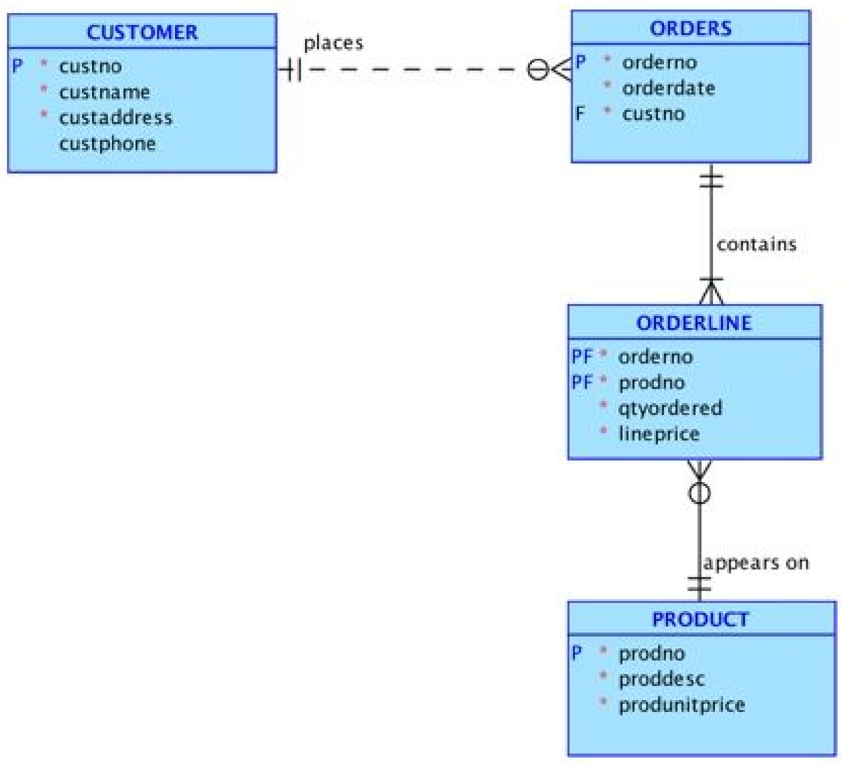
Logical Modelling Sample Solutions

Customer Orders

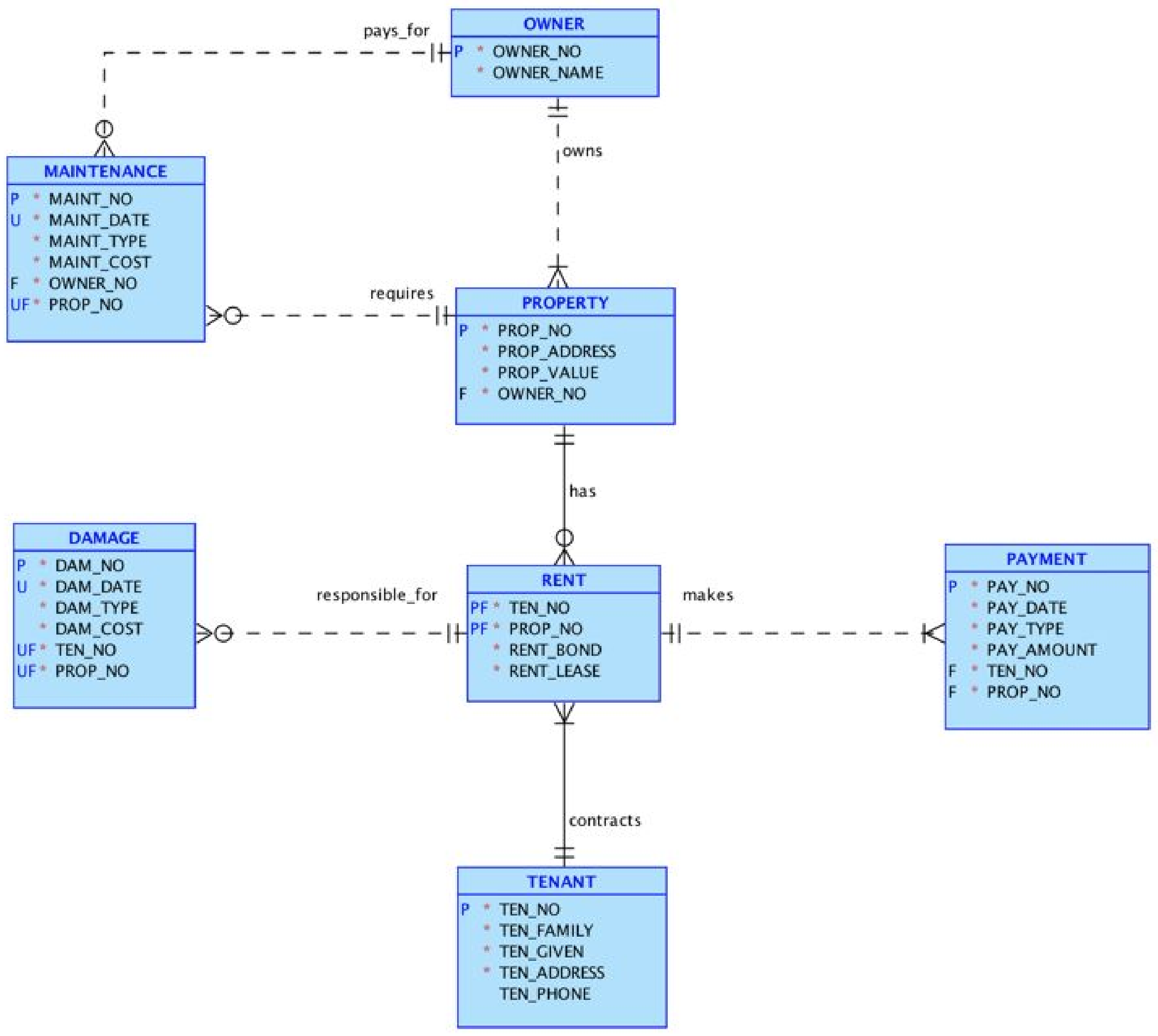


Rental System

Note:

- Surrogate keys of MAINT_NO and DAM_NO added to MAINTENANCE and DAMAGE
- Since the model is to be implemented in Oracle date and time can be stored in a single DATE type attribute

Did you arrive at this model?



Remember that every model you create via top down design must have the final entities checked to ensure they are in 3NF with no insert, update or delete anomalies. If you apply this to MAINTENANCE you will see:

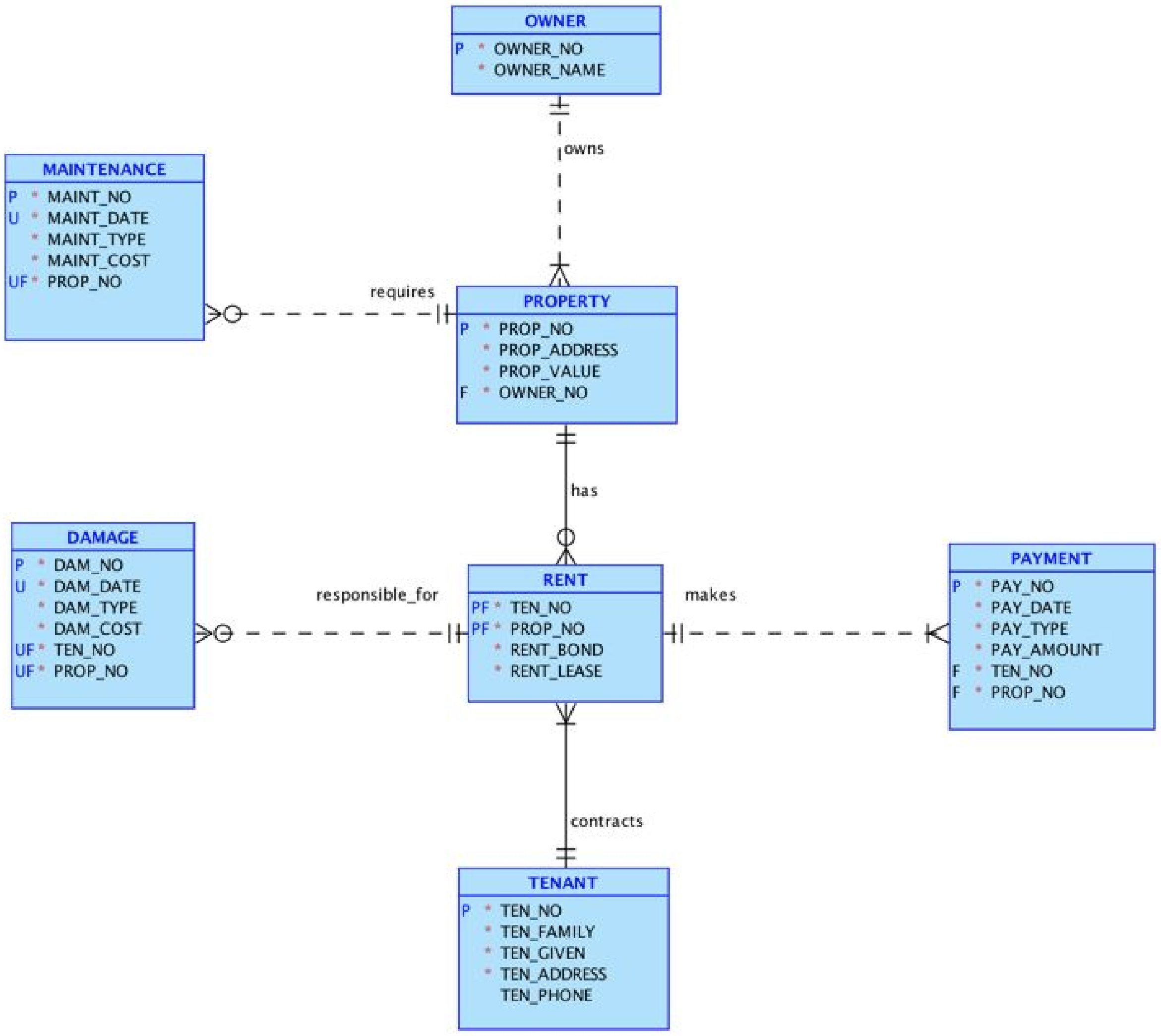
MAINTENANCE (<u>maint_no</u>, maint_date, maint_type, maint_cost, *owner_no*, *prop_no*)
prop_no -> owner_no is a transitive dependency so this entity is not in 3NF

The 3NF form would be:
MAINTENANCE (<u>maint_no</u>, maint_date, maint_type, maint_cost, *prop_no*)

PROPERTY (<u>prop_no</u>, owner_no)

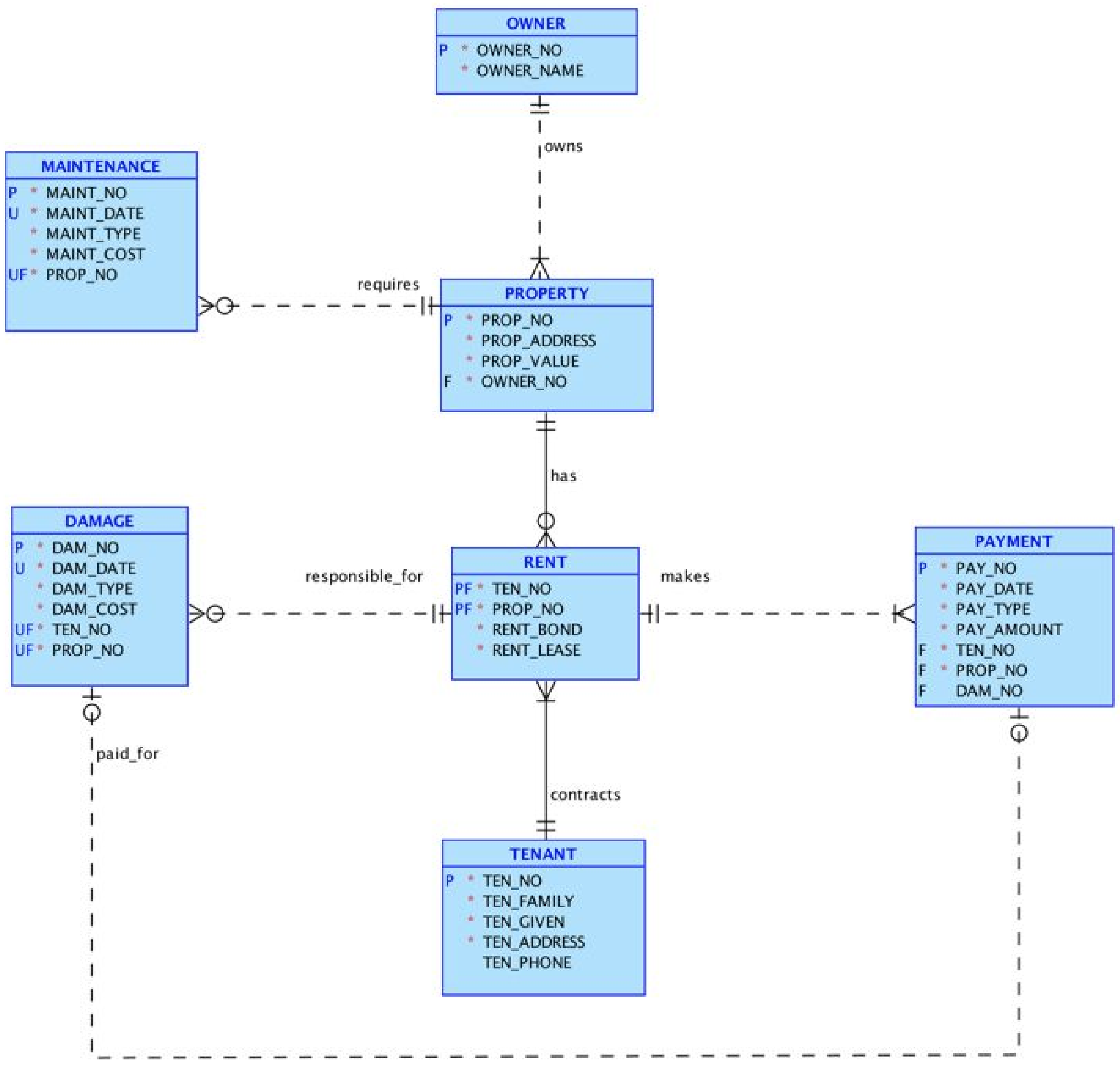
The relationship between MAINTENANCE and OWNER is a redundant relationship (see Coronel & Morris 5-4d (ed 12) 5.4.4 (ed 11)). Sometimes such relationships are deliberately left in the design as they simplify the design, here it would more easily allow a connection between maintenance and the owner who should be charged for the maintenance. Deciding to maintain such a relationship, involves overheads (here a redundant owner_no in Maintenance) and needs to be a clear and *documented* conscious decision.

The model without this relationship would be:



This model can be improved by connecting DAMAGE and PAYMENT - in this way you will be able to connect a given payment with a given act of damage. Here we will assume a business rule that says a damage if billed to a tenant must be paid in full in one payment.

An improved model to track payment for a damage:

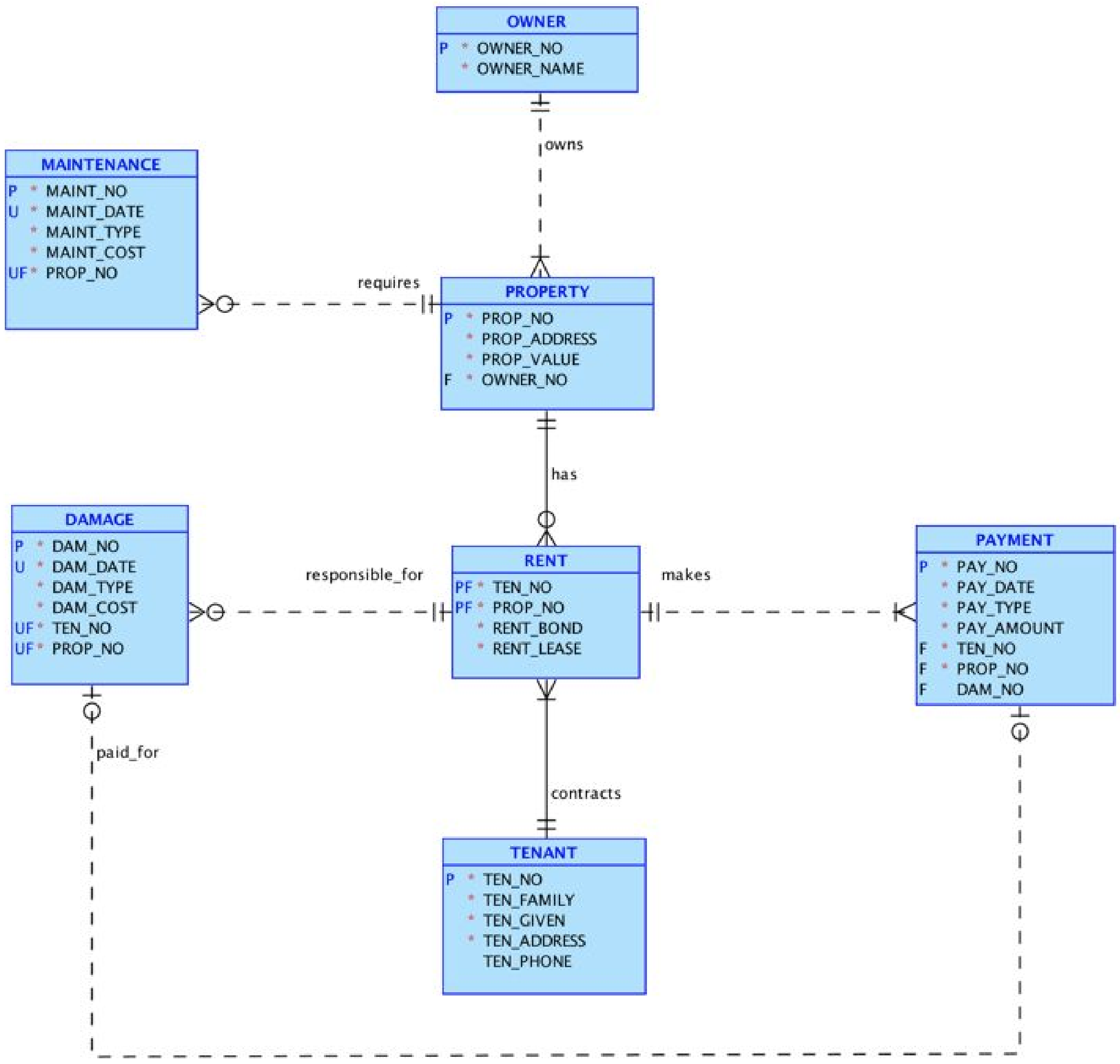


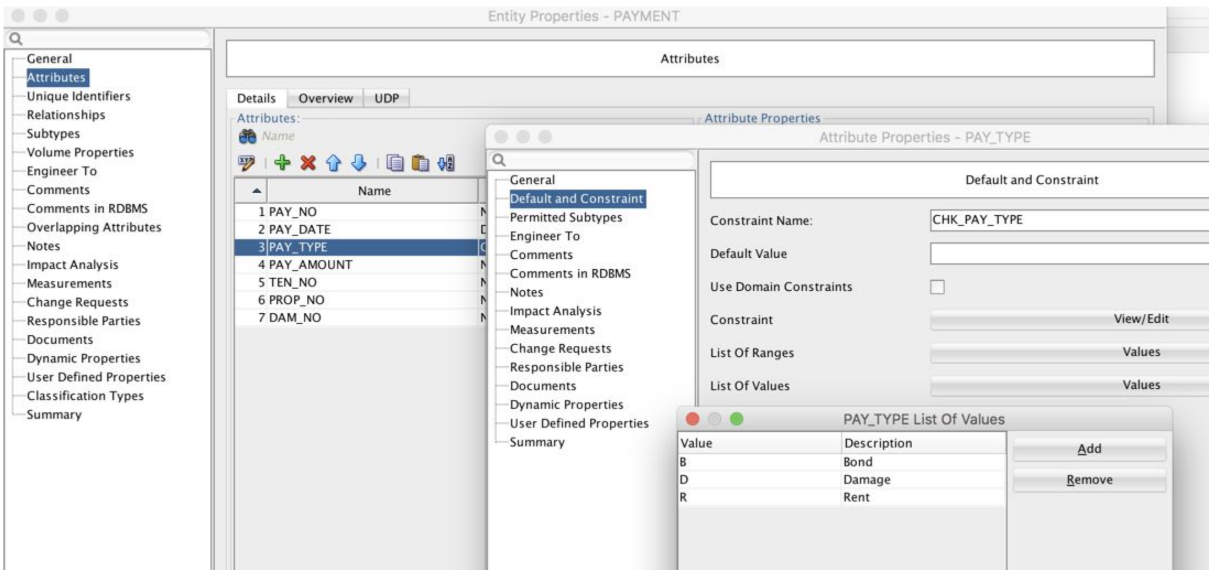
Part of the case study indicates that the payment type must be recorded as Rent, Bond or Damage. This should be enforced in one of two ways:

- by adding a CONSTRAINT (see Alexandria on how to add such a constraint), or
- by using a LOOKUP table.

A constraint becomes part of the database structure, if the user wishes to add a new option, say 'F' for future rent payments then the structure must be altered. On the other hand if a lookup table is used all that needs to happen is a new row needs to be added to the lookup table with the appropriate values.

Controlling pay_type via a CONSTRAINT (CHK_PAY_TYPE)





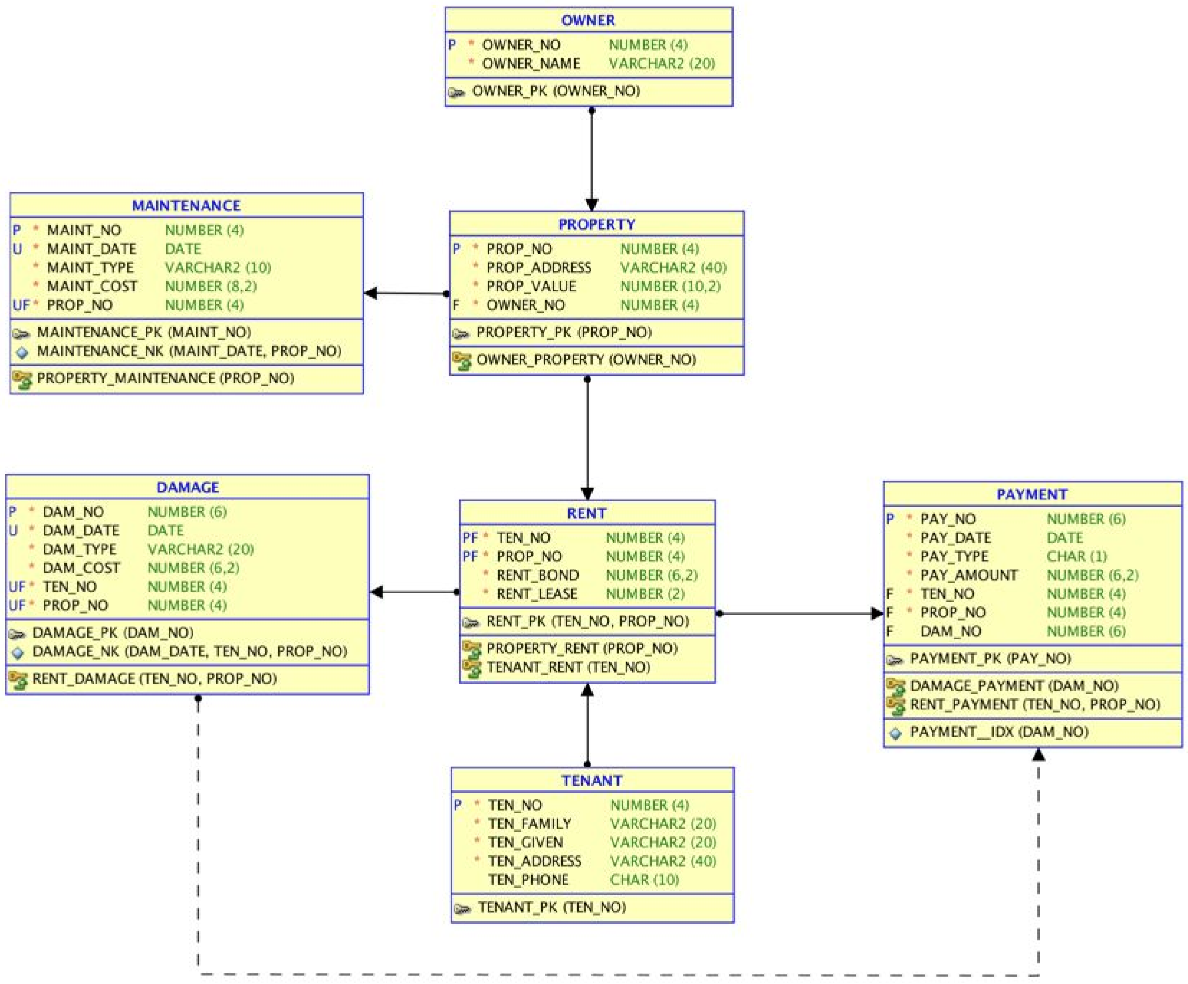
Attributes:

	Name
1	PAY_NO
2	PAY_DATE
3	PAY_TYPE
4	PAY_AMOUNT
5	TEN_NO
6	PROP_NO
7	DAM_NO

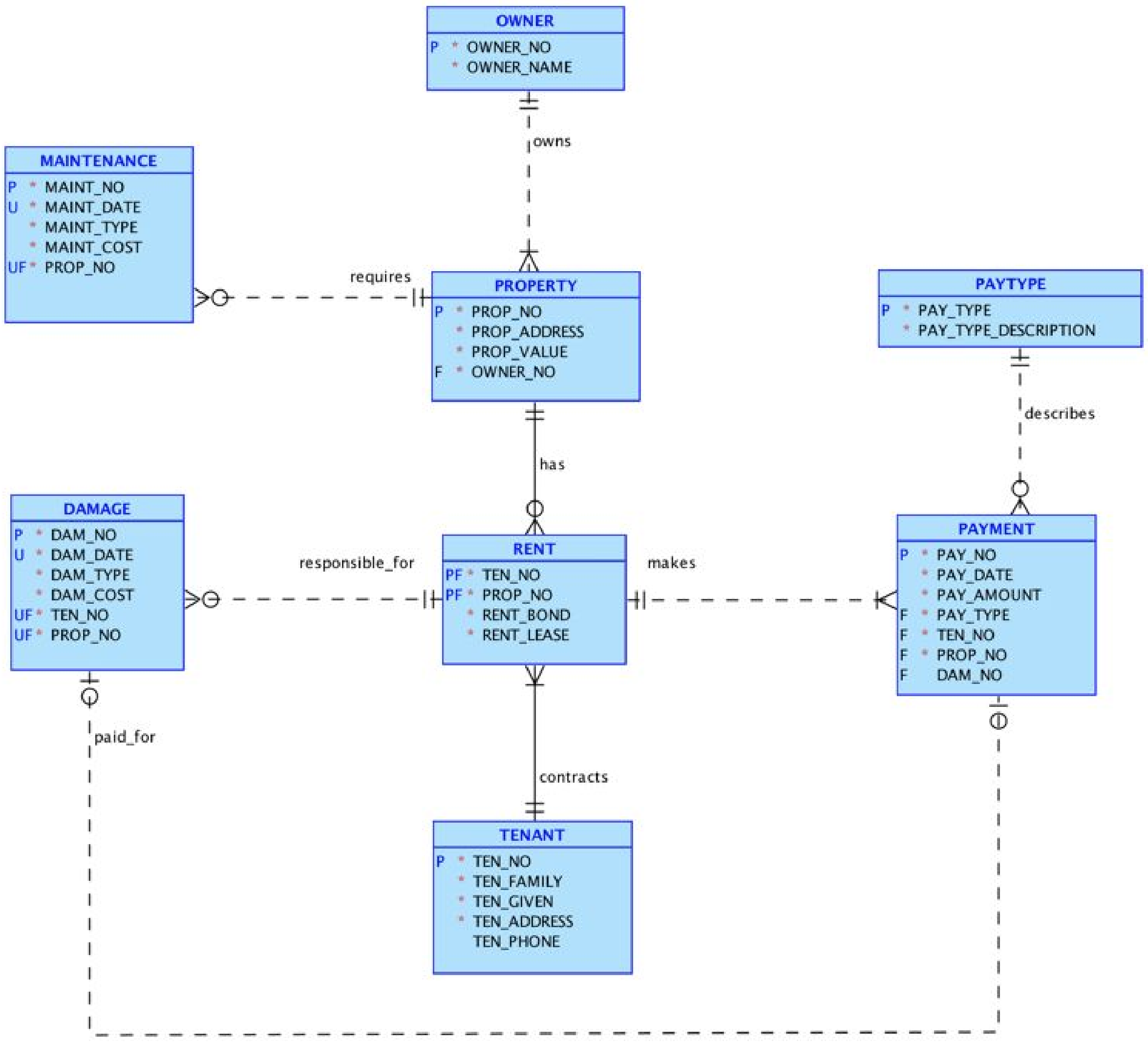
Default and Constraint

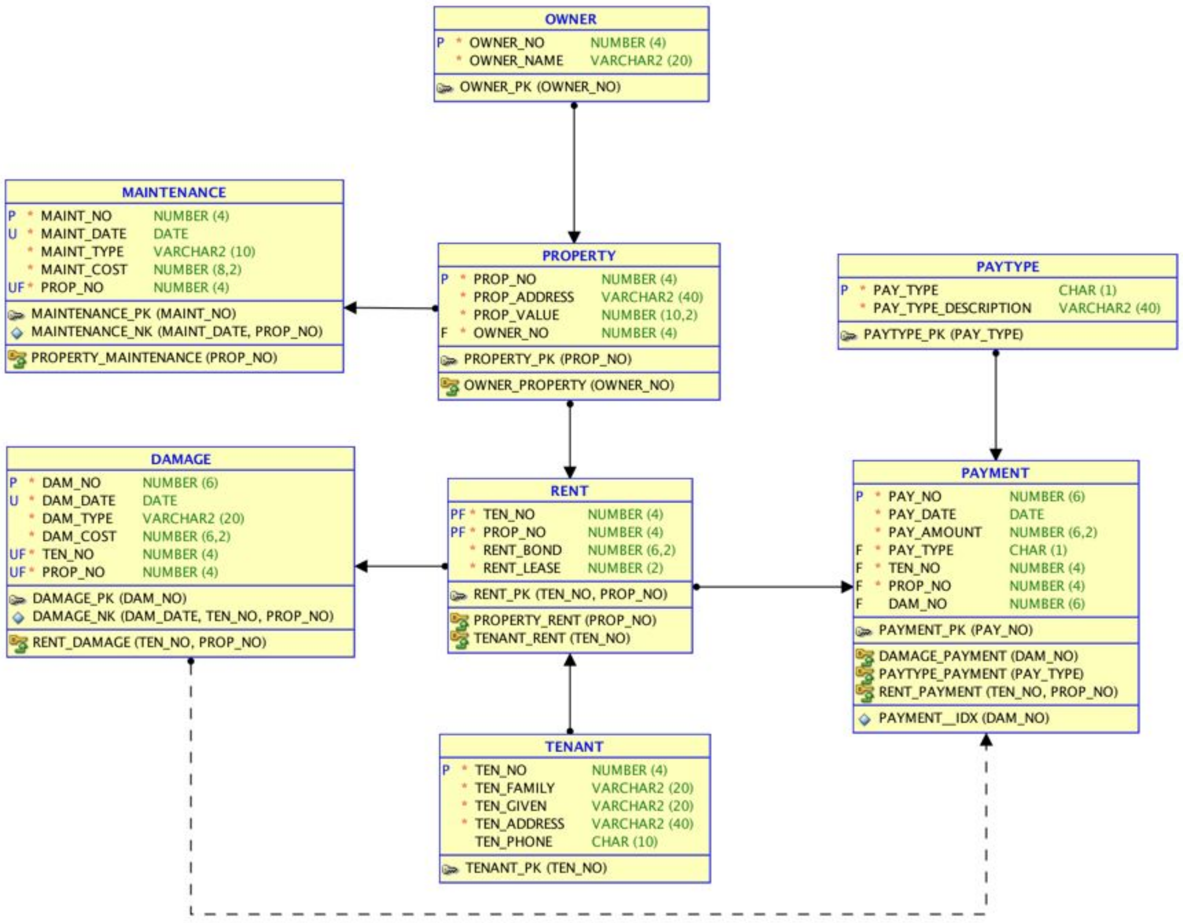
Constraint Name: CHK_PAY_TYPE
Default Value:
Use Domain Constraints: ☐
Constraint
List Of Ranges
List Of Values

Value	Description
B	Bond
D	Damage
R	Rent


OWNER
P * OWNER_NO NUMBER (4)
* OWNER_NAME VARCHAR2 (20)
OWNER_PK (OWNER_NO)
MAINTENANCE
P * MAINT_NO NUMBER (4)
U * MAINT_DATE DATE
* MAINT_TYPE VARCHAR2 (10)
* MAINT_COST NUMBER (8,2)
UF * PROP_NO NUMBER (4)
MAINTENANCE_PK (MAINT_NO)
MAINTENANCE_NK (MAINT_DATE, PROP_NO)
PROPERTY_MAINTENANCE (PROP_NO)
PROPERTY
P * PROP_NO NUMBER (4)
* PROP_ADDRESS VARCHAR2 (40)
* PROP_VALUE NUMBER (10,2)
F * OWNER_NO NUMBER (4)
PROPERTY_PK (PROP_NO)
OWNER_PROPERTY (OWNER_NO)
DAMAGE
P * DAM_NO NUMBER (6)
U * DAM_DATE DATE
* DAM_TYPE VARCHAR2 (20)
* DAM_COST NUMBER (6,2)
UF * TEN_NO NUMBER (4)
UF * PROP_NO NUMBER (4)
DAMAGE_PK (DAM_NO)
DAMAGE_NK (DAM_DATE, TEN_NO, PROP_NO)
RENT_DAMAGE (TEN_NO, PROP_NO)
RENT
PF * TEN_NO NUMBER (4)
PF * PROP_NO NUMBER (4)
* RENT_BOND NUMBER (6,2)
* RENT_LEASE NUMBER (2)
RENT_PK (TEN_NO, PROP_NO)
PROPERTY_RENT (PROP_NO)
TENANT_RENT (TEN_NO)
PAYMENT
P * PAY_NO NUMBER (6)
* PAY_DATE DATE
* PAY_TYPE CHAR (1)
* PAY_AMOUNT NUMBER (6,2)
F * TEN_NO NUMBER (4)
F * PROP_NO NUMBER (4)
F DAM_NO NUMBER (6)
PAYMENT_PK (PAY_NO)
DAMAGE_PAYMENT (DAM_NO)
RENT_PAYMENT (TEN_NO, PROP_NO)
PAYMENT__IDX (DAM_NO)
TENANT
P * TEN_NO NUMBER (4)
* TEN_FAMILY VARCHAR2 (20)
* TEN_GIVEN VARCHAR2 (20)
* TEN_ADDRESS VARCHAR2 (40)
TEN_PHONE CHAR (10)
TENANT_PK (TEN_NO)

Controlling pay_type via a LOOKUP TABLE (PAYTYPE)





The schema files for both approaches are provided as part of this solution.

To capture the full run of your schema file you must insert a SPOOL and ECHO on command at the top of your file, and a SPOOL OFF and ECHO off at the end of your script.

For example - added to the top of your script:

```
-- Capture run of script in file called rental-run.txt
set echo on
SPOOL rental-run.txt

-- Database Teaching Team
-- Logical Rental Model Schema script file
-- Schema file includes example constraint to control pay_type

-- Added drop commands for sequences
```

here add drop sequence commands, sql developer does not add these

….. the SQL Developer Generated Script goes here …...

```
-----------------------------------------------------
set echo off
SPOOL off
```

Please check the sample solution schema files to see this in action.