Database Systems 2

Lecture 18

Normalisation

The Process of Normalization

Formal technique for analyzing a relation based on its primary key and functional dependencies between its attributes.

Often executed as a series of steps. Each step corresponds to a specific normal form, which has known properties.

As normalization proceeds, relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.

Unnormalized Form (UNF)

A table that contains one or more repeating groups.

To create an unnormalized table:

- transform data from information source (e.g. paper based form) into table format with columns and rows.

First Normal Form (1NF)

A relation in which:

- the intersection of each row and column contains one and only one value, (i.e. is atomic)

and:

- One or more of the columns is designated as the primary key.

To get from UNF to 1NF

Identify repeating group(s) in unnormalized table which repeats for the key attribute(s).

Nominate an attribute (or group of attributes) to act as the key for the unnormalized table.

Get rid of repeating group

Remove repeating group by:

- entering appropriate data into the empty columns of rows containing repeating data (‘flattening’ the table).

- Note: some texts separate the repeating group out to it's own table at this stage, although technically it should be done when you go from 1NF to 2NF.

ClientRental UNF To 1NF By Flattening

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
		PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
		PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
		PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	**John Kay**	PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	**Aline Stewart**	PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	**Aline Stewart**	PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

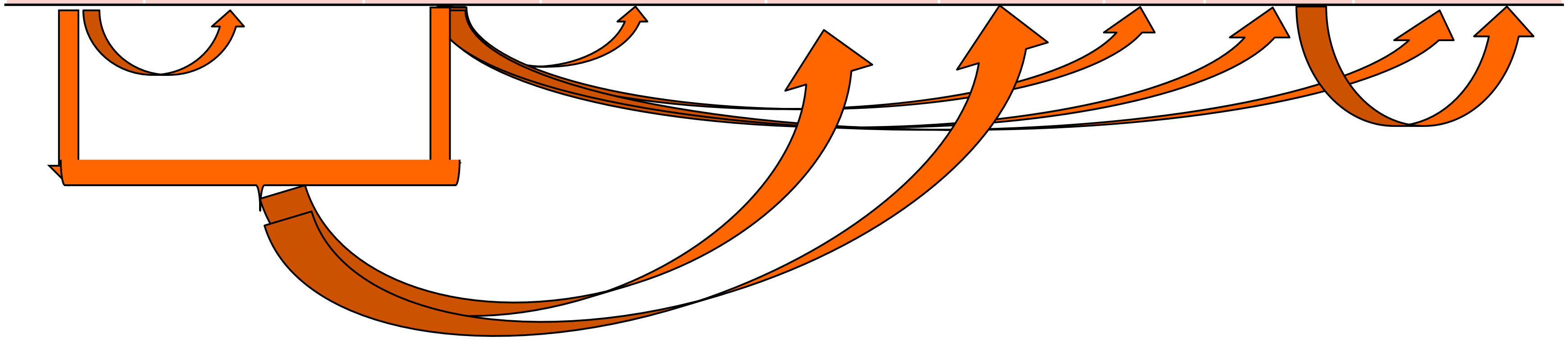
Now choose a primary key

Might be fairly easy to spot.

If it isn't, got through the following process.

ClientRental Functional Dependencies

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	John Kay	PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	Aline Stewart	PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	Aline Stewart	PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw



What about rentStart and rentFinish?

ClientRental Functional Dependencies

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	John Kay	PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	Aline Stewart	PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	Aline Stewart	PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

- clientNo → cName
- propertyNo → pAddress, rent, ownerNo, oName
- ownerNo → oName
- clientNo, propertyNo → cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName

There are other possible dependencies:

- clientNo, rentStart → cName, propertyNo, pAddress, rentFinish, rent, ownerNo, oName
- propertyNo, rentStart → clientNo, cName, pAddress, rentFinish, rent, ownerNo, oName

ClientRental Functional Dependencies

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	John Kay	PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	Aline Stewart	PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	Aline Stewart	PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

No one field can act as a primary key for the whole table:

What are the possible composite keys?

ClientRental Primary Key

Candidate keys are

- (clientNo, propertyNo)
- (clientNo, rentStart)
- (propertyNo, rentStart)

Choose (clientNo, propertyNo) as Primary key

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	John Kay	PG16	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	Aline Stewart	PG36	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	Aline Stewart	PG16	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

Client rental 1NF relation

Write down the ClientRental relation using standard notation

ClientRental (<u>clientNo, propertyNo</u>, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

WE NOW HAVE A TABLE IN 1NF.

Second Normal Form (2NF)

Based on concept of full functional dependency:

- $A_1, \ldots, A_n$ and B are attributes of a relation,
- B is fully dependent on $A_1, \ldots, A_n$ if B is functionally dependent on $A_1, \ldots, A_n$ but not on any proper subset of $A_1, \ldots, A_n$.

2NF - A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the whole primary key.

Getting from 1NF to 2NF

Identify primary key for the 1NF relation.

Identify functional dependencies in the relation.

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

ClientRental Example: 1NF to 2NF

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	owner No	oName
CR76	PG4	John Kay	6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

Partial dependencies are:

- clientNo → cName
- propertyNo → pAddress, rent, ownerNo, oName

ClientRental Example: *clientNo → cName*

Create new relation **Client**, with primary key clientNo

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Remove cName from the **ClientRental** relation

clientNo	propertyNo		pAddress	rentStart	rentFinish	rent	owner No	oName
CR76	PG4		6 Lawrence St	1-Jul-00	31-Aug-01	350	CO40	Tina Murphy
CR76	PG16		5 Novar Dr	1-Sep-02	1-Sep-02	450	CO93	Tony Shaw
CR56	PG4		6 Lawrence St	1-Sep-99	10-Jun-00	350	CO40	Tina Murphy
CR56	PG36		2 Manor Rd	10-Oct-00	1-Dec-01	375	CO93	Tony Shaw
CR56	PG16		5 Novar Dr	1-Nov-02	10-Aug-03	450	CO93	Tony Shaw

ClientRental Example: propertyNo -> pAddress, rent, ownerNo, oName

Create new relation **PropertyOwner**, with primary key propertyNo

<u>propertyNo</u>	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St	350	CO40	Tina Murphy
PG16	5 Novar Dr	450	CO93	Tony Shaw
PG36	2 Manor Rd	375	CO93	Tony Shaw

Remove attributes pAddress, rent, ownerNo, oName from the **ClientRental** relation

<u>clientNo</u>	<u>propertyNo</u>			rentStart	rentFinish			
CR76	PG4			1-Jul-00	31-Aug-01			
CR76	PG16			1-Sep-02	1-Sep-02			
CR56	PG4			1-Sep-99	10-Jun-00			
CR56	PG36			10-Oct-00	1-Dec-01			
CR56	PG16			1-Nov-02	10-Aug-03			

ClientRental Example: 2NF relations

Tidy up, and re-name the ClientRental relation “Rental”

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-00	31-Aug-01
CR76	PG16	1-Sep-02	1-Sep-02
CR56	PG4	1-Sep-99	10-Jun-00
CR56	PG36	10-Oct-00	1-Dec-01
CR56	PG16	1-Nov-02	10-Aug-03

Write down the 2NF relations:

Client (clientNo, cName)
PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)
Rental (clientNo, propertyNo, rentStart, rentFinish)

Third Normal Form (3NF)

Based on concept of transitive dependency:

- A, B and C are attributes of a relation such that A → B and B → C,
- then C is transitively dependent on A through B. (Provided that A is not functionally dependent on B or C).

3NF - A relation that is in 1NF and 2NF and in which no non-primary-key attribute is transitively dependent on the primary key.

Getting from 2NF to 3NF

Identify the primary key in the 2NF relation.

Identify functional dependencies in the relation.

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

ClientRental Example: 2NF to 3NF

Consider the relation:

PropertyOwner (<u>propertyNo</u>, pAddress, rent, ownerNo, oName)

We have functional dependencies

- propertyNo → ownerNo
- ownerNo → oName

So oName is transitively dependent on propertyNo, the primary key

ClientRental Example: Remove Transitive Dependency On Primary Key

Create new relation **Owner**, with primary key ownerNo and attribute oName

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

Remove oName from PropertyOwner relation

propertyNo	pAddress	rent	ownerNo	
PG4	6 Lawrence St	350	CO40	
PG16	5 Novar Dr	450	CO93	
PG36	2 Manor Rd	375	CO93	

ClientRental Example: 3NF Relations

Tidy up, and re-name PropertyOwner relation "PropertyForRent"

propertyNo	pAddress	rent	ownerNo*
PG4	6 Lawrence St	350	CO40
PG16	5 Novar Dr	450	CO93
PG36	2 Manor Rd	375	CO93

Write down the 3NF relations:

Client (clientNo, cName)
Rental (clientNo, propertyNo, rentStart, rentFinish)
PropertyOwner (propertyNo, pAddress, rent, ownerNo)
Owner (ownerNo, oName)

How To Show it in the Exam

We can represent normalisation:

- with text
- with a diagram
- with a table

Zero Normal Form

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

First Normal Form

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

Second Normal Form

Client (clientNo, cName)

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

Rental (clientNo*, propertyNo*, rentStart, rentFinish)

Third Normal Form

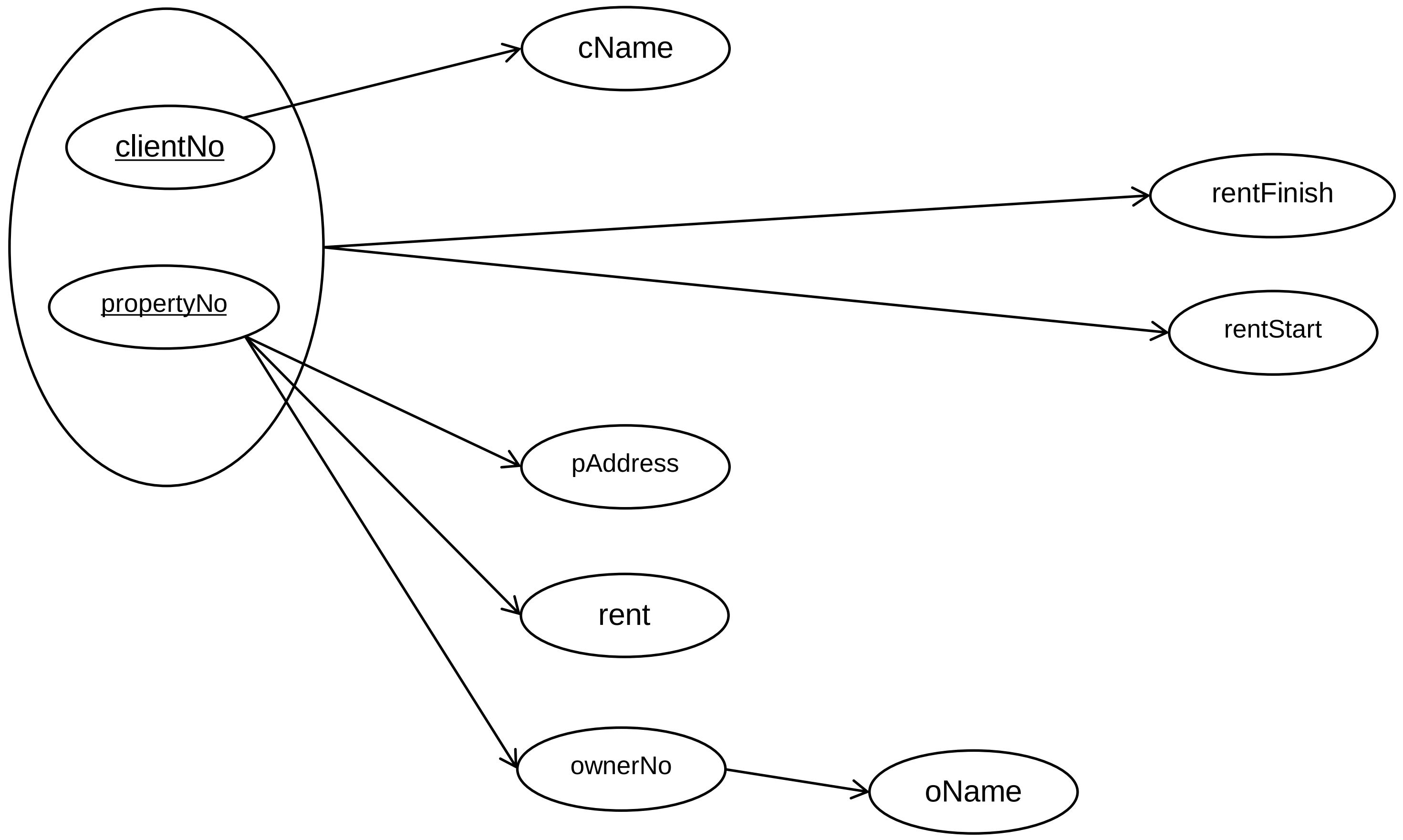
Client (clientNo, cName)

Rental (clientNo, propertyNo, rentStart, rentFinish)

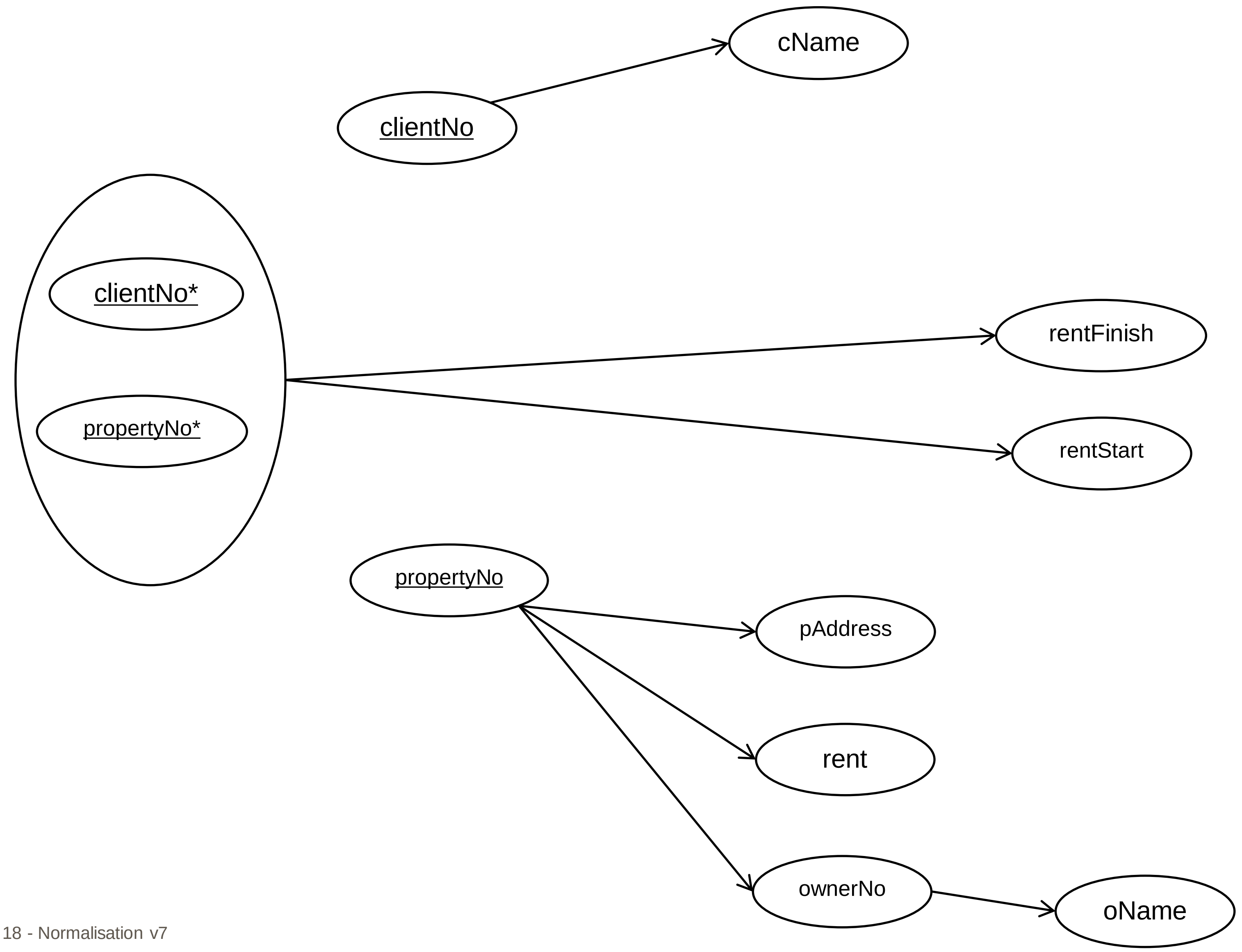
PropertyOwner (propertyNo, pAddress, rent, ownerNo*)

Owner (ownerNo, oName)

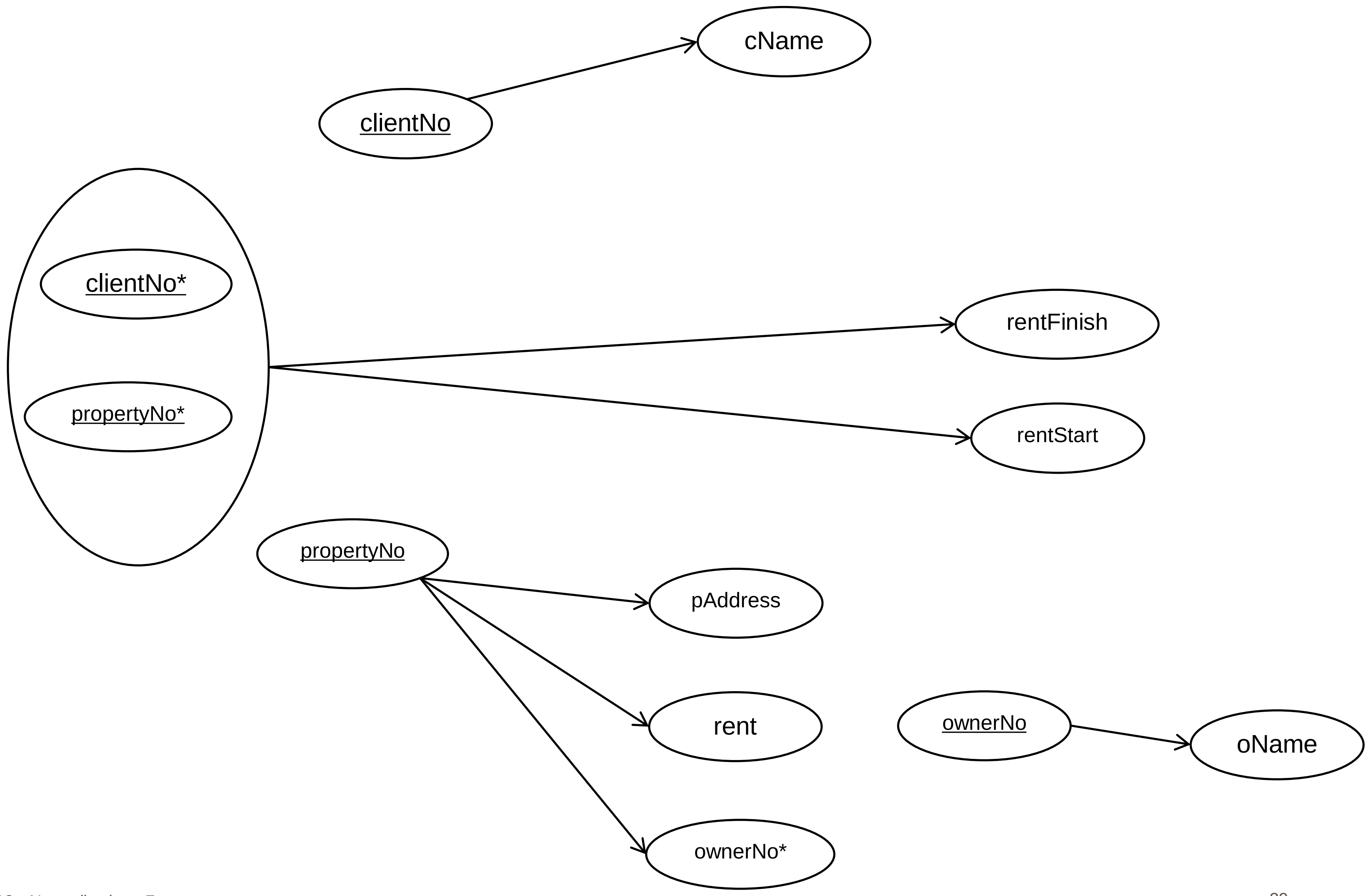
1NF


clientNo
propertyNo
cName
rentFinish
rentStart
pAddress
rent
ownerNo
oName

2NF


cName
clientNo
clientNo*
propertyNo*
rentFinish
rentStart
propertyNo
pAddress
rent
ownerNo
oName

3NF


clientNo
cName
clientNo*
propertyNo*
rentFinish
rentStart
propertyNo
pAddress
rent
ownerNo*
ownerNo
oName

0NF	1NF	2NF	3NF	
clientNo	clientNo	clientNo*	clientNo*	Rental
propertyNo	propertyNo	propertyNo*	propertyNo*	
cName	cName	rentStart	rentStart	
pAddress	pAddress	rentFinish	rentFinish	
rentStart	rentStart			
rentFinish	rentFinish			
Rent	Rent			
OwnerNo	OwnerNo			
oName	oName			
		clientNo	clientNo	Client
		cName	cName	
		propertyNo	propertyNo	PropertyOwner
		pAddress	pAddress	
		Rent	Rent	
		OwnerNo	OwnerNo*	
		oName		
			OwnerNo	Owner
			oName	

Exam Questions

Boyce–Codd Normal Form (BCNF)

Based on functional dependencies that take into account all candidate keys in a relation, however BCNF also has additional constraints compared with general definition of 3NF.

BCNF - A relation is in BCNF if and only if <u>every</u> determinant in the relation is a candidate key for the whole relation.

Boyce–Codd normal form (BCNF)

For a relation with only one candidate key, 3NF and BCNF are equivalent.

To test whether a relation is in BCNF, we identify all the determinants and make sure that they are all candidate keys for the whole relation.

If they aren't, this can give rise to the following situation.

Boyce–Codd normal form (BCNF)

It may be possible for there to be a functional dependency A $\rightarrow$ B in a relation, such that:

B is part of a primary key attribute, and A is not a candidate key:

3NF will allow this. BCNF will not.

BCNF requires that you remove that dependency by separating out into a further table.

Boyce–Codd normal form (BCNF)

Violation of BCNF is quite rare.

Potential to violate BCNF may occur in a relation that:

- contains two (or more) composite candidate keys;
- the candidate keys overlap (ie. have at least one attribute in common).
- there is a functional dependency in a relation whose determinant is not a candidate key, but whose dependent is part of one of the candidate keys.

BCNF Example

Consider the relation:

ClientInterview (<u>clientNo</u>, <u>interviewdate</u>, interviewTime, staffNo, roomNo)

<u>clientNo</u>	<u>interviewDate</u>	interviewTime	staffNo	roomNo
CR76	13-May-02	10:30	SG5	G101
CR56	13-May-02	12:00	SG5	G101
CR74	13-May-02	12:00	SG37	G102
CR56	01-Jul-02	10:30	SG5	G102

A member of staff will be allocated a given room, during the day, while they are interviewing clients.

A room may be allocated to several members of staff throughout the day.

A client is only interviewed once on a given date, but maybe called back for other interviews.

The member of staff may be given a different room on a different day.

BCNF Example: Functional dependencies

clientNo	interviewDate	intervie		
CR76	13-May-02	10:30		
CR56	13-May-02	12:00		
CR74	13-May-02	12:00		G102
CR56	01-Jul-02	10:30	SG5	G102

There are three candidate keys for this relation. They all overlap on the interviewDate attribute.

Functional dependencies are

fd1: clientNo, interviewdate → interviewTime, staffNo,roomNo

fd2: staffNo, interviewdate, interviewTime → clientNo, roomNo

fd3: roomNo, interviewdate, interviewTime → staffNo,clientNo

fd4: staffNo, interviewdate → roomNo

BCNF Example

ClientInterview relation is in 3NF

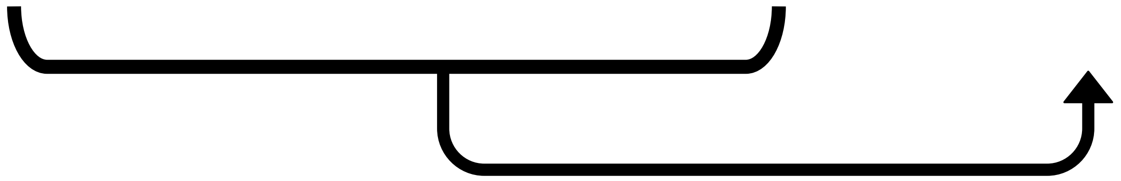
Is ClientInterview relation in BCNF?

fd1:	the determinant, (clientNo, interviewdate), is the primary key
fd2 and fd3:	both determinants are candidate keys for the whole table
fd4:	its determinant (staffNo, InterviewDate) is NOT a candidate key for the whole table, but it's dependent (room_No) IS part of one of the other candidate keys.

It is fd4 that will give rise to the possibility of update anomalies:

fd4: staffNo, interviewdate → roomNo

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-02	10:30	SG5	G101
CR56	13-May-02	12:00	SG5	G101
CR74	13-May-02	12:00	SG37	G102
CR56	01-Jul-02	10:30	SG5	G102



Where is the redundancy?

BCNF Example

Create new relation **StaffRoom** with the attributes from fd4:

- Determinant attributes form the primary key of the new relation
- Include dependent attributes in the relation

Now can you see the duplication?

interviewDate	staffNo	roomNo
13-May-02	SG5	G101
13-May-02	SG5	G101
13-May-02	SG37	G102
01-Jul-02	SG5	G102

interviewDate	staffNo	roomNo
13-May-02	SG5	G101
13-May-02	SG37	G102
01-Jul-02	SG5	G102

BCNF Example

Remove the dependent attributes of fd4 from the ClientInterview relation and rename it **Interview.**

<u>clientNo</u>	<u>interviewDate</u>	interviewTime	staffNo
CR76	13-May-02	10:30	SG5
CR56	13-May-02	12:00	SG5
CR74	13-May-02	12:00	SG37
CR56	01-Jul-02	10:30	SG5

Write down the BCNF relations:

StaffRoom (<u>staffNo</u>, <u>interviewdate</u>, roomNo)

Interview (<u>clientNo</u>, <u>interviewdate</u>, interviewTime, staffNo)