySQL Schema new_schema1 MySQL Schema

Tables (5 items)

Add Table Landlord Lease Office Property Tenant

Views (0 items)

Add View

Office - Table

Table	Columns	Indexes	Foreign Keys	Triggers	Partitioning	Options	Inserts	Privileges
Office	Office_ID	INT	PK	NN	UQ			
	Office_Name	VARCHAR(45)		NN	UQ			
	Office_Location	VARCHAR(45)		NN	UQ			

Figure 8: To test my schema I decided I would actually create, and test it using MySQL.

References:

Mahmud Alam – Lecturer

Ayesha Falki - Lecturer

Topic 2 - Entity-Relationship Modelling (1).pptx

Creating ERD Tutorial (Draw.io).pdf

draw.io

Appendices:

- 1) ass1Oisins3952320.pdf (this file)
- 2) entityTable.drawio
- 3) ERD.drawio
- 4) FK.drawio
- 5) Junction.drawio