Updated 12/9/2022

Topic 2: Database design
L21: Normalization

Wolfgang Gatterbauer

CS3200 Database design (fa22)

https://northeastern-datalab.github.io/cs3200/fa22s3/

11/28/2022

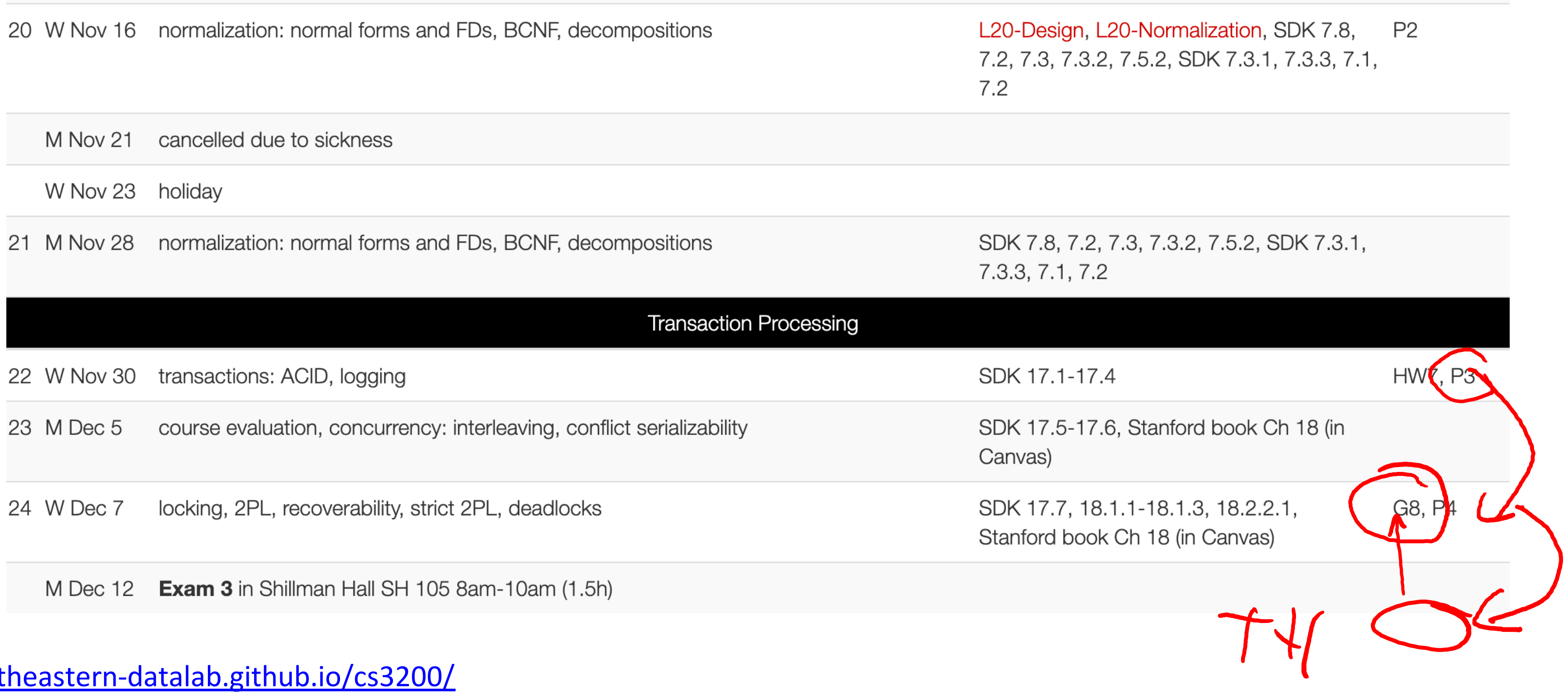
Class warm-up

- Notifications
  - postings on Piazza (anonymous, but no notification) vs Canvas (not anonymous, but notifications): Suggestion: all contents on Piazza, Canvas announcement for alerts
- Outline remaining 4 classes (we needed to shift topics)
- Exam2 open discussion

- Today: normalization
- Then: transactions

Suggested changes to calendar

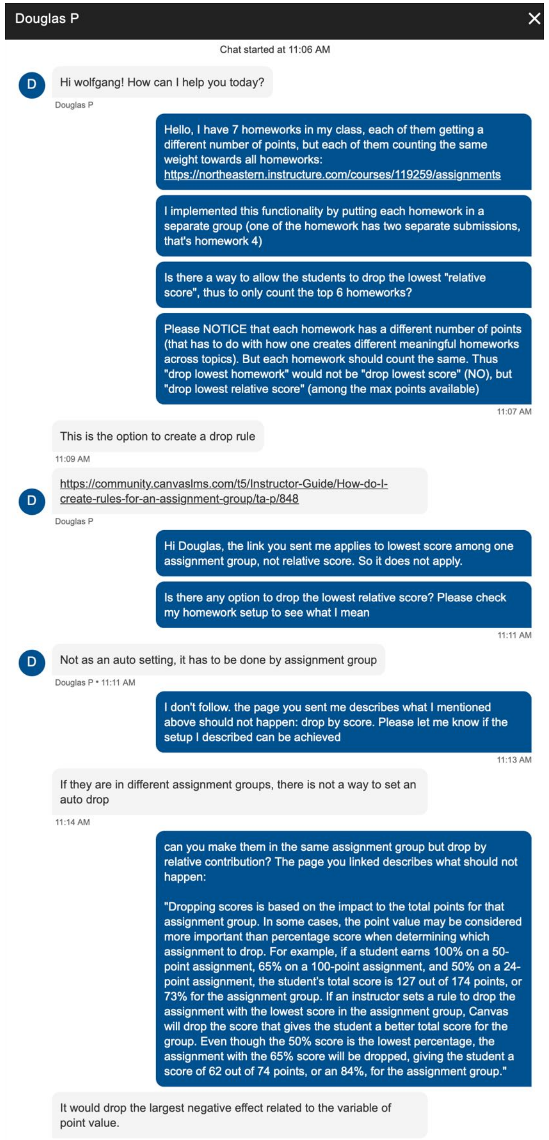
- G9 removed, G8 pushed to THU next week
- P3 and P4 pushed by 1 week. TBD: by < 1 week
- No HW7: to discuss dropping a whole HW
- Alternative: lots of in-class SQL and design practice



20	W Nov 16	normalization: normal forms and FDs, BCNF, decompositions	L20-Design, L20-Normalization, SDK 7.8, 7.2, 7.3, 7.3.2, 7.5.2, SDK 7.3.1, 7.3.3, 7.1, 7.2	P2
	M Nov 21	cancelled due to sickness		
	W Nov 23	holiday		
21	M Nov 28	normalization: normal forms and FDs, BCNF, decompositions	SDK 7.8, 7.2, 7.3, 7.3.2, 7.5.2, SDK 7.3.1, 7.3.3, 7.1, 7.2	
Transaction Processing				
22	W Nov 30	transactions: ACID, logging	SDK 17.1-17.4	HW7, P3
23	M Dec 5	course evaluation, concurrency: interleaving, conflict serializability	SDK 17.5-17.6, Stanford book Ch 18 (in Canvas)	
24	W Dec 7	locking, 2PL, recoverability, strict 2PL, deadlocks	SDK 17.7, 18.1.1-18.1.3, 18.2.2.1, Stanford book Ch 18 (in Canvas)	G8, P4
	M Dec 12	**Exam 3** in Shillman Hall SH 105 8am-10am (1.5h)		

TH

Dropping lowest contribution HW not possible if HWs have different points



Douglas P

Chat started at 11:06 AM

Hi wolfgang! How can I help you today?

Douglas P

Hello, I have 7 homeworks in my class, each of them getting a different number of points, but each of them counting the same weight towards all homeworks: https://northeastern.instructure.com/courses/119259/assignments

I implemented this functionality by putting each homework in a separate group (one of the homework has two separate submissions, that's homework 4)

Is there a way to allow the students to drop the lowest "relative score", thus to only count the top 6 homeworks?

Please NOTICE that each homework has a different number of points (that has to do with how one creates different meaningful homeworks across topics). But each homework should count the same. Thus "drop lowest homework" would not be "drop lowest score" (NO), but "drop lowest relative score" (among the max points available)

11:07 AM

This is the option to create a drop rule

11:09 AM

https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-create-rules-for-an-assignment-group/ta-p/848

Douglas P

Hi Douglas, the link you sent me applies to lowest score among one assignment group, not relative score. So it does not apply.

Is there any option to drop the lowest relative score? Please check my homework setup to see what I mean

11:11 AM

Not as an auto setting, it has to be done by assignment group

Douglas P • 11:11 AM

I don't follow. the page you sent me describes what I mentioned above should not happen: drop by score. Please let me know if the setup I described can be achieved

11:13 AM

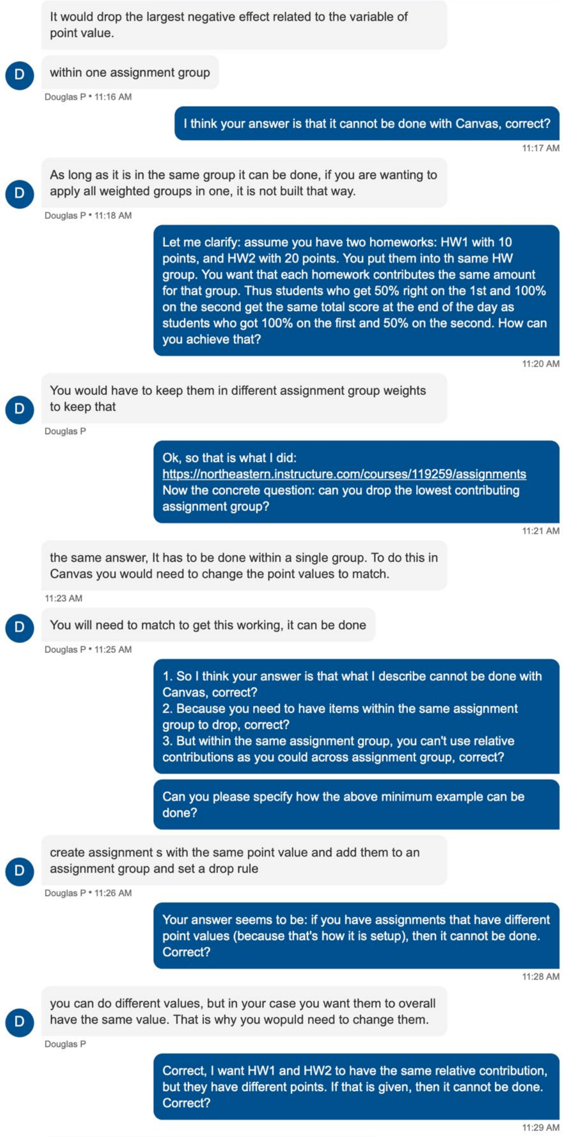
If they are in different assignment groups, there is not a way to set an auto drop

11:14 AM

can you make them in the same assignment group but drop by relative contribution? The page you linked describes what should not happen:

"Dropping scores is based on the impact to the total points for that assignment group. In some cases, the point value may be considered more important than percentage score when determining which assignment to drop. For example, if a student earns 100% on a 50-point assignment, 65% on a 100-point assignment, and 50% on a 24-point assignment, the student's total score is 127 out of 174 points, or 73% for the assignment group. If an instructor sets a rule to drop the assignment with the lowest score in the assignment group, Canvas will drop the score that gives the student a better total score for the group. Even though the 50% score is the lowest percentage, the assignment with the 65% score will be dropped, giving the student a score of 62 out of 74 points, or an 84%, for the assignment group."

It would drop the largest negative effect related to the variable of point value.



It would drop the largest negative effect related to the variable of point value.

within one assignment group

Douglas P • 11:16 AM

I think your answer is that it cannot be done with Canvas, correct?

11:17 AM

As long as it is in the same group it can be done, if you are wanting to apply all weighted groups in one, it is not built that way.

Douglas P • 11:18 AM

Let me clarify: assume you have two homeworks: HW1 with 10 points, and HW2 with 20 points. You put them into th same HW group. You want that each homework contributes the same amount for that group. Thus students who get 50% right on the 1st and 100% on the second get the same total score at the end of the day as students who got 100% on the first and 50% on the second. How can you achieve that?

11:20 AM

You would have to keep them in different assignment group weights to keep that

Douglas P

Ok, so that is what I did:
https://northeastern.instructure.com/courses/119259/assignments
Now the concrete question: can you drop the lowest contributing assignment group?

11:21 AM

the same answer, It has to be done within a single group. To do this in Canvas you would need to change the point values to match.

11:23 AM

You will need to match to get this working, it can be done

Douglas P • 11:25 AM

1. So I think your answer is that what I describe cannot be done with Canvas, correct?
2. Because you need to have items within the same assignment group to drop, correct?
3. But within the same assignment group, you can't use relative contributions as you could across assignment group, correct?

Can you please specify how the above minimum example can be done?

create assignment s with the same point value and add them to an assignment group and set a drop rule

Douglas P • 11:26 AM

Your answer seems to be: if you have assignments that have different point values (because that's how it is setup), then it cannot be done. Correct?

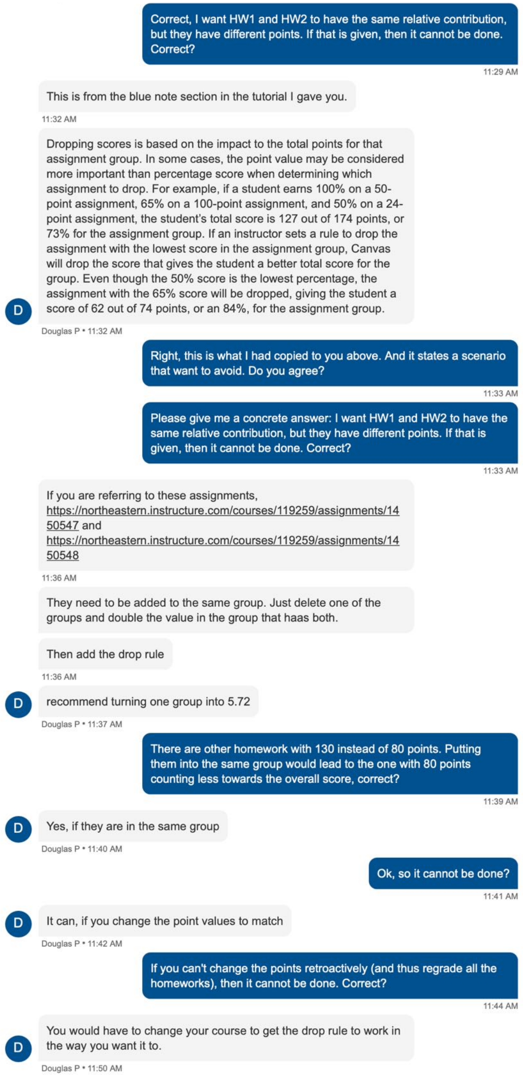
11:28 AM

you can do different values, but in your case you want them to overall have the same value. That is why you wopuld need to change them.

Douglas P

Correct, I want HW1 and HW2 to have the same relative contribution, but they have different points. If that is given, then it cannot be done. Correct?

11:29 AM



Correct, I want HW1 and HW2 to have the same relative contribution, but they have different points. If that is given, then it cannot be done. Correct?

11:29 AM

This is from the blue note section in the tutorial I gave you.

11:32 AM

Dropping scores is based on the impact to the total points for that assignment group. In some cases, the point value may be considered more important than percentage score when determining which assignment to drop. For example, if a student earns 100% on a 50-point assignment, 65% on a 100-point assignment, and 50% on a 24-point assignment, the student's total score is 127 out of 174 points, or 73% for the assignment group. If an instructor sets a rule to drop the assignment with the lowest score in the assignment group, Canvas will drop the score that gives the student a better total score for the group. Even though the 50% score is the lowest percentage, the assignment with the 65% score will be dropped, giving the student a score of 62 out of 74 points, or an 84%, for the assignment group.

Douglas P • 11:32 AM

Right, this is what I had copied to you above. And it states a scenario that want to avoid. Do you agree?

11:33 AM

Please give me a concrete answer: I want HW1 and HW2 to have the same relative contribution, but they have different points. If that is given, then it cannot be done. Correct?

11:33 AM

If you are referring to these assignments, https://northeastern.instructure.com/courses/119259/assignments/1450547 and https://northeastern.instructure.com/courses/119259/assignments/1450548

11:36 AM

They need to be added to the same group. Just delete one of the groups and double the value in the group that haas both.

Then add the drop rule

11:36 AM

recommend turning one group into 5.72

Douglas P • 11:37 AM

There are other homework with 130 instead of 80 points. Putting them into the same group would lead to the one with 80 points counting less towards the overall score, correct?

11:39 AM

Yes, if they are in the same group

Douglas P • 11:40 AM

Ok, so it cannot be done?

11:41 AM

It can, if you change the point values to match

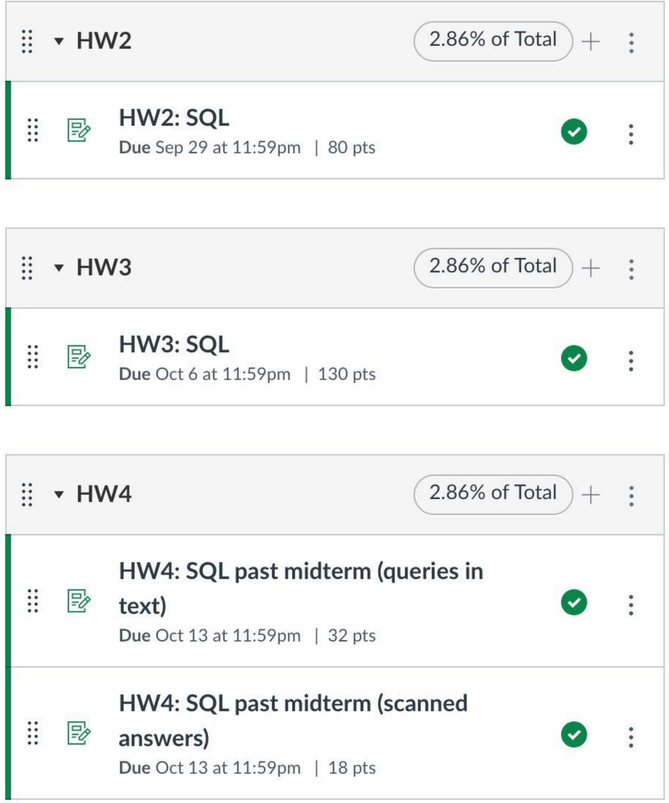
Douglas P • 11:42 AM

If you can't change the points retroactively (and thus regrade all the homeworks), then it cannot be done. Correct?

11:44 AM

You would have to change your course to get the drop rule to work in the way you want it to.

Douglas P • 11:50 AM



HW2 — 2.86% of Total

HW2: SQL
Due Sep 29 at 11:59pm | 80 pts

HW3 — 2.86% of Total

HW3: SQL
Due Oct 6 at 11:59pm | 130 pts

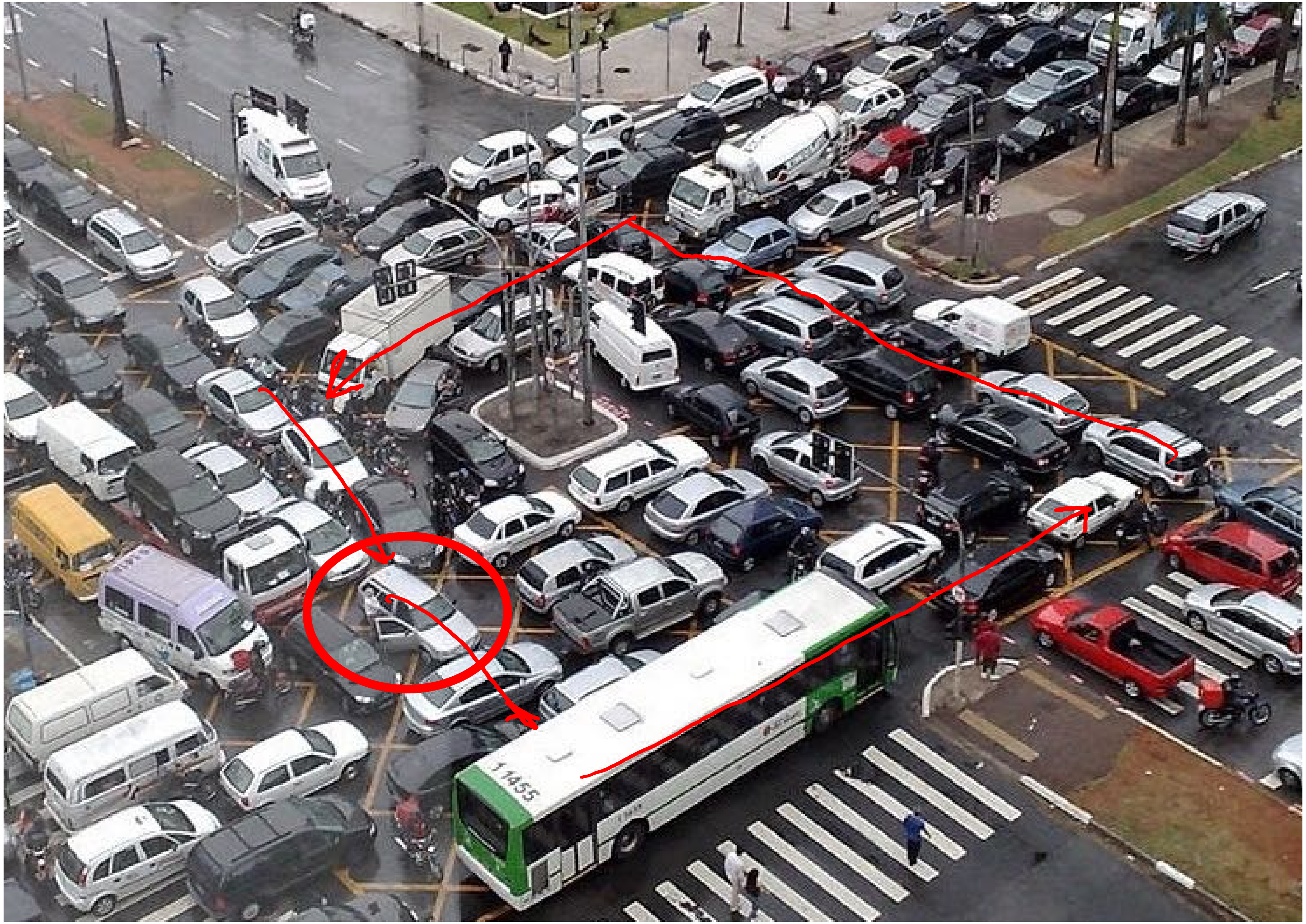
HW4 — 2.86% of Total

HW4: SQL past midterm (queries in text)
Due Oct 13 at 11:59pm | 32 pts

HW4: SQL past midterm (scanned answers)
Due Oct 13 at 11:59pm | 18 pts

Deadlock: What is going one here? And why? And how to resolve?

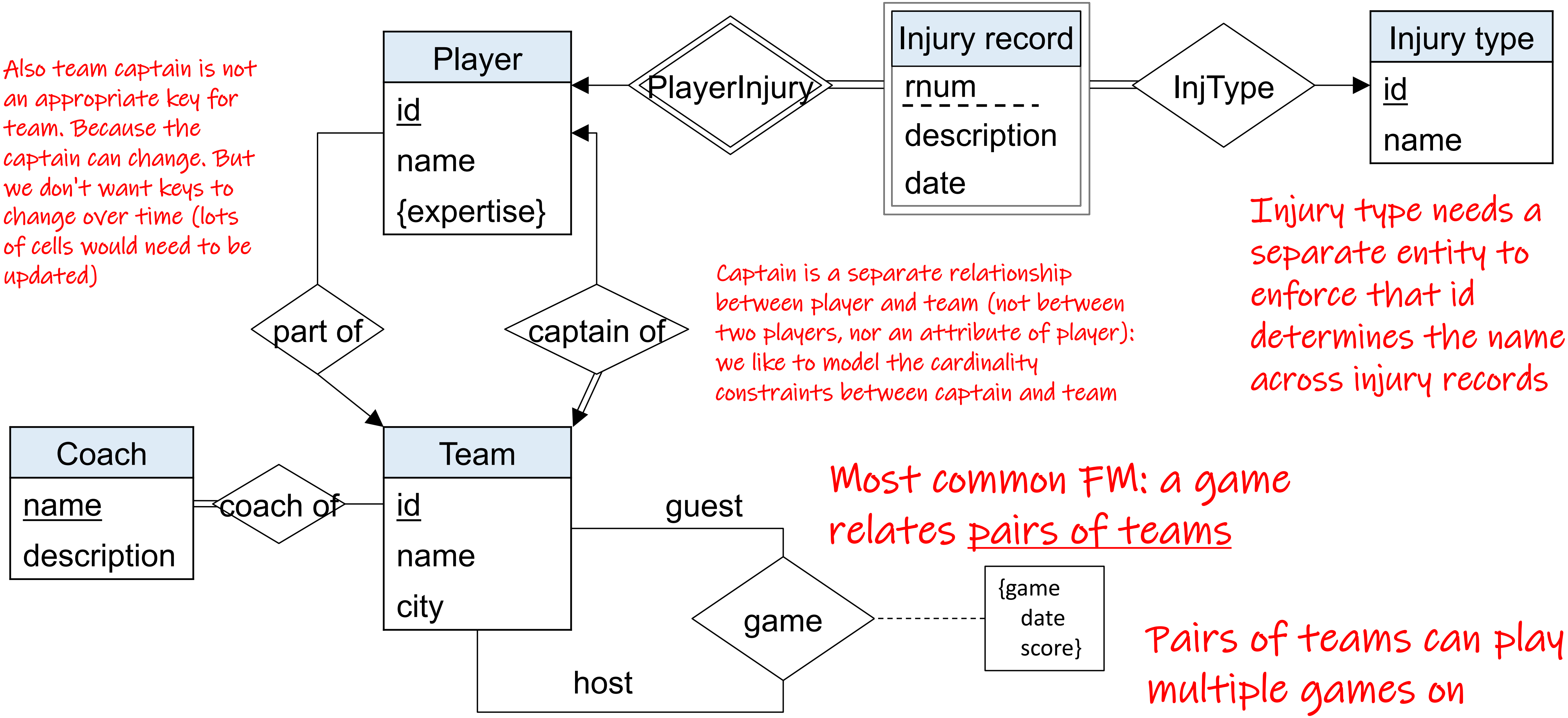
Deadlock: What is going one here? And why? And how to resolve?



Exam2 discussion

Q1

Also team captain is not an appropriate key for team. Because the captain can change. But we don't want keys to change over time (lots of cells would need to be updated)



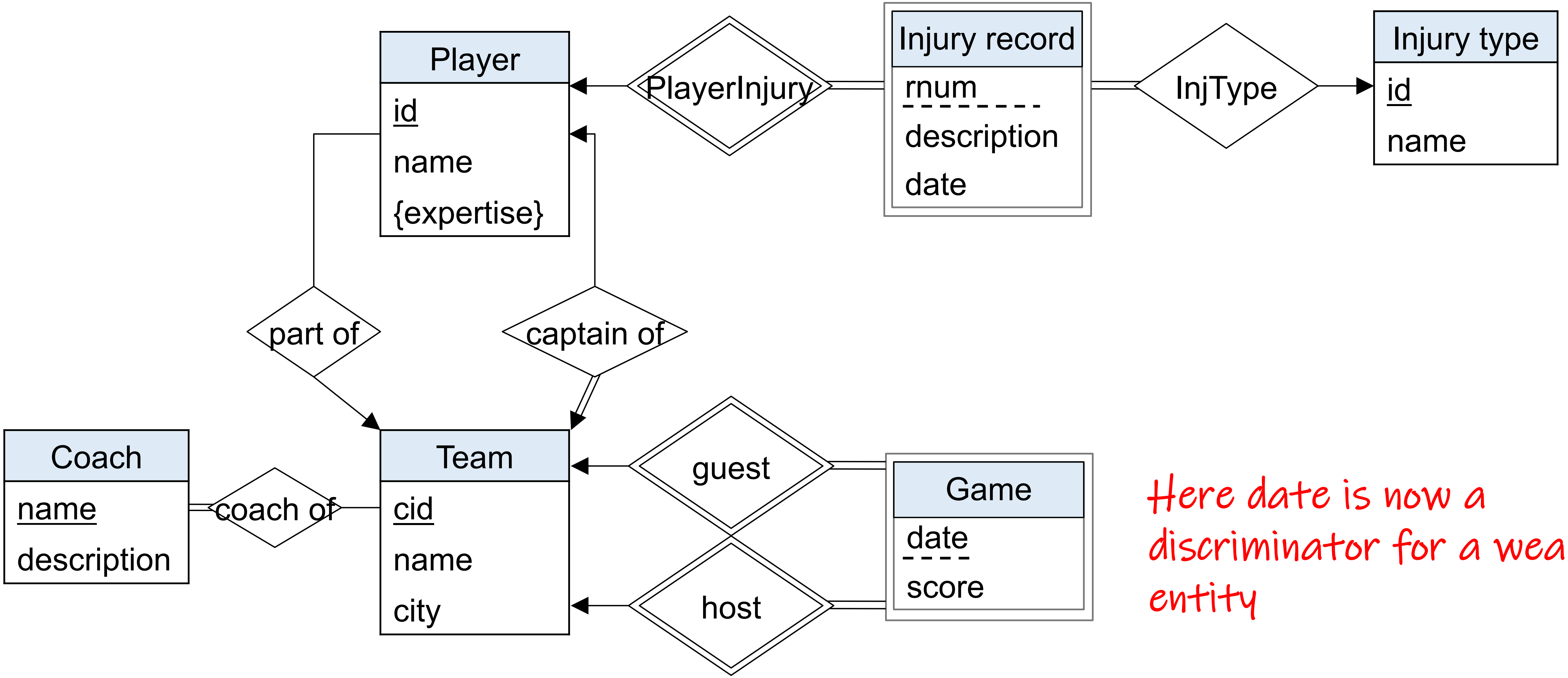
Captain is a separate relationship between player and team (not between two players, nor an attribute of player): we like to model the cardinality constraints between captain and team

Injury type needs a separate entity to enforce that id determines the name across injury records

Most common FM: a game relates <u>pairs of teams</u>

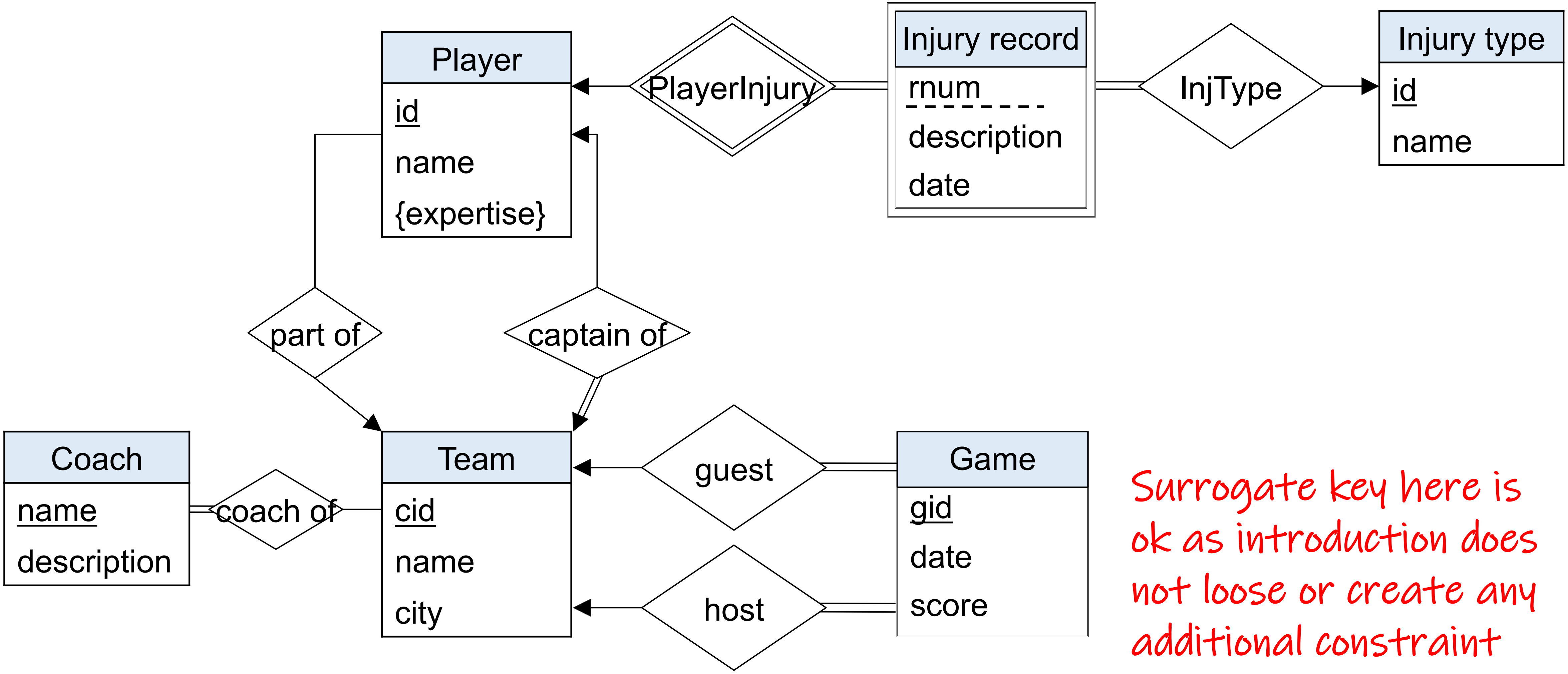
Pairs of teams can play multiple games on different dates, not correct to underline date

Q1



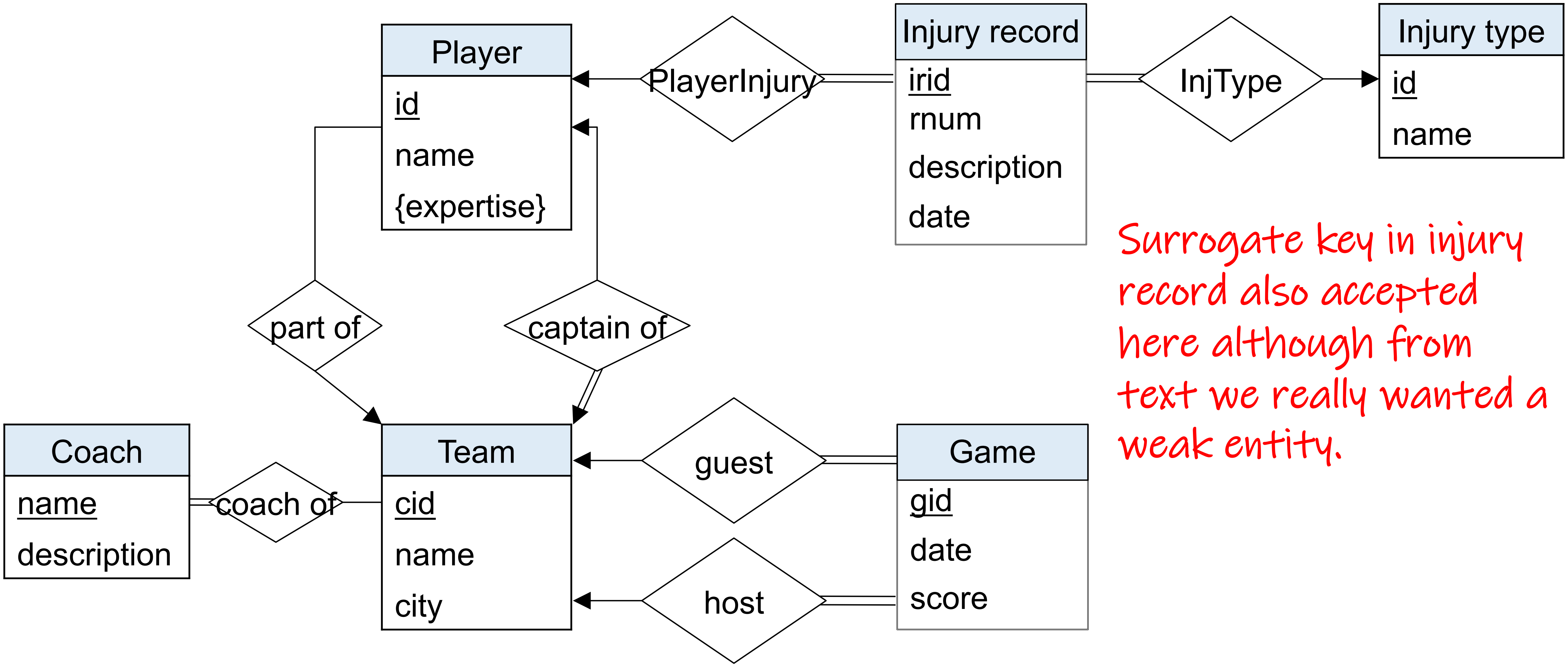
Here date is now a discriminator for a weak entity

Q1



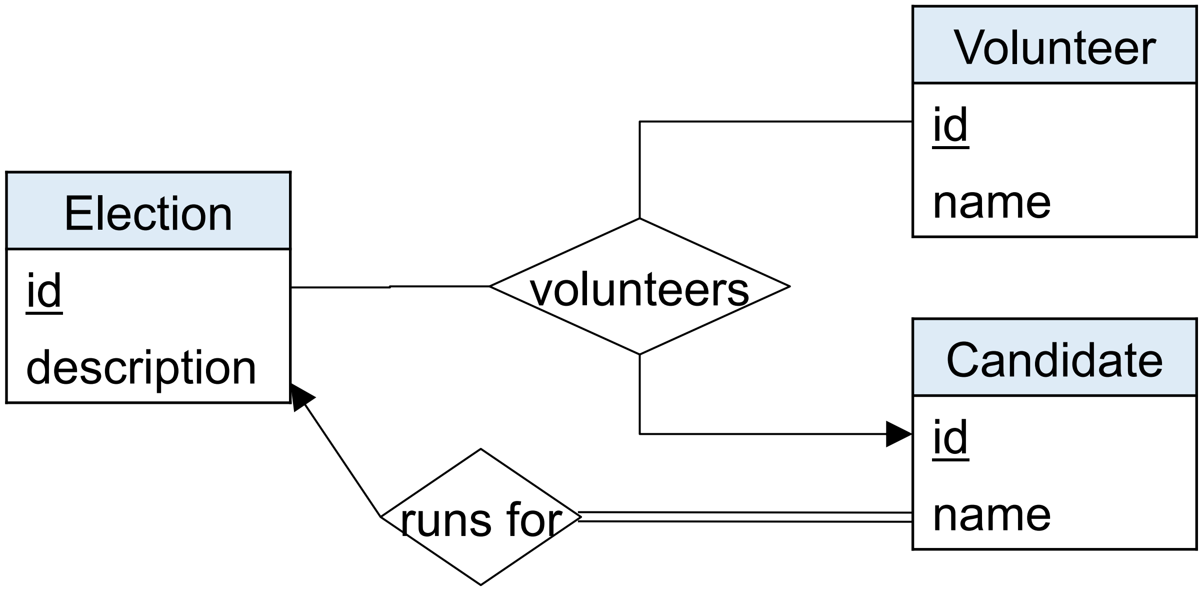
Surrogate key here is ok as introduction does not loose or create any additional constraint

Q1



Surrogate key in injury record also accepted here although from text we really wanted a weak entity.

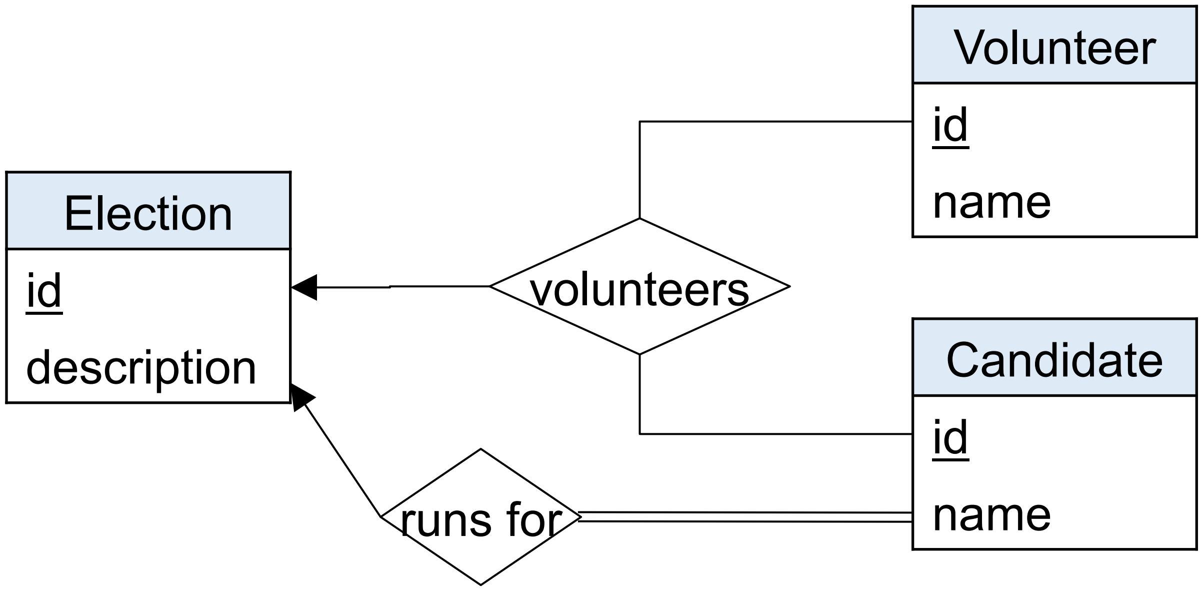
Q2



Mandatory participation for volunteer is not correct since there is not constraint that every candidate needs to have at least one volunteer.

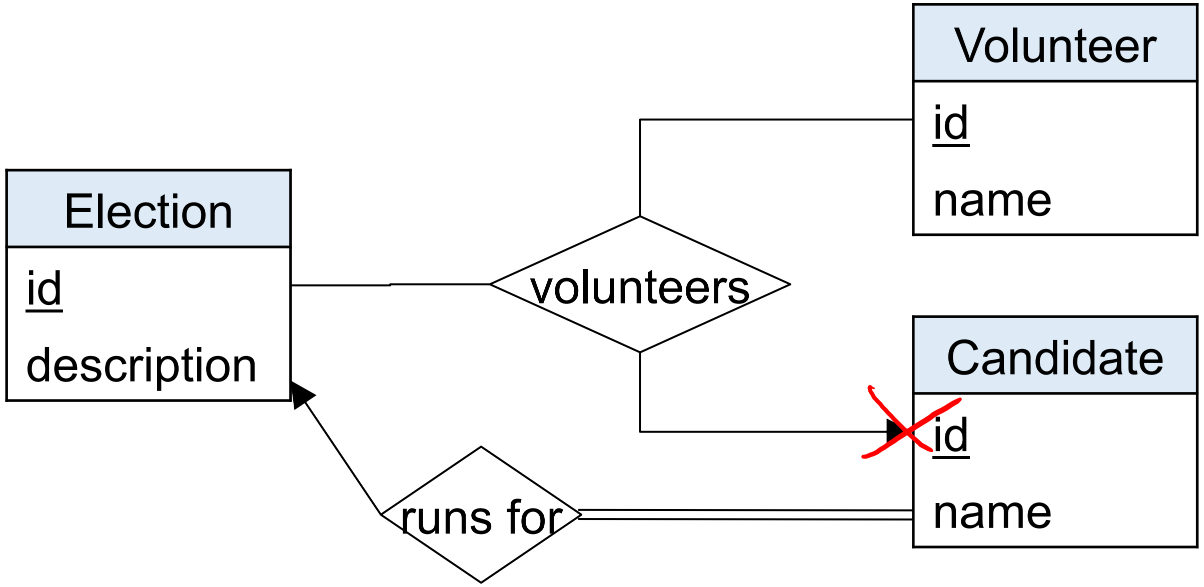
We need to model two different relationships: you could have a candidate without volunteers (cp. to chasm trap)

Q2

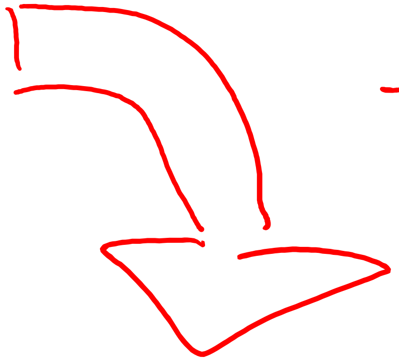


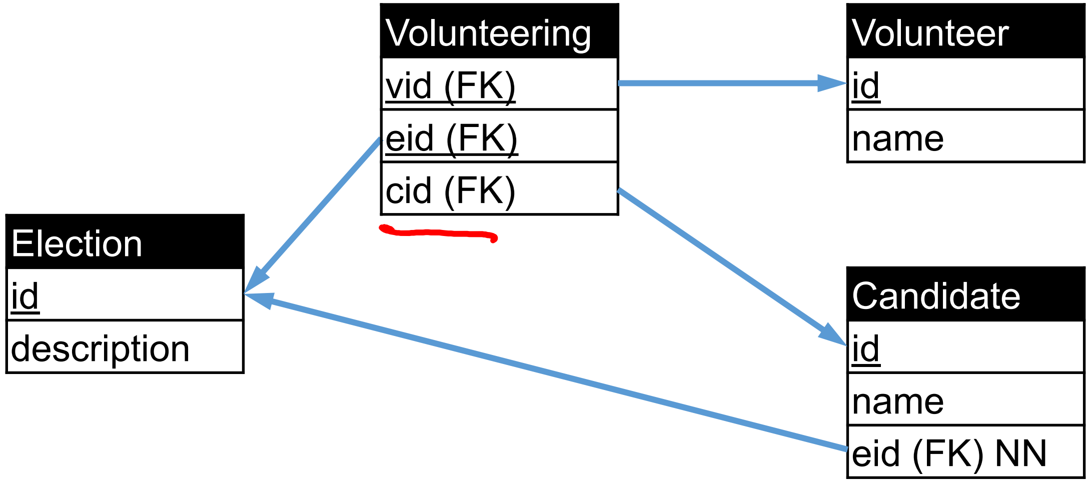
Arrows are not correct here.
Volunteers could work for different
candidates for the same election.

Q3: Q2 into relations



V	E	C
A	1	B
A	2	B
~~A~~	~~1~~	~~C~~
A	1	C ✓





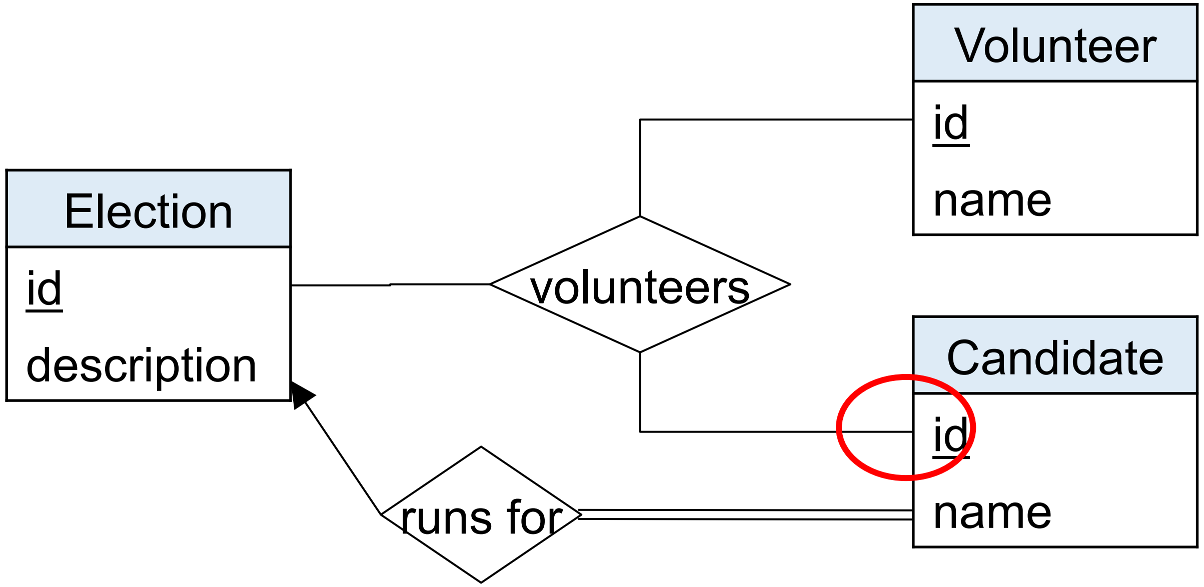
Volunteering
vid (FK)
eid (FK)
cid (FK)

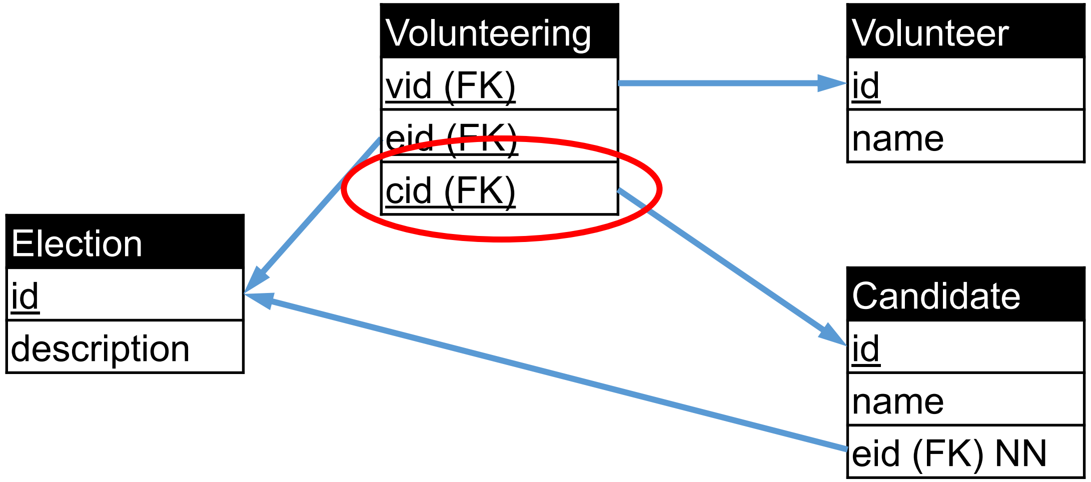
Volunteer
id
name

Election
id
description

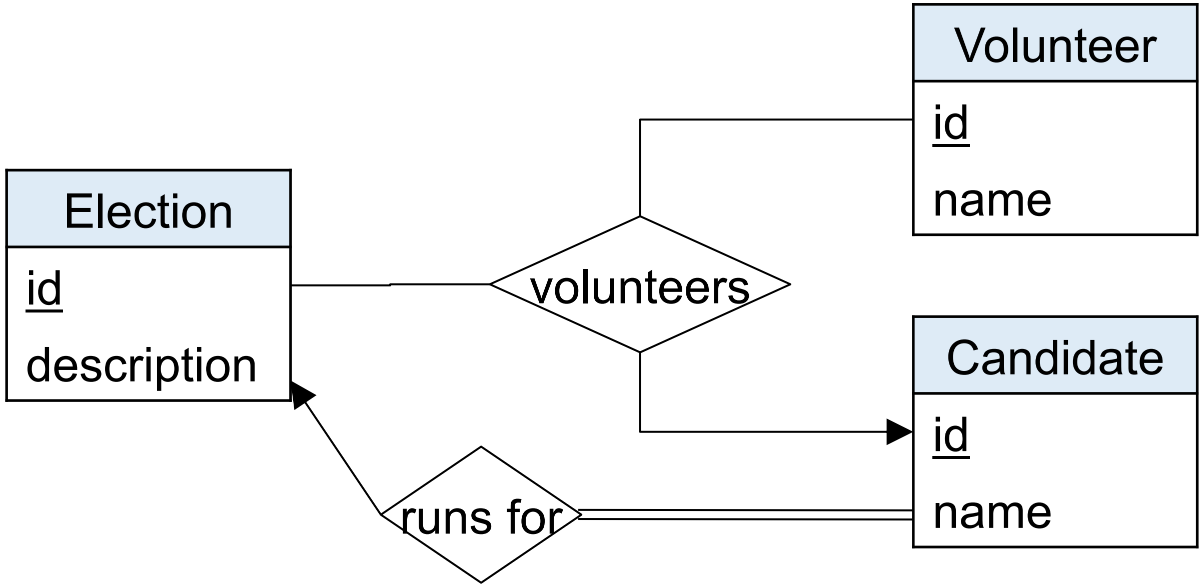
Candidate
id
name
eid (FK) NN

Q3: Q2 into relations





Q3: Q2 into relations



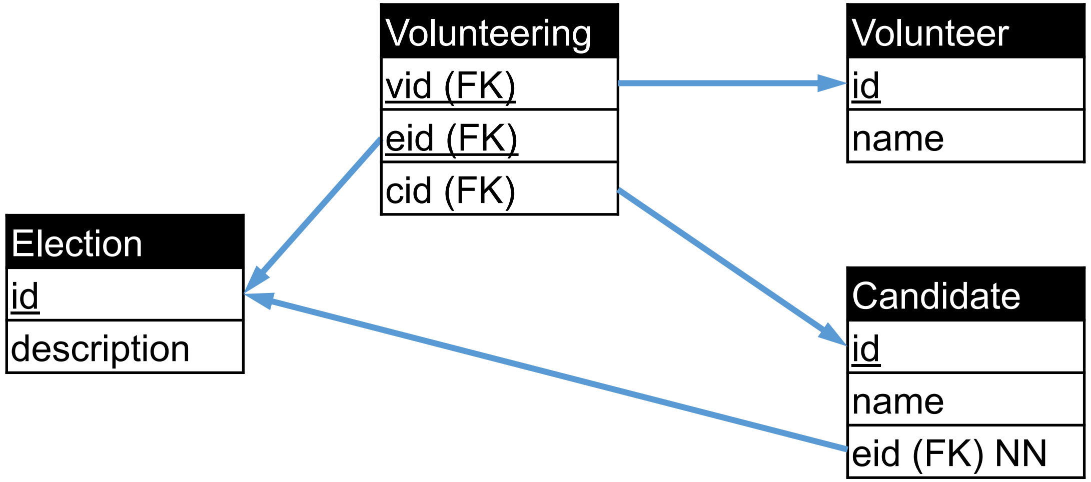
Volunteer
id
name

Election
id
description

volunteers

Candidate
id
name

runs for



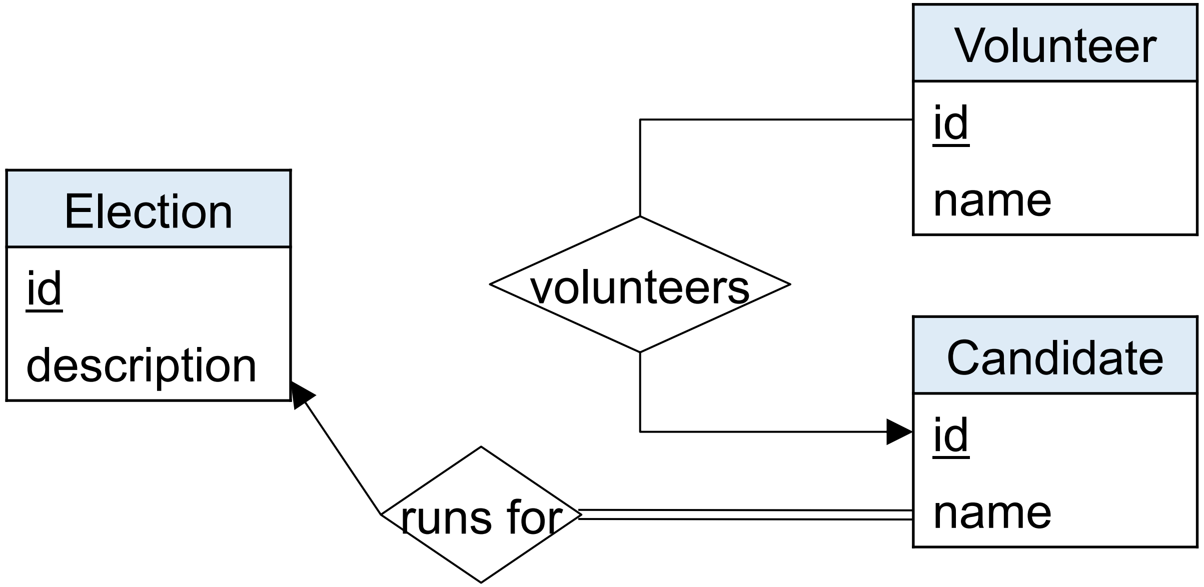
Volunteering
vid (FK)
eid (FK)
cid (FK)

Volunteer
id
name

Election
id
description

Candidate
id
name
eid (FK) NN

Q3: Q2 into relations



Volunteer
id
name

volunteers

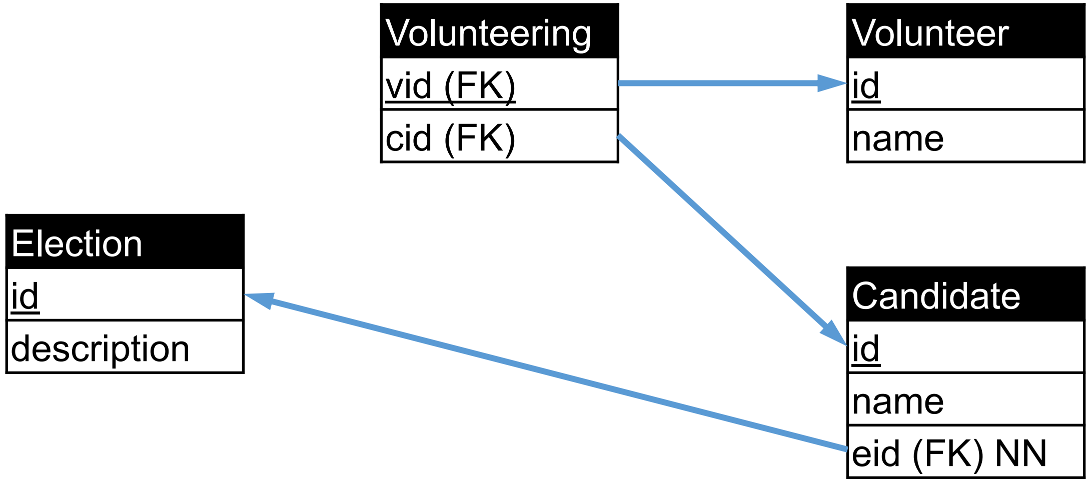
Election
id
description

Candidate
id
name

runs for

What have we lost here?

We lost the constraint that volunteering is possible only for max 1 candidate per election!



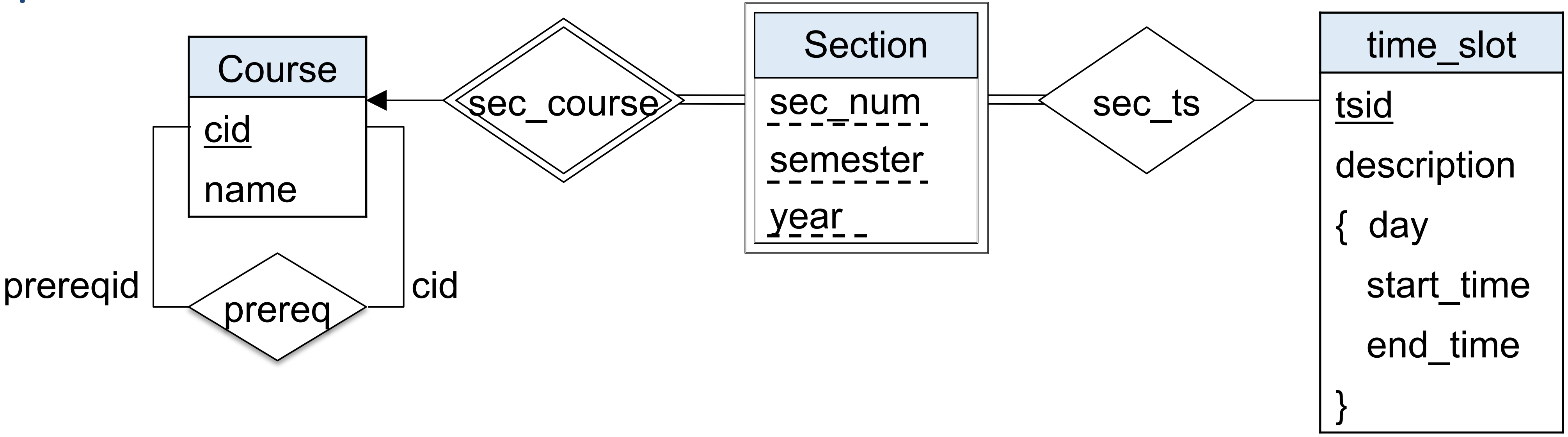
Volunteering
vid (FK)
cid (FK)

Volunteer
id
name

Election
id
description

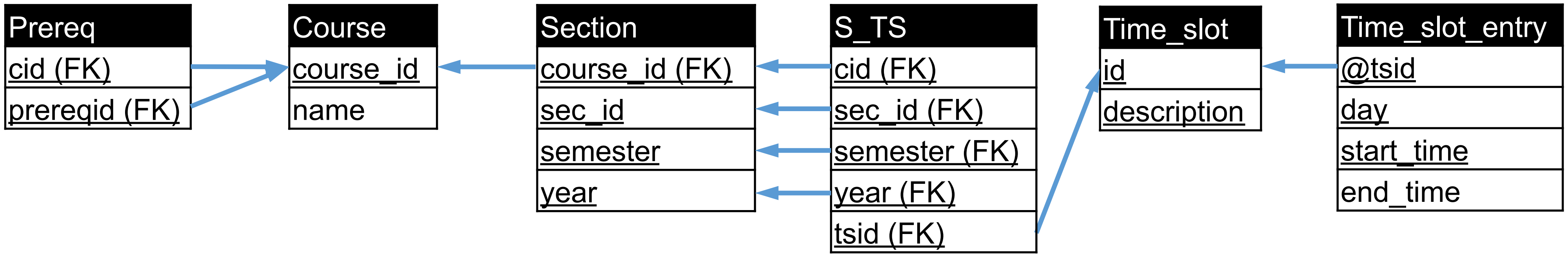
Candidate
id
name
eid (FK) NN

Q4



No NN needed (all indicated ones would be wrong)

we need S_TS table for
many-to-many relationship



Prereq
cid (FK)
prereqid (FK)

Course
course_id
name

Section
course_id (FK)
sec_id
semester
year

S_TS
cid (FK)
sec_id (FK)
semester (FK)
year (FK)
tsid (FK)

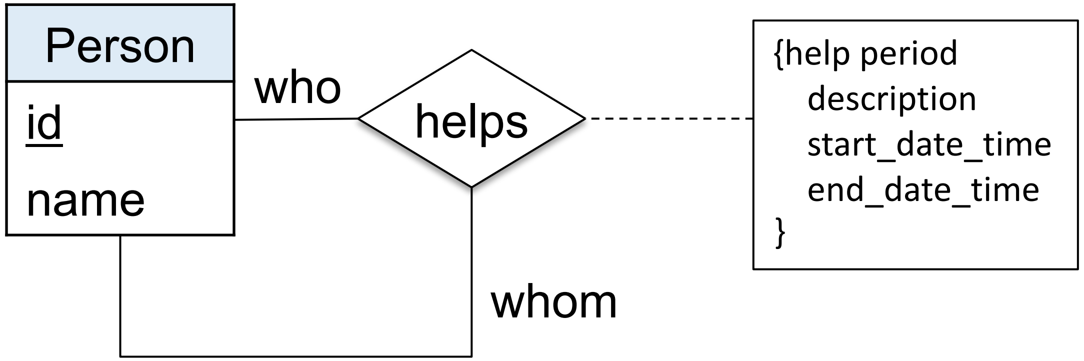
Time_slot
id
description

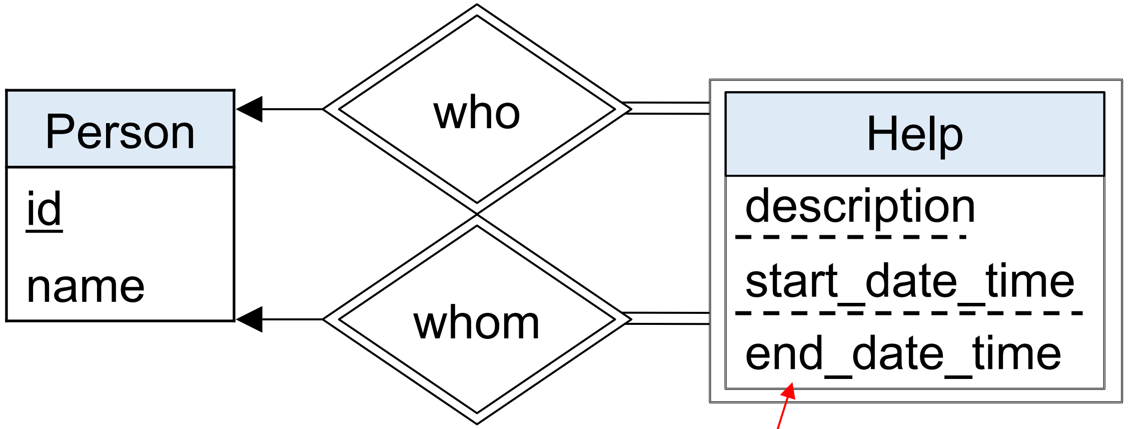
Time_slot_entry
@tsid
day
start_time
end_time

Mandatory participation of section with time slot cannot be represented.

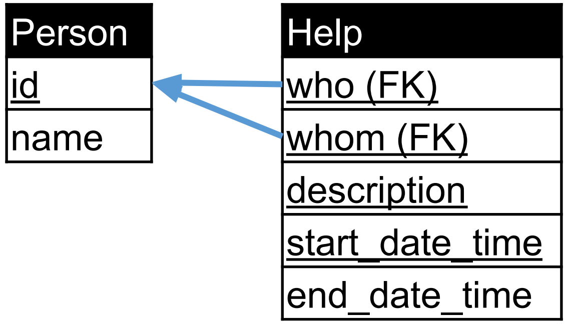
we need time_slot_entry table
for multivalued attribute

Q5





Many mistakes in PK for help. Use of surrogate key was here not allowed due to explicit instructions: "Please do not invent any surrogate key here", otherwise would have been ok.



Person
id
name

Help
who (FK)
whom (FK)
description
start_date_time
end_date_time

end_date_time could optionally be part of the primary key (or discriminator if modeled as weak entity) but better not: you want to record a helping task that has not yet finished (and you don't help the person starting on the same day with identical description). All the other attribute need to be part.

Only one person entity because the "roles" of helping and helped person may depend on the particular task

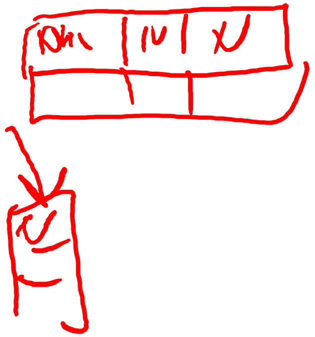
No NN since all FKs are part of PK

Q10: customers who purchased on very few days in 2009

```
WITH Order_days as
    (SELECT customer_name
    FROM ORDERS
    WHERE order_date_year = 2009
    group by customer_name, order_date_month, order_date_day)
select C.customer_name, count(O.customer_name) days
from customer C
left join order_days O on C.customer_name = O.customer_name
group by C.customer_name
order by days asc
```

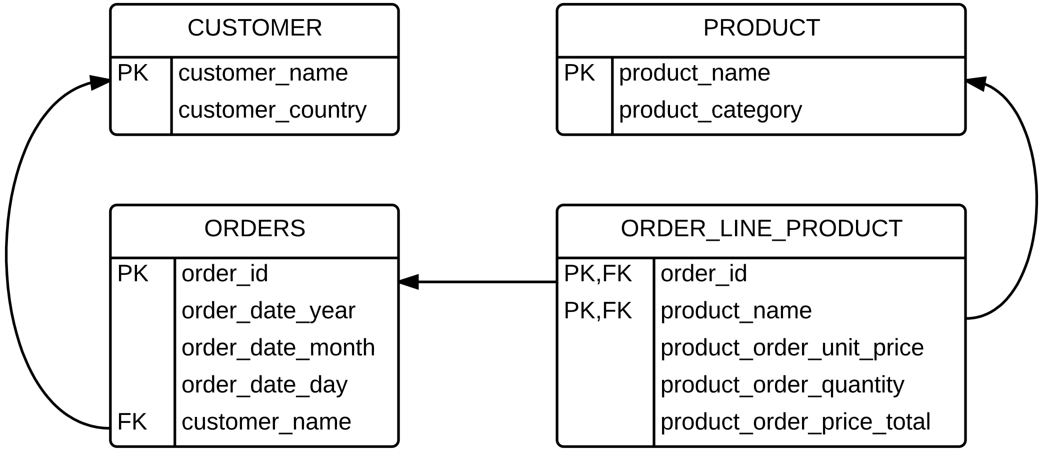
COUNT *

COUNT N



0 → Null

1 → Null 1 null



CUSTOMER	
PK	customer_name
	customer_country

PRODUCT	
PK	product_name
	product_category

ORDERS	
PK	order_id
	order_date_year
	order_date_month
	order_date_day
FK	customer_name

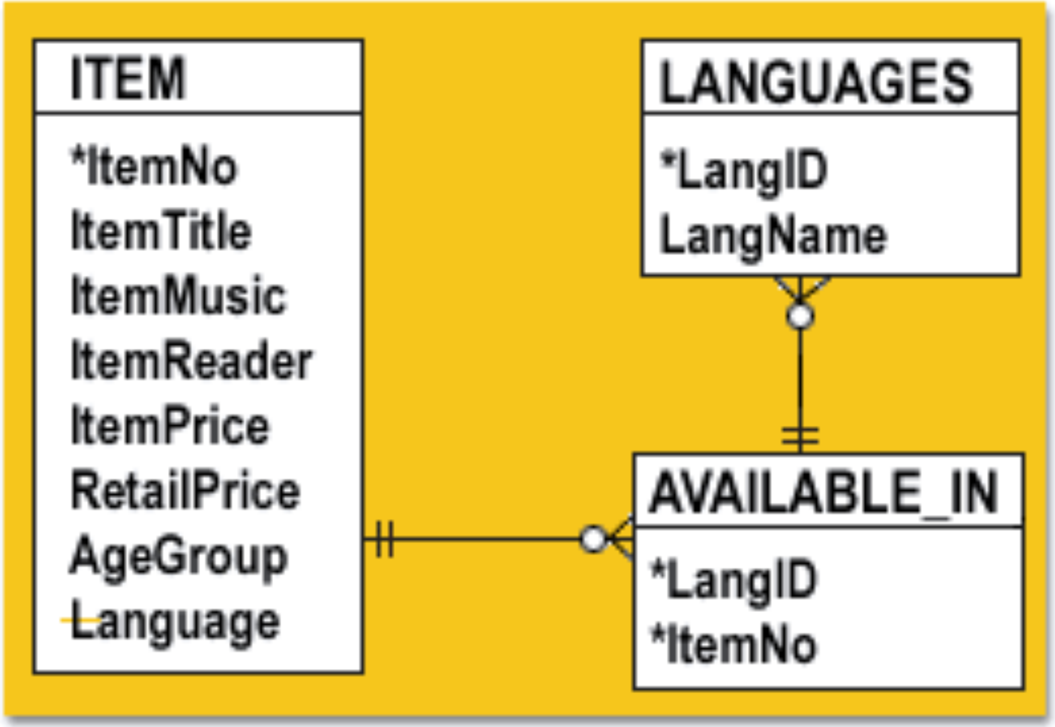
ORDER_LINE_PRODUCT	
PK,FK	order_id
PK,FK	product_name
	product_order_unit_price
	product_order_quantity
	product_order_price_total

```
SELECT c.customer_name,
    COUNT(DISTINCT (order_date_month, order_date_day)) days
FROM (Customer c LEFT JOIN Orders o
ON o.customer_name = c.customer_name
AND o.order_date_year = 2009)
GROUP BY c.customer_name
ORDER BY days
```

count({(null, null)}) will count 1 tuple!

Design

FM even in online tutorials



ITEM
- *ItemNo
- ItemTitle
- ItemMusic
- ItemReader
- ItemPrice
- RetailPrice
- AgeGroup
- Language

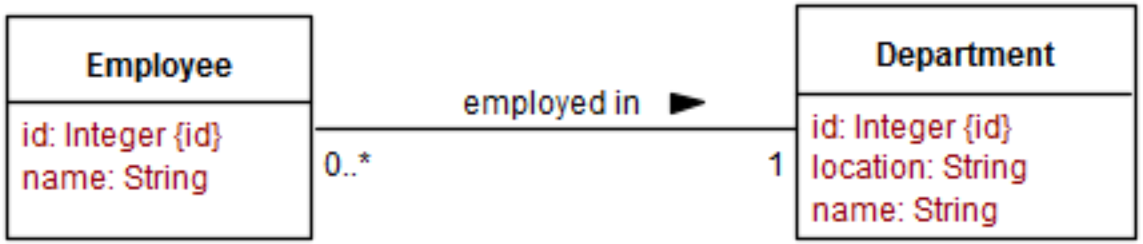
LANGUAGES
- *LangID
- LangName

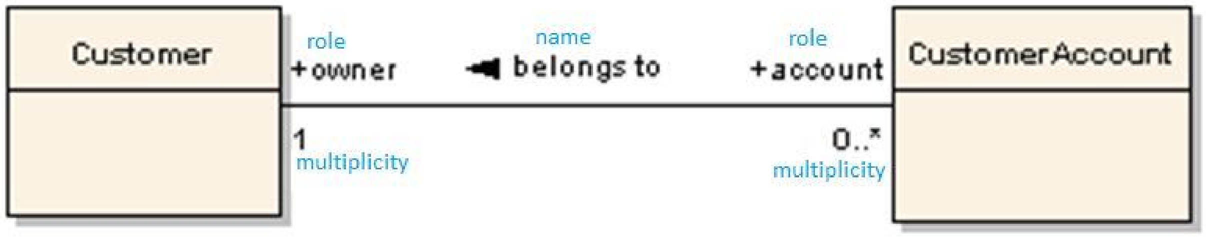
AVAILABLE_IN
- *LangID
- *ItemNo

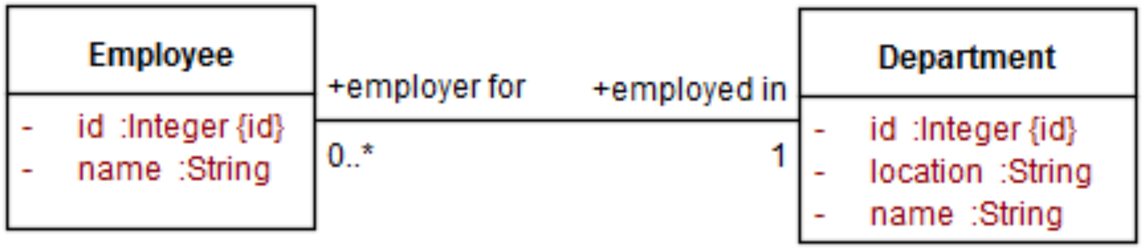
Anything wrong here?

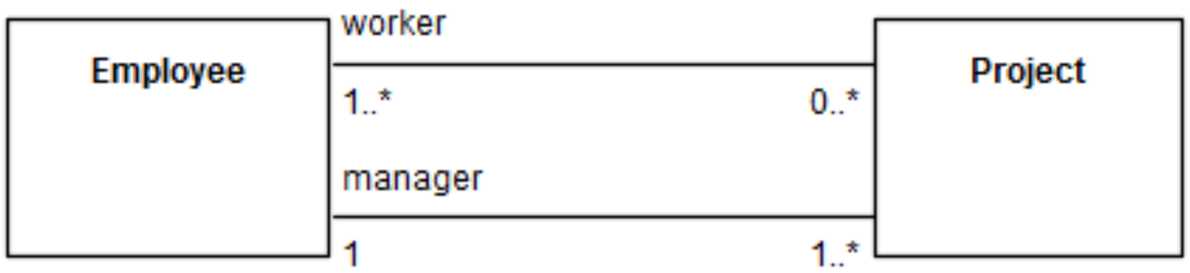
Source: https://www.relationaldbdesign.com/database-analysis/module2/primary-foreign-key-relations.php

UML notation and the name "role"



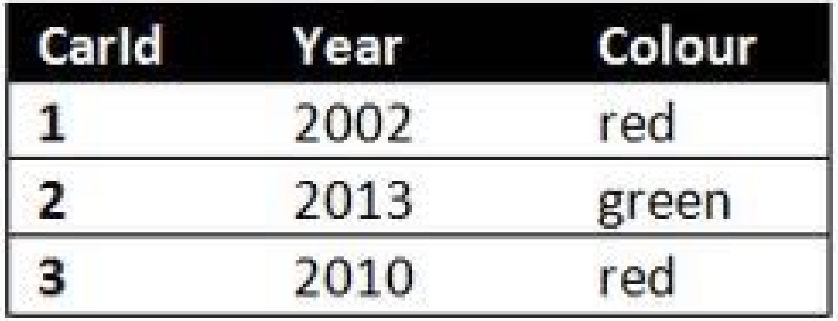




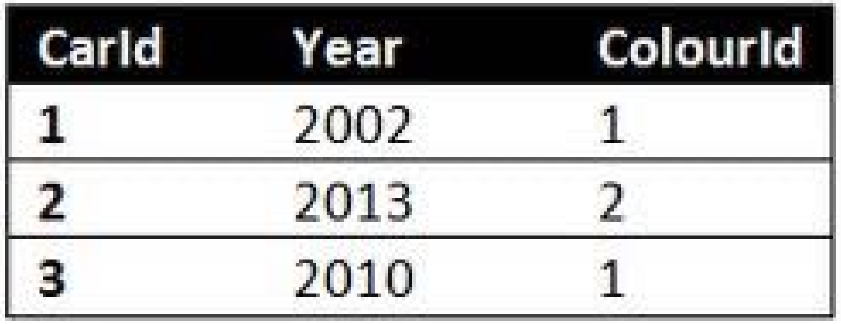


Source: https://training-course-material.com/training/Data_Modelling_with_UML

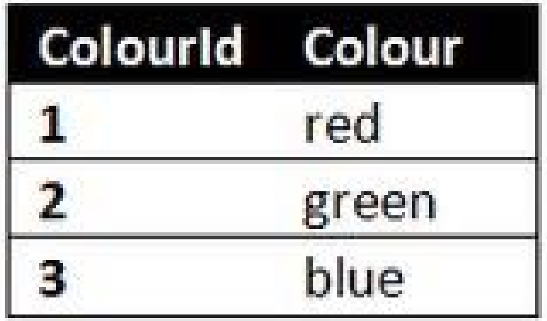
Attributes vs. entities (attributes vs. tables)



CarId	Year	Colour
1	2002	red
2	2013	green
3	2010	red



CarId	Year	ColourId
1	2002	1
2	2013	2
3	2010	1



ColourId	Colour
1	red
2	green
3	blue

Source: https://training-course-material.com/training/Data_Modelling_with_UML

2. Finding FDs

What we will learn about next

- “Good” vs. “Bad” FDs: Intuition
- Finding FDs
- Closures
- PRACTICE: Compute the closures

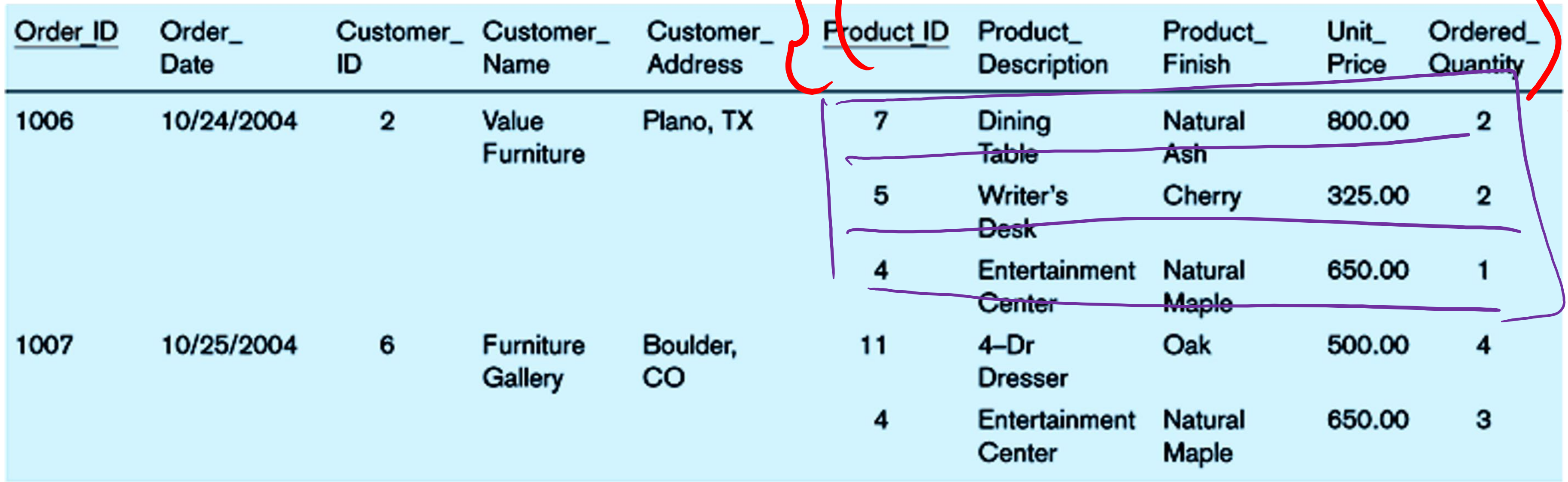
1NF

- First normal form: A relation that has a primary key and in which there are no repeating groups
  - No multivalued attributes
  - Every attribute value is atomic (single fact in each table cell)

- All relations are in 1NF

- Normalization steps (from tabular view of data):
  - *Goal: create new relations from the tabular view*
  - Action: remove repeating groups
  - Action: select the primary key

Example: Convert To 1NF

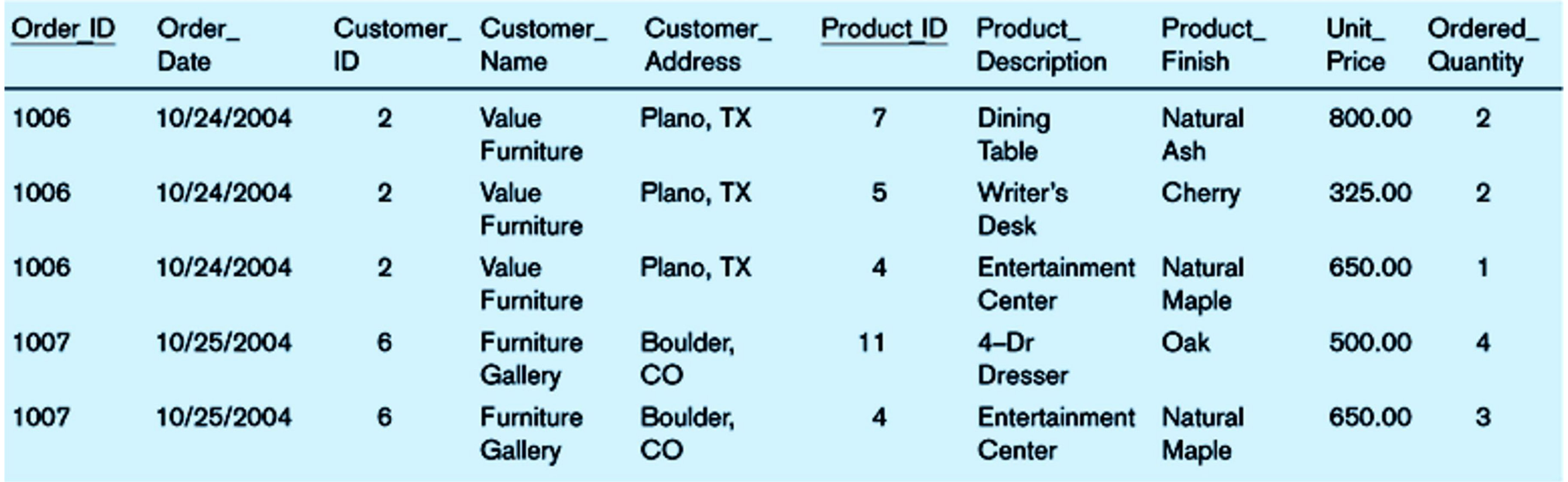


Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

- Normalization steps (from tabular view of data):
  - Goal: create a relation from the tabular view
  - Action: remove repeating groups
  - Action: select the primary key

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

Action: Remove Repeating Groups

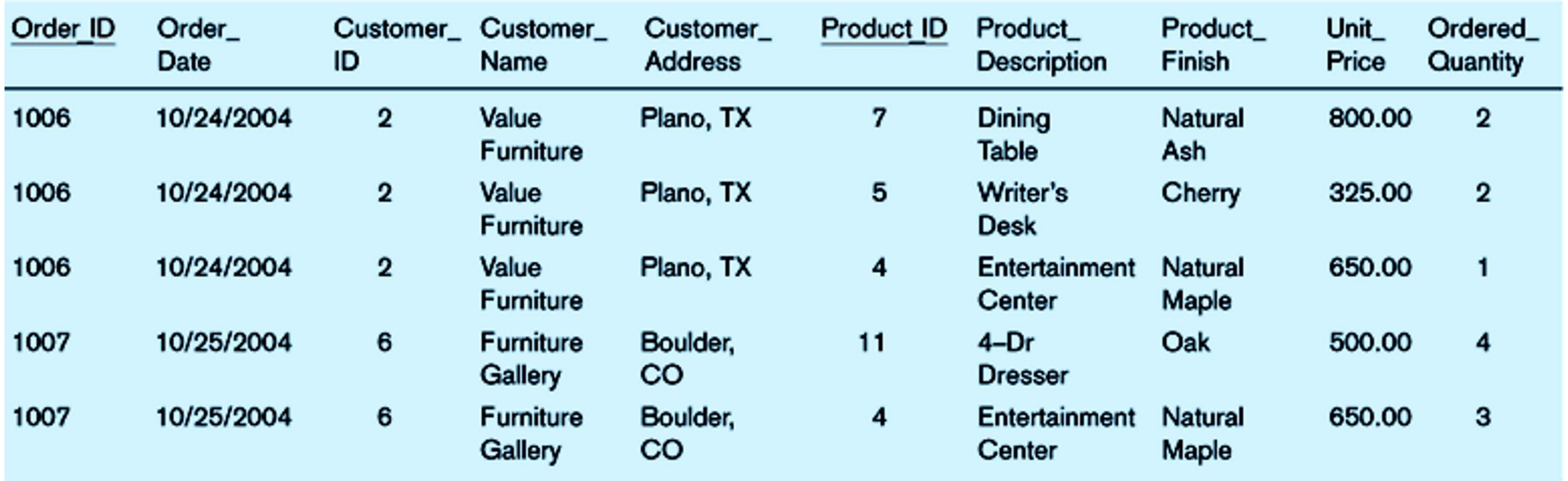


Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

- Is the data view a relation now? ?
- Is it well-structured? ?

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

Action: Remove Repeating Groups

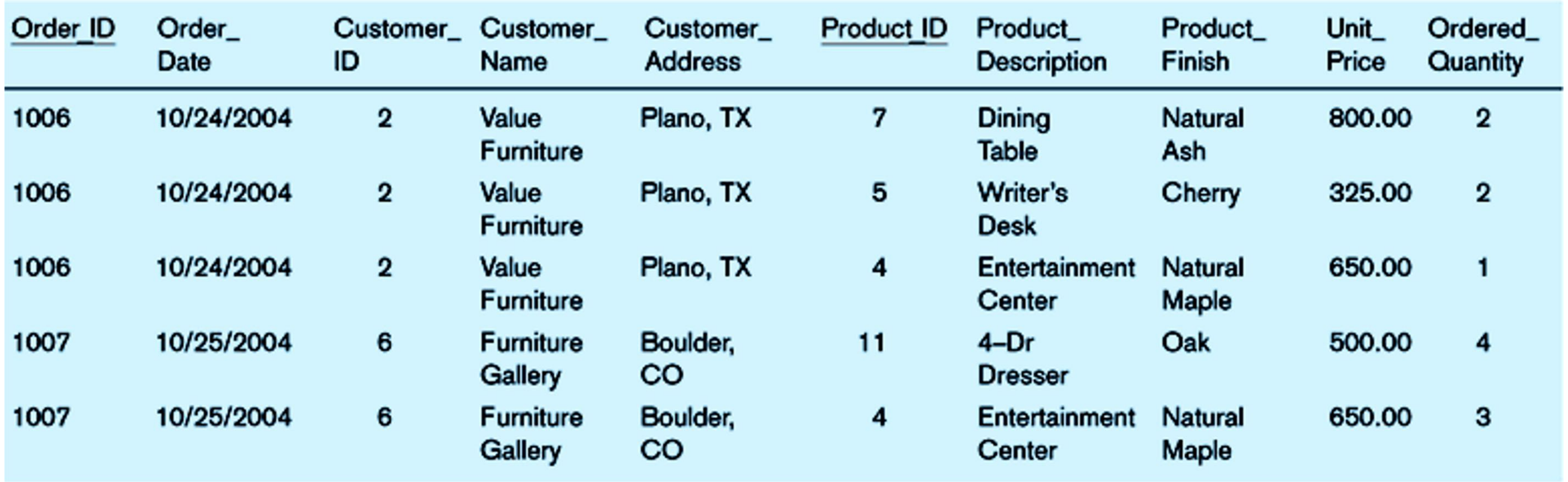


Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address	Product_ID	Product_Description	Product_Finish	Unit_Price	Ordered_Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

- Is the data view a relation now?
  - Answer: yes
- Is it well-structured?
  - Answer: no

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

What are the anomalies in this table?



Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

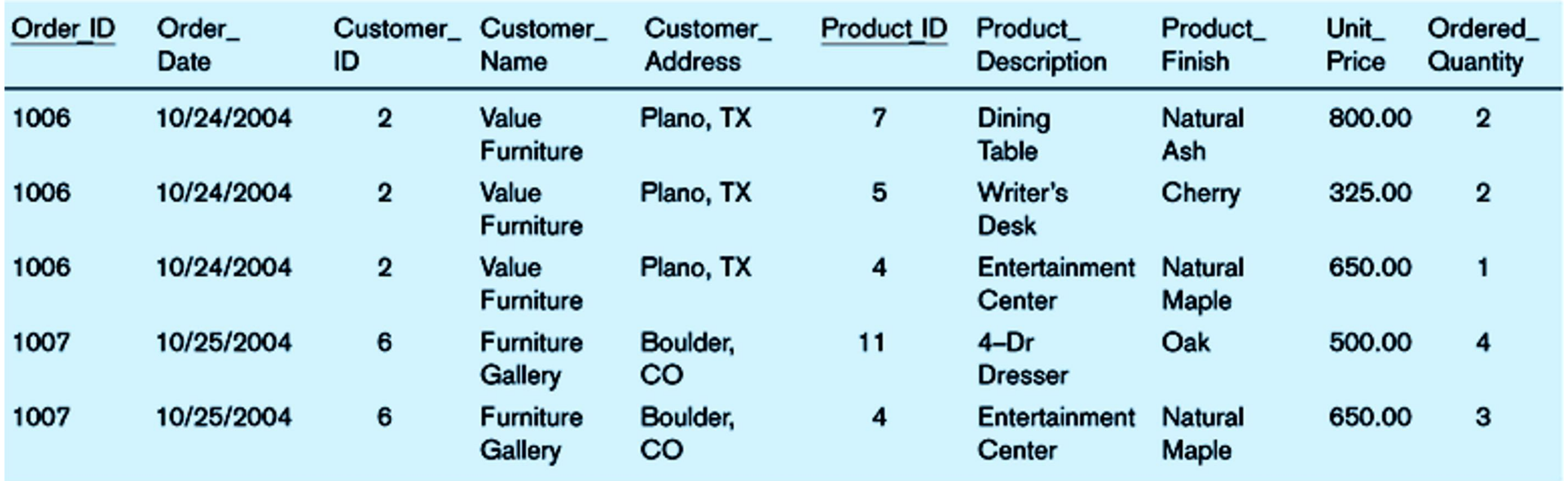
- **Update anomaly**
- **Insert anomaly**

?

- **Deletion anomaly**

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

What are the anomalies in this table?



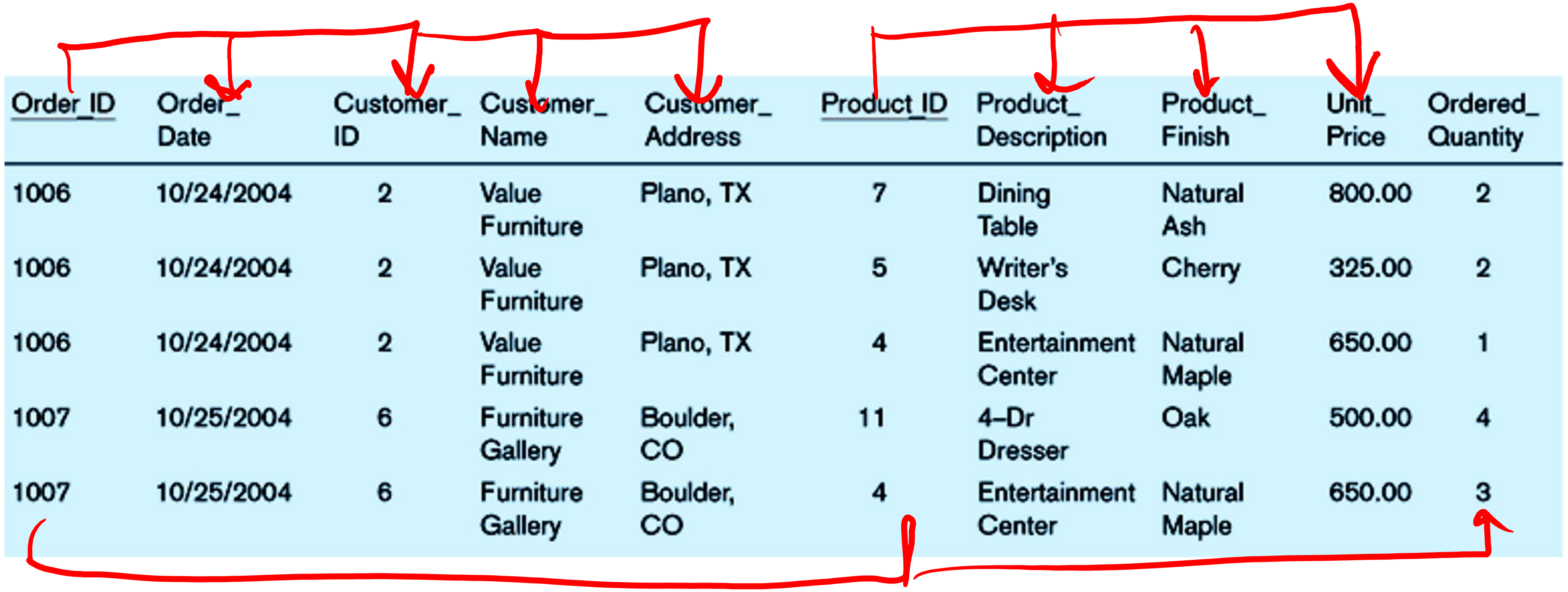
Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

- **Update anomaly** Changing the price of product ID 4 requires update in several records
- **Insert anomaly** If new product is ordered for order 1007 of existing customer, customer data must be re-entered, causing duplication
- **Deletion anomaly** If we delete the Dining Table from Order 1006, we lose information concerning this item's finish and price

Why do these anomalies exist? Because there are multiple themes (entity types) in one relation. -> duplication, and unnecessary dependency between entities

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

What are the anomalies in this table?



Order_ID	Order_ Date	Customer_ ID	Customer_ Name	Customer_ Address	Product_ID	Product_ Description	Product_ Finish	Unit_ Price	Ordered_ Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

- **Update anomaly** Changing the price of product ID 4 requires update in several records
- **Insert anomaly** If new product is ordered for order 1007 of existing customer, customer data must be re-entered, causing duplication
- **Deletion anomaly** If we delete the Dining Table from Order 1006, we lose information concerning this item's finish and price

Why do these anomalies exist? Because there are multiple themes (entity types) in one relation. -> duplication, and unnecessary dependency between entities

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Action: Select A Primary Key

- Identify FDs and CKs (candidate keys = minimal superkeys)
- Four determinants and functional dependencies
  - Order_ID → Order_Date, Customer_ID, Customer_Name, Customer_Address
  - Customer_ID → Customer_Name, Customer_Address
  - Product_ID → Product_Description, Product_Finish, Unit_Price
  - Order_ID, Product_ID → Ordered_Quantity
- Select a PK from CKs
  - (Order_ID, Product_ID)

Remark: notice that (order_id, product_id) determine all other attributes (because it is a candidate key, here even chosen as primary key). However, that is not shown explicitly since it follows from the other FDs

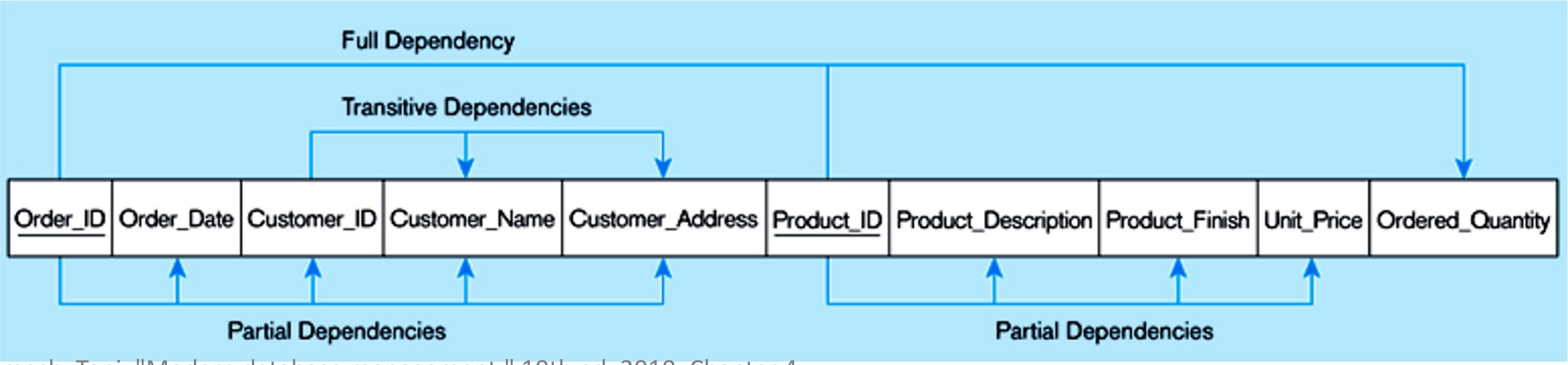


Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

Next Step: Convert To 2NF

- 2NF: A relation in 2NF in which every non-key attribute is fully functionally dependent on the primary key

- Partial FD: A FD in which one or more nonkey attributes are functionally dependent on part (but not all) of the PK

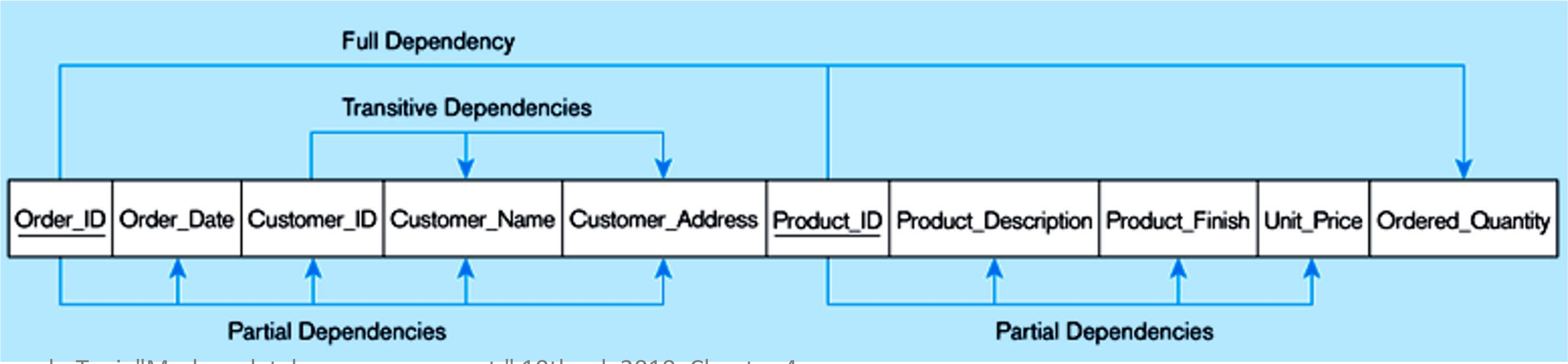


Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

Getting A Relation To 2NF

- Create a new relation for each primary key attribute that is a determinant in a partial dependency
  - That attribute is the primary key in the new relation
- Move the nonkey attributes that are dependent on this primary key attribute(s) from the old relation to the new relation
- Exercise: Convert 1NF relation to 2NF

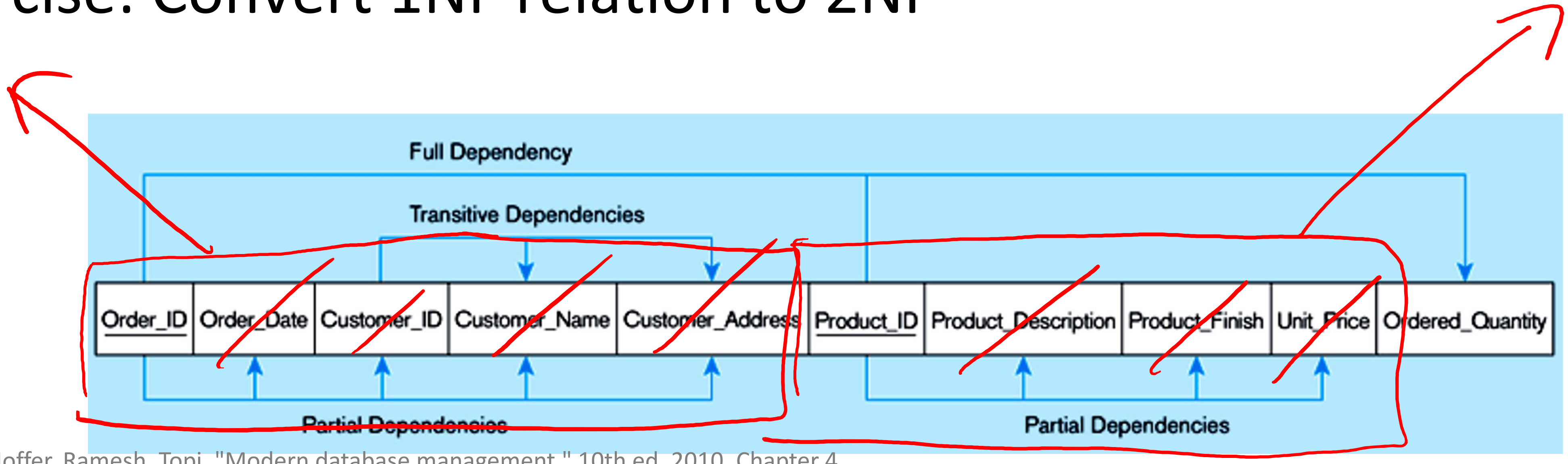
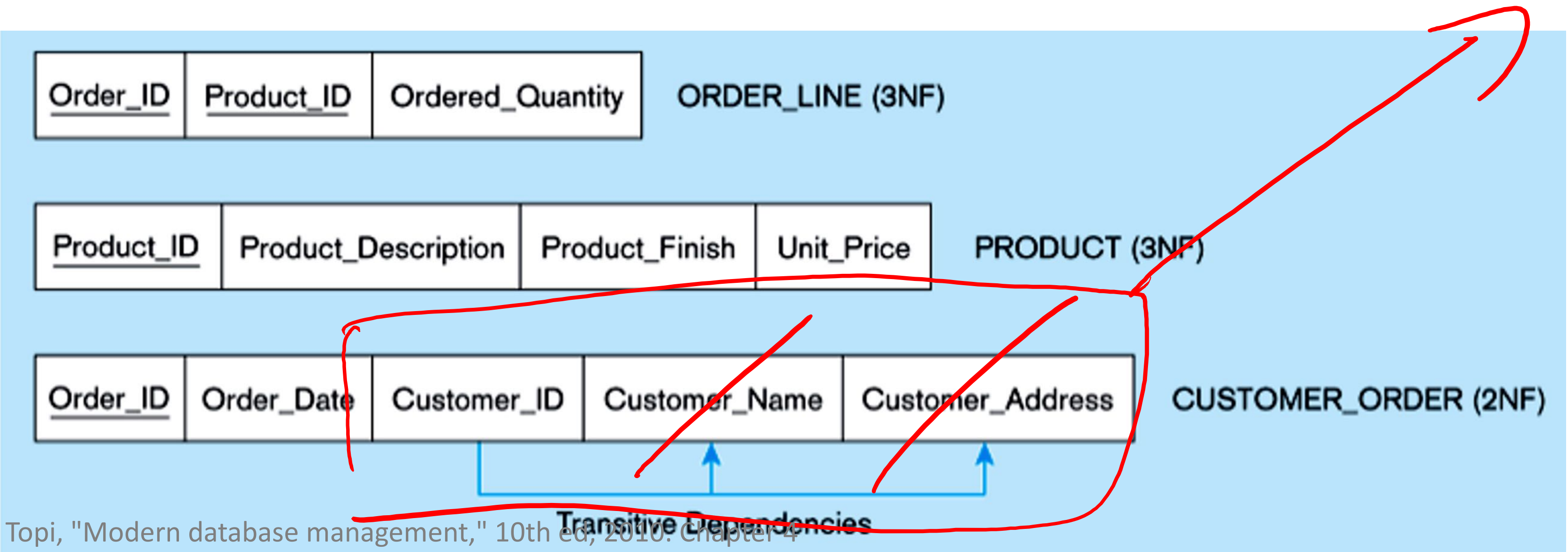


Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

A 1NF Relation Is In 2NF if

- The PK consists of only one attribute. There cannot be a partial dependency in such a relation
- (or) no nonkey attributes exist in the relation (thus all attributes in the relation are components of the PK). There are no FDs in such a relation
- (or) every nonkey attribute is functionally dependent on the full set of PK attributes.



Order_ID | Product_ID | Ordered_Quantity ORDER_LINE (3NF)

Product_ID | Product_Description | Product_Finish | Unit_Price PRODUCT (3NF)

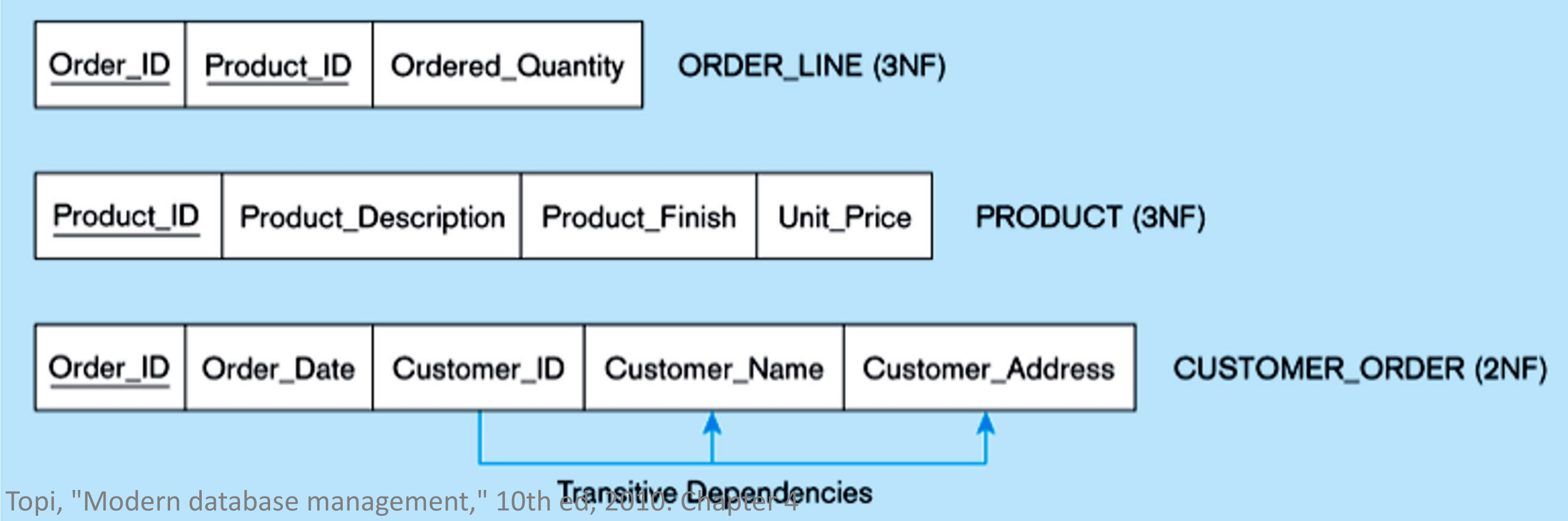
Order_ID | Order_Date | Customer_ID | Customer_Name | Customer_Address CUSTOMER_ORDER (2NF)

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

3NF

- 3NF: A relation that is in 2NF and has no transitive dependencies present
- Transitive dependency: An FD between two (or more) nonkey attributes
  - FD between the PK and one or more nonkey attributes that are dependent on the PK via another nonkey attribute
- Transitive dependency example:

Transitivity:
a<b & b<c ⇒ a<c



Order_ID | Product_ID | Ordered_Quantity ORDER_LINE (3NF)

Product_ID | Product_Description | Product_Finish | Unit_Price PRODUCT (3NF)

Order_ID | Order_Date | Customer_ID | Customer_Name | Customer_Address CUSTOMER_ORDER (2NF)

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

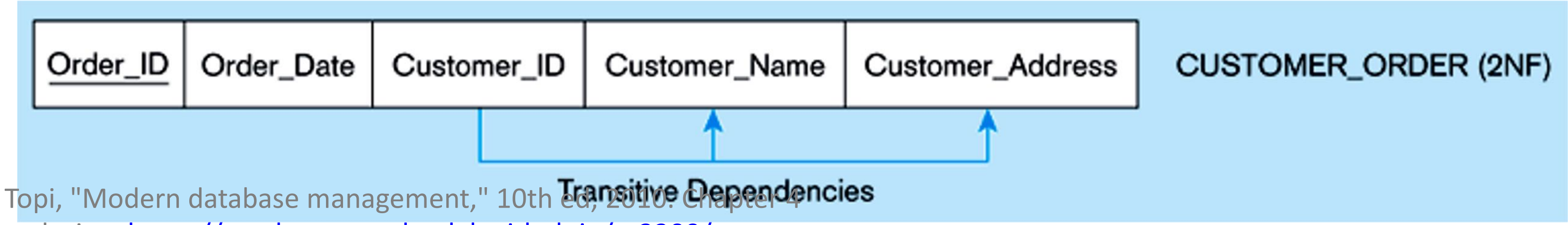
Removing Transitive Dependencies

- For each nonkey attribute(s) that is a determinant in a relation, create a new relation.
  - That attribute becomes the PK of the new relation

- Move all of the attributes that are functionally dependent on the attribute from the old to the new relation

- Leave the attribute (which serves as a PK in the new relation in the old relation to serve as a FK that allows us to associate the two relations

- Exercise: Convert relation below to 3NF



Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address

CUSTOMER_ORDER (2NF)

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010: Chapter 4

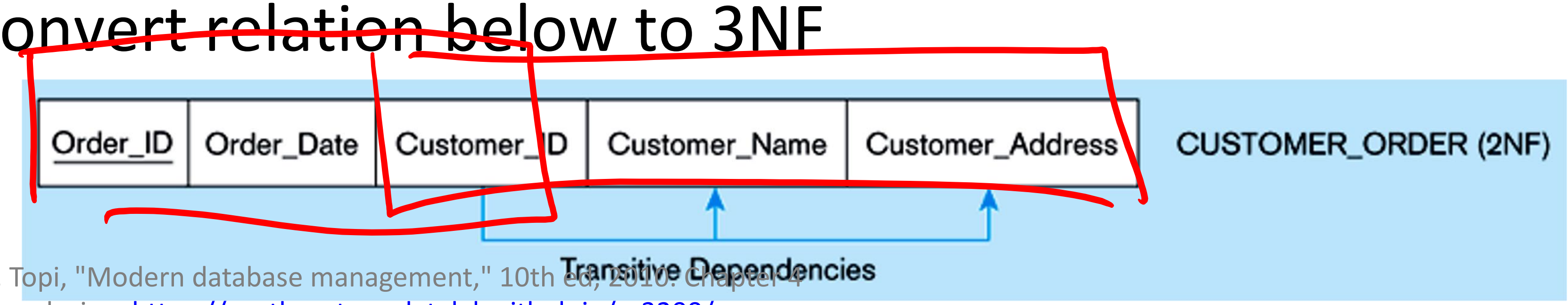
Removing Transitive Dependencies

- For each nonkey attribute(s) that is a determinant in a relation, create a new relation.
  - That attribute becomes the PK of the new relation

- Move all of the attributes that are functionally dependent on the attribute from the old to the new relation

- Leave the attribute (which serves as a PK in the new relation in the old relation to serve as a FK that allows us to associate the two relations

- Exercise: Convert relation below to 3NF



Order_ID | Order_Date | Customer_ID | Customer_Name | Customer_Address

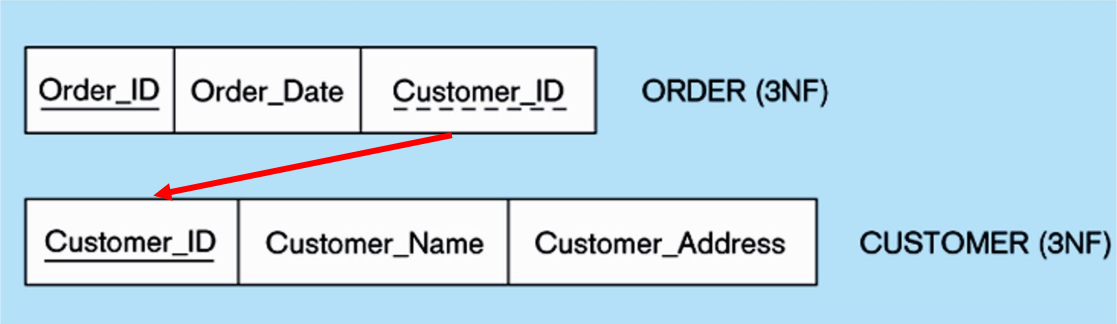
CUSTOMER_ORDER (2NF)

Transitive Dependencies

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

Third Normal Form

- Example converted to 3NF:



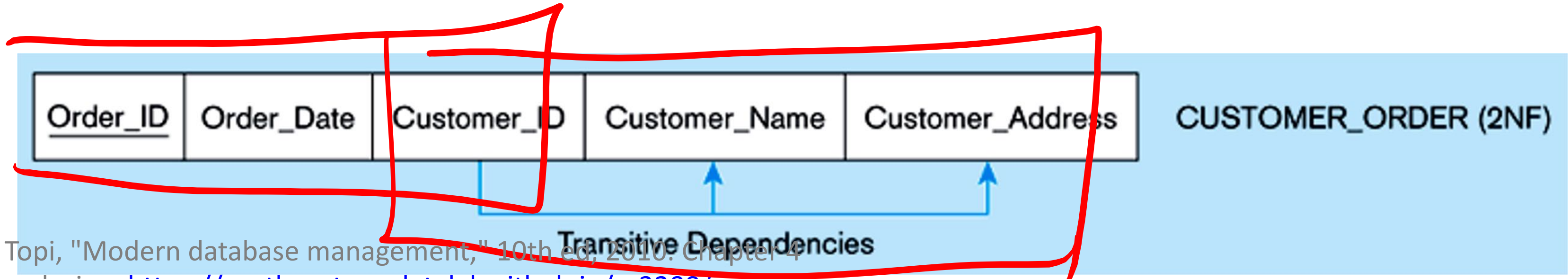
Order_ID	Order_Date	Customer_ID

ORDER (3NF)

Customer_ID	Customer_Name	Customer_Address

CUSTOMER (3NF)

- Original example in 2NF:



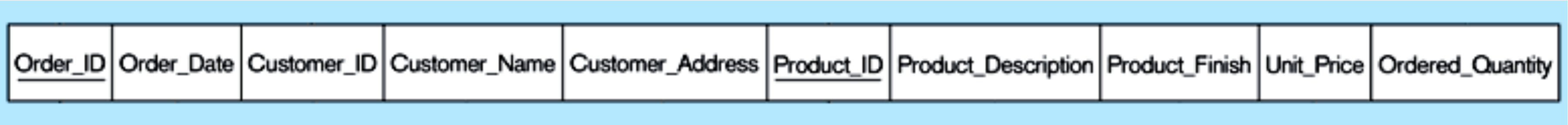
Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address

CUSTOMER_ORDER (2NF)

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4

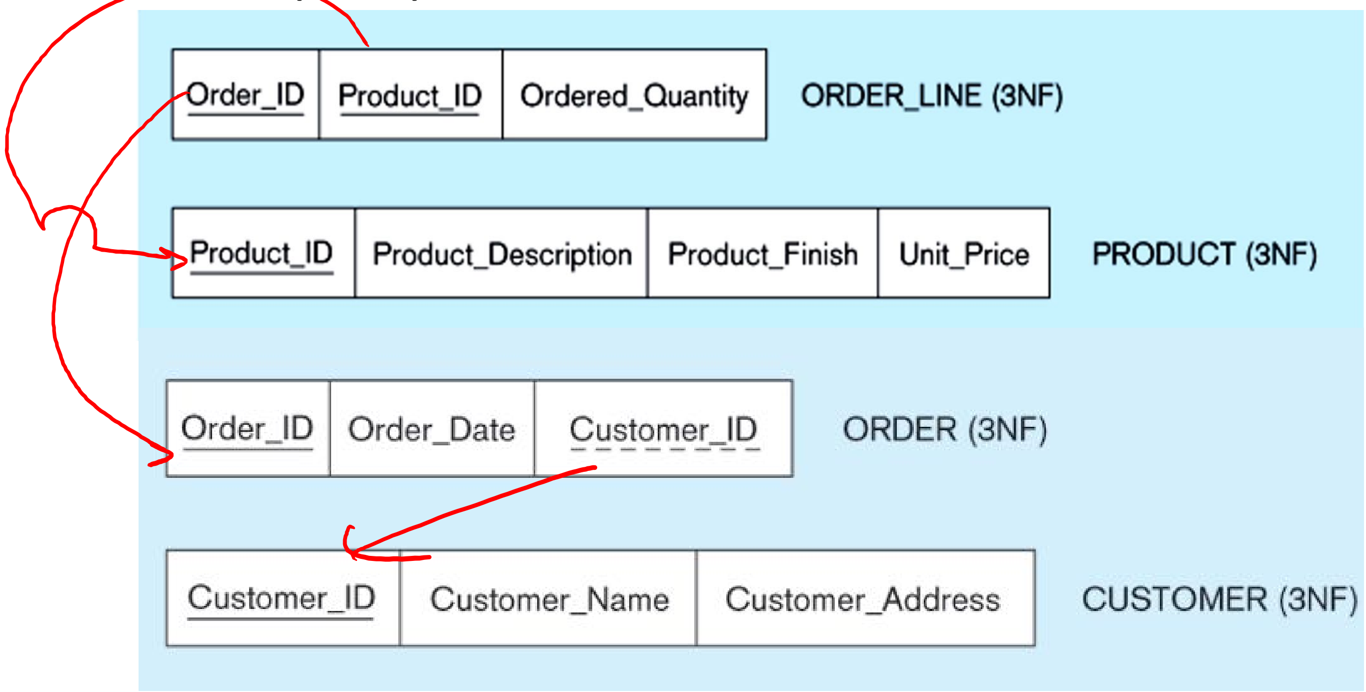
Full Example: From 1NF to 3NF

1NF before starting normalization:



Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address	Product_ID	Product_Description	Product_Finish	Unit_Price	Ordered_Quantity

After (3NF):



Order_ID	Product_ID	Ordered_Quantity

ORDER_LINE (3NF)

Product_ID	Product_Description	Product_Finish	Unit_Price

PRODUCT (3NF)

Order_ID	Order_Date	Customer_ID

ORDER (3NF)

Customer_ID	Customer_Name	Customer_Address

CUSTOMER (3NF)

Figure Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Chapter 4
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Normalization Summary

- Data normalization is the process of decomposing relations with anomalies to produce smaller, well-structured relations

- Goals of normalization include:
  - Minimize data redundancy
  - Simplifying the enforcement of referential integrity constraints
  - Simplify data maintenance (inserts, updates, deletes)
  - Improve representation model to match "the real world"

Quick recap FDs

- **Functional Dependency (FD)**: The value of one set of attributes (the **determinant**) uniquely determines the value of another set of attributes (the **dependents**)
- A **superkey (SK)** is as a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent.
- A **candidate key (CK)** is a non-redundant (minimal) SK (sometimes called just "**a key**")
  - **Prime attribute**: belonging to some candidate key (the opposite is sometimes called a "**nonkey attribute**")
- **Partial FD**: FD in which some non-prime attributes are functionally dependent on part (but not all) of any CK
- **Transitive FD**: An FD between *two (or more) nonkey attributes* (important for distinction 3NF vs BCNF!)
- **3NF**: no partial nor transitive FD

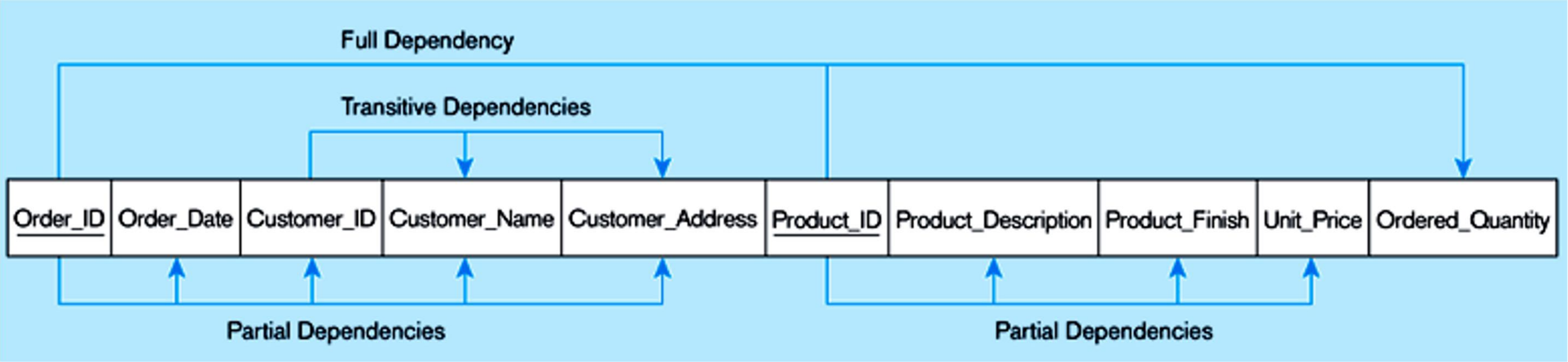


Fig Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

Complete Normalization Practice!

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-03	10:00	need to replace crockery	SG37	Ann Beech	M231 JGR
		22-Apr-04	09:00	in good order	SG14	David Ford	M533 HDR
		1-Oct-04	12:00	damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-04	13:00	replace living room carpet	SG14	David Ford	M533 HDR
		24-Oct-04	14:00	good condition	SG37	Ann Beech	N721 HFR

Can a database store this information? Is it in 1NF?

?

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-03	10:00	6 Lawrence St, Glasgow	need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-04	09:00	6 Lawrence St, Glasgow	in good order	SG14	David Ford	M533 HDR
PG4	1-Oct-04	12:00	6 Lawrence St, Glasgow	damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-04	13:00	5 Novar Dr, Glasgow	replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-04	14:00	5 Novar Dr, Glasgow	good condition	SG37	Ann Beech	N721 HFR

No! Only now a database can store the information: 1NF
But we still need a primary key. What should it be?

?

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-03	10:00	6 Lawrence St, Glasgow	need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-04	09:00	6 Lawrence St, Glasgow	in good order	SG14	David Ford	M533 HDR
PG4	1-Oct-04	12:00	6 Lawrence St, Glasgow	damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-04	13:00	5 Novar Dr, Glasgow	replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-04	14:00	5 Novar Dr, Glasgow	good condition	SG37	Ann Beech	N721 HFR

Now 1NF + PK !

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg

Find and draw all FDs ?

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection

<u>propertyNo</u>	<u>iDate</u>	iTime	pAddress	comments	staffNo	sName	carReg

?

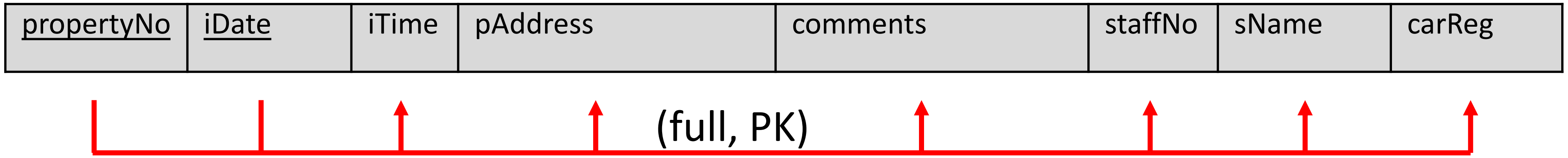
Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection



propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg

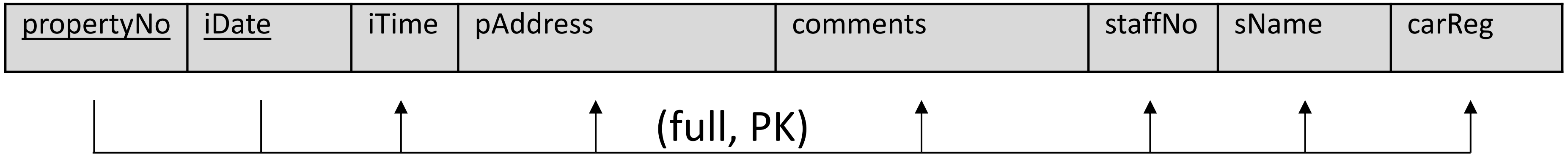
Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection



What can we infer about properties and staff?

?

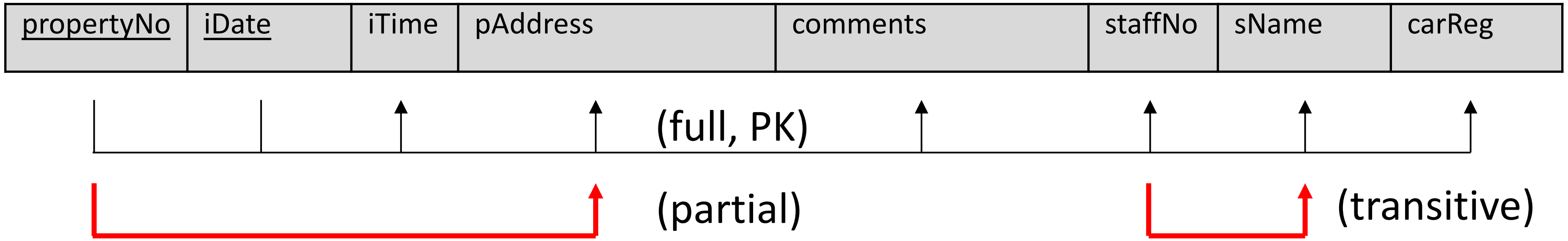
Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection



What can we infer about
properties and staff?

Members of DreamHome inspect properties

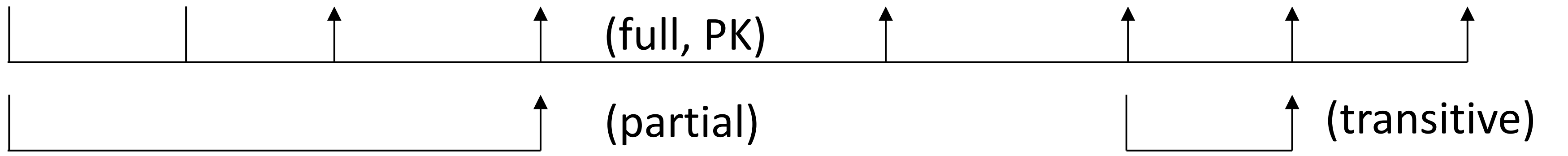
- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

?

Members of DreamHome inspect properties

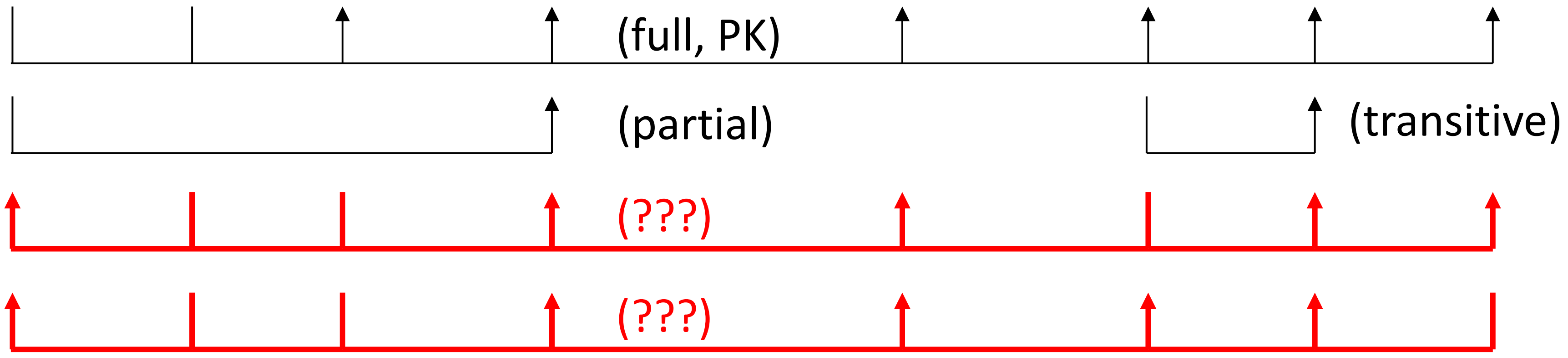
- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

Members of DreamHome inspect properties

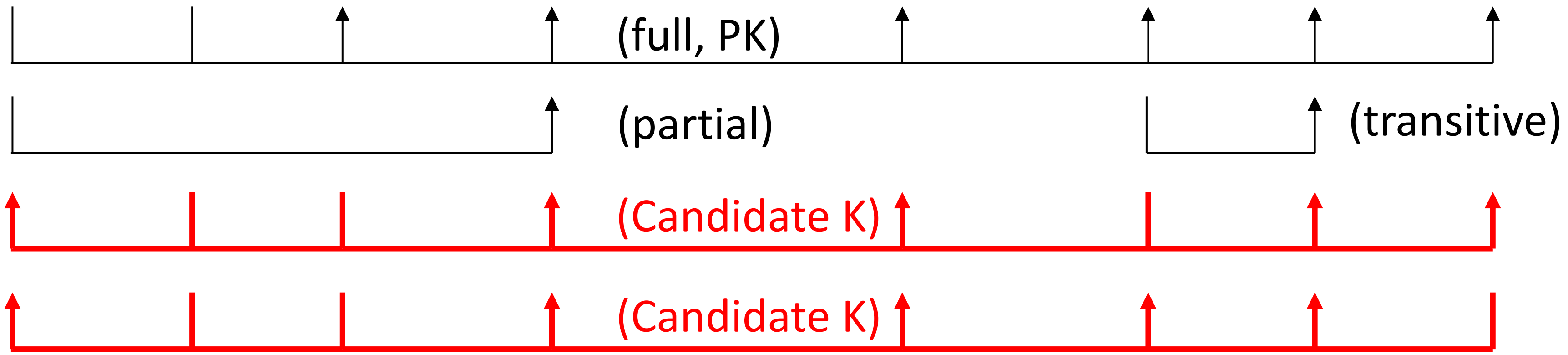
- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection

<u>propertyNo</u>	<u>iDate</u>	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

Recall Candidate Key (CD): "CK is a non-redundant (minimal) set of attributes upon which all attributes are functionally dependent."

Members of DreamHome inspect properties

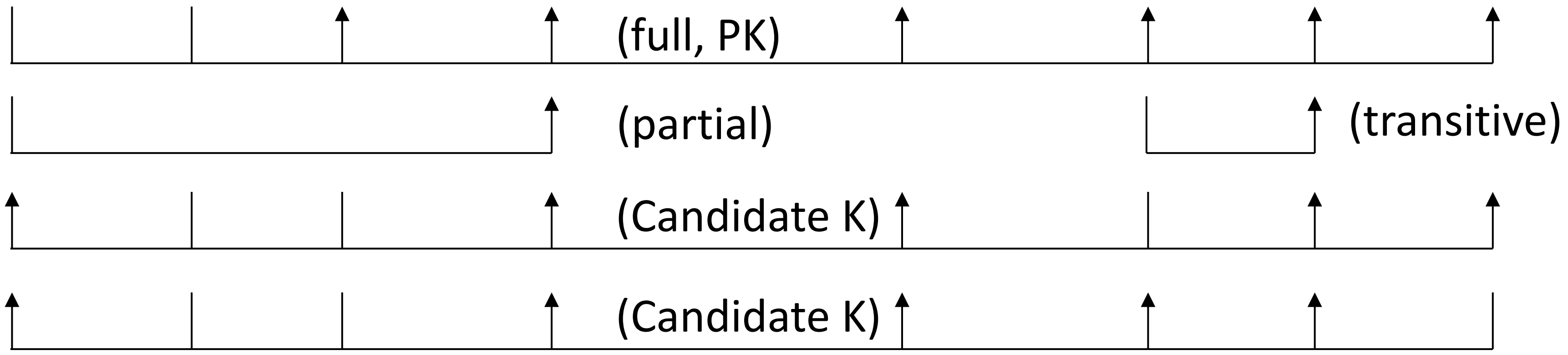
- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

?

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

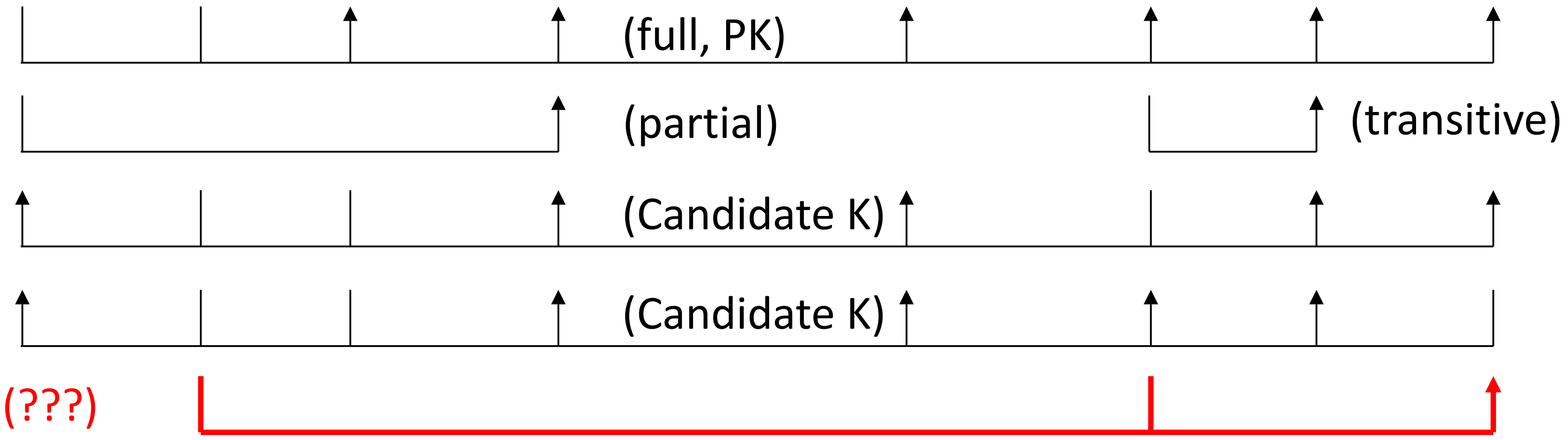
Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

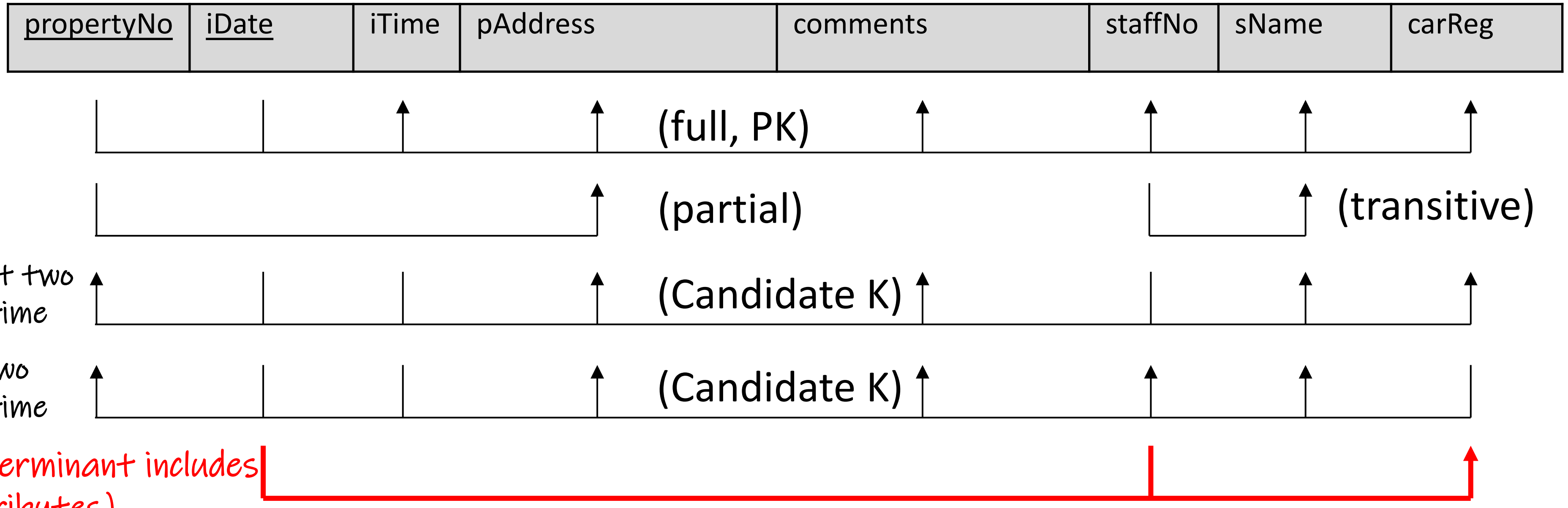
Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.

Example: DreamHome Rental

StaffPropertyInspection



A person cannot be at two places at the same time

A car cannot be at two places at the same time

(not partial: determinant includes non-primary attributes)
(not transitive: determinant does includes non-primary attributes)

Recall partial FD: "FD in which non-prime attributes are functionally dependent on part (but not all) of any Candidate Key"

Members of DreamHome inspect properties

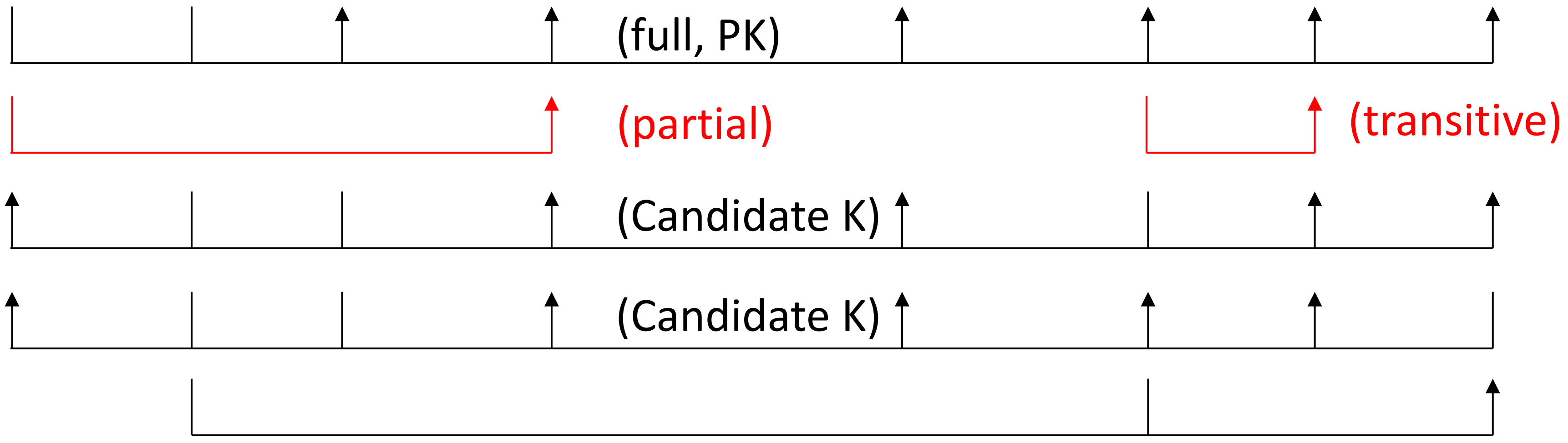
- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental

StaffPropertyInspection

<u>propertyNo</u>	<u>iDate</u>	iTime	pAddress	comments	staffNo	sName	carReg



A person cannot be at two places at the same time

A car cannot be at two places at the same time

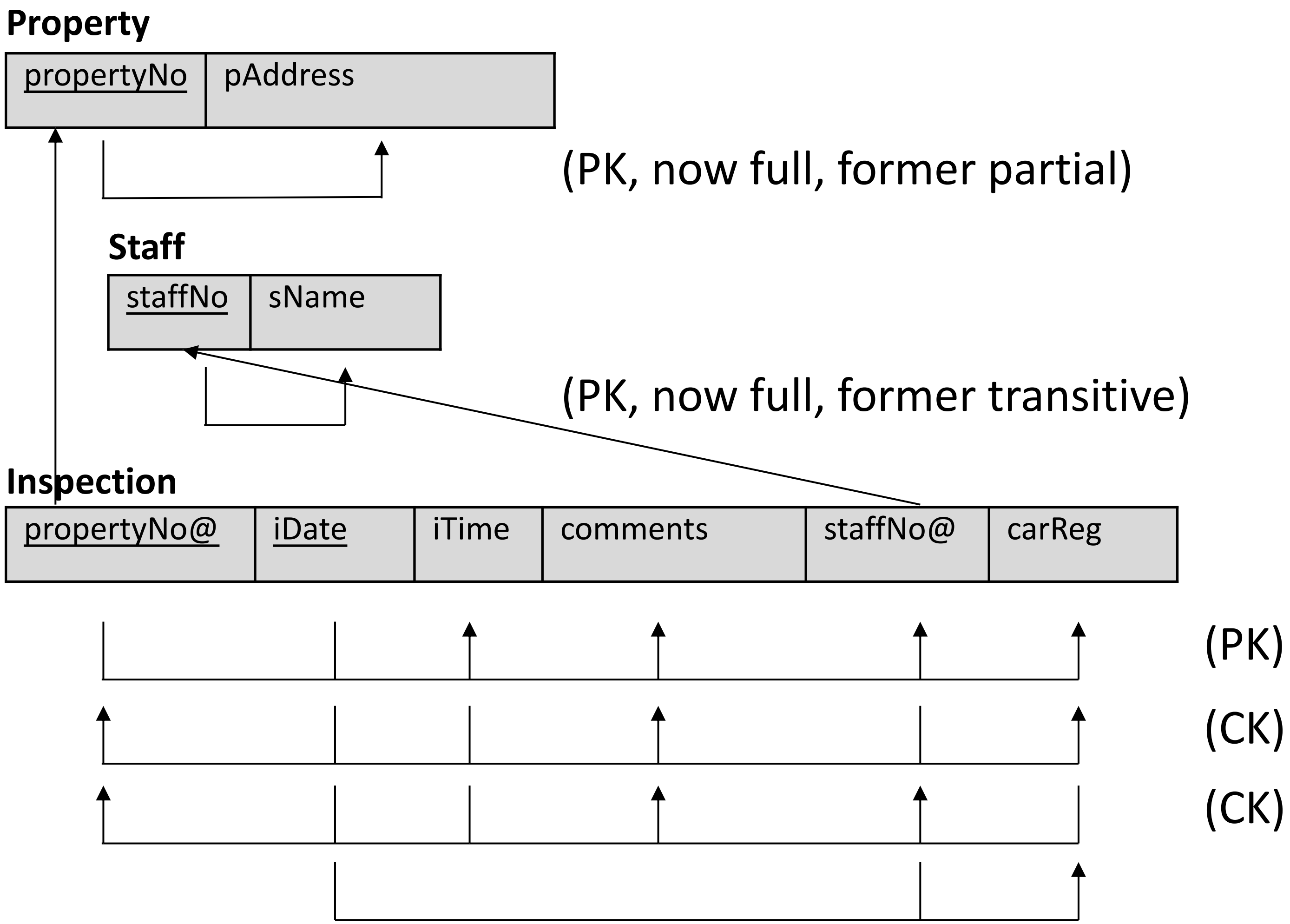
How do we decompose? ?

Members of DreamHome inspect properties

- When staff are required to undertake these inspections, they are allocated a company car for use on the day of the inspections. (One car per person & day)
- However, a car may be allocated to several members of staff as required throughout the working day.
- A member of staff may inspect several properties on a given date, but a property is only inspected once on a given date.

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental



A person cannot be at two places at the same time

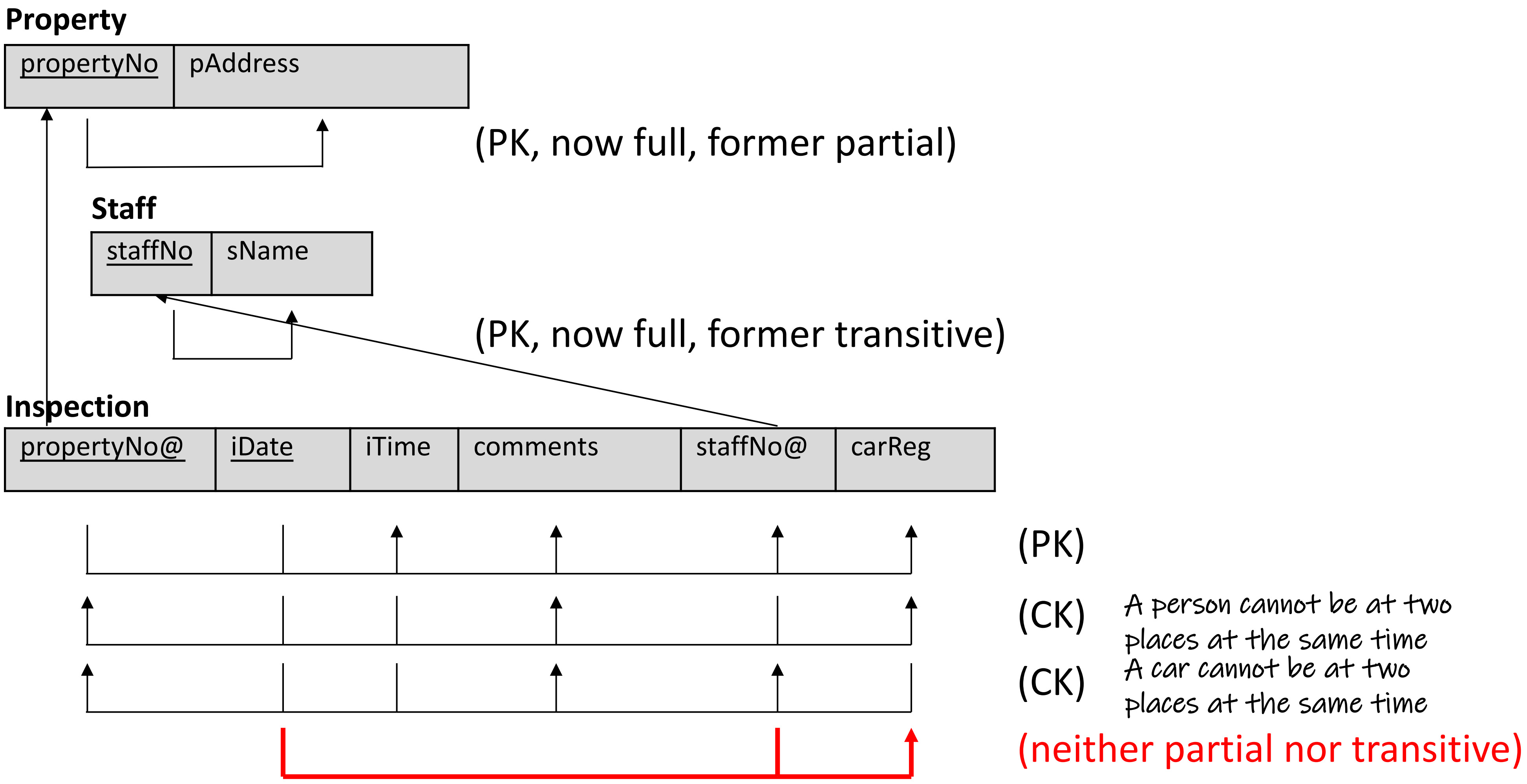
A car cannot be at two places at the same time

What next?

?

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental



Property

<u>propertyNo</u>	pAddress

(PK, now full, former partial)

Staff

<u>staffNo</u>	sName

(PK, now full, former transitive)

Inspection

<u>propertyNo@</u>	<u>iDate</u>	iTime	comments	staffNo@	carReg

(PK)

(CK) A person cannot be at two places at the same time

(CK) A car cannot be at two places at the same time

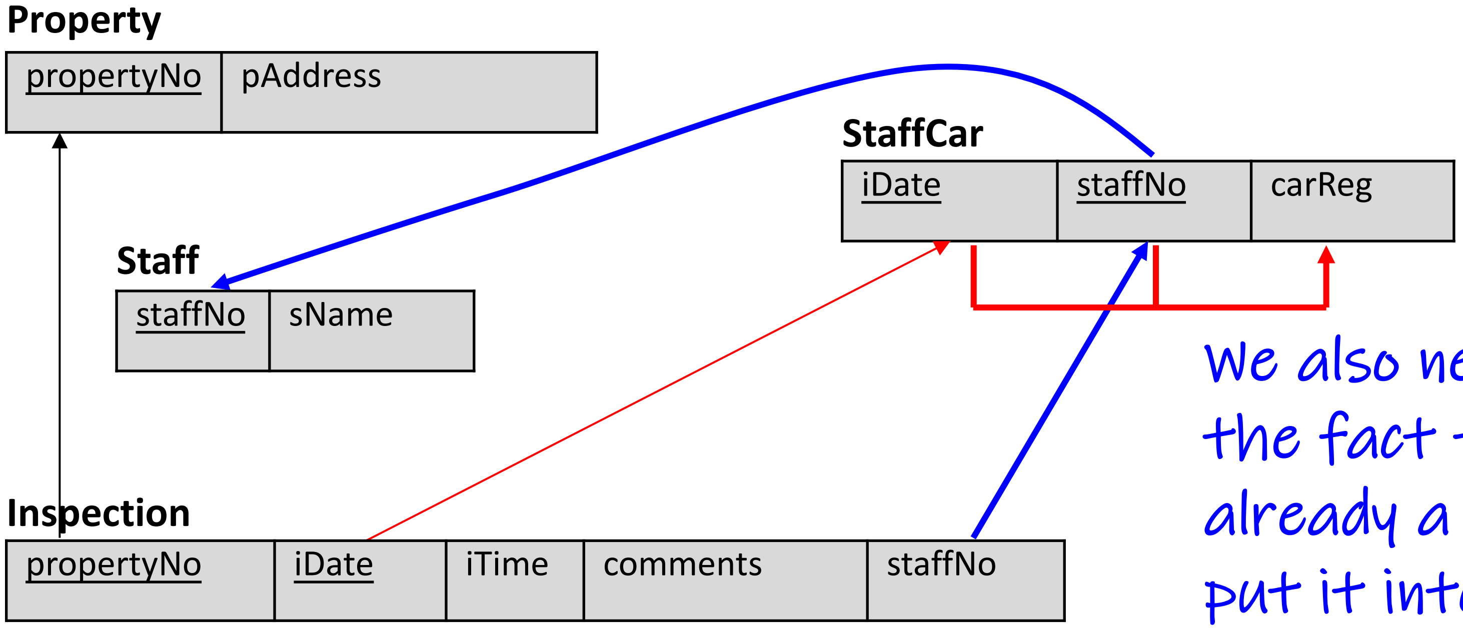
(neither partial nor transitive)

If we only want to decompose into 3NF, then we are done. The shown FD can also create anomalies and another NF called BCNF (Boyce-Codd NF) still decomposes that too. How?

?

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental



Property

propertyNo	pAddress

StaffCar

iDate	staffNo	carReg

Staff

staffNo	sName

Inspection

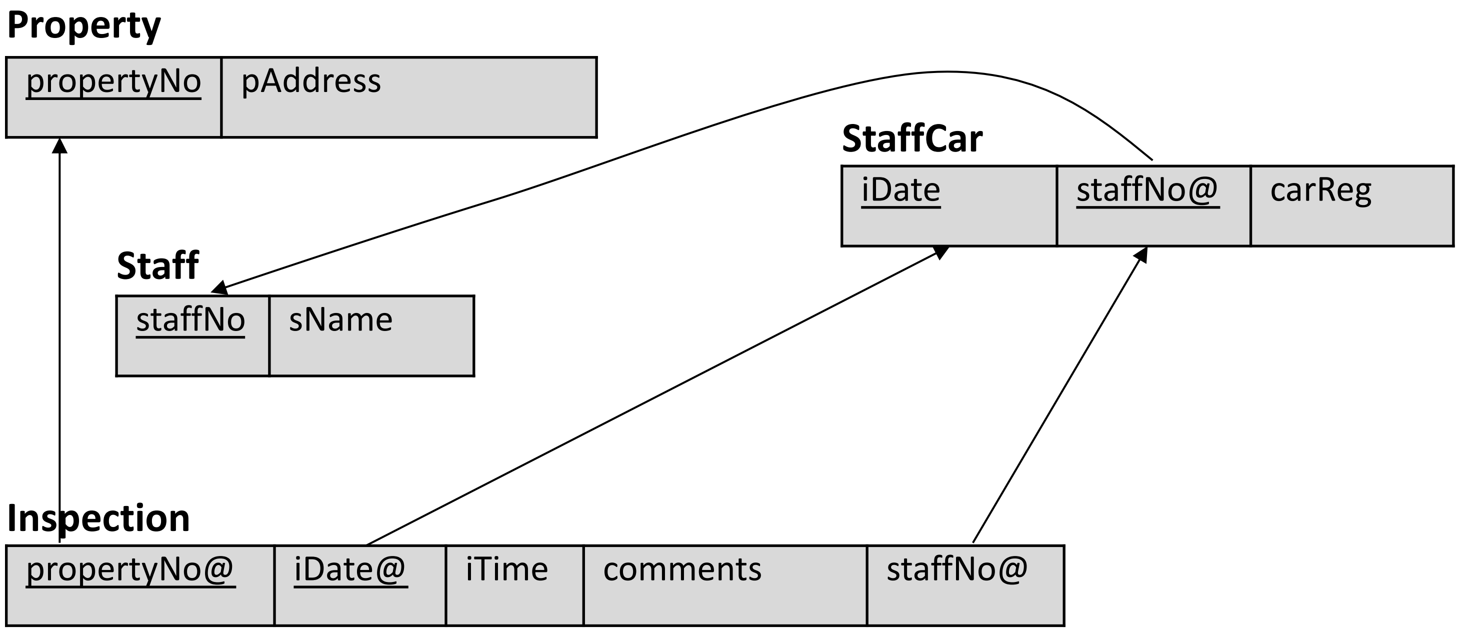
propertyNo	iDate	iTime	comments	staffNo

(PK)

We also need to keep track of the fact that "staffno@" was already a foreign key before we put it into another table

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental



Property

propertyNo | pAddress

StaffCar

iDate | staffNo@ | carReg

Staff

staffNo | sName

Inspection

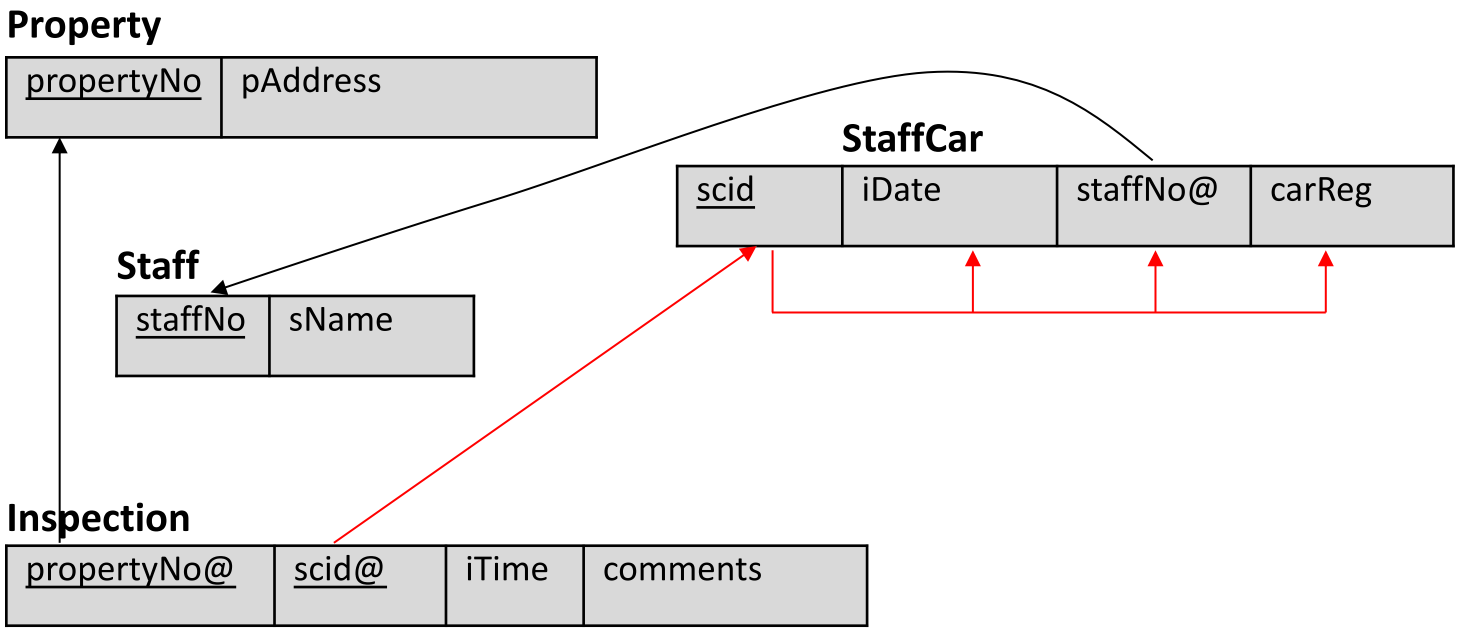
propertyNo@ | iDate@ | iTime | comments | staffNo@

Extra question: We now have a composite FK (idate, staffno) from INSPECTION to STAFFCAR. Thus (idate, staffno) is a composite PK in STAFFCAR. Assume we like to replace it with a surrogate key. How would the resulting completely normalized tables look like?

?

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Example: DreamHome Rental



Property

propertyNo	pAddress

StaffCar

scid	iDate	staffNo@	carReg

Staff

staffNo	sName

Inspection

propertyNo@	scid@	iTime	comments

This is now fully normalized.
Downside: we need to join INSPECTION with STAFFCAR every time we like to find out about when a property (by "properyNo") was last inspected

Source: Connolly, Begg: Database systems, 4th ed, p. 423, 2005.