DATABASES

INTRODUCTION TO DATA SCIENCE

TIM KRASKA

teaching
datascience
.org

OTHER ANNOUNCEMENTS

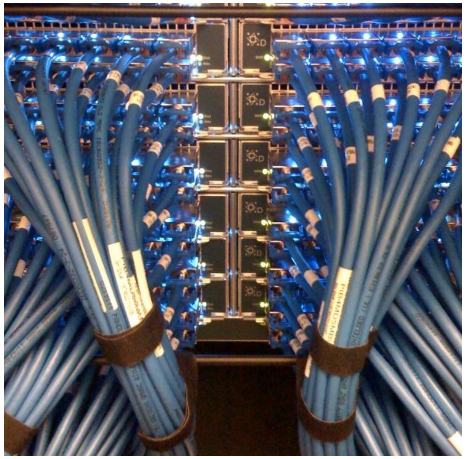
Want to get involved in research?

We are offering several independent studies and summer research internship.

Sign-up available on: http://database.cs.brown.edu/

or directly: http://tinyurl.com/zxznf92

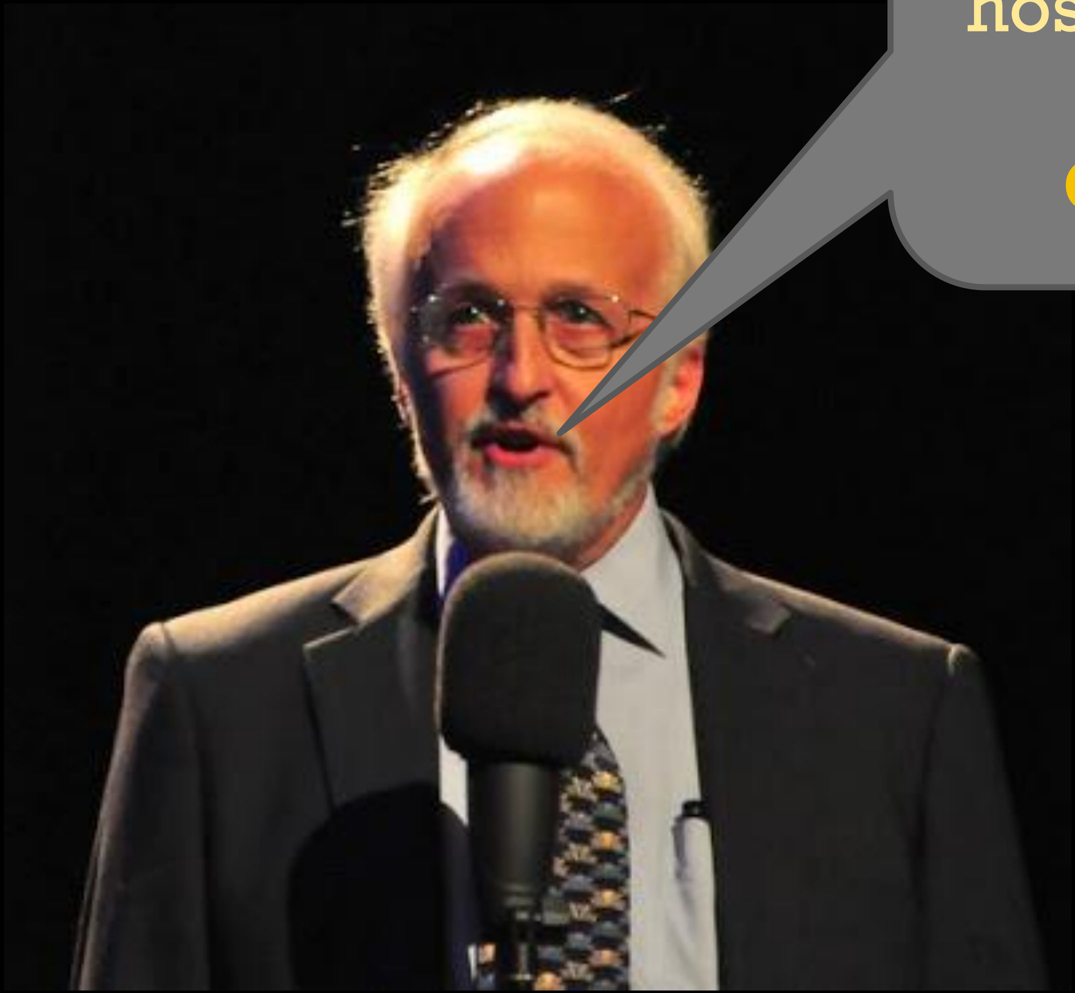
Possible Topics:



Infiniband

Tupleware

Interactive Data Exploration



Do you want to drink from the fire hose??? Then take my

CS227 class

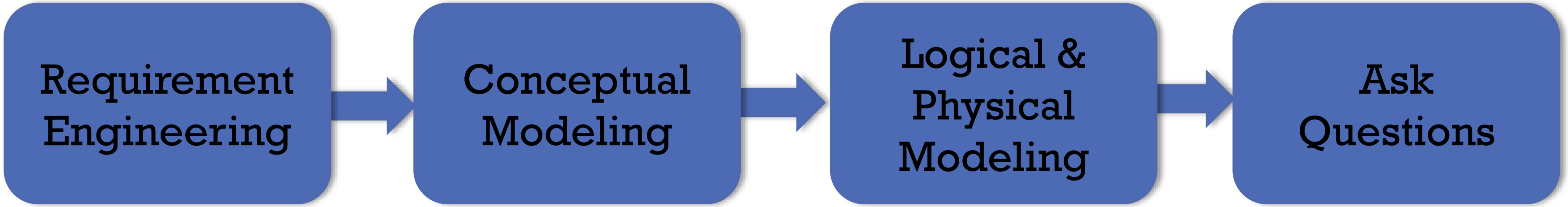
BOOK

So far:

Database System Concepts
Sixth Edition by Silberschatz.

Pieces of chapters 1, 2, 3, 6, 7 and (18)

DATABASES FOR DATA SCIENTIST



Book of duty

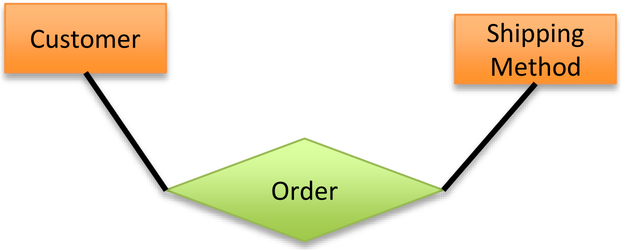
Conceptual Design (ER)

- Logical design (schema)
- Physical design (index, hints)

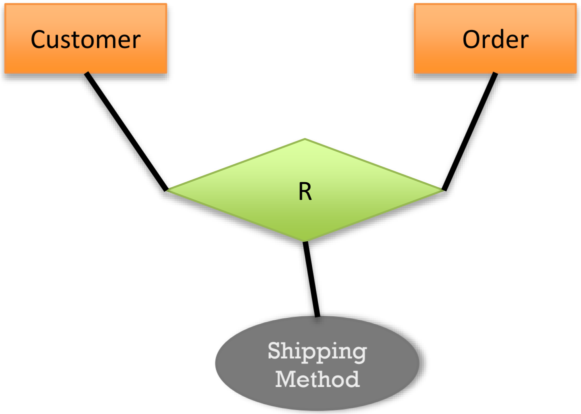
CLICKER QUESTION I

- A customer can have several orders
- An order belongs to a single customer
- Every order has exactly one shipping method (e.g., Post, Fexed, UPS,…)

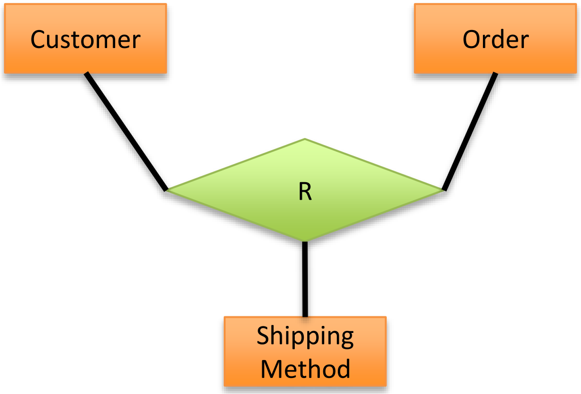
(A)



(B)

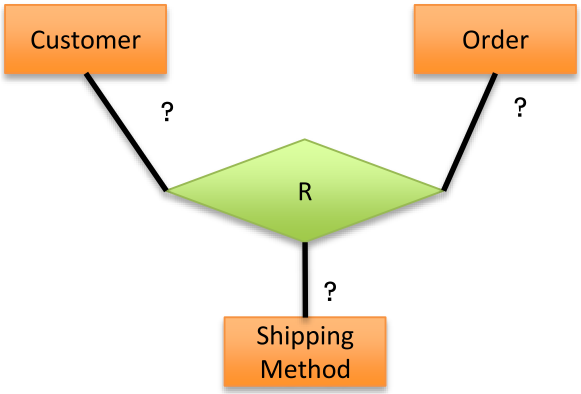


(C)

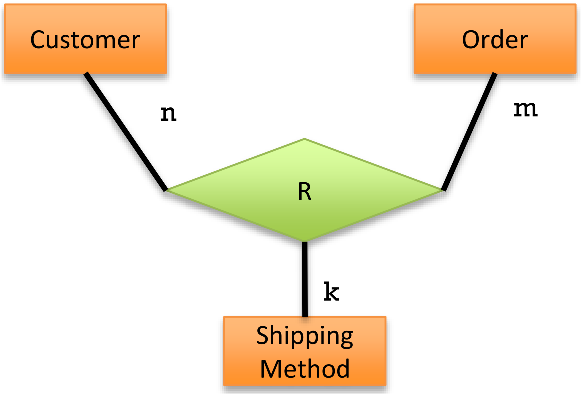


CLICKER QUESTION II

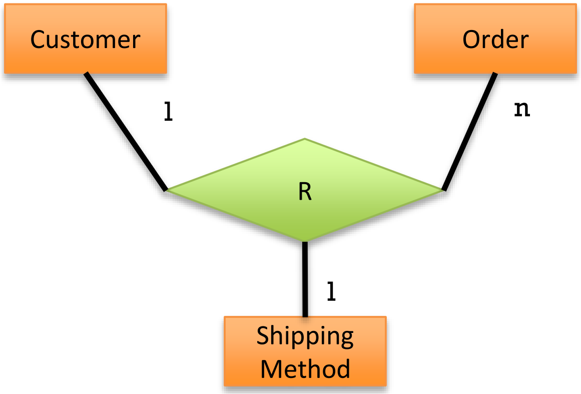
- A customer can have several orders
- An order belongs to a single customer
- Every order has exactly one shipping method (e.g., Post, Fexed, UPS,…)



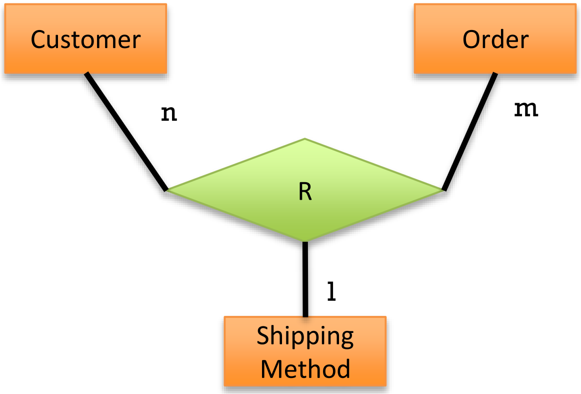
(1)



(2)



(3)

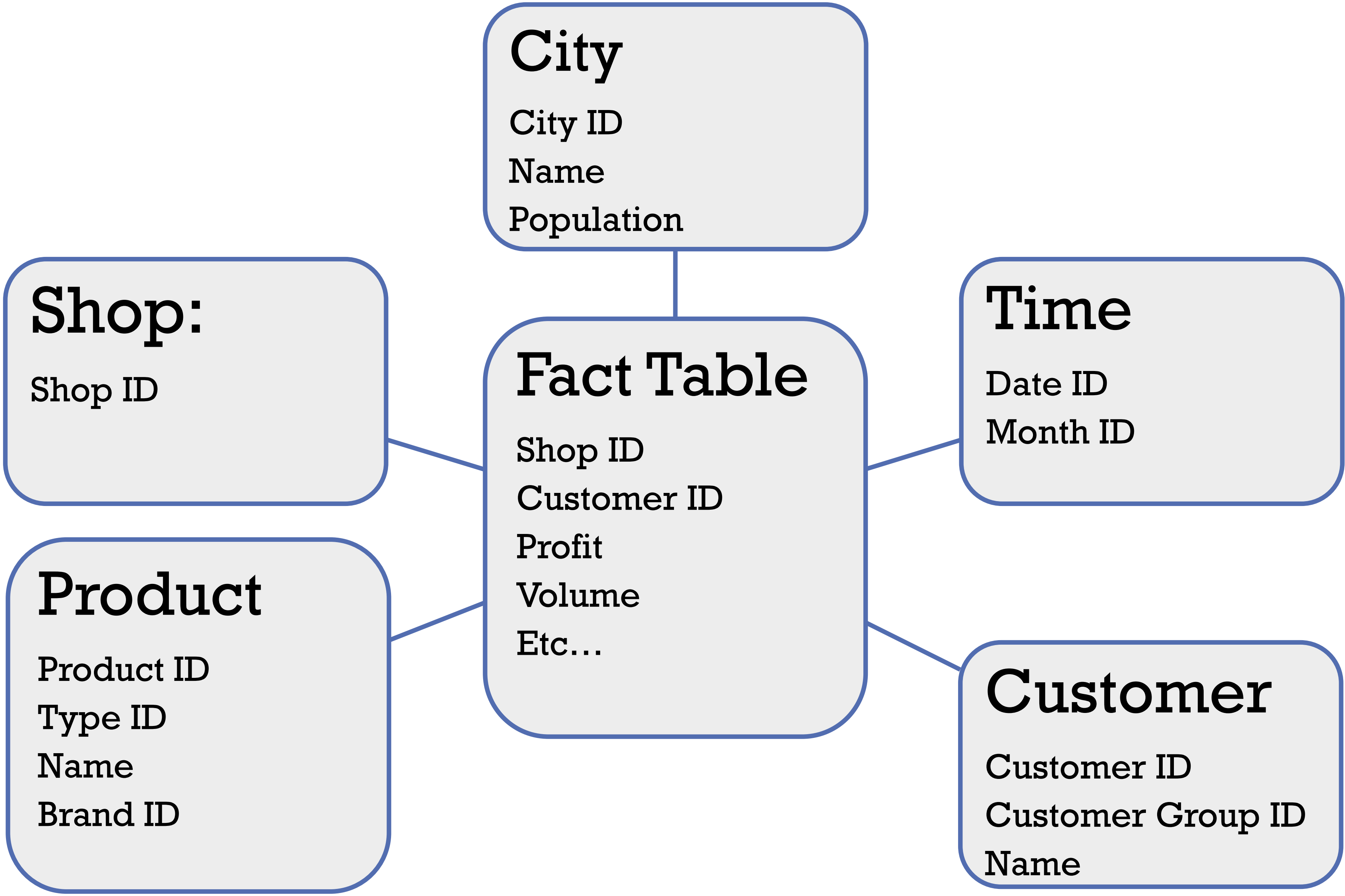


PROBLEM

- **You are the new Data Scientist at Evil Market**
- Evil Market is tracking all customer purchases with their membership or credit card
- They also have data about their customers (estimated income, family status,…) from surveys they have done in the past
- Recently, they are trying to improve their image for young mothers
- As a start they want to know the following information for mothers under 30 for 2015:
  - How much do they spend at Evil Market?
  - How does this compare to all customers under 30?
  - What are their favorite products?
  - Did they spend more in 2015 than in 2014?

Your first project: Design the schema for Evil Market to analyze Evil Market's purchase logs!

STAR SCHEMA



City

City ID

Name

Population

Shop:

Shop ID

Fact Table

Shop ID

Customer ID

Profit

Volume

Etc…

Time

Date ID

Month ID

Product

Product ID

Type ID

Name

Brand ID

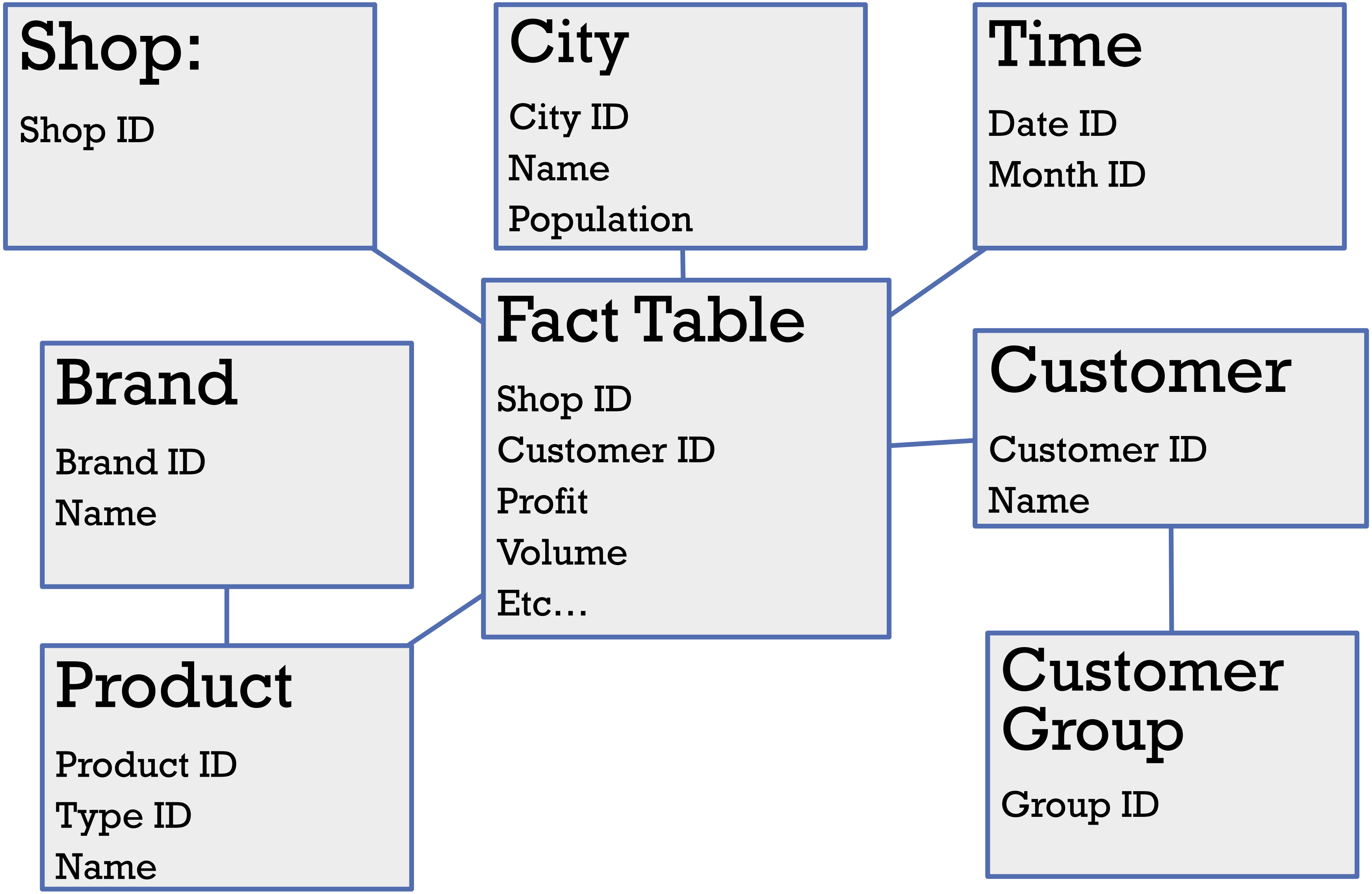
Customer

Customer ID

Customer Group ID

Name

SNOWFLAKE SCHEMA



Shop:

Shop ID

City

City ID
Name
Population

Time

Date ID
Month ID

Fact Table

Shop ID
Customer ID
Profit
Volume
Etc…

Brand

Brand ID
Name

Customer

Customer ID
Name

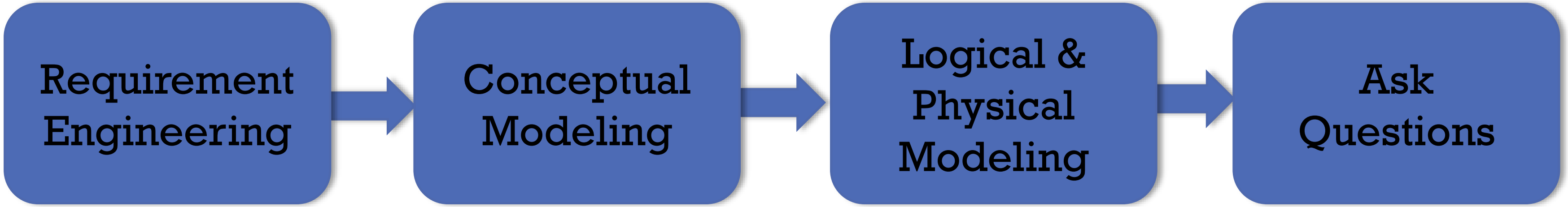
Product

Product ID
Type ID
Name

Customer Group

Group ID

DATABASES FOR DATA SCIENTIST



Book of duty

Conceptual Design (ER)

- Logical design (schema)
- Physical design (index, hints)

SQL: RELATIONAL ALGEBRA

FORMAL DEFINITION OF REL. ALGEBRA

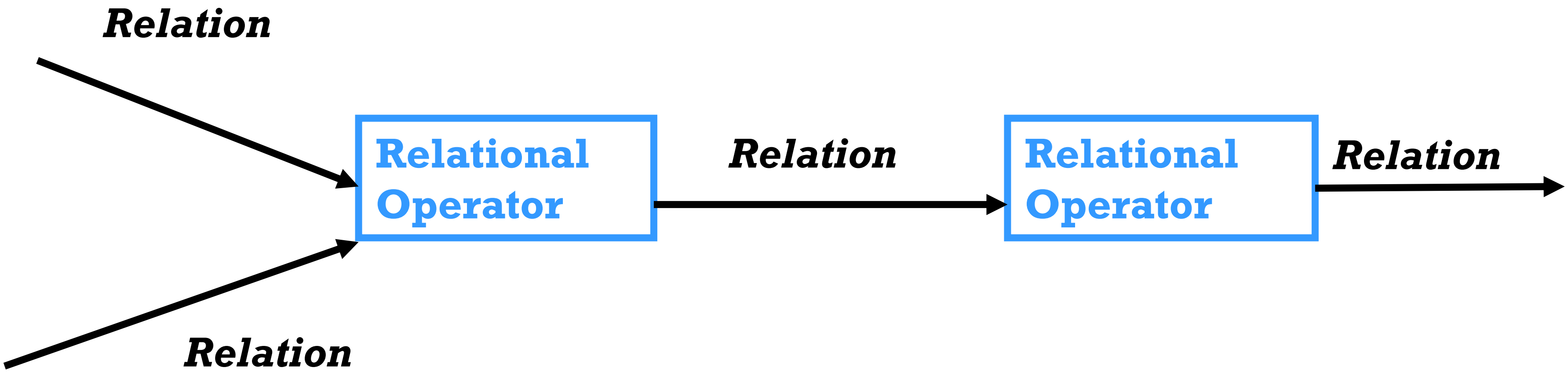
Atoms (basic expressions)

- A **relation** in the database
- A **constant relation**

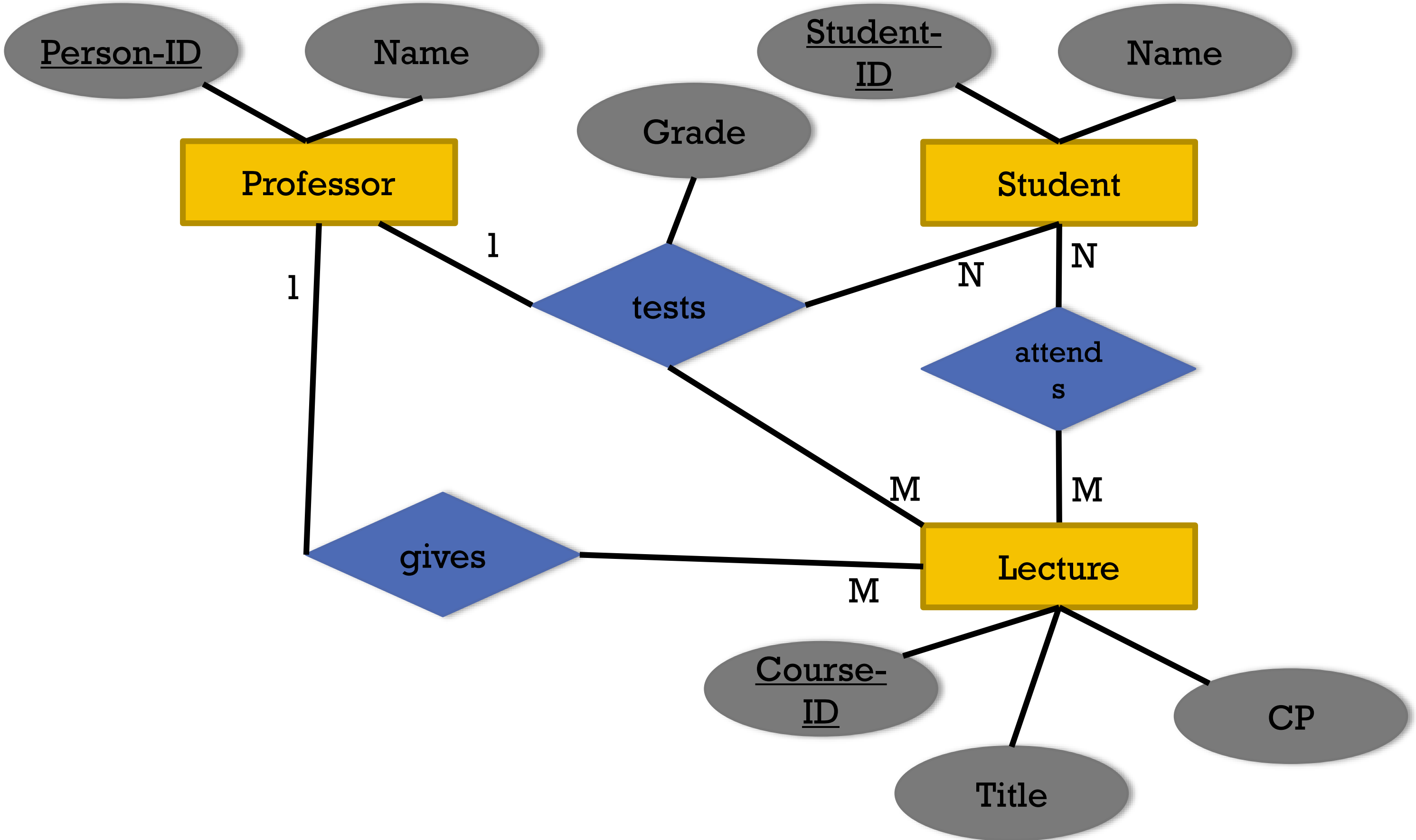
Operators (composite expressions)

- **Selection**: σ (E1)
- **Projection**: Π (E1)
- **Cartesian Product**: E1 x E2
- **Rename**: ρ_V(E1), $\rho_{A \leftarrow B}$ (E1)
- **Union**: E1 $\cup$ E2
- **Minus**: E1 - E2

CLOSURE PROPERTY / COMPOSABILITY



Professor(Person-ID:integer, Name:varchar(30), Level:varchar(2))
Student(Student-ID:integer, Name:varchar(30), Semester:integer)
Lecture(Course-ID:varchar(10), Title:varchar(50), CP:float)
Gives(Person-ID:integer, Course-ID:varchar(10))
Attends(Student-ID:integer, Course-ID:varchar(10))
Tests(Student-ID:integer, Course-ID:varchar(10), Person-ID:integer, Grade:char(2))



SELECTION AND PROJECTION

Professor(Person-ID:integer, Name:varchar(30), Level:varchar(2))
Student(Student-ID:integer, Name:varchar(30), Semester:integer)

Selection

$\sigma_{\text{Semester} > 10}$ (Student)		
Student-ID	Name	Semester
24002	Xenokrates	18
25403	Jonas	12

Projection

Π_{Level} (Professor)
Level
FP
AP

CARTESIAN PRODUCT

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

x

R	
D	E
d_1	e_1
d_2	e_2

=

Result				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_1	c_1	d_2	e_2
a_2	b_2	c_2	d_1	e_1
a_2	b_2	c_2	d_2	e_2

CARTESIAN PRODUCT (CTD.)

Professor x Attends

Professor				**Attends**	
Person-ID	Name	Level	Room	Student-ID	Course-ID
2125	Ugur	FP	226	26120	5001
...	...	...	...	...	...
2125	Ugur	FP	226	29555	5001
...	...	...	...	...	...
2137	Jeff	AP	7	29555	5001

- Huge result set (n * m)
- Usually only useful in combination with a selection (-> Join)

NATURAL JOIN

Two relations:

- $R(A_1,\ldots, A_m, B_1,\ldots, B_k)$
- $S(B_1,\ldots, B_k, C_1,\ldots, C_n)$

$$R \bowtie S = \Pi_{A1,\ldots, Am, R.B1,\ldots, R.Bk, C1,\ldots, Cn}(\sigma_{R.B1=S.B1 \wedge\ldots\wedge R.Bk = S.Bk}(RxS))$$

$R \bowtie S$											
$R - S$				$R \cap S$				$S - R$			
A_1	A_2	…	A_m	B_1	B_2	…	B_k	C_1	C_2	…	C_n
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

⋮

THREE-WAY NATURAL JOIN

(Student $\bowtie$ attends) $\bowtie$ Lecture

(Student $\bowtie$ attends) $\bowtie$ Lecture						
Student-ID	Name	Semester	Course-NR	Title	CP	Person-ID
26120	Fichte	10	CS1951a	Intro to Data Science	2	9999
27550	Jonas	12	CS18	Programming	2	2134
28106	Carnap	3	CS19	More Programming	3	2126
…	…	…	…	…	…	…

THETA-JOIN

Two Relations:

- R(A1, …, An)
- S(B1, …, Bm)

$$R \bowtie_{\theta} S = \sigma_{\theta} (R \text{ x } S)$$

$R \bowtie_{\theta} S$							
R				S			
A_1	A_2	…	A_n	B_1	B_2	…	B_m
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

JOIN VARIANTS

- natural join

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Result				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1

- left outer join

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟕

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Result				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	-	-

JOIN VARIANTS

- right outer join

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

⟕

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Result				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
-	-	c_3	d_2	e_2

JOIN VARIANTS

- (full) outer join

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\bowtie$ (full outer)

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Result				
A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	-	-
-	-	c_3	d_2	e_2

- left semi join

L		
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\ltimes$

R		
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

Result		
A	B	C
a_1	b_1	c_1

JOIN VARIANTS

- right semi join

	L	
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

$\rtimes$

	R	
C	D	E
c_1	d_1	e_1
c_3	d_2	e_2

=

	Resultat	
C	D	E
c_1	d_1	e_1

RENAME OPERATOR ρ

Renaming of relation names

- Needed to process self-joins and recursive relationships
- E.g., two-level dependencies of lectures („grandparents “)

Π☹🗁🖏⚐❐♏❐♏❑◆♓⬧♓◆♏☎

σ☹🗎🖏♍❑◆❐⬧♏🖅♓♎🖬👍💧🗎🗎🖮🗀 ∧

☹🗎🖏◘❐♏❐♏❑◆♓⬧♓◆♏🖬☹🗁🖏♍❑◆❐⬧♏🖅♓♎ ☎

ρ☹🗁☎☼♏❑◆♓❐♏⬧✆ ⌧ ρ☹🗎

☎☼♏❑◆♓❐♏⬧✆

✆

✆

☼♏❑◆♓❐♏⬧

course-id	prerequisite
CS1951A	CS160
CS1951A	CS320
CS2270	CS1270
CS1270	CS160

Renaming of attribute names

ρ

SET DIFFERENCE (–)

Notation: *Relation$_1$* - *Relation$_2$*

R - S valid only if:

1. *R, S have same number of columns (arity)*
2. *R, S corresponding columns have same domain (compatibility)*

Example:

$$(\Pi_{bname}(\sigma_{amount \geq 1000}(loan))) - (\Pi_{bname}(\sigma_{balance < 800}(account)))$$

loan

bname	lno	amount
Downtown	L-17	1000
Redwood	L-23	2000
Perry	L-15	1500
Downtown	L-14	500
Perry	L-16	300

account

bname	acct_no	balance
Mianus	A-215	700
Brighton	A-201	900
Redwood	A-222	700
Brighton	A-217	850

= Result?

(A)

bname
Mianus
Redwood

(B)

bname
Downtown
Redwood
Perry

(C)

bname
Downtown
Perry

INTERSECTION

$\Pi_{\text{Person-ID}}(\text{Lecture}) \cap \Pi_{\text{Person-ID}}(\sigma_{\text{Level=FP}}(\text{Professor}))$

Only works if both relations have the same schema

- Same attribute names and attribute domains

Intersection can be simulated with minus:

$\mathbf{R \cap S = R - (R - S)}$

Union works similarly…

CODD`S THEOREM

3 Languages:

- **Relational Algebra**
- **Tuple Relational Calculus** (safe expressions only)
- **Domain Relational Calculus** (safe expressions only)

are equivalent.

Impact of Codd`s theorem:

- SQL is based on the **relational calculus**
- SQL implementation is based on **relational algebra**
- **Codd`s theorem shows that SQL implementation is correct and complete.**

NOT COVERED

Set Division

Aggregate Functions

Codd´s Proof

…