A Relation

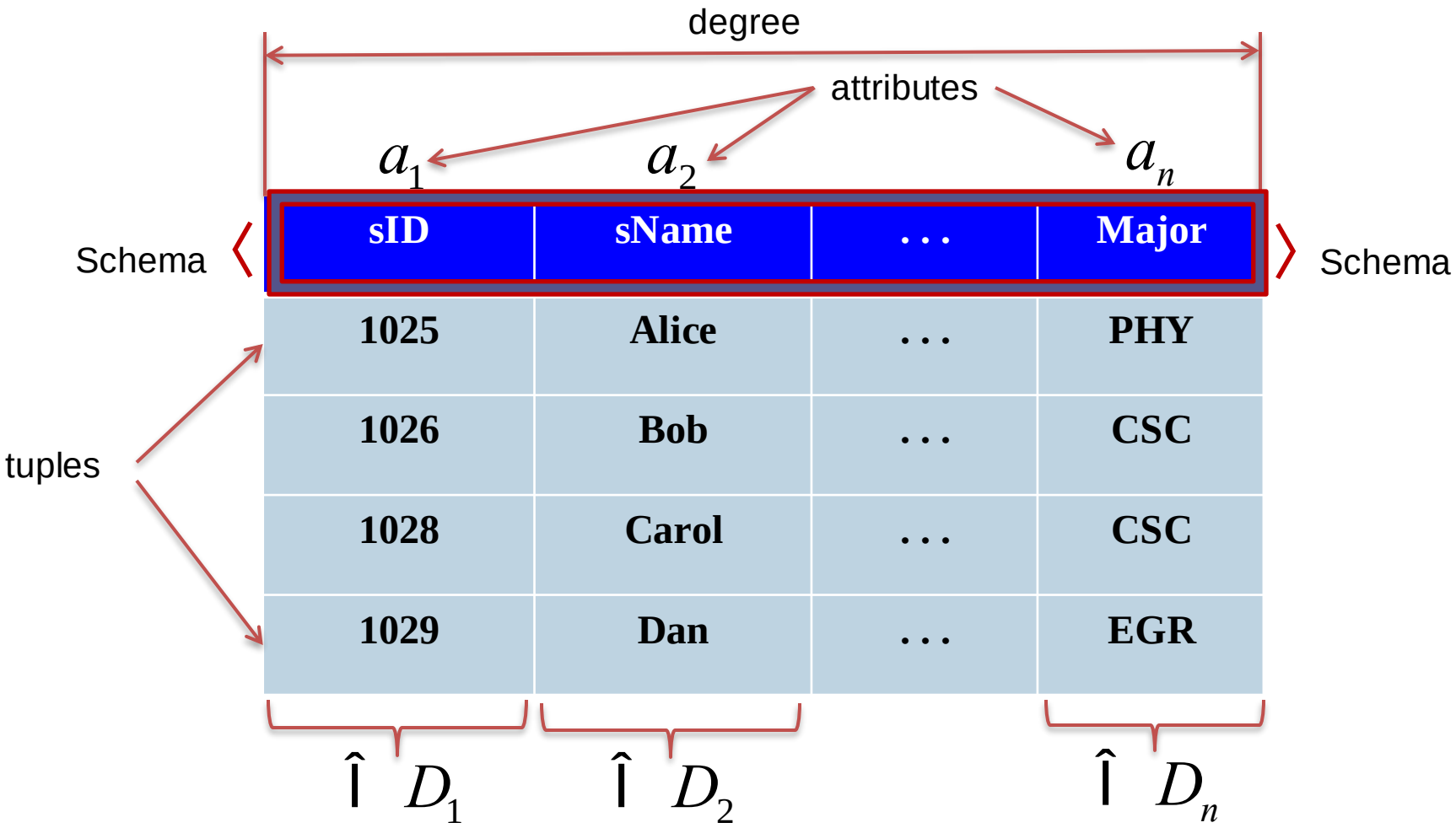


Diagram from B. Signer

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE

Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Functional Dependency

The value of one or more attributes determines the value of one or more other attributes

Let R be a relation schema, and $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$ *are sets of attributes. The functional dependency with* $\boldsymbol{\alpha}$ *as the determinant and* $\boldsymbol{\beta}$ *as the dependent is*

$$\boldsymbol{\alpha \rightarrow \beta}$$

This functional dependency is said to hold on relation $\boldsymbol{R}$ if and only if for any valid relation $\boldsymbol{r}(\boldsymbol{R})$, whenever any two tuples $\boldsymbol{t_1}$ *and* $\boldsymbol{t_2}$ *of* $\boldsymbol{r}$ agree on the attributes $\boldsymbol{\alpha}$, they also agree on $\boldsymbol{\beta}$:

$$\boldsymbol{t_1[\alpha] = t_2[\alpha] \rightarrow t_1[\beta] = t_2[\beta]}$$

$$t_1[\alpha] = t_2[\alpha] \rightarrow t_1[\beta] = t_2[\beta]$$

A	B
4	1
5	1
7	3

A → B does hold, but B → A does NOT hold:

when t_1 = (4,1) ,t_2 = (5,1)

then t_1[B] =t_2 [B] ,

but t_1 [A] ≠ t_2 [A]

Functional Dependence Example 2

Student (SSN, sName, address, Hscode, Hsname, Hscity, GPA, priority)

SSN → sName
SSN → address

(assuming a student doesn't move during enrollment)

HScode → HSname, HScity
HSname, HScity → HScode

(assuming no two HS in same city have the same name)

SSN → GPA ,
GPA → priority , therefore
SSN → priority

(an example of our old friend Transitivity!)

Relational Database Keys

Key type	Definition
Composite key	Key that is composed of more than one attribute
Superkey	Key that can uniquely identify any row in the table
Candidate key	Candidate key: minimal superkey
Primary key	A candidate key selected for uniquely identify all other attribute values in any given row; cannot have NULL values
Foreign key	Primary key of one table that has been placed into another table to create a common attribute
Secondary key	Key used strictly for data retrieval purposes

FIGURE 3.2 AN EXAMPLE OF A SIMPLE RELATIONAL DATABASE

Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Figure 3.3 - An Illustration of Integrity Rules

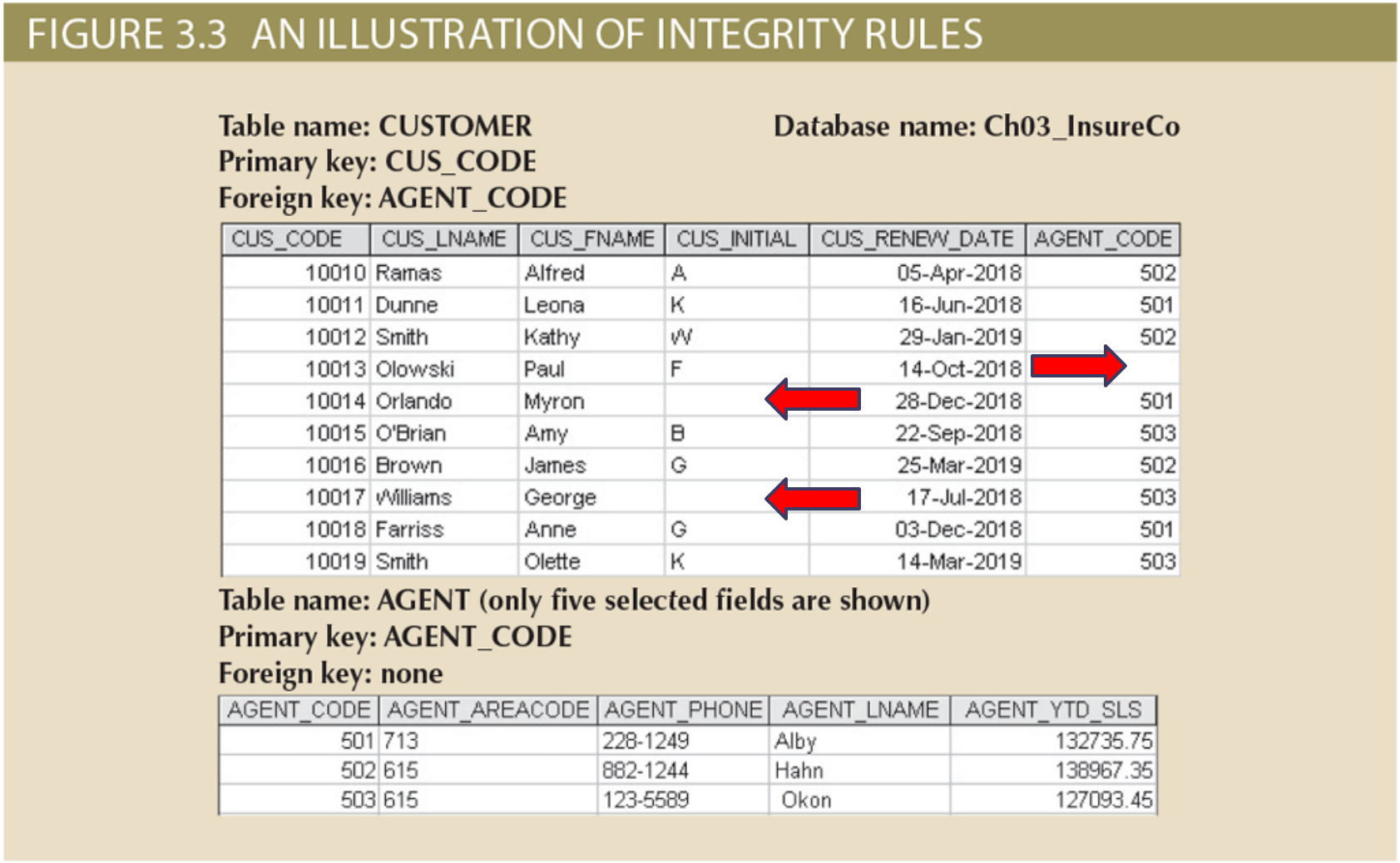


FIGURE 3.3 AN ILLUSTRATION OF INTEGRITY RULES

Table name: CUSTOMER **Database name: Ch03_InsureCo**
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2018	502
10011	Dunne	Leona	K	16-Jun-2018	501
10012	Smith	Kathy	W	29-Jan-2019	502
10013	Olowski	Paul	F	14-Oct-2018	
10014	Orlando	Myron		28-Dec-2018	501
10015	O'Brian	Amy	B	22-Sep-2018	503
10016	Brown	James	G	25-Mar-2019	502
10017	Williams	George		17-Jul-2018	503
10018	Farriss	Anne	G	03-Dec-2018	501
10019	Smith	Olette	K	14-Mar-2019	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

NULL is …

nothing,
هیچ چی, tiada apa-apa,
nada, nichts,שום דבר ,ingenting
沒什麼, нічого, waxba, কিছু না,

faic, 아무것도 아님, नथिंग , 何も ,
hakuna kitu, không có gì,
Hiç bir şey, لا شئ,
dim byd

Codd's Original RA Operators

Set Operators		Relation Operators	
Union	∪	Selection	σ
Difference	−	Projection	Õ
Intersection	∩	Join	⨝
		Cartesian Product	×
		Division	÷

One more RA Operator

Set Operators	Relation Operators
Union	Selection σ
Difference $-$	Projection
Intersection	Join
Rename ρ	Cartesian Product $\times$
	Division $\div$

Soccer Recruitment DB

College (cName, state, enrollment)

Player (pID, pName, yellowCards, Highschool)

Tryout (pID, cName, pPosition, decision)

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Layout idea from J. Widom

Selection: σ

Colleges with more than 12,000 students:

$$\sigma_{enr>12000}\ College$$

Players with no Yellow Cards:

$$\sigma_{yCard=no}\ Player$$

Players trying out for goalie at ASU:

$$\sigma_{pPos=goalie \wedge cName=ASU}\ Tryout$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

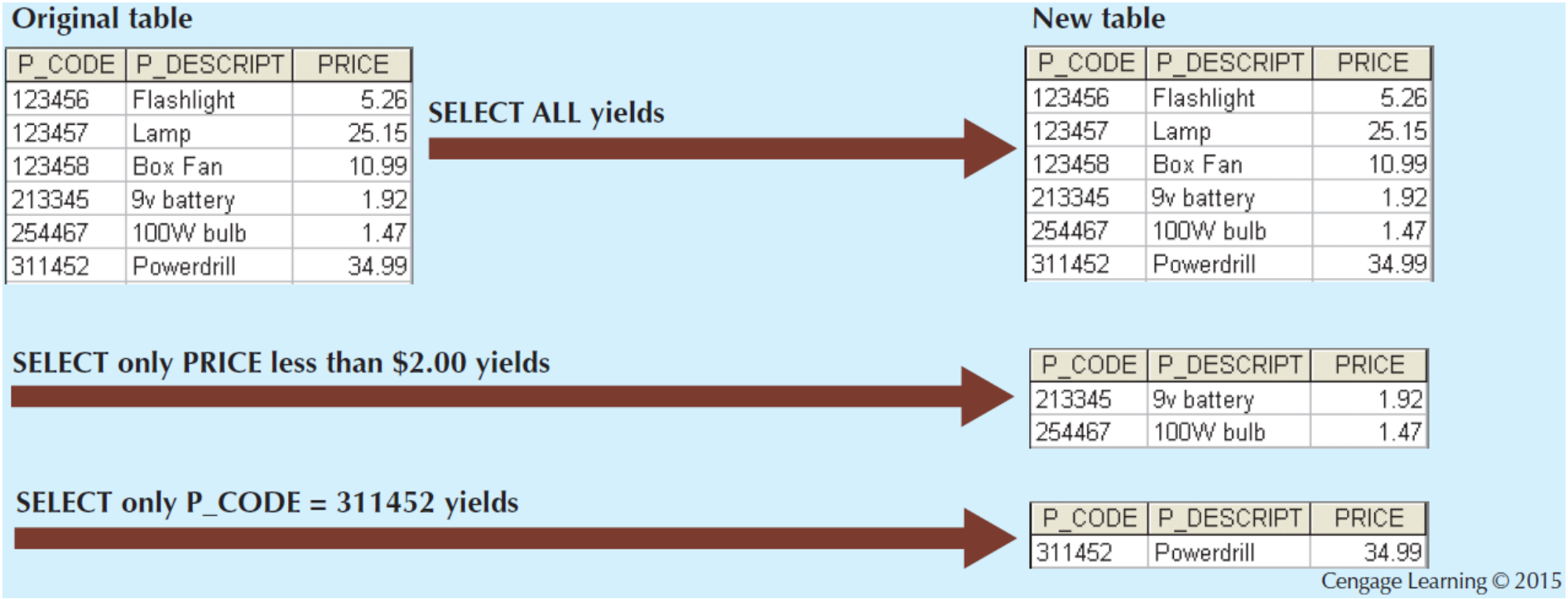
pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Layout idea from J. Widom

Figure 3.4 - Select



Projection: Õ

Players and Yellow cards only

$$\tilde{O}_{pID,yCard}(Player)$$

Eliminate Cname from Tryout

$$\tilde{O}_{pID,pPos,dec}(Tryout)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

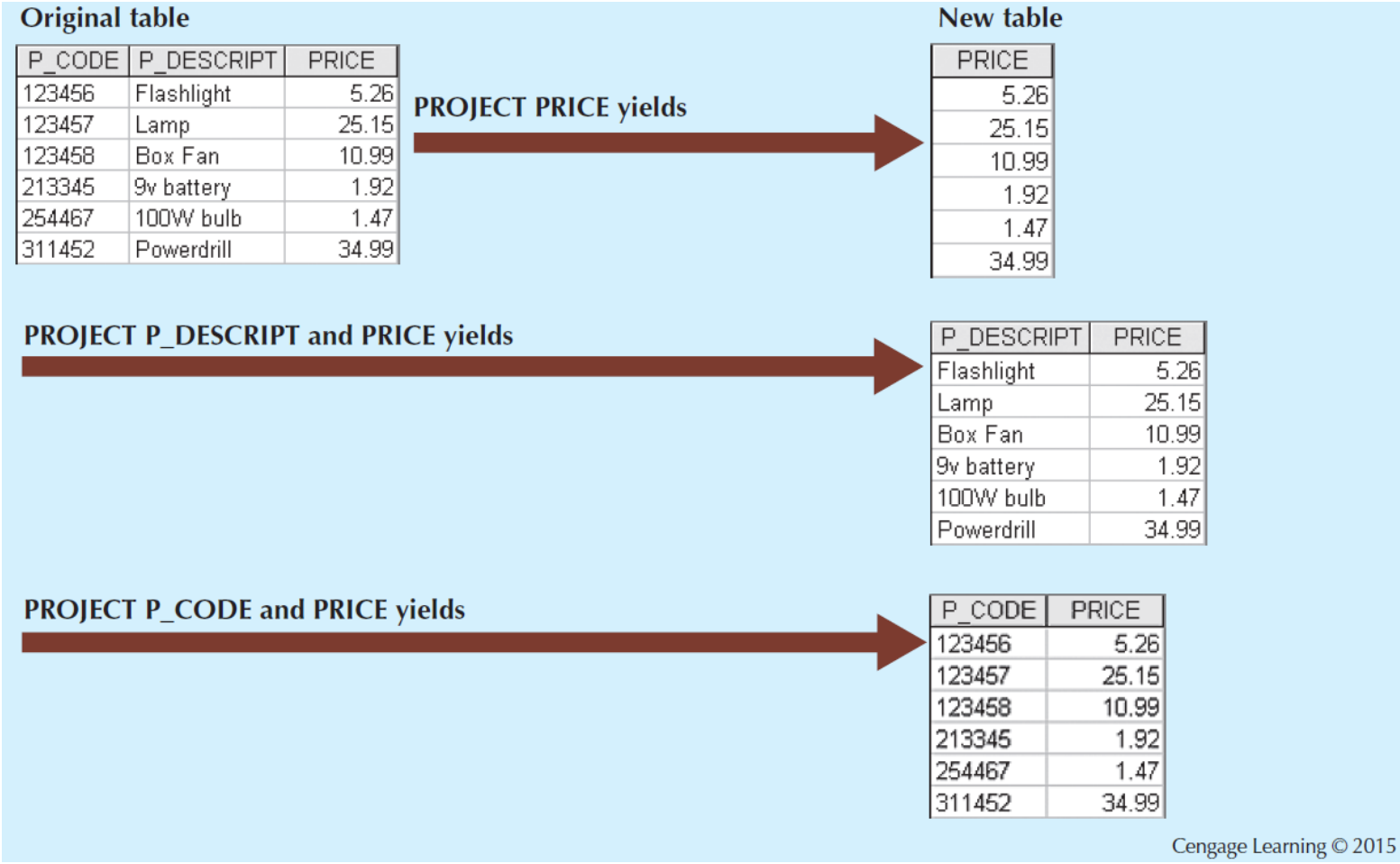
Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Figure 3.5 - Project



Selection & Projection together

ID and Name of players with Yellow cards

$$\prod_{pID,\, pName} \left(\sigma_{yCard=yes}\, Player \right)$$

Names of Colleges that chose players and the player's position:

$$\prod_{cName,pPos} \left(\sigma_{dec=yes}\, Tryout \right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Cartesian Product: ×

College

cName	state	enr
Akron	OH	12000
Maryland	MD	37000
UCLA	CA	40675
N.Dame	IN	8452

×

Player

pID	pName	yCard	HS
1	Alice	0	AHS
2	Bob	1	DHS
3	Carol	2	RHS
4	Dave	0	UHS

=

cName	state	enr	pID	pName	yCard	HS
Akron	OH	12000	1	Alice	0	AHS
Akron	OH	12000	2	Bob	1	DHS
Akron	OH	12000	3	Carol	2	RHS
Akron	OH	12000	4	Dave	0	UHS
Maryland	MD	37000	1	Alice	0	AHS
Maryland	MD	37000	2	Bob	1	DHS
Maryland	MD	37000	3	Carol	2	RHS
Maryland	MD	37000	4	Dave	0	UHS
UCLA	CA	40675	1	Alice	0	AHS
UCLA	CA	40675	2	Bob	1	DHS
UCLA	CA	40675	3	Carol	2	RHS
UCLA	CA	40675	4	Dave	0	UHS

⋮

Figure 3.9 - Product

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	AISLE	SHELF
23	W	5
24	K	9
25	Z	6

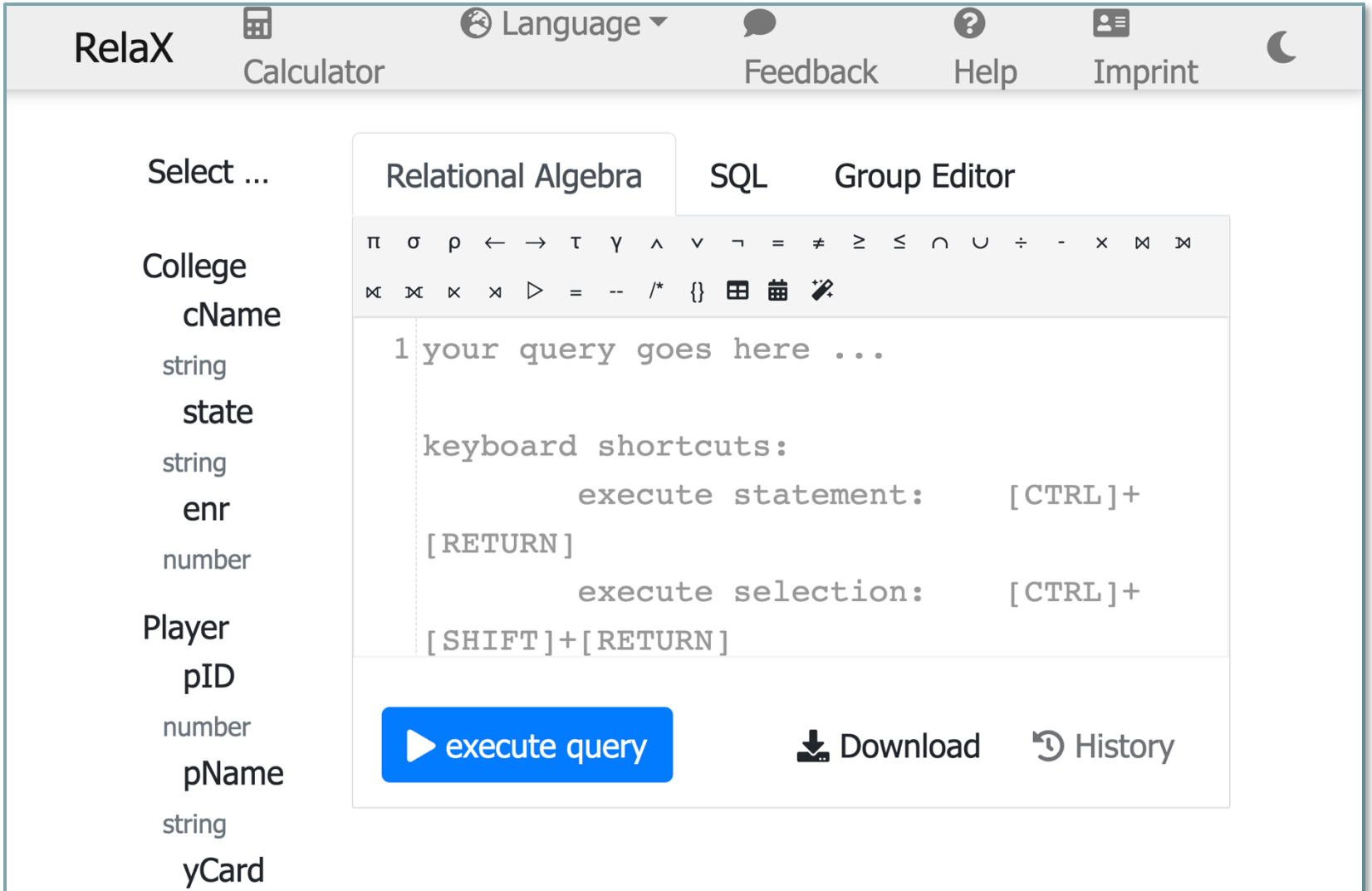
yields

P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

Activity for Relational Modeling part 1

Go to the Relational Algebra calculator setup with our tables

https://bit.ly/3RMhfUk



Activity for Relational Modeling part 1

Enter relational expressions to extract these:

a. ID's of players from LSU

b. Names of players from High Schools with fewer than 600 students

c. ID's of players trying out for goalie

d. Colleges in FL

e. Colleges and position from Tryout

f. Remove the yCard attribute from Player

g. Name and ID of players who tried out for goalie

h. Colleges that selected players

i. Size of High Schools that have Players without yellow cards

j. Why does this only produce one tuple?

$$\pi\ pPos, dec\ (\sigma\ pPos='goalie'\ (Tryout))$$

*“ In specifying a relational algebra, much like specification of an integer algebra, we are able to use symbols in place of relations to solve queries. These operators are subject to the same algebraic properties that integer algebra operators (+, -, *, /) are. As a result, we can assume certain laws that always apply to a relation,* ***any*** *relation, undergoing that operation. For example, in integer algebra we know that addition and multiplication are* ***associative*** *in that we can change the grouping of operands and not change the result:*

$$a + (b + c) = (a + b) + c$$

Similarly, in relational algebra we know that natural join is associative and thus know that A join B join C can be executed in any order. These properties and laws create the power to re-write query formulations and be ***guaranteed*** *to get the same results.”*

“ Another practical advantage to (Relational Algebra) is in the specification of database constraints. First, understanding the relational algebra enables you to determine the simplest way to formulate the constraint. Second, by formulating the constraint in formal logic, you can immediately clarify any ambiguity in intent from the business subject matter experts who formulated the business rule in loose English and avoid bugs.”

-- *Todd Everett in DBA StackExchange*

“He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast.”

-- Leonardo da Vinci

Union Compatible

For the **Union**, **Difference**, and **Intersection** operators the two relations must be ***Union Compatible:***

If two relations *A* and *B* have

1. The same number of attributes, and
2. Attribute A_i has a domain compatible with that of B_i

Then *A* and *B* are said to be *Union Compatible* .

Difference: −

ID's of players who tried out for goalie at LSU and failed, <u>but did not</u> try out for goalie at ASU.

$$\left(\prod_{pID}\left(\sigma_{cName='LSU' \wedge pPos='goalie' \wedge dec='no'} Tryout\right)\right) - \left(\prod_{pID}\left(\sigma_{cName='A\&M' \wedge pPos='goalie'} Tryout\right)\right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Difference example 2

Colleges that did not have any successful player tryouts (including colleges that didn't have *any* tryouts):

$$\prod_{cName}(College) - \left(\prod_{cName}\left(\sigma_{dec='yes'}\, Tryout\right)\right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Figure 3.8 - Difference

MySQL : Use NOT IN instead of MINUS

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

DIFFERENCE

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Martin	Lopez

Union: ∪

ID's of players who tried out for goalie at LSU and succeeded, or who tried out for goalie at ASU and failed

$$\left(\prod_{pID}\left(\sigma_{cName='LSU' \wedge pPos='goalie' \wedge dec='yes'} Tryout\right)\right) \cup \left(\prod_{pID}\left(\sigma_{cName='ASU' \wedge pPos='goalie' \wedge dec='no'} Tryout\right)\right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Figure 3.6 - Union

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00
123458	Box Fan	10.99

yields

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

Intersect: ∩

ID's of players who tried out for goalie at LSU and succeeded AND who tried out for goalie at ASU and failed

$$\left(\prod_{pID}\left(\sigma_{cName='LSU' \wedge pPos='goalie' \wedge dec='yes'} Tryout\right)\right) \cap \left(\prod_{pID}\left(\sigma_{cName='ASU' \wedge pPos='goalie' \wedge dec='no'} Tryout\right)\right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

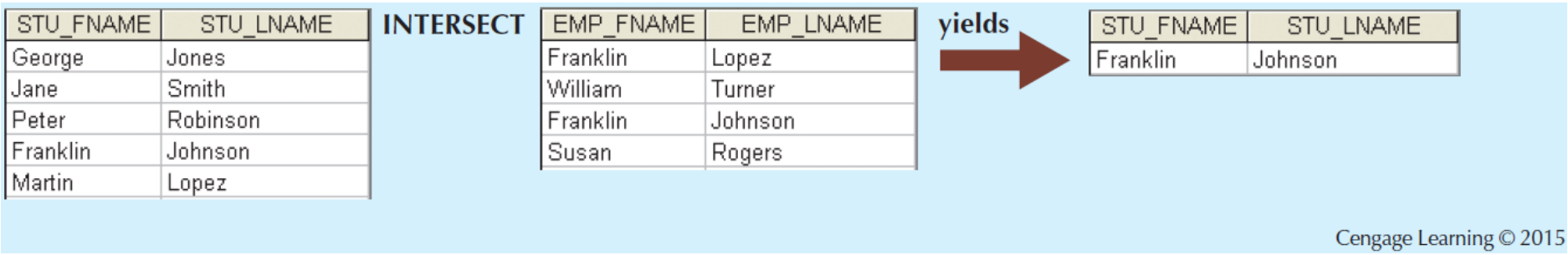
Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Figure 3.7 - Intersect



STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

INTERSECT

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
Franklin	Johnson

Rename: ρ

Find the size of the smallest high school (using a copy of Player under the new name s)

$$\tilde{O}_{HS}(Player) -$$

$$\tilde{O}_{Player.HS}(s_{Player.HS>s.HS}(Player \acute{} r_{s}(Player)))$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

$$\Pi_{HS}(Player) - \Pi_{Player.HS}\left(\sigma_{Player.HS>s.HS}\left(Player \times r_{S}(Player)\right)\right)$$

Player.pID	Player.pName	Player.yCard	Player.HS	s.pID	s.pName	s.yCard	s.HS
10001	Andy	no	1200	10001	Andy	no	1200
10001	Andy	no	1200	20002	Blake	no	1600
10001	Andy	no	✓ 1200	30003	Chuck	no	600
10001	Andy	no	1200	40004	Dave	yes	1600
10001	Andy	no	✓ 1200	50005	Ed	yes	300
10001	Andy	no	✓ 1200	60006	Ford	no	250
20002	Blake	no	✓ 1600	10001	Andy	no	1200
20002	Blake	no	1600	20002	Blake	no	1600
20002	Blake	no	✓ 1600	30003	Chuck	no	600
20002	Blake	no	1600	40004	Dave	yes	1600
20002	Blake	no	✓ 1600	50005	Ed	yes	300
20002	Blake	no	✓ 1600	60006	Ford	no	250
30003	Chuck	no	600	10001	Andy	no	1200
30003	Chuck	no	600	20002	Blake	no	1600
30003	Chuck	no	600	30003	Chuck	no	600
30003	Chuck	no	600	40004	Dave	yes	1600
30003	Chuck	no	✓ 600	50005	Ed	yes	300
30003	Chuck	no	✓ 600	60006	Ford	no	250
40004	Dave	yes	✓ 1600	10001	Andy	no	1200
40004	Dave	yes	1600	20002	Blake	no	1600
40004	Dave	yes	✓ 1600	30003	Chuck	no	600
40004	Dave	yes	1600	40004	Dave	yes	1600
40004	Dave	yes	✓ 1600	50005	Ed	yes	300
40004	Dave	yes	✓ 1600	60006	Ford	no	250
50005	Ed	yes	300	10001	Andy	no	1200
50005	Ed	yes	300	20002	Blake	no	1600
50005	Ed	yes	300	30003	Chuck	no	600
50005	Ed	yes	300	40004	Dave	yes	1600
50005	Ed	yes	300	50005	Ed	yes	300
50005	Ed	yes	✓ 300	60006	Ford	no	250
60006	Ford	no	250	10001	Andy	no	1200
60006	Ford	no	250	20002	Blake	no	1600
60006	Ford	no	250	30003	Chuck	no	600
60006	Ford	no	250	40004	Dave	yes	1600
60006	Ford	no	250	50005	Ed	yes	300
60006	Ford	no	250	60006	Ford	no	250

$$\tilde{O}_{HS}(Player) - \tilde{O}_{Player.HS}\left(\sigma_{Player.HS>s.HS}\left(Player \times \rho_{S}(Player)\right)\right)$$

Player.pID	Player.pName	Player.yCard	Player.HS	s.pID	s.pName	s.yCard	s.HS
10001	Andy	no	1200	30003	Chuck	no	600
10001	Andy	no	1200	50005	Ed	yes	300
10001	Andy	no	1200	60006	Ford	no	250
20002	Blake	no	1600	10001	Andy	no	1200
20002	Blake	no	1600	30003	Chuck	no	600
20002	Blake	no	1600	50005	Ed	yes	300
20002	Blake	no	1600	60006	Ford	no	250
30003	Chuck	no	600	50005	Ed	yes	300
30003	Chuck	no	600	60006	Ford	no	250
40004	Dave	yes	1600	10001	Andy	no	1200
40004	Dave	yes	1600	30003	Chuck	no	600
40004	Dave	yes	1600	50005	Ed	yes	300
40004	Dave	yes	1600	60006	Ford	no	250
50005	Ed	yes	300	60006	Ford	no	250

$$\tilde{O}_{HS}(Player) - \tilde{O}_{Player.HS}\left(s_{Player.HS>s.HS}\left(Player \times r_{S}(Player)\right)\right)$$

Player.HS
1200
1600
600
300
250

−

Player.HS
1200
1600
600
300

=

Player.HS
250

Relational Algebra is SET based! No duplicates!

Three forms of Rename: ρ

Expression E under the name X $\rho_{x}(E)$

Attributes of E renamed $\rho_{A_1, A_2, \ldots, A_n}(E)$

Everything renamed $\rho_{x(A_1, A_2, \ldots, A_n)}(E)$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

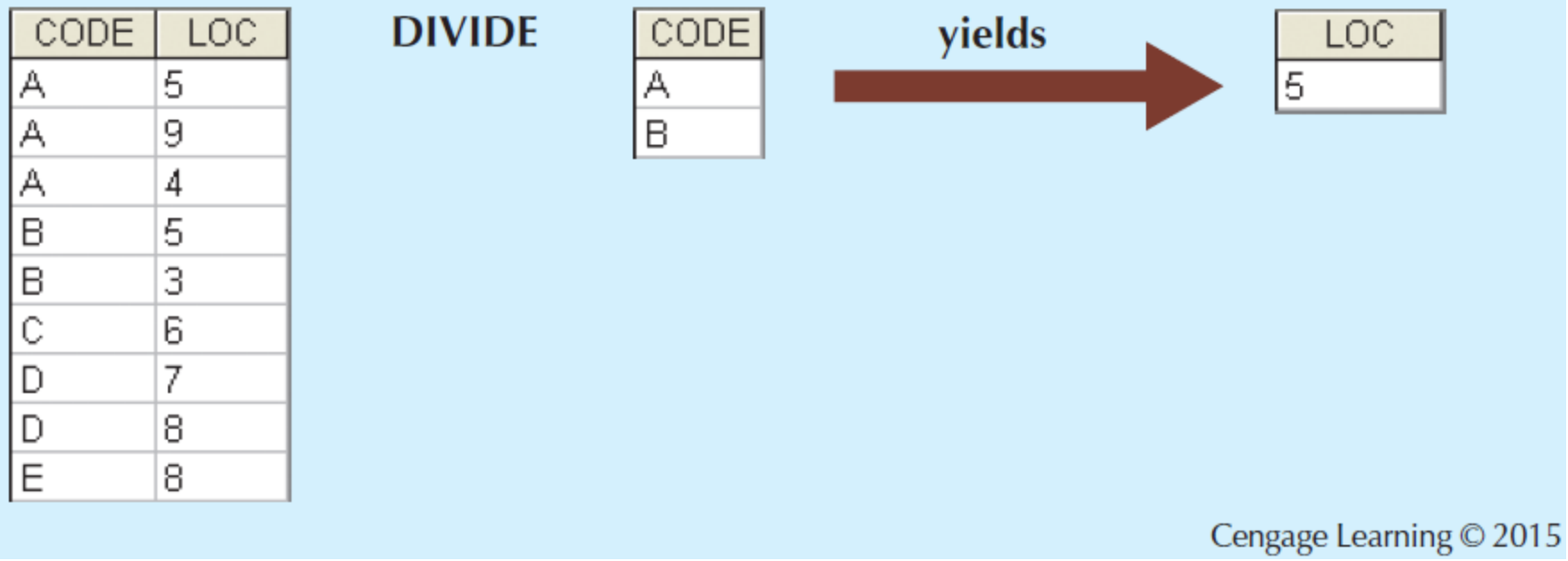
Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Division operator: ÷

Often used for queries using “ALL”.

Which tuples of relation CODE are associated with ALL the tuples of relation B.



Division: ÷

PlayerGoals

pName	Goals	vName
Joe	1	Pepsi
Ann	1	TD-Garden
Joe	2	AA
Jim	3	United
Ann	2	Pepsi
Joe	1	TD-Garden
Ann	0	AA
Joe	2	United
Jim	0	Pepsi
Ann	1	United

Venues

vName	City	State
TD-Garden	Boston	MA
United	Chicago	IL
AA	Dallas	TX
Pepsi	Denver	CO

1. Who played in all of the venues?

$$\Pi_{pName,vName}(PlayerGoals) \div \Pi_{vName}(Venues) \rightarrow (Joe, Ann)$$

2. Who played & scored in all of the venues?

$$\Pi_{pName,vName}(\sigma_{Goals>0}(PlayerGoals)) \div \Pi_{vName}(Venues) \rightarrow (Joe)$$

Strikers accepted by FSU who didn't tryout anywhere else

$$\text{FSUStrikers} = \prod_{\text{pID}} \left(\sigma_{cName='FSU' \wedge pPos='striker' \wedge dec='yes'} Tryout\right)$$

$$\text{OtherStrikers} = \prod_{pID} \left(\sigma_{cName \neq 'FSU' \wedge pPos='striker'} Tryout\right)$$

$$\text{FSUYesOnlyStrikers} = \text{FSUStrikers} - \text{OtherStrikers}$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Natural Join □□

Find the names of players with no yellow cards and the colleges where they tried out.

$$\tilde{O}_{pName,cName}(s_{yCard=0}(Player \square\square Tryout))$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Natural Join ⋈

Find the names of all students from high schools with less than 1000 students who tried out for the position of 'mid' at colleges with more than 10,000 students and failed at least once.

$$\Pi_{pName}\left(\sigma_{HS<1000 \wedge pPos='mid' \wedge dec='no' \wedge enr>10000}\left((College \bowtie Tryout) \bowtie Player\right)\right)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Natural Join ⋈

Find the IDs and names of students who registered but didn't tryout anywhere.

$$\Pi_{pID,pName}((\Pi_{pID} Player - \Pi_{pID} Tryout) \bowtie Player)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

Player

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Theta Join $\bowtie_q$

If the Player schema included the State in which the Player went to High School, we could have:

Find the names of students who tried out in the same state as their high school

$$\Pi_{pName}((College \bowtie Tryout) \bowtie_{state=HSst} Player)$$

College

cName	state	enr
LSU	LA	18000
ASU	AZ	12000
OSU	OK	22000
FSU	FL	18000

PlayerSt

pID	pName	yCard	HS
10001	Andy	no	1200
20002	Blake	no	1600
30003	Chuck	no	600
40004	Dave	yes	1600
50005	Ed	yes	300
60006	Ford	no	250

Tryout

pID	cName	pPos	dec
10001	LSU	goalie	no
10001	ASU	mid	yes
20002	FSU	strike	yes
30003	OSU	mid	no
40004	ASU	goalie	no
50005	LSU	mid	no

Theta Join: $\bowtie_q$

When using theta join, the two relations need not have attributes with the same name if the theta join condition involves attributes from each.

Find which Trucks will be able to tow which trailers.

$$Truck \bowtie_{MaxTowWeight \geq MaxGrossWeight} Trailer$$

Truck

Model	MaxTowWeight
Tundra	10000
Sierra	11200
Silverado	11200
F-150	7600
Titan	9390
RAM150	8750

Trailer

Model	MaxGross Weight
Mech-10	7800
Mech-20	9200
Deca	10000
Mini	7200

Self Join, aided by ρ (rename)

Sometimes we need to compare two tuples p and q of one relation.

Who scored more points than the player with jersey number 10 for any of the regular season games?"

$$S = \rho_X(PlayerStats) \bowtie_{(X.Game=Y.Game)} \rho_Y\ (PlayerStats)$$

$$A = \Pi_{(X.Jersey)}(\sigma_{(X.pts>Y.pts)\wedge(Y.Jersey=10)})\ (S)$$

PlayerStats

Jersey	pName	Game	Pts
10	Andy	1	12
12	Blake	1	22
14	Chuck	1	8
10	Andy	2	14
12	Blake	2	24
16	Dave	2	25

Outer Join – Example

- Relation *instructor2*

ID	name	dept_name
10101	Srinivasan	CompSci
12121	Wu	Finance
15151	Mozart	Music

- Relation *teaches2*

ID	course_id
10101	CS-101
12121	FIN-201
76766	BIO-101

Outer Join – Example

instructor □□ *teaches*

ID	name	dept_name	course_id
10101	Srinivasan	CompSci	CS-101
12121	Wu	Finance	FIN-201

(*Left*) *instructor* ⟕ *teaches*

ID	name	dept_name	course_id
10101	Srinivasan	CompSci	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	*null*

Outer Join – Example

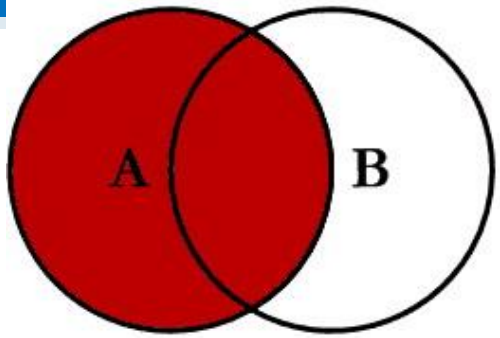
(*Right*) *instructor* ⟖ *teaches*

ID	name	dept_name	course_id
10101	Srinivasan	CompSci	CS-101
12121	Wu	Finance	FIN-201
76766	*null*	*null*	BIO-101

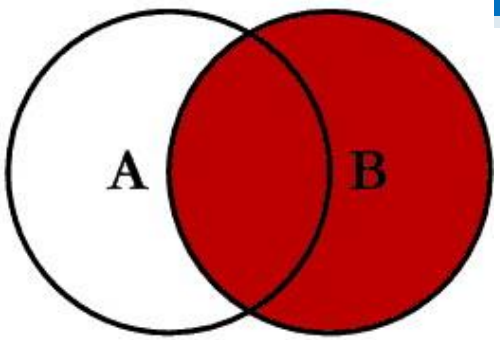
(*Full*) *instructor* ⟗ *teaches*

ID	name	dept_name	course_id
10101	Srinivasan	CompSci	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	*null*
76766	*null*	*null*	BIO-101

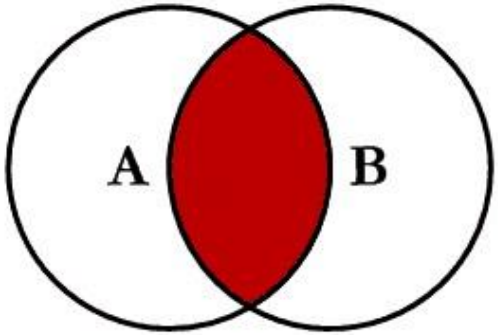
SQL JOINS



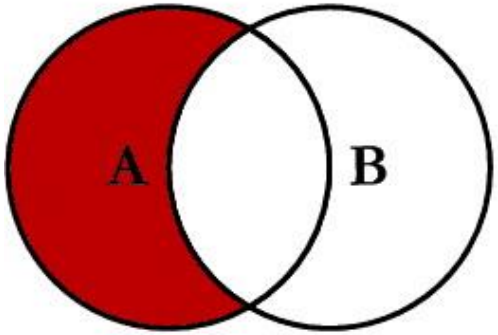

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



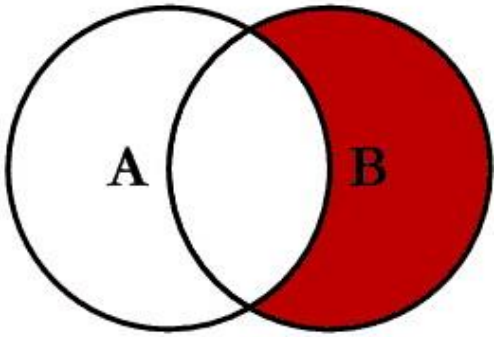

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



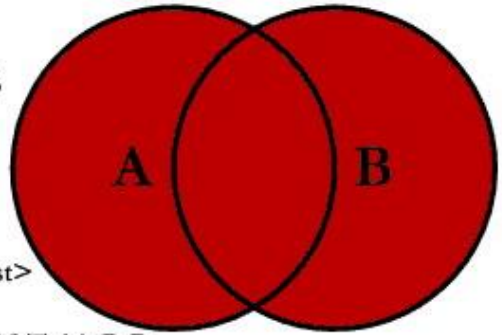

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



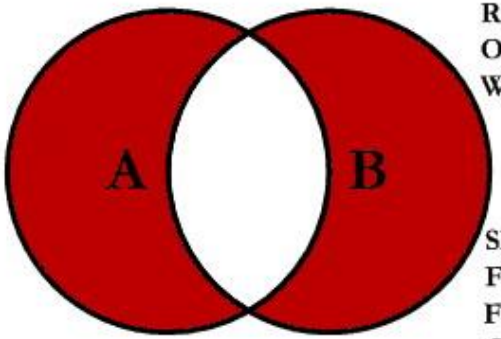

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```