INFSCI 1022
Database Management Systems

Today’s Evil Plan

- Learn about what is ‘NORMAL’ when it comes to databases
- Attribute dependencies
- Normal forms 1-5

Excel != Database

Format Painter | Formatting | Table | Styles | Clear | Filter | Select

Clipboard | Font | Alignment | Number | Styles | Cells | Editing

D94 | Congenital anomalies

	A	B	C	D	E	F	G	H
1	**FieldType**	**DataType**	**Data**	**DataField**	**FieldOption**	**Where from**	**What we broke by re-desgning the deliv**	
2	textfield	Text	Alpha and Freetext:	Place of birth	Hospital			
3				Place of birth	Birth Center			
4				Place of birth	clinic/office			
5				Place of birth	home birth planned			
6				Place of birth	home birth, unplanned			
7				Place of birth	other			
8			Date:	Date of first prenatal visit		Epic		
9			Date:	date of last prenatal visit		Epic		
10			Numeric:	number of prenatal visits		Epic		
11			Date:	LMP		PCM EDD Control		
12			Numeric:	living children		PCM Pregnancy History cont	x	
13			Date:	date of last live birth		PCM Pregnancy History control		
14			Numeric:	live births now deceased		PCM Pregnancy History control		
15			Numeric:	total number of other pregnancy outcomes		PCM Pregnancy History control		
16			Date:	date of last Other pregnancy outcome		PCM Pregnancy History control		
17			Numeric:	mother's weight at delivery	LBs NOT kg	admission assessment form		
18				pregnancy risk factors	GDM	dta	x	
19					Pre-pregnancy Diabetes	dta	x	
20					G HTN	dta	x	
21					HTN prior to pregnancy	dta	x	
22					previous preterm delivery	PCM Pregnancy History cont	x	
23					previous poor pregnancy outcome	PCM Pregnancy History cont	x	
24					pregnancy resulted from infertility treatm	DTA	x	
25					vaginal bleeding prior to onset of labor	DTA	x	
26					prev CS	PCM Pregnancy History cont	x	
27					# of prev CS	PCM Pregnancy History control		
28				Infections	Herpes	dta	x	
29					Chlamydia	dta	x	
30					Gonorrhae	dta	x	

Sheet1 | Birth cert | del form | operative del | shoulder | MD assessment | Anesth | powern ...

Format Painter
Clipboard | Font | Alignment | Number | Formatting | Table | Styles | Cells | Clear | Filter | Select | Editing

C56 | spontaneous labor augmented

	A	B	C	D	E	F	G	H
1		DTA type						
2								
3								
4								
5								
6								
7								
8								
9			chorio		659.33 ... antepartum condition or complication			
10			prolonged ROM	SROM	658.21 ... delivered, with or without mention of antepartum condition			
11				AROM	658.31 ... delivered, with or without mention of antepartum condition			
12			did this fetus require intrauterine resuscitive measures durnig labor?	y/n				
13								
14								
15			if GBS pos	Treated< 4 hours prior to delivery				
16				Treated > 4 hours prior to delivery				
17				untreated				
18			Past pregnancy complications	None				
19				3-4th degree lac				
20				Blood transfusion				
21				Eclampsia				
22				CS				
23				PPH				
24				GDM				
25				Fetal anomaly				
26				Pre-eclampsia				
27				Pre-gest DM				
28				Pre-term labor				
29				PPROM				

Sheet1 | Birth cert | del form | operative del | shoulder | MD assessment | Anesth | powern(...

G158 x

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	DTA Type		Section Name	DTA Display Name	Nomenclature	ICD-9 Code	L+D ST	del note	Anesth	PP ST	Antepartum	Pass to peds	Pu fo thi
2	DTA type	source					Demographic						
3	Calculation:		Essential elements	Age			Gyn	x	x	x	x	x	x
4	Calculation:			G			Infectious	x	x			x	x
5	Calculation:			P			Hematological	x	x				x
6	Calculation:			EDC			Medical	x	x	x	x	x	x
7	Alpha:			multiple Gestation	Singleton		Social	x	x	x	x	x	x
8					twins	651.03 ... antepartum condition or complication	OB	x	x	x	x	x	x
9					triplets	651.43 ... antepartum condition or complication		x	x	x	x	x	x
10					quads	651.23 ... antepartum condition or complication		x	x	x	x	x	x
11	Numeric:		for each fetus (per multiples above)	EFW (Estimated Fetal Weight)				x	x		x		x
12	Alpha and Freetext:		for each fetus (per multiples above)	Position	vertex/breech/etc.			x	x	x	x	x	x
13	Alpha:			Cx Exam	dilation/effacement /station		x		x		x		
14	Numeric:			BP			x	x	x	x	x		
15	Alpha and Freetext:			pre-E sympt	HA		x	x	x	x	x		
16					Vis changes		x	x	x				
17					N/V		x	x	x				
18					Abd pain								
19													

Pitfalls in Relational Database Design

- Relational database design requires that we find a "good" collection of relation schemas (tables).
- A bad design may lead to
  – Repetition of Information.
  – Inability to represent certain information.

Design Goals

- Avoid redundant data
- Ensure that relationships among **entities** are represented
- Ensure that relationships among **attributes** are represented
- Facilitate the checking of updates for violation of database integrity constraints.

Example

- Consider the relation schema:

Lending-schema = (branch-name, branch-city, assets, customer-name, loan-number, amount)

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	Brooklyn	9000000000	Jones	L-17	1000
Redwood	Palo Alto	2100000000	Smith	L-23	2000
Perryridge	Horseneck	1700000000	Hayes	L-15	1500
Downtown	Brooklyn	9000000000	Jackson	L-14	1500

What's Wrong?

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	**Brooklyn**	**9000000000**	Jones	L-17	1000
Redwood	Palo Alto	2100000000	Smith	L-23	2000
Perryridge	Horseneck	1700000000	Hayes	L-15	1500
Downtown	**Brooklyn**	**9000000000**	Jackson	L-14	1500

- Redundancy:
  - Data for *branch-name, branch-city,* assets are repeated for each loan that a branch makes
  - Wastes space
  - Complicates updating, introducing possibility of inconsistency of *assets* value

What's Wrong?

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	Brooklyn	9000000000	Jones	L-17	1000
Redwood	Palo Alto	2100000000	Smith	L-23	2000
Perryridge	Horseneck	1700000000	Hayes	L-15	1500
Downtown	Brooklyn	9000000000	Jackson	L-14	1500
Midtown	**Pittsburgh**	**2349000000**	**NULL**	**NULL**	**NULL**

- Null values
  - Cannot store information about a branch if no loans exist
  - Can use null values, but they are difficult to handle.

Decomposition

- Decompose the relation schema *Lending-schema* into:
  - *Branch-schema = (branch_name, branch_city, assets)*
  - *Loan-info-schema = (customer_name, loan_number, branch_name, amount)*
- All attributes of an original schema (*R)* must appear in the decomposition (R_1, R_2): $\boldsymbol{R = R_1 \cup R_2}$

Decompositon

branch-name	branch-city	assets	customer-name	loan-number	amount
Downtown	Brooklyn	9000000000	Jones	L-17	1000
Redwood	Palo Alto	2100000000	Smith	L-23	2000
Perryridge	Horseneck	1700000000	Hayes	L-15	1500
Downtown	Brooklyn	9000000000	Jackson	L-14	1500

branch-name	branch-city	assets
Downtown	Brooklyn	9000000000
Redwood	Palo Alto	2100000000
Perryridge	Horseneck	1700000000

branch-name	customer-name	loan-number	amount
Downtown	Jones	L-17	1000
Redwood	Smith	L-23	2000
Perryridge	Hayes	L-15	1500
Downtown	Jackson	L-14	1500

Dependency

- A dependency occurs in a database when information stored in the same database table uniquely determines other information stored in the same table.
- You can also describe this as a relationship where knowing the value of one attribute (or a set of attributes) is enough to tell you the value of another attribute (or set of attributes) in the same table.

Functional Dependency

- Dependency = Functional Dependency
- If there is a dependency in a database such that attribute B is dependent upon attribute A, you would write this as "A -> B".

Functional Dependency Example

In a table listing employee characteristics including Social Security Number (SSN) and name, it can be said that name is dependent upon SSN (or SSN -> name) because an employee's name can be uniquely determined from their SSN. However, the reverse statement (name -> SSN) is not true because more than one employee can have the same name but different SSNs.

http://databases.about.com/od/specificproducts/a/Database-Dependency.htm

Functional Dependencies

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

Consider this relation – can you identify at least one functional dependency here?

Functional Dependencies

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

The functional dependency **{Book}** → **{Author}** applies; that is, if we know the book, we know the author

Trivial Functional Dependencies

- Occurs when you describe a functional dependency of an attribute on a collection of attributes that includes the original attribute.
- For example, “{A, B} -> B” is a trivial functional dependency, as is “{name, SSN} -> SSN”.
- This type of functional dependency is called trivial because it can be derived from common sense. It is obvious that if you already know the value of B, then the value of B can be uniquely determined by that knowledge.

Full Functional Dependencies

- Occur when you already meet the requirements for a functional dependency and the set of attributes on the left side of the functional dependency statement cannot be reduced any farther.
- For example, “{SSN, age} -> name” is a functional dependency, but it is not a full functional dependency because you can remove age from the left side of the statement without impacting the dependency relationship.

Transitive Dependencies

- Occur when there is an indirect relationship that causes a functional dependency.
- For example, ”A -> C” is a transitive dependency when it is true only because both “A -> B” and “B -> C” are true.

Transitive Dependencies

Book	Genre	Author	Author Nationality
Twenty Thousand Leagues Under the Sea	Science Fiction	Jules Verne	French
Journey to the Center of the Earth	Science Fiction	Jules Verne	French
Leaves of Grass	Poetry	Walt Whitman	American
Anna Karenina	Literary Fiction	Leo Tolstoy	Russian
A Confession	Religious Autobiography	Leo Tolstoy	Russian

Consider this relation – can you identify a transitive dependency here?

Transitive Dependencies

The functional dependency **{Book} → {Author Nationality}** applies; that is, if we know the book, we know the author's nationality. Furthermore:

- {Book} → {Author}
- {Author} does not → {Book}
- {Author} → {Author Nationality}

Therefore {Book} → {Author Nationality} is a transitive dependency.

Transitive dependency occurred because a non-key attribute (Author) was determining another non-key attribute (Author Nationality).

Multivalued Dependencies

- Occur when the presence of one or more rows in a table implies the presence of one or more other rows in that same table.
- For example, imagine a car company that manufactures many models of car, but always makes both red and blue colors of each model.
  - If you have a table that contains the model name, color and year of each car the company manufactures, there is a multivalued dependency in that table.
  - If there is a row for a certain model name and year in blue, there must also be a similar row corresponding to the red version of that same car.

Multivalued Dependencies

Make	Model	Color	Year
Ford	Focus	Blue	2014
Ford	Focus	Red	2014
Ford	Explorer	Blue	2015
Ford	Explorer	Blue	2015

Importance of Dependencies

- Dependencies provide the **basic building blocks used in database normalization**.
  - For a table to be in second normal form (2NF), there must be no case of a non-prime attribute in the table that is functionally dependent upon a subset of a candidate key.
  - For a table to be in third normal form (3NF), every non-prime attribute must have a non-transitive functional dependency on every candidate key.
  - For a table to be in Boyce-Codd Normal Form (BCNF), every functional dependency (other than trivial dependencies) must be on a superkey.
  - For a table to be in fourth normal form (4NF), it must have no multivalued dependencies.

Normalization

- Normalization is the process of efficiently organizing data in a database.
- Goals of the normalization process:
  - Eliminating redundant data
  - Ensuring that data dependencies make sense (good relationships)

Normal Forms

- A series of guidelines for ensuring that databases are normalized.
- Numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF).
- In practical applications, you'll often see 1NF, 2NF, and 3NF, with only the occasional 4NF.

Normal Forms

- Forth normal form is rarely seen.
- Fifth normal form is almost never seen.

Caveat…

- Normal forms are **guidelines** and guidelines only.
- Sometimes, it becomes necessary to stray from them to meet practical business requirements.
- When variations take place, it is still extremely important to evaluate any possible ramifications they could have on your system and account for possible inconsistencies.

First Normal Form (1NF)

- Eliminate duplicate columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Eliminate duplicate columns from the same table.

- Referred to as the atomicity of a table.
- Tables that comply with this rule are said to be atomic.
- Let's explore this principle with a classic example - a table within a human resources database that stores the manager-subordinate relationship.
- For the purposes of our example, we'll impose the business rule that each manager may have one or more subordinates while each subordinate may have only one manager.

Example – Step 1

Assume that that each manager may have one or more subordinates while each subordinate may have only one manager.

manager	subordinate_1	subordinate_2	subordinate_3	subordinate_4
John	Mary	Josh	David	Jane

Example – Step 2

manager	subordinates
John	Mary, Josh, David, Jane

Example – Step 2

manager	subordinates
John	Mary, Josh, David, Jane

- The subordinates column is still duplicative and non-atomic.
- What happens when we need to add or remove a subordinate?

Example – Step 3

- Here's a table that satisfies the first rule of 1NF:

manager	subordinates
John	Mary
John	Josh
John	David
John	Jane

- In this case, each subordinate has a single entry, but managers may have multiple entries.

Example – Step 4

- Remember the second rule of 1NF: identify each row with a unique column or set of columns (the primary key)?

manager_id	manager	subordinate_id	subordinate
5	John	1	Mary
5	John	2	Josh
5	John	3	David
5	John	4	Jane

Second Normal Form (2NF)

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

2NF Example

2NF attempts to reduce the amount of redundant data in a table by extracting it, placing it in new table(s) and creating relationships between those tables.

employees	
employee_id	**name**
1	Mary
2	Josh
3	David
4	Jane
5	John

manager_subordinates	
manager_id	**subordinate_id**
5	1
5	2
5	3
5	4

2NF Example

You were hired to develop a POS system. Your client gave you a spreadsheet that contains customers' information. How do you convert it to database entities/tables that comply with 2NF?

Customers						
CustNum	FirstName	LastName	Address	City	State	Zip
1	Mary	Doe	743 Evergreen St.	Pittsburgh	PA	15217
2	Josh	Smith	134 Phillips Avenue	Pittsburgh	PA	15217
3	David	Burke	456 Hobart Street	Pittsburgh	PA	15217
4	Jane	Brown	7645 Liberty Ave.	Pittsburgh	PA	15222
5	John	Black	134 Phillips Avenue	Pittsburgh	PA	15217

2NF Example

What's wrong with just leaving the table as-is?

Customers						
CustNum	FirstName	LastName	Address	City	State	Zip
1	Mary	Doe	743 Evergreen St.	Pittsburgh	PA	15217
2	Josh	Smith	**134 Phillips Avenue**	**Pittsburgh**	**PA**	**15217**
3	David	Burke	456 Hobart Street	Pittsburgh	PA	15217
4	Jane	Brown	7645 Liberty Ave.	Pittsburgh	PA	15222
5	John	Black	**134 Phillips Avenue**	**Pittsburgh**	**PA**	**15217**

2NF Example

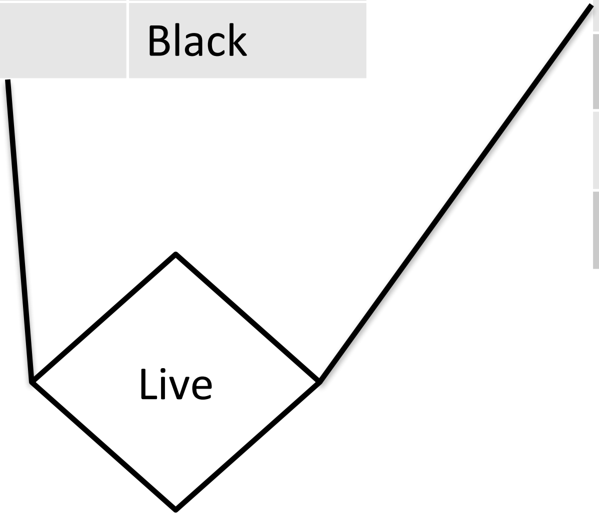
- In a 2NF-compliant database structure, this redundant information is extracted and stored in a separate table.
- We’ll create two tables – ***Customers*** and ***Addresses***

Customers						
CustNum	FirstName	LastName	Address	City	State	Zip
1	Mary	Doe	743 Evergreen St.	Pittsburgh	PA	15217
2	Josh	Smith	**134 Phillips Avenue**	**Pittsburgh**	**PA**	**15217**
3	David	Burke	456 Hobart Street	Pittsburgh	PA	15217
4	Jane	Brown	7645 Liberty Ave.	Pittsburgh	PA	15222
5	John	Black	**134 Phillips Avenue**	**Pittsburgh**	**PA**	**15217**

2NF Example

Customers		
CustNum	FirstName	LastName
1	Mary	Doe
2	Josh	Smith
3	David	Burke
4	Jane	Brown
5	John	Black

Addresses				
AddressID	Address	City	State	Zip
1	743 Evergreen St.	Pittsburgh	PA	15217
2	134 Phillips Avenue	Pittsburgh	PA	15217
3	456 Hobart Street	Pittsburgh	PA	15217
4	7645 Liberty Ave.	Pittsburgh	PA	15222



But wait… There is more!

Customers

CustNum	FirstName	LastName
1	Mary	Doe
2	Josh	Smith
3	David	Burke
4	Jane	Brown
5	John	Black

Live

Addresses

AddressID	Address
1	743 Evergreen St.
2	134 Phillips Avenue
3	456 Hobart Street
4	7645 Liberty Ave.

Located in

ZipCodes

ZipID	City	State	Zip
1	Pittsburgh	PA	15217
3	Pittsburgh	PA	15222

Third Normal Form (3NF)

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.

Example – 3NF

- Consider the following table that's part of our POS system.
- This table contains information about customer orders.

order_num	cust_num	unit_price	quantity	total
X43565	1	2.75	4	11.00
Y43525	2	1000.00	28	28000.00
U43746	3	53.07	6	318.42
L86549	4	100.00	154	15400.00

Example – 3NF

Requirements of 1NF.

- Are there any duplicative columns?
- Do we have a primary key?

Requirements of 2NF.

- Are there any subsets of data that apply to multiple rows?

order_num	cust_num	unit_price	quantity	total
X43565	1	2.75	4	11.00
Y43525	2	1000.00	28	28000.00
U43746	3	53.07	6	318.42
L86549	4	100.00	154	15400.00

Example – 3NF

Are all of the columns fully dependent upon the primary key?

order_num	cust_num	unit_price	quantity	total
X43565	1	2.75	4	11.00
Y43525	2	1000.00	28	28000.00
U43746	3	53.07	6	318.42
L86549	4	100.00	154	15400.00

Example – 3NF

The **customer number** varies with the order number and it doesn't appear to depend upon any of the other fields.

order_num	cust_num	unit_price	quantity	total
X43565	**1**	2.75	4	11.00
Y43525	**2**	1000.00	28	28000.00
U43746	**3**	53.07	6	318.42
L86549	**4**	100.00	154	15400.00

Example – 3NF

- **Unit price** could be dependent upon the customer number if we charged each customer a set price.
- However, we could sometimes charge the same customer different prices.
- Therefore, the **unit price** is fully dependent upon the order number.

order_num	cust_num	unit_price	quantity	total
X43565	1	**2.75**	4	11.00
Y43525	2	**1000.00**	28	28000.00
U43746	3	**53.07**	6	318.42
L86549	4	**100.00**	154	15400.00

Example – 3NF

- The **quantity** of items also varies from order to order, so it is dependent of order_num

order_num	cust_num	unit_price	quantity	total
X43565	1	2.75	**4**	11.00
Y43525	2	1000.00	**28**	28000.00
U43746	3	53.07	**6**	318.42
L86549	4	100.00	**154**	15400.00

Example – 3NF

The **total** can be derived by multiplying the unit price by the quantity, therefore it's **NOT** fully dependent upon the primary key.

order_num	cust_num	unit_price	quantity	total
X43565	1	2.75	4	**11.00**
Y43525	2	1000.00	28	**28000.00**
U43746	3	53.07	6	**318.42**
L86549	4	100.00	154	**15400.00**

Example – 3NF

- We must remove **total** from the table to comply with the third normal form.

order_num	cust_num	unit_price	quantity
X43565	1	2.75	4
Y43525	2	1000.00	28
U43746	3	53.07	6
L86549	4	100.00	154

Example – 3NF

Before 3NF normalization:

- SELECT order_num, **total** FROM orders;

After 3NF normalization:

- SELECT order_num, **unit_price * quantity AS total** FROM orders;

order_num	cust_num	unit_price	quantity
X43565	1	2.75	4
Y43525	2	1000.00	28
U43746	3	53.07	6
L86549	4	100.00	154

Another Example – 3NF

Suppose we have relation SUPPLIER_PART

SUPPLIER_PART(SUPP_ID, PART_ID, SNAME, QUANTITY)

with the following assumptions:

1. SUPP_ID is unique for every supplier.
2. SNAME is unique for every supplier.
3. QUANTITY is the accumulated quantities of a part supplied by a supplier.
4. A supplier can supply more than one part.
5. A part can be supplied by more than one supplier

https://wiki.engr.illinois.edu/download/attachments/199370487/cs411-05c-relationaldesign.pdf

Another Example – 3NF

SUPPLIER_PART(SUPP_ID, PART_ID, SNAME, QUANTITY)

We can find the following **non-trivial** functional dependencies:

1. SUPP_ID →SNAME
2. SNAME →SUPP_ID
3. (SUPP_ID, PART_ID) → QUANTITY
4. (SNAME, PART_ID) → QUANTITY

https://wiki.engr.illinois.edu/download/attachments/199370487/cs411-05c-relationaldesign.pdf

Another Example – 3NF

SUPPLIER_PART(SUPP_ID, PART_ID, SNAME, QUANTITY)

The candidate keys are:

1. SUPP_ID, PART_ID
2. SNAME, PART_ID

https://wiki.engr.illinois.edu/download/attachments/199370487/cs411-05c-relationaldesign.pdf

Another Example – 3NF

SUPPLIER_PART(SUPP_ID, PART_ID, SNAME, QUANTITY)

Yay! The relation is in 3NF.

The Boyce-Codd Normal Form (3.5NF)

- Meet all the requirements of the third normal form.
- Every determinant must be a candidate key.

The Boyce-Codd Normal Form (3.5NF)

- BCNF addresses **unlikely** situations which 3NF does not handle.
- The definition of 3NF does not deal with a relation that:
  - has multiple candidate keys, where
  - those candidate keys are composite, and
  - the candidate keys overlap (i.e., have at least one common attribute)

http://defiant.yk.psu.edu/~lxn/IST_210/normal_form_definitions.html

Example - BCNF

Consider the following table **ClientInterview.** This relation is already in 3NF

ClientInterview				
clientNo	**interviewDate**	**interviewTime**	**staffNo**	**roomNo**
CR76	13-May-02	10.30	SG5	G101
CR76	13-May-02	12.00	SG5	G101
CR74	13-May-02	12.00	SG37	G102
CR56	1-Jul-02	10.30	SG5	G102

Example - BCNF

ClientInterview (ClientNo, interviewDate, interviewTime, staffNo, roomNo)

Functional Dependencies:

1. clientNo, interviewDate, interviewTime, staffNo, roomNo (**PK**)
2. staffNo, interviewDate, interviewTime, clientNo (**CK**)
3. roomNo, interviewDate, interviewTime, clientNo, staffNo (**CK**)
4. staffNo, interviewDate, roomNo (**not a CK**)

Example - BCNF

As a consequence the ***ClientInterview*** relation may suffer from update anomalies

- For example, two rows have to be updated if the roomNo changes for staffNo SG5 on the 13-May-02.

ClientInterview				
clientNo	**interviewDate**	**interviewTime**	**staffNo**	**roomNo**
CR76	13-May-02	10.30	SG5	G101
CR76	13-May-02	12.00	SG5	G101
CR74	13-May-02	12.00	SG37	G102
CR56	1-Jul-02	10.30	SG5	G102

Example - BCNF

To transform the ClientInterview relation to BCNF, we must remove the violating functional dependency by creating two new relations called ***Interview*** and ***StaffRoom***:

- Interview (clientNo, interviewDate, interviewTime, staffNo)
- StaffRoom(staffNo, interviewDate, roomNo)

Example - BCNF

Interview			
clientNo	**interviewDate**	**interviewTime**	**staffNo**
CR76	13-May-02	10.30	SG5
CR76	13-May-02	12.00	SG5
CR74	13-May-02	12.00	SG37
CR56	1-Jul-02	10.30	SG5

StaffRoom		
staffNo	**interviewDate**	**roomNo**
SG5	13-May-02	G101
SG5	13-May-02	G101
SG37	13-May-02	G102
SG5	1-Jul-02	G102

Example - BCNF

Today's Court Bookings

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

https://en.wikipedia.org/wiki/Boyce%E2%80%93Codd_normal_form

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

- Each row in the table represents a court booking at a tennis club that has one hard court (Court 1) and one grass court (Court 2)
- A booking is defined by its Court and the period for which the Court is reserved
- Additionally, each booking has a Rate Type associated with it. There are four distinct rate types:
    - SAVER, for Court 1 bookings made by members
    - STANDARD, for Court 1 bookings made by non-members
    - PREMIUM-A, for Court 2 bookings made by members
    - PREMIUM-B, for Court 2 bookings made by non-members

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

The table's possible key combinations are:

- S_1 = {Court, Start Time}
- S_2 = {Court, End Time}
- S_3 = {Rate Type, Start Time}
- S_4 = {Rate Type, End Time}
- S_5 = {Court, Start Time, End Time}
- S_6 = {Rate Type, Start Time, End Time}
- S_7 = {Court, Rate Type, Start Time}
- S_8 = {Court, Rate Type, End Time}
- S_T = {Court, Rate Type, Start Time, End Time}

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

- *Start Time* and *End Time* attributes have no duplicate values for each of them
- However, it is possible that in some other days two different bookings on court 1 and court 2 could *start at the same time* or *end at the same time*.
- This is the reason why {Start Time} and {End Time} cannot be considered as the table's superkeys.
- However, only S1, S2, S3 and S4 are candidate keys

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

The table's possible key combinations are:

- **S_1 = {Court, Start Time}**
- **S_2 = {Court, End Time}**
- **S_3 = {Rate Type, Start Time}**
- **S_4 = {Rate Type, End Time}**
- S_5 = {Court, Start Time, End Time}
- S_6 = {Rate Type, Start Time, End Time}
- S_7 = {Court, Rate Type, Start Time}
- S_8 = {Court, Rate Type, End Time}
- S_T = {Court, Rate Type, Start Time, End Time}

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

- The table does not adhere to BCNF.
- The dependency **Rate Type** $\longrightarrow$ **Court**, in which the determining attribute (Rate Type), is **NOT** a candidate key.
- Dependency **Rate Type** $\longrightarrow$ **Court** is respected as a Rate Type should only ever apply to a single Court.

Rate Types

<u>Rate Type</u>	Court	Member Flag
SAVER	1	Yes
STANDARD	1	No
PREMIUM-A	2	Yes
PREMIUM-B	2	No

Today's Bookings

Member Flag	Court	Start Time	End Time
Yes	1	09:30	10:30
Yes	1	11:00	12:00
No	1	14:00	15:30
No	2	10:00	11:30
No	2	11:30	13:30
Yes	2	15:00	16:30

Normalization Exercise 1

petID	petName	petType	petAge	ownerName	visitDate	procedure
246	Rover	Dog	12	Sam Cook	JAN 13/2002 MAR 27/2002 APR 02/2002	01 - RABIES VACCINATION 10 - EXAMINE and TREAT WOUND 05 - HEART WORM TEST
298	Spot	Dog	2	Terry Kim	JAN 21/2002 MAR 10/2002	08 - TETANUS VACCINATION 05 - HEART WORM TEST
341	Morris	Cat	4	Sam Cook	JAN 23/2001 JAN 13/2002	01 - RABIES VACCINATION 01 - RABIES VACCINATION
519	Tweedy	Bird	2	Terry Kim	Apr 30, 2002 May 16, 2004	20 - ANNUAL CHECK UP 12 - EYE WASH

Adapted from
http://cs.senecac.on.ca/~dbs201/pages/Norm_Exercise1.doc

Normalization Exercise 2

INVOICE

FROM: HILLTOP ANIMAL HOSPITAL
DATE: 01/13/2014
INVOICE # 987

TO: MR. RICHARD COOK
123 Phillips Avenue
Pittsburgh, PA 15218

Pet	Procedure	Amount
Rover	Rabies Vaccination	30.00
Morris	Rabies Vaccination	24.00
	Subtotal:	54.00
	PA Tax (7%)	3.78
	Amount Owing	**$57.78**

Adapted from http://cs.senecac.on.ca/~dbs201/pages/Norm_Exercise2.doc

Normalization Exercise 3

Gallery Customer History Form

Customer Name: Jackson, Elizabeth

Address: 123 Murray Avenue
Pittsburgh, PA 15218

Phone: (412) 256-8612

Purchases Made

Artist	Title	Purchase Date	Sales Price
03 - Carol Channing	Laugh with Teeth	09/17/2000	7000.00
15 - Dennis Frings	South toward Emerald Sea	05/11/2000	1800.00
03 - Carol Channing	At the Movies	02/14/2002	5500.00
15 - Dennis Frings	Black square	07/15/2003	2200.00