Institución Universitaria
Acreditada en Alta Calidad

ANDRES
MARTINEZ
GUTIERREZ

2018-2

DATABASE ADMINISTRATION ADVANCED

AGENDA

- Relational Model
- Common mistakes designing databases
- Functional dependency
- Normalization

Naming Conventions

Naming Conventions

- Avoid quotes ("First Name”)
- Lowercase (first_name, last_name)
- Data types are not names (text, timestamp)
- Underscores to separate words
- Not abbreviations
- Avoid reserved words (User, order, lock, table)

Naming Conventions

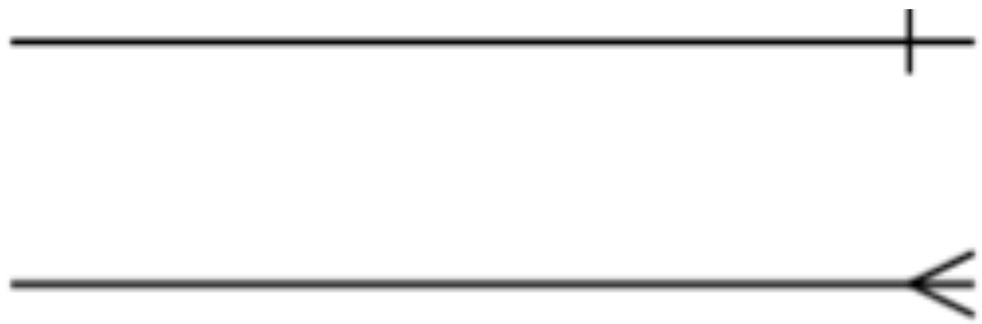
- Tables, views in plural (subscriptions, cohorts
- Primary keys (id)
- Foreign keys ("#{prefix_ot_table}_id" pet_id, user_id, cohort_id)
- Not prefix, not suffix (TB_ , SP_, VW_)

Naming Conventions

Don't	Do
tblContact	contacts
tblEmployee	employees
"All Employees"	all_employees
wordCount	word_count
wordcount	word_count
mid_nm	mid_name
MovieExect	movie_executive
MovieStar	movie_star

EDR Notation

EDR Notation



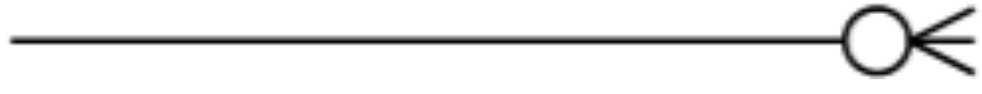
One

Many

One (and only one)

Zero or one

One or many



Zero or many

Functional Dependency

Functional dependency

- Campo clave determina a un no clave

- **A -> B** ("A determina a B" o "B es funcionalmente dependiente de A”)

- **A, B -> C** ("A y B determinan C" o "C es funcionalmente dependiente de A y B")

Functional dependency

- **Dependencia Parcial:**

  - Parte de la clave determina un campo no clave.
    - *Sólo se presenta en claves* ***COMPUESTAS***

- **Dependencia Transitiva:**

  - Un campo no clave determina a otro campo no clave

Functional dependency

- Dependencia Parcial (Ejemplo):

Student_id	Course_id	Student_name	Course_name

- Which would be the primary key?

- Is there any partial dependency? Why?

Functional dependency

- Dependencia Transitiva (Ejemplo):

Book	Author	Length	Author_age

- Which would be the primary key?
- Is there any transitive dependency? Why?

Normalization

http://www.bkent.net/Doc/simple5.htm
http://www.tutorialspoint.com/dbms/database_normalization.htm
http://www.mahipalreddy.com/dbdesign/dbqa.htm#fd
https://mariadb.com/kb/en/library/database-normalization-4th-normal-form/
https://www.tutorialcup.com/dbms/fourth-normal-form.htm
https://www.guru99.com/database-normalization.html

Normalization

- Reduce the space needed to describe your data
- To prevent arbitrary and artificial data absurdities
- To restrict how many of something can be related to something else
- Process to remove anomalies
  - **Insertion, Update, Deletion**

Normalization

- **Normalized:** Designed to *minimize* redundancy
- **Denormalized:** Designed to *optimize* read time

“Denormalization is commonly used to create highly scalable systems”

Premature Optimization

“premature optimization is the root of all evil”

Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: ***premature optimization is the root of all evil***. Yet we should not pass up our opportunities in that critical 3%.”

Donald Knuth

1st Normal Form

1st Normal Form

- 1st Normal Form
- Attributes in a relation must have atomic domains.
- The values in an atomic domain are indivisible units.
- **"Remove repeating groups"**

1st Normal Form

Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

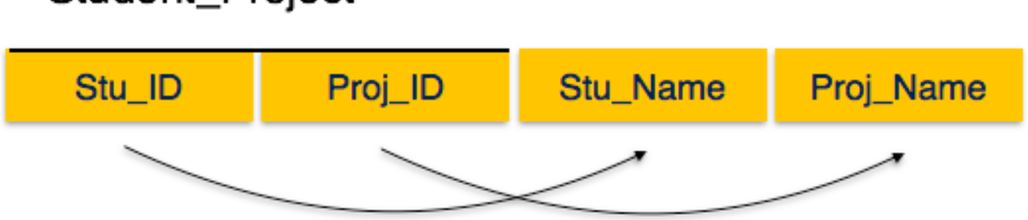
2nd Normal Form

2nd Normal Form

- 1st Normal form.
- Every non-prime attribute should be fully functionally dependent on prime key attribute
- **"Remove partial dependencies"**

2nd Normal Form

Student_Project



Student

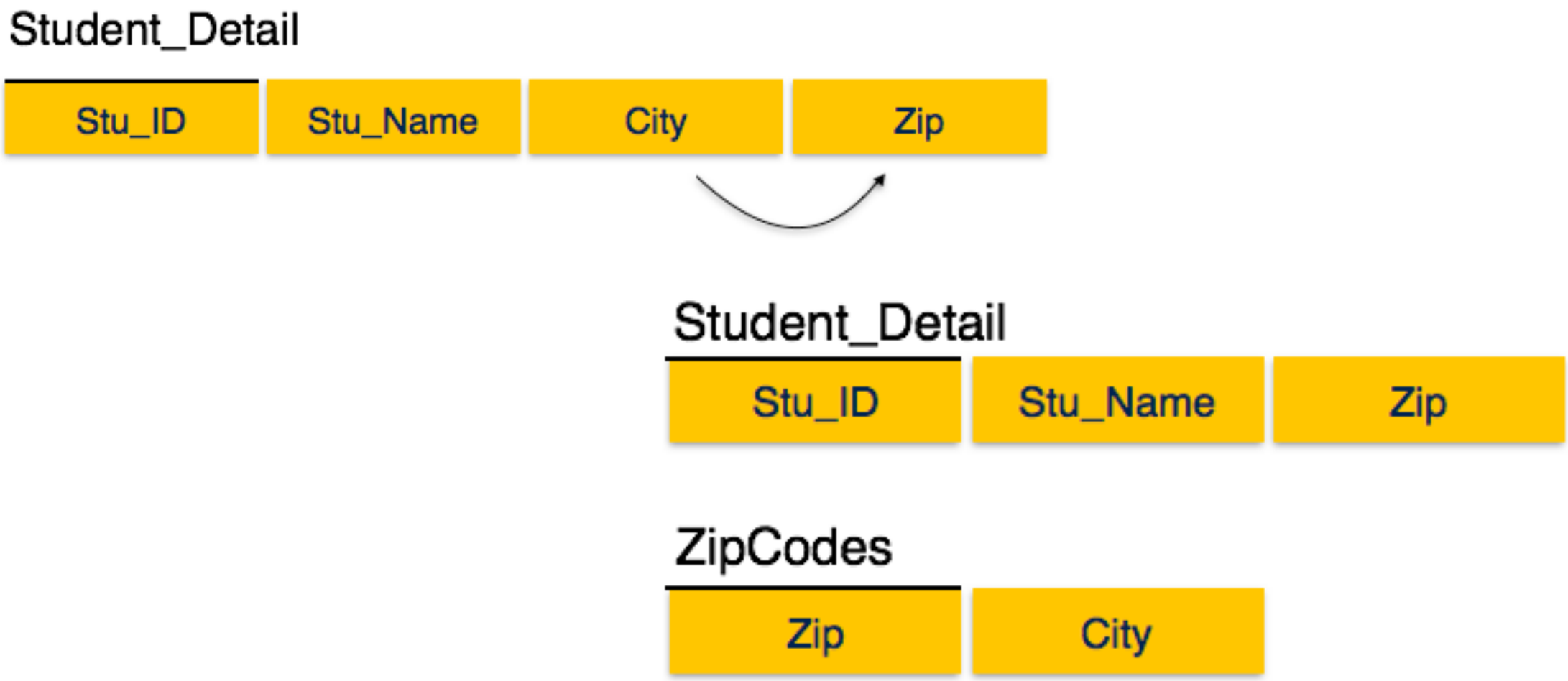
Project

3rd Normal Form

3rd Normal Form

- 2nd Normal form.
- No non-prime attribute is transitively dependent on prime key attribute.
- **"Remove transitive dependencies"**
  - “A record is in second and third normal forms if every field is either part of the key or provides a (single-valued) fact about exactly the **whole key** and nothing else”

3rd Normal Form



3rd Normal Form

Tournament Winners

Tournament	*Year*	*Winner*	*Winner Date of Birth*
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Tournament Winners

Tournament	*Year*	*Winner*
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth

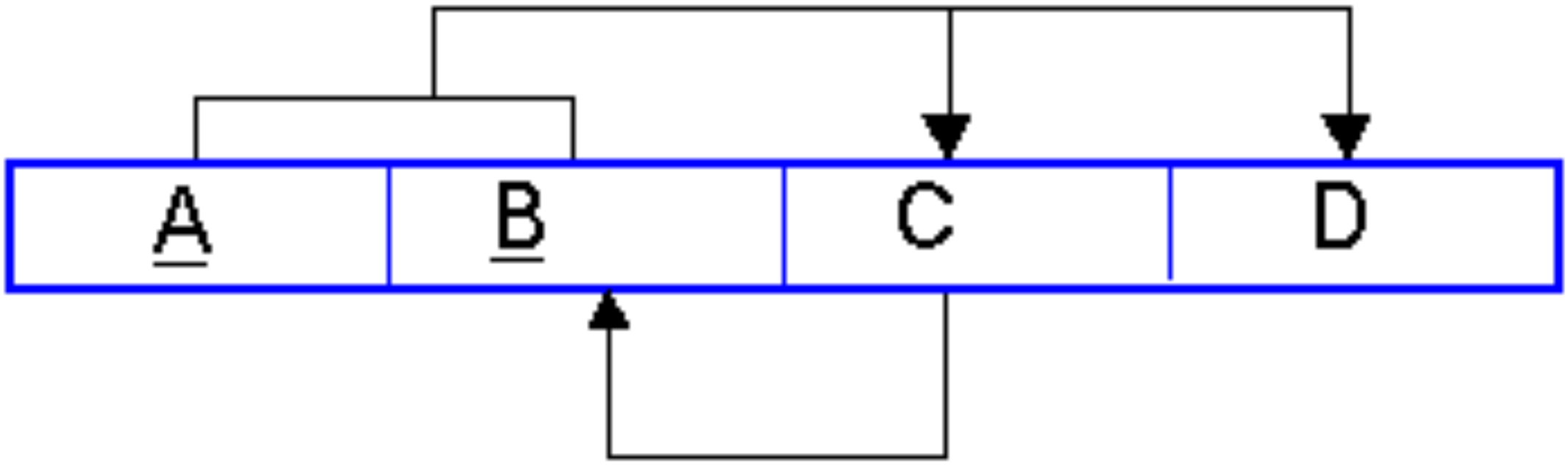
Winner	*Date of Birth*
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

Boyce-Codd Normal Form

Boyce-Codd Normal Form

- *Usually* tables that are in Third Normal Form are already in Boyce Codd Normal Form
- ***A non key attribute is a determinant of a key attribute***

Boyce-Codd Normal Form



Boyce-Codd Normal Form

S_Num	T_Code	Offering#	Review Date
123599	FIT104	01764	2nd March
123599	PIT305	01765	12th April
123599	PIT107	01789	2nd May
346700	FIT104	01764	3rd March
346700	PIT305	01765	7th May

Boyce-Codd Normal Form

StudentReview

S_Num	Offering#	Review Date
123599	01764	2nd March
123599	01765	12th April
123599	01789	2nd May
346700	01764	3rd March
346700	01765	7th May

OfferingTeacher

Offering#	T_Code
01764	FIT104
01765	PIT305
01789	PIT107

Boyce-Codd Normal Form

Nearest Shops

Person	*Shop Type*	*Nearest Shop*
Davidson	Optician	Eagle Eye
Davidson	Hairdresser	Snippets
Wright	Bookshop	Merlin Books
Fuller	Bakery	Doughy's
Fuller	Hairdresser	Sweeney Todd's
Fuller	Optician	Eagle Eye

Boyce-Codd Normal Form

Shop Near Person

Person	*Shop*
Davidson	Eagle Eye
Davidson	Snippets
Wright	Merlin Books
Fuller	Doughy's
Fuller	Sweeney Todd's
Fuller	Eagle Eye

Shop

Shop	*Shop Type*
Eagle Eye	Optician
Snippets	Hairdresser
Merlin Books	Bookshop
Doughy's	Bakery
Sweeney Todd's	Hairdresser

4th Normal Form

4th Normal Form

- 3rd Normal form
- A record type should not contain two or more independent multi-valued facts about an entity.
- **"Remove multivalued dependencies"** (An item depends on more than one value)

4th Normal Form

- Each instructor takes only one course
- Each course can have one or more instructors
- Each student can have several instructors per course
- Each student can take one or more courses

Student	Course	Instructor
Conrad Pienaar	Biology	Nkosizana Asmal
Dingaan Fortune	Mathematics	Kader Dlamini
Gerrie Jantjies	Science	Helen Ginwala
Mark Thobela	Biology	Nkosizana Asmal
Conrad Pienaar	Science	Peter Leon
Alicia Ncita	Science	Peter Leon
Quinton Andrews	Mathematics	Kader Dlamini
Dingaan Fortune	Mathematics	Helen Ginwala

4th Normal Form

- The only possible key is a combination of all three attributes, as shown below.
- No other combination will uniquely identify a particular record.
- **Multivalued dependency**: Each value of the first attribute, there are one or more associated values of the second attribute

4th Normal Form

- For each value of student, there were many values of course
- Each value of student, there are one or more associated values of instructor.

4th Normal Form

EMPLOYEE	SKILL	LANGUAGE
Smith	cook	
Smith	type	
Smith		French
Smith		German
Smith		Greek

EMPLOYEE	SKILL	LANGUAGE
Smith	cook	French
Smith	type	German
Smith	type	Greek

EMPLOYEE	SKILL	LANGUAGE
Smith	cook	French
Smith	type	German
Smith		Greek

EMPLOYEE	SKILL	LANGUAGE
Smith	cook	French
Smith	cook	German
Smith	cook	Greek
Smith	type	French
Smith	type	German
Smith	type	Greek

5th Normal Form

5th Normal Form

- 4th Normal form
- Its information content cannot be reconstructed from several smaller record types
- **"Remove join dependencies"**

5th Normal Form

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car
Jones	Ford	truck
Brown	Ford	car
Brown	GM	car
Brown	Totota	car
Brown	Totota	bus

AGENT	COMPANY
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

COMPANY	PRODUCT
Ford	car
Ford	truck
GM	car
GM	truck
Toyota	car
Toyota	bus

AGENT	PRODUCT
Smith	car
Smith	truck
Jones	car
Jones	truck
Brown	car
Brown	bus

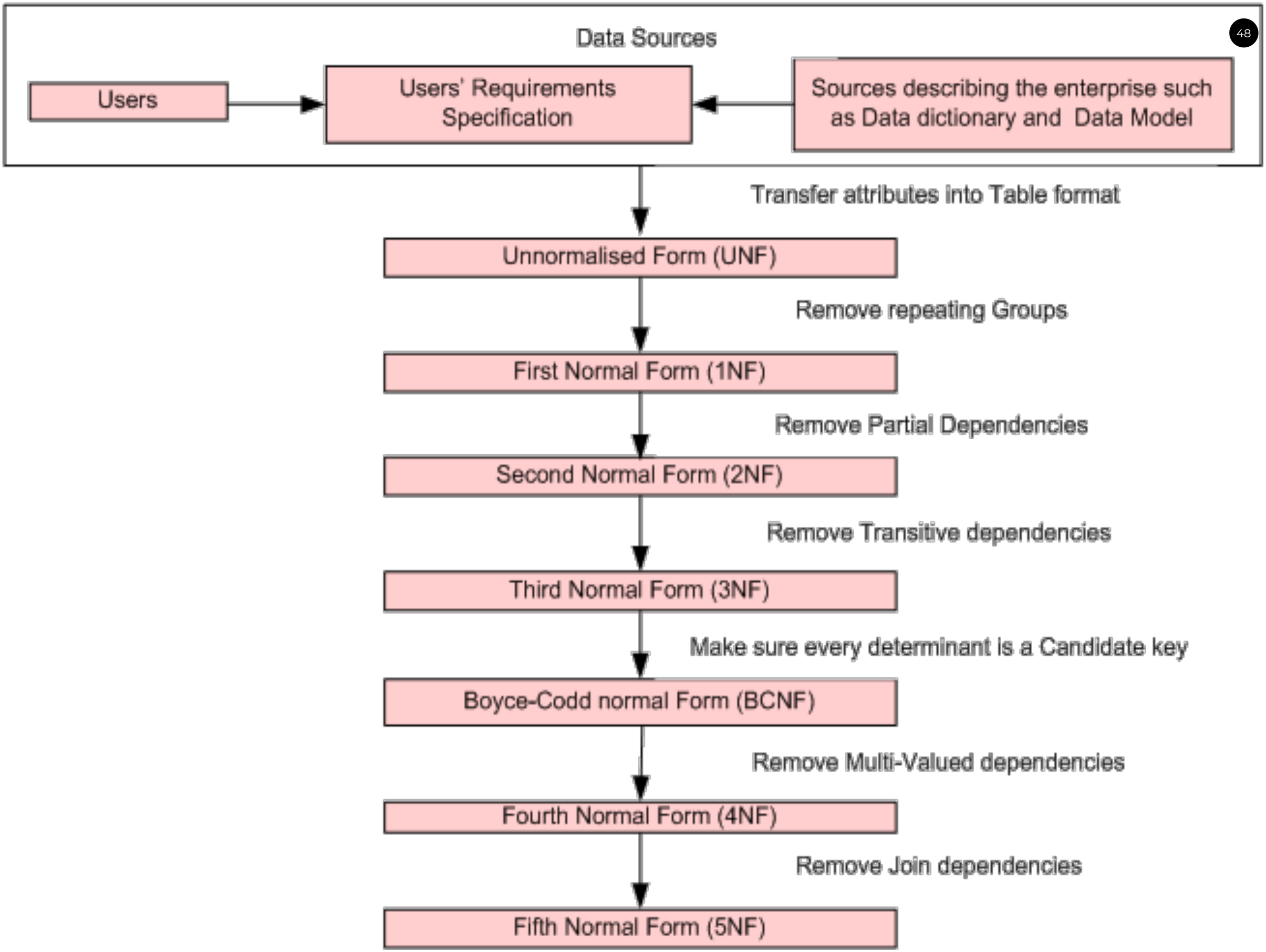
5th Normal Form

- If an agent sells a certain product, and he represents a company making that product, then he sells that product for that company

- If an **agent sells product**, and agent **represents company**, and the **company makes that product**, then **agent sells that product for that company.**

5th Normal Form

- Agent represents a Company and sells some Product that the company makes.
- **"Which agent sells which product from which company?"**


Data Sources
Users
Users' Requirements Specification
Sources describing the enterprise such as Data dictionary and Data Model
Transfer attributes into Table format
Unnormalised Form (UNF)
Remove repeating Groups
First Normal Form (1NF)
Remove Partial Dependencies
Second Normal Form (2NF)
Remove Transitive dependencies
Third Normal Form (3NF)
Make sure every determinant is a Candidate key
Boyce-Codd normal Form (BCNF)
Remove Multi-Valued dependencies
Fourth Normal Form (4NF)
Remove Join dependencies
Fifth Normal Form (5NF)

Sales Records

Sales Records:

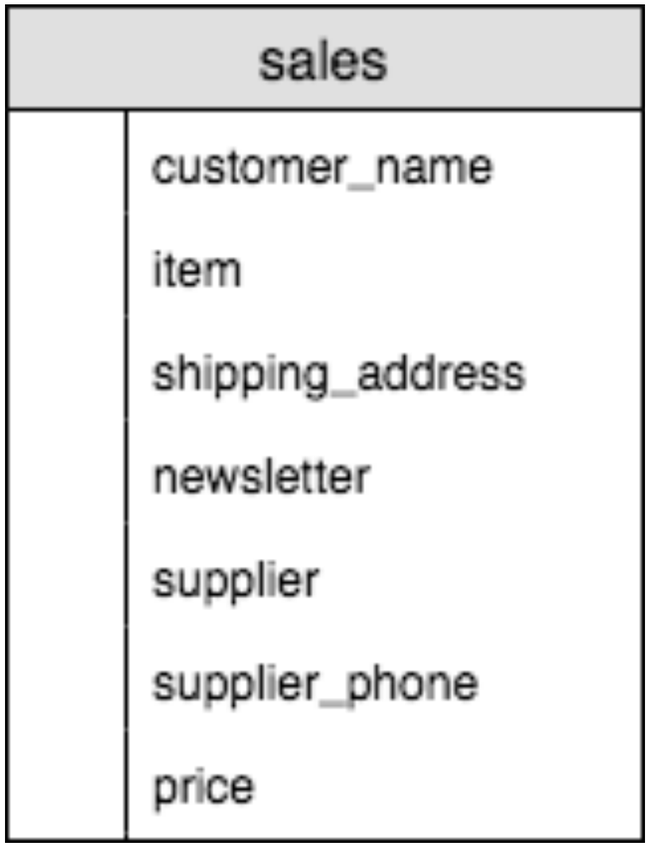
Cust Name	Item	Shipping Address	Newsletter	Supplier	Supplier Phone	Price
Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
Roger Banks	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300
Evan Wilson	Xbox One, PS Vita	28 Rock Av, Denver	Xbox News, PlayStation News	Wholesale	Toll Free	450
Alan Smith	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300

Unnormalized

Sales Records:

Cust Name	Item	Shipping Address	Newsletter	Supplier	Supplier Phone	Price
Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
Roger Banks	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300
Evan Wilson	Xbox One, PS Vita	28 Rock Av, Denver	Xbox News, PlayStation News	Wholesale	Toll Free	450
Alan Smith	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300

customer_name → item, shipping_address, newsletter, supplier, supplier_phone, price



1st Normal Form. Atomic values

- Let's find columns which don't have atomic values:
  - **Items**
  - **Newsletter**
- Don't add ***new*** columns at the moment

1st Normal Form. Atomic values

customer_name	item	shipping_address	newsletter	supplier	supplier_phone	price
Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
Roger Banks	Play Station 4	47 Campus Rd, Boston	Play Station News	Sony	(800) BUY-SONY	300
Evan Wilson	Xbox One	28 Rock Av, Denver	Xbox News	Wholesale	Toll Free	250
Evan Wilson	Ps Vita	28 Rock Av, Denver	Play Station News	Wholesale	Toll Free	200
Alan Smith	Play Station 4	47 Campus Rd, Boston	Play Station News	Sony	(800) BUY-SONY	300

sales
customer_name
item
shipping_address
newsletter
supplier
supplier_phone
price

Functional dependencies

customer_name, item, newsletter → shipping_address, supplier, supplier_phone, price

- It’s hard to identify which columns determine the entire row
- It could be customer and item but likely the customer may buy similar items
- That’s the reason I picked customer, item and newsletter as candidate keys

Partial dependencies

customer_name, item, newsletter → shipping_address, supplier, supplier_phone, price

- Which columns depends partially on the candidate key?
- Shipping address just depends on the customer
- Supplier, phone, price depends on the item

Functional dependencies

- Keep in mind that a customer could have multiple addresses

customer_name	item	shipping_address	newsletter	supplier	supplier_phone	price
Alan Smith	[illegible]	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
Roger Banks	Play Station 4	47 Campus Rd, Boston	Play Station News	Sony	(800) BUY-SONY	300
Evan Wilson	Xbox One	28 Rock Av, Denver	Xbox News	Wholesale	Toll Free	250
Evan Wilson	Ps Vita	28 Rock Av, Denver	Play Station News	Wholesale	Toll Free	200
Alan Smith	[illegible]	47 Campus Rd, Boston	Play Station News	Sony	(800) BUY-SONY	300

Functional dependencies

customer_name → newsletter

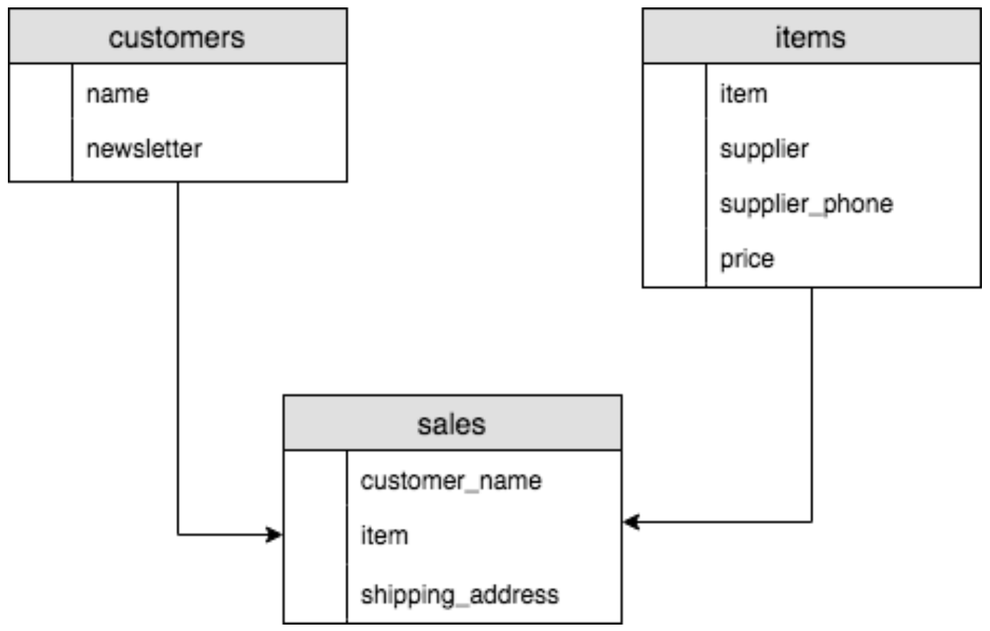
item → supplier, supplier_phone, price

customer_name, item → shipping_address

customer_name	newsletter	
Alan Smith	Xbox News	
Roger Banks	Play Station News	
Evan Wilson	Xbox News	
Evan Wilson	Play Station News	
Alan Smith	Play Station News	

item	supplier	supplier_phone	price
Xbox One	Microsoft	(800) BUY-XBOX	250
Play Station 4	Sony	(800) BUY-SONY	300
Ps Vita	Wholesale	Toll Free	200

customer_name	item	shipping_address
Alan Smith	Xbox One	35 Palm St, Miami
Roger Banks	Play Station 4	47 Campus Rd, Boston
Evan Wilson	Xbox One	28 Rock Av, Denver
Evan Wilson	Ps Vita	28 Rock Av, Denver
Alan Smith	Play Station 4	47 Campus Rd, Boston



Transitive dependencies

- There is just one table with a composite key and there are no more partial dependencies.
- Our model is in 2nd Normal form
- Let’s take a look if there are any transitive dependencies

Transitive dependencies

- Supplier phone doesn’t depend on the item, depends on the supplier.
- Newsletter doesn’t depend on customer directly and also a customer can subscribe to multiple newsletters

Transitive dependencies

- So… A customer could have multiple newsletters and a newsletter is used by multiple customers.

Functional dependencies

customer_name → {}

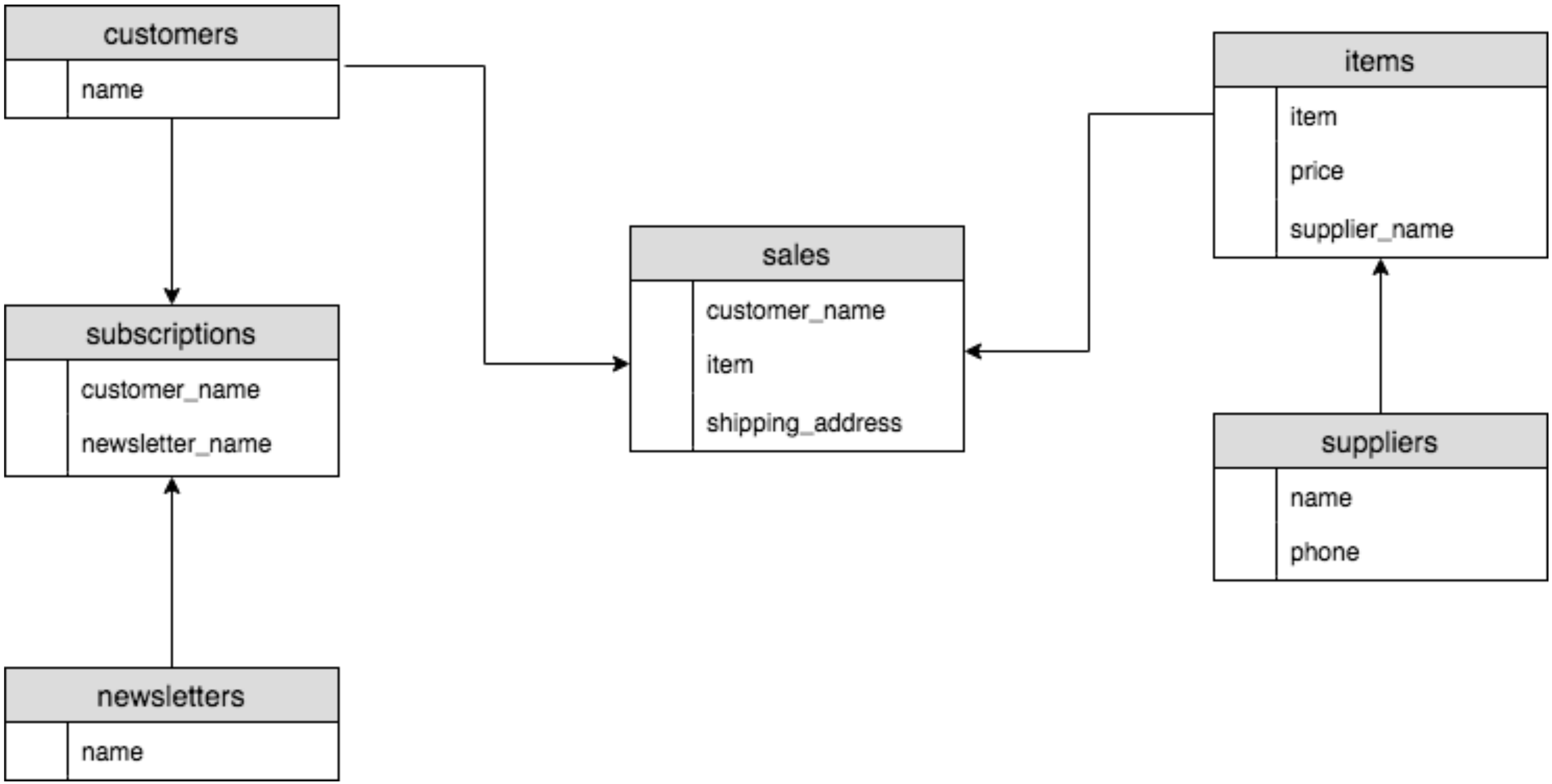
newsletter → {}

customer_name, newsletter → {}

item → price, supplier

item → supplier, supplier_phone, price

customer_name, item → shipping_address

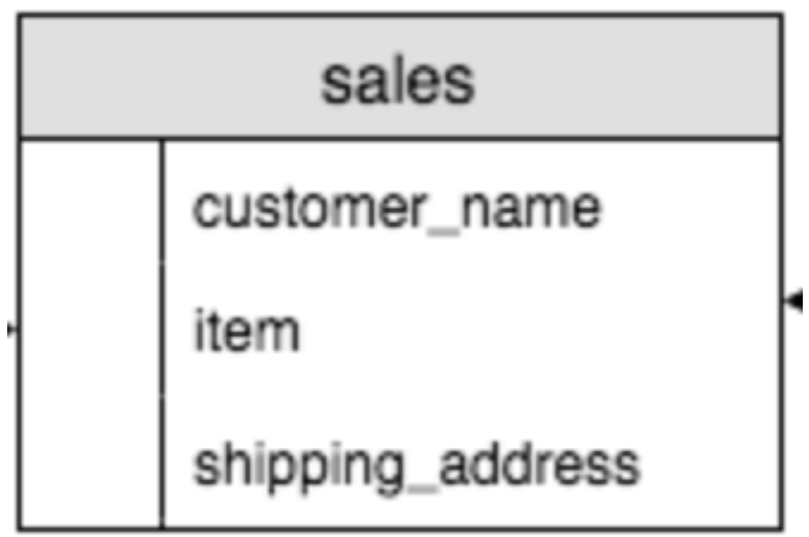

customers
name
subscriptions
customer_name
newsletter_name
newsletters
name
sales
customer_name
item
shipping_address
items
item
price
supplier_name
suppliers
name
phone

customer_name				
Alan Smith				
Roger Banks				
Evan Wilson				
item	**price**	**supplier_name**		
Xbox One	250	Microsoft		
Play Station 4	300	Sony		
Ps Vita	200	Wholesale		
customer_name	**item**	**shipping_address**		
Alan Smith	Xbox One	35 Palm St, Miami		
Roger Banks	Play Station 4	47 Campus Rd, Boston		
Evan Wilson	Xbox One	28 Rock Av, Denver		
Evan Wilson	Ps Vita	28 Rock Av, Denver		
Alan Smith	Play Station 4	47 Campus Rd, Boston		

supplier	**supplier_phone**
Microsoft	(800) BUY-XBOX
Sony	(800) BUY-SONY
Wholesale	Toll Free
newsletter_name	
Xbox News	
Play Station News	
customer_name	**newsletter_name**
Alan Smith	Xbox News
Roger Banks	Play Station News
Evan Wilson	Xbox News
Alan Smith	Play Station News
Evan Wilson	Play Station News

Let’s analyze

- Let’s take a look at the sales table
- Do you think do we have any partial dependency?
- Do you think there is Boyce Codd normal form?



Let’s analyze

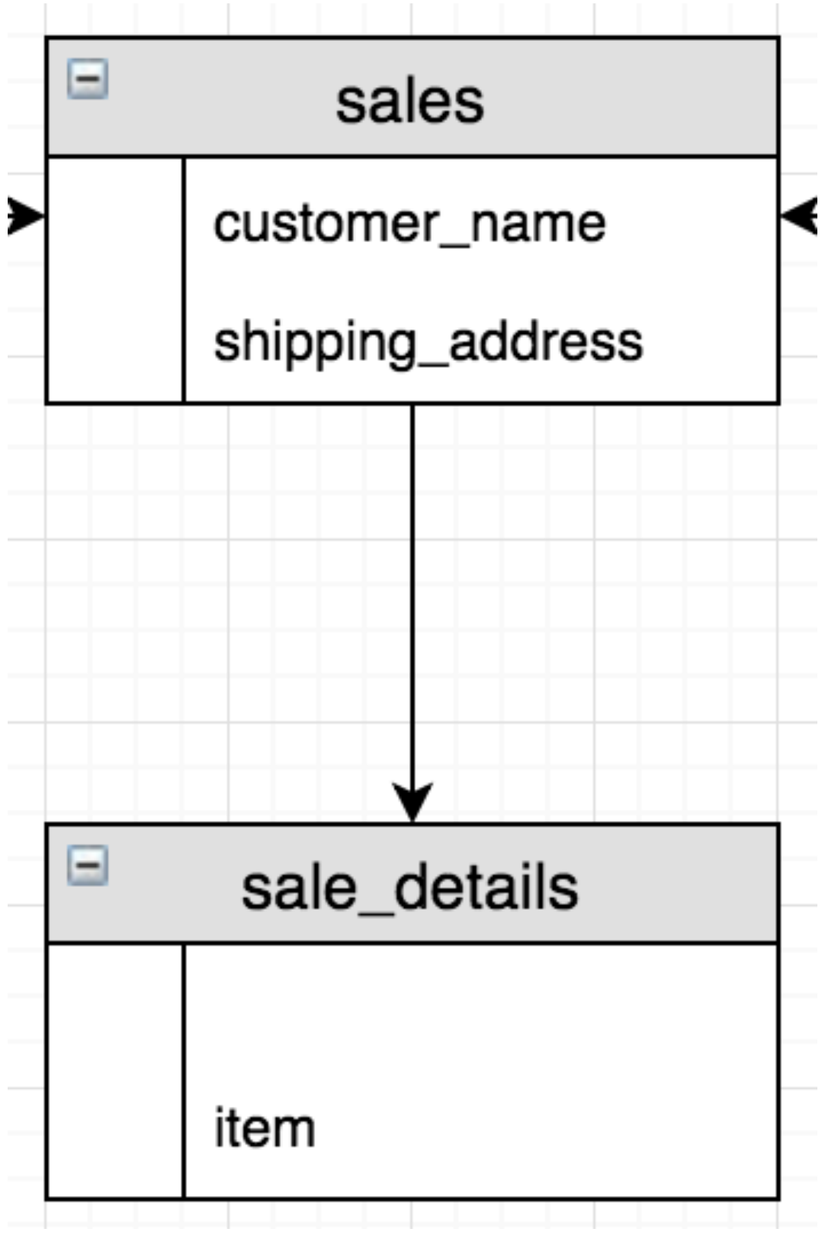
- Some people agree and disagree
- Truth is if the customer buys the same product in a future transaction, the candidate key will be violated
- Also, is hard to track when an item was purchased

Let's analyze

customer_name	item	shipping_address
Alan Smith	Xbox One	35 Palm St, Miami
Roger Banks	Play Station 4	47 Campus Rd, Boston
Evan Wilson	Xbox One	28 Rock Av, Denver
Evan Wilson	Ps Vita	28 Rock Av, Denver
Alan Smith	Play Station 4	47 Campus Rd, Boston
Roger Banks	Play Station 4	47 Campus Rd, Boston

Let's analyze

- A new table is necessary, is similar to a bill. A bill has many items
- But, there is a problem: What would be your foreign key?



Let’s analyze

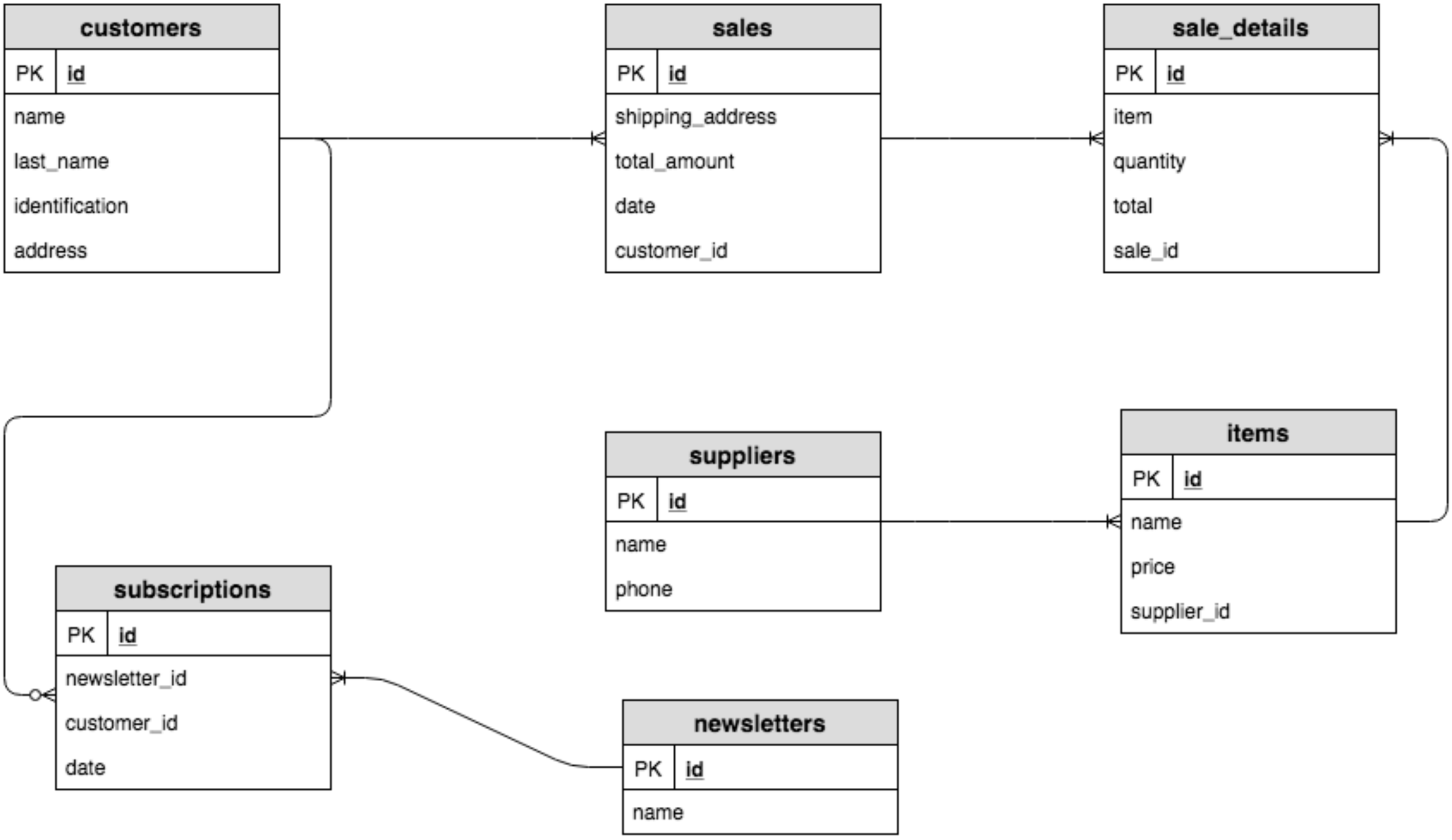
- Also if the customer comes again and wants to deliver to the same address, your primary key won’t be a primary key since the record will be duplicated.
- You need a unique identifier for each transaction.
- A bill has an id in real life.

Let’s analyze

- This is one of the reasons to use integer primary keys which would be auto-incremental.
- Also, the price of the item could change over the time, but the price emitted in one bill can’t change by law

Let’s analyze

- Also, you may want to have some new columns like quantity for each item.
- Ideally you may want to have a total_paid column which represents the total of the bill and avoid calculating every time the total of the bill.
- A date of the purchase is also necessary.


customers
PK id
name
last_name
identification
address
sales
PK id
shipping_address
total_amount
date
customer_id
sale_details
PK id
item
quantity
total
sale_id
suppliers
PK id
name
phone
items
PK id
name
price
supplier_id
subscriptions
PK id
newsletter_id
customer_id
date
newsletters
PK id
name

Final thoughts

- You can safely say that your data is clean and you have integrity in your data.
- I’m not a big fan of tables which only has 1 column (2 with the id column) like **newsletter**

Final thoughts

- If you're looking for performance you should remove the subscription table and add the customer id to the newsletter table
- Also, you need to validate if an item could be supplied by different suppliers, in that case is a many-to-many relationship and need to be careful. Let's assume that we have single providers for each item

Compare!

- You may want to compare to this normalization https://youtu.be/UrYLYV7WSHM vs the one I did, doing that you will know the differences and realize that there are multiple of ways of thinking and come with different solutions.
- In my case there are some mistakes which can't happen in real life.

Thank you!