

(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)

Step 1. Identify key attribute for the ClientRental unnormalized table: **clientNo**

Step 2. Identify the repeating group:

(**propertyNo, pAddress, rentStart, rentFinish, rent, ownerNo, oName**)

Step 3. Remove the repeating group by

- Approach 1: 'Flattening' the table

ClientRental

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

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

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)

*rentStart is needed to be part of the composite primary key because a client can rent the same property some years later. For example, CR56 can rent PG16 in 2016 again.*

The above relation suffers from update anomalies. Suppose we need to change the name of client number CR56, we have to update 3 tuples. If only one tuple is updated, this results in an inconsistency in the database.

### 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. We will use the answer of approach 1 above to demonstrate the process of normalization.

1NF:

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

i.e.  $\text{clientNo}, \text{propertyNo}, \text{rentStart} \rightarrow \text{cName}, \text{pAddress}, \text{rentFinish}, \text{rent}, \text{ownerNo}, \text{oName}$

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:

$\text{clientNo} \rightarrow \text{cName}$

$\text{propertyNo} \rightarrow \text{pAddress}, \text{rent}, \text{ownerNo}, \text{oName}$

~~$\text{rentStart} \rightarrow ?$~~

Step 2. If partial dependencies exist on the primary key, remove them by placing them in a new relation along with a copy of their determinant.

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

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

ClientRental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)

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

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

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

**Figure 14.14** Second normal form relations derived from the ClientRental relation.

*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.*

The PropertyOwner relation from 2NF above still suffers from update anomalies. If we want to update the name of an owner, we have to update more than one tuples. Failing to update all related records will result in an inconsistent state.

### 2NF to 3NF: To remove transitive dependencies

Step 1. Identify the transitive dependencies in the relation, if any. In other words, look among the non-primary-key attributes to see if any functional dependencies can be derived.

One transitive dependency is identified from PropertyOwner:  
ownerNo → oName

Step 2. If transitive dependencies exist on the primary key, remove them by placing them in a new relation along with a copy of their determinant.

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

Client		Rental			
clientNo	cName	clientNo	propertyNo	rentStart	rentFinish
CR76	John Kay	CR76	PG4	1-Jul-07	31-Aug-08
CR56	Aline Stewart	CR76	PG16	1-Sep-08	1-Sep-09
		CR56	PG4	1-Sep-06	10-Jun-07
		CR56	PG36	10-Oct-07	1-Dec-08
		CR56	PG16	1-Nov-09	10-Aug-10

PropertyForRent				Owner	
propertyNo	pAddress	rent	ownerNo	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93		

**Figure 14.17** A summary of the 3NF relations derived from the ClientRental relation.

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

- 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

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

- 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)