

(The answers are written in BLUE, the rest is mainly explanation to the answer)

COMP211 Database Design

Solution for Class Practice 7.1 Normalization

ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul 07	31-Aug-08	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-08	1-Sep-09	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-06	10-June-07	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-07	1-Dec-08	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-09	10-Aug-10	450	CO93	Tony Shaw

Given the above instance of ClientRental relation, illustrate the process of normalization by converting the table shown to Boyce-Codd Normal Form (BCNF). State any assumptions you make about the data shown in this table.

UNF:

ClientRental (clientNo, cName, propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

UNF to 1NF: To remove repeating groups (i.e. make sure that intersection of each row and column contains one and only one value)

The format of the resulting **1NF relations** is as follows (note that the PK has been changed):
ClientRental (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

1NF to 2NF: To remove partial dependencies

A relation with a single-attribute primary key, that is, without composite primary key, is automatically in at least 2NF.

Step 1. Identify the partial dependencies (i.e. functionally depends on part of the determinant) in the relation, if any.

Two partial dependencies are identified:

clientNo → cName

propertyNo → pAddress, rent, ownerNo, oName

~~rentStart → ?~~

(The answers are written in BLUE, the rest is mainly explanation to the answer)

The format of the resulting **2NF relations** is as follows:

ClientRental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

***Note:** In fact, it is possible to have rent in both PropertyOwner and ClientRental. The rent in PropertyOwner records the current rent of a property whereas the rent in ClientRental is the historical record of the rent in different periods of time.*

2NF to 3NF: To remove transitive dependencies

One transitive dependency is identified from PropertyOwner:

ownerNo \rightarrow oName

The format of the resulting **3NF relations** is as follows:

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

Owner (ownerNo, oName)

ClientRental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)

The normalization process has decomposed the original ClientRental relation using a series of steps. This results in a lossless-join decomposition, which is reversible using the natural join operation.

3NF to BCNF: To remove functional dependencies where the determinant is not a candidate key

Step 1. For each relation, identify all the functional dependencies and make sure that all the determinants are candidate keys. If there are determinants not being candidate key, remove them by placing them in a new relation

Client (clientNo, cName)

clientNo \rightarrow cName

- Already in BCNF as there is only a single determinant, which is the candidate key.

ClientRental (clientNo, propertyNo, rentStart, rentFinish)

clientNo, propertyNo, rentStart \rightarrow rentFinish

(The answers are written in BLUE, the rest is mainly explanation to the answer)

- Already in BCNF as all the determinants are candidate keys.

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

propertyNo \rightarrow pAddress, rent, ownerNo

- Already in BCNF as there is only a single determinant, which is the candidate key.

Owner (ownerNo, oName)

ownerNo \rightarrow oName

- Already in BCNF as there is only a single determinant, which is the candidate key.

The format of the resulting **BCNF relations** is as follows:

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

Owner (ownerNo, oName)

ClientRental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)