CSCB20: Lecture 1

January 7, 2019 8:20 AM

what is a Database?

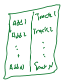
- It is a collection of interrelated data.
- The data is relevant to an enterprise (FB, Google, Amazon)
  - eg amazon shows you items related to you.

what is a database management system (DBMS)?

- A database and a set of programs to access database.
- Provides ways to store and retrieve database information.
- Must be convenient and efficient.
- Two things database should do efficiently: write, read

Where are databases used?

- Enterprise Information
  - Sales
  - Accounting
  - HR - Human Resource
  - Manufacturing
  - Online retailers
- Banking and Finance
  - Banking
  - Credit card transaction
  - finance
- Other applications?
  - Universities
  - Airlines
  - Telecommunication
  - more!



when did databases occur?

- 1960's data storage changed from tape to direct access. → has memory address (location)
- This allowed shared interactive data use.
- Early databases were navigational which was very inefficient for searching
- Edgar Codd created a new system in the 1970s based on the relational model.
- Late 1970's and early 1980's SQL was developed based on a relational model, which is the foundation of current databases and what we will study. → relational model is improvement of navigational model ⇒ Table form.
- In 2000's, with increasingly large data sets, new XML databases and NoSQL databases are becoming more prevalent.

Why use databases?

- Commercialized management of large amounts of data.
- Ability to update and maintain data
- keep track of relationships between subsets of the data
- efficient access and searching capabilities
- multiple users can access and share data.
- Ability to limit access to a certain portion of the data according to user type and enables security of data.
- Minimizes redundancy of multiple data sets
- Enables consistency constraints
- Allows users an abstract view of the data which hides the details of how the data are stored and maintained.

Data Abstraction - how?

- Physical level:
  - lowest level, how the data are actually stored
  - Usually in complex low-level data structures
- Logical Level: Ways to connect data (eg. kindle users are likely to purchase ebooks.
  - what data are stored in the database and what relationships exist between the data.
  - Implementing the simple structure of the logical level may require complex physical low level structures.
- View level - logical table [$a+b+c+d=17$], what is $a+b$? $x=a+b$ then $x+c+d=17$ | Intermediate table that can be developed. - describes only a small portion of the data base.
  - Highest level of abstraction
  - Allows users to simplify their interaction with the database system.
  - Can have many views, why is this good?

eg.

Name	Quiz 1
A	⋮
⋮	
Z	

An intermediate table, give every student with first name starting with a "C".

New table

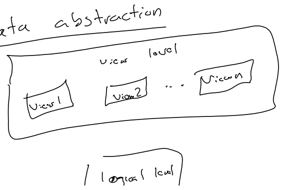
Name	Quiz 1
C	⋮
⋮	
C	

} views!

why do we want views?

- simplifies interaction with users.

Data abstraction



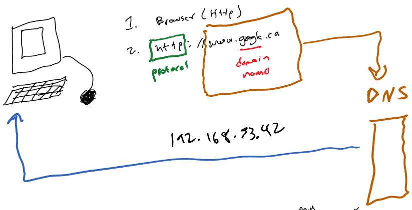
Protocle: allows to communicate with a specific rules.

eg. browser - Http

Prof - english

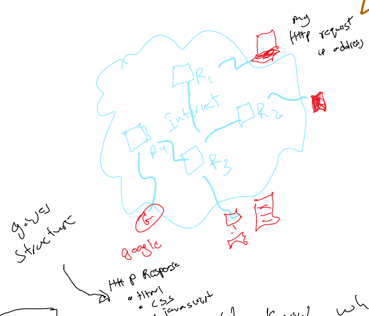
speak in the protocle

Browser understands Http protocol.



DNS → Domain Name System: Machine that allows you to convert string into #.

= IP address of DNS is provided by your IP service provider eg. rogers, bell ...



IP address: # that identifies a location of hosts / destination

IP addresses are hard to memorize, so we map it to a domain using DNS.

Think of DNS as a machine to translate IP address to domains vice versa

You don't know which path you take to get to your end destination. You just know the destination. Each router takes the best next path. Think of it like a post office. whichever gets you to your destination quicker.

your browser on your computer understands Http. And sends Http request

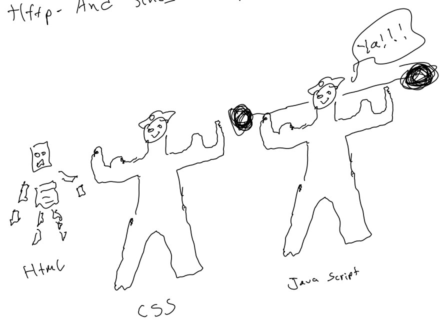
On google's side - a web server to understands Http. And sends Http response.

Html - skeleton

CSS - look and feel

Just Html and CSS provides a picture

Javascript - makes it interactive

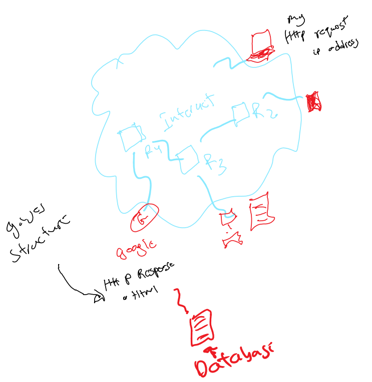


To ban an IP Address: block the end point IP Address

- Use VPN to by pass
- Eg to watch US netflix, use VPN to use a US router.

Where does databases fit in the web?

- Username and passwords



Our journey today starts from databases.

:)

Terminology

Database Schema: The logical design of the database

Database instance: snapshot of the data in the database

Relation Schema:

Relational model

Table = relation
row = tuples
column = attributes

Database is a collection of tables each having a unique name

Instance of a database is the info stored at a particular moment in time.

Schema: the overall design of the database

instances changes quick, schema usually doesn't

domain? limits of column such as credits and salary. Actual range for which the values are being pulled off to create a table.

Database Schema: The logical design of the database
Database instance: snapshot of the data in the database
Relation Schema:

How do we uniquely refer to a tupple or row in a schema?
Super key: set of attributes that allow you to differentiate one row from another row.

What are possible super keys for the instructor relations?
instructor (ID, name, dept_name, salary) ← still a super key cause id is present and unique.

Teaches relation

ID is not a super key.

Good super key: ID and course ID? no cause multiple selection possible

Better to take everything.

Candidate key: chooses a minimal super key.
Not long of attribute!

Eg: Super key for relation instructor

$\underbrace{\{ID\}}_{SK1}, \underbrace{\{name, dept\_name\}}_{SK2}, \underbrace{\{ID, name\}}_{SK3}$

which is a SK?

A. {ID}, {Name, dept_name}

contains enough attributes to uniquely identify

January 14, 2019 9:05 AM

Admin

First Quiz Jan 25^{th},

6am ~ 9am release on Friday.

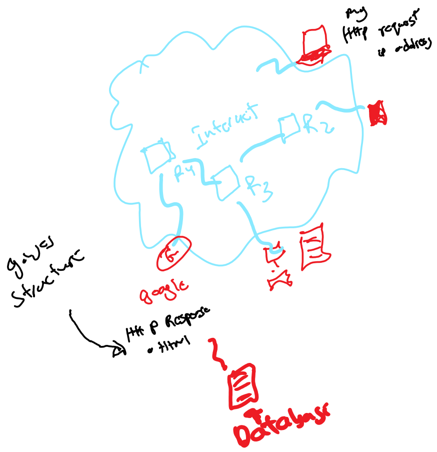
Everything available sometime this week.

Abbas office hours

Mon: 11:30 - 2:30 - IC349

Fri: 3:00Pm - 5:00Pm - IC349

Week 2

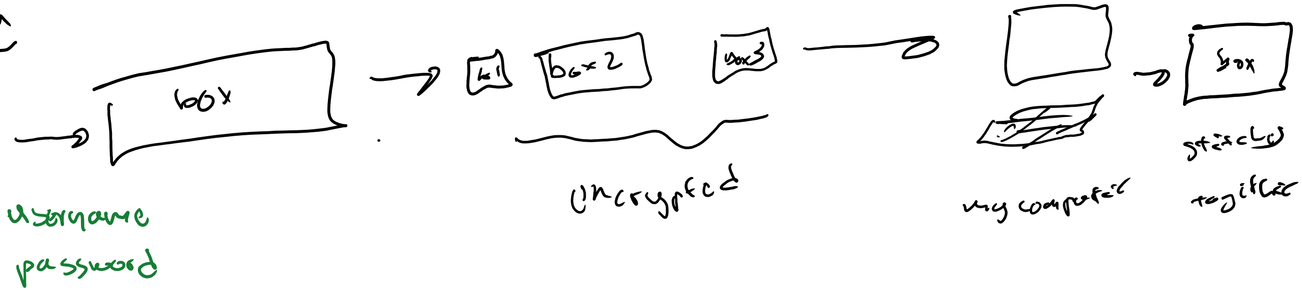


- DNS tells you destination but not the route.
  IP address

- Router breaks the file/movie into small packets and sends it.
  - 5gb could take a lot of time
  - Saves bandwidth, instead of sending 5gb
  - router has finite memory, can only hold a certain # of packets
  - if too many incoming traffic to the buffer overflows and packet drops.
  - dropped packets can get resent.

- browser takes all these packets and stitches it back together.

Encryption



Relational Algebra = Sudo Code

Example of a Relation

attributes (or columns)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

tuples (or rows)

Relation Schema: instructor(ID, name, dept_name, salary)

Domain	Type
id	integer
name	string
salary	integer constrained ≥ 0

relational database → data in a table

Terminology

Q. What is a *superkey*?

A. A set of one or more attributes that *uniquely identify* a tuple in the relation.

Q. What is a *candidate* key?

A. A *minimal* super key.

Q. What is a *primary* key?

A. A candidate key chosen to distinguish between

Minimal != length or cardinality

Minimal = enough attributes that if one is removed then you can't uniquely differentiate 1 row from another

Example of a Relation

attributes (or columns)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

tuples (or rows)

Relation Schema: instructor(ID, name, dept_name, salary)

Key: by definition is minimal

Super: key any attributes that is minimal or not is a super key

{ID} → key & super key

{ID, Name} → Super key

{ID, Name, Salary} → Super key

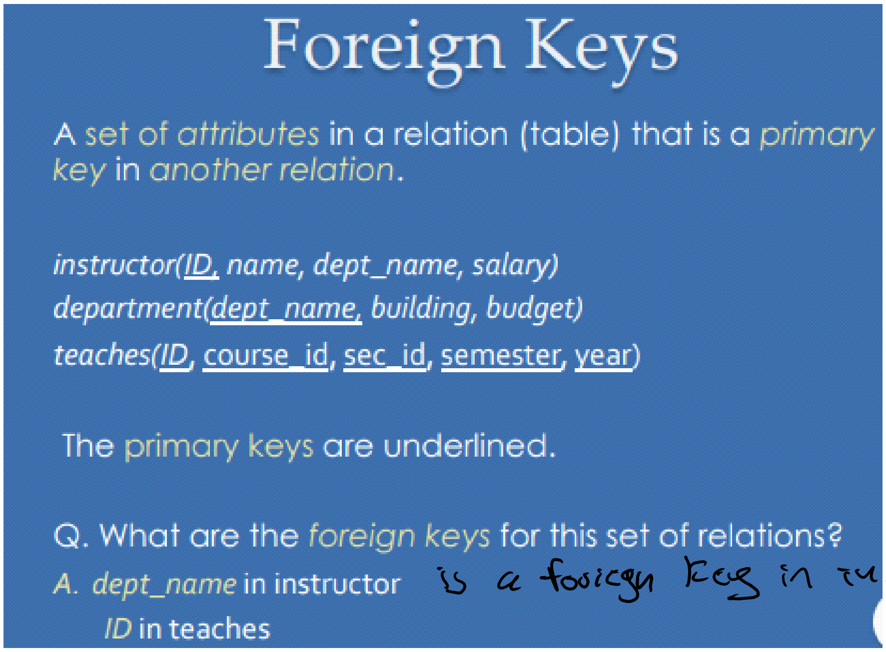
{Name, Department, Salary} → Minimal

{Name, Department} → not a key

January 14, 2019 9:42 AM

Primary key:

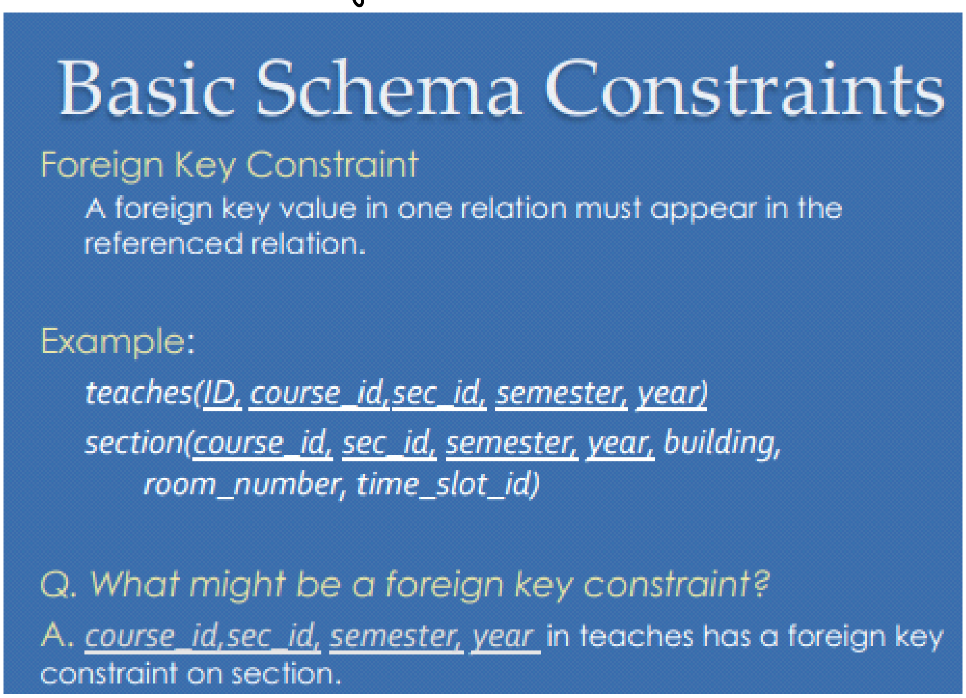
A candidate key chosen to distinguish between tuples

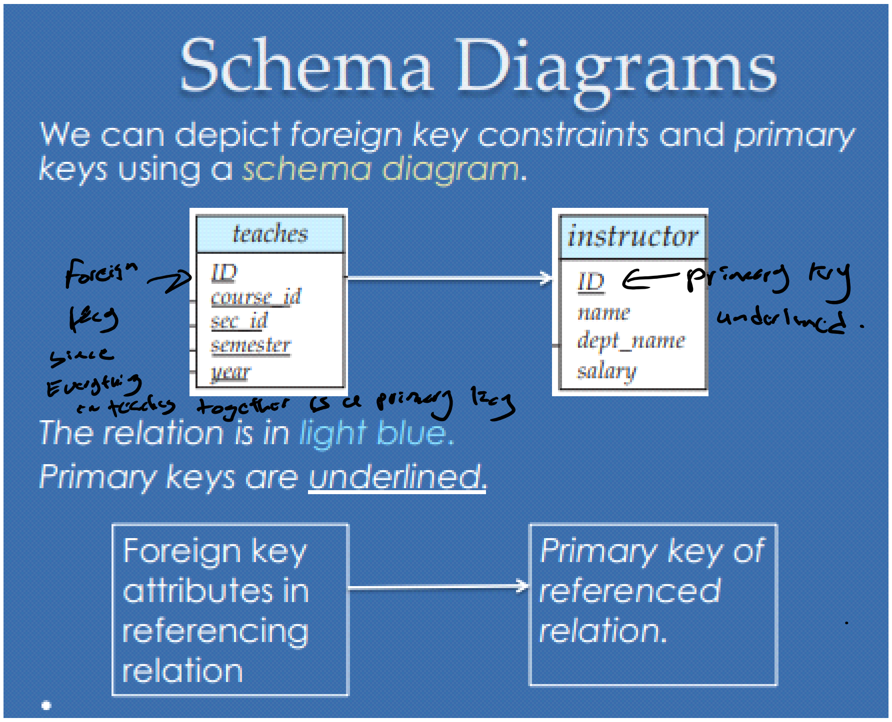


is a foreign key in instructor

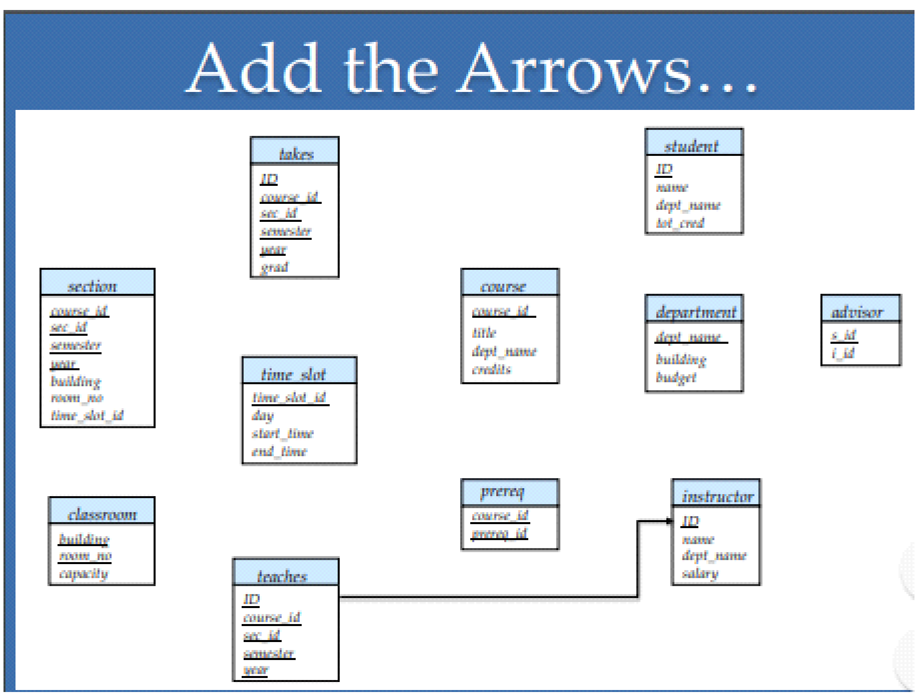
underlined are primary keys for each relation

Important: because you can connect different tables together.




Schema Diagrams
We can depict foreign key constraints and primary keys using a schema diagram.
teaches
ID
course_id
sec_id
semester
year
instructor
ID
name
dept_name
salary
Foreign key
since Everything in teaches together is a primary key
primary key underlined.
The relation is in light blue.
Primary keys are underlined.
Foreign key attributes in referencing relation
Primary key of referenced relation.

Ex


Add the Arrows…
takes
ID
course_id
sec_id
semester
year
grad
student
ID
name
dept_name
tot_cred
section
course_id
sec_id
semester
year
building
room_no
time_slot_id
time_slot
time_slot_id
day
start_time
end_time
course
course_id
title
dept_name
credits
department
dept_name
building
budget
advisor
s_id
i_id
classroom
building
room_no
capacity
prereq
course_id
prereq_id
instructor
ID
name
dept_name
salary
teaches
ID
course_id
sec_id
semester
year

Relational Operations

We have a set of tables or relations.

Now what? How do we get information from them?

We perform *queries*.

> Simple Query:
>
> *select tuples from a relation satisfying a predicate*

Results in a new relation that is a subset of the original.

Why is it useful that the result is a relation?

Selection

table x

Notation is $\sigma_p(x)$. $\qquad \sigma_p(x)$

predicate

p is the *selection predicate*
x is the *relation*

p is a *boolean* formula of *terms* and *connectives.*

Connectives: $\wedge$ (and), $\vee$ (or), ~ (not)
Operators: <, >, ≤, ≥, =, ≠
Terms:

- attribute operator attribute
- attribute operator constant

Selection

Notation is $\sigma_p(x)$.

$\boxed{\sigma_{salary >= 85000}(instructor)}$

Every row that meets predicate p

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Instructor Relation

Select the tuples with attribute salary at least 85000 from the instructor relation.

Selection

Notation is $\sigma_p(x)$.

$\boxed{\sigma_{salary >= 85000}(instructor)}$

ID	name	dept_name	salary
12121	Wu	Finance	90000
22222	Einstein	Physics	95000
33456	Gold	Physics	87000
83821	Brandt	Comp. Sci.	92000

To remove colums use projection.

Select the tuples with attribute salary at least 85000 from the instructor relation.

January 14, 2019 9:58 AM

Projection

Symbol is Π

Selection of attributes.

$\Pi_{ID,\ salary}(instructor)$

output

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Select all tuples from the instructor relation with attributes ID and salary.

Join only works with two tables. Binary. Selection works on one or more

- look at columns exactly identical in T_1 and T_2
- look at common values and shared attributes.
- if 2 Comp Sci then rows from table 1 will be combined with 2 in table 2.

Natural Join

Combine two relations into a single relation.

The tuples are joined if the attributes common to both relations are equal.

T_1 T_2

instructor ⋈ department

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

⋈

dept_name	building	budget
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Natural Join

The tuples are joined if the attributes common to both relations are equal.

instructor ⋈ department

output of the natural join

ID	name	salary	dept_name	building	budget
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
12121	Wu	90000	Finance	Painter	120000
15151	Mozart	40000	Music	Packard	80000
22222	Einstein	95000	Physics	Watson	70000
32343	El Said	60000	History	Painter	50000
33456	Gold	87000	Physics	Watson	70000
45565	Katz	75000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
76543	Singh	80000	Finance	Painter	120000
76766	Crick	72000	Biology	Watson	90000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000

Which common attribute(s) are these relations joined on?

Cartesian Product

This is the *cross product* of two relations.

Q. What is the cross product of {a, b} and {c, d}?

A. {a, b} X {c, d} produces {(a, c), (a, d), (b, c), (b, d)}

T_1 T_2

The cross product produces all possible pairs of rows of the two relations.

Q. Can you see a problem?

A. If the two relations have attributes in common, how do we tell which relation each attribute is from?

row in T_2

$m \times n$

row in T_1

Cartesian Product Example

Relations *r*, *s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

8 rows

r x *s*:

cartesian join

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

4

4

Renaming Attributes

Allows us to refer to a relation, (say E) by more than one name.

$\rho_x(E)$ Given table E

returns the expression E under the name X

prepend every column with character x

Example

Relations r

A	B
a	1
b	2

cartesian join

$r \times \rho_s(r)$ prepend

column A of table s

r.A	r.B	s.A	s.B
a	1	a	1
a	1	b	2
b	2	a	1
b	2	b	2

s.A	s.B
a	1
b	2

Union

Relations r, s:

For $r \cup s$ to be valid.

1. r, s must have the *same arity* (same number of attributes)
2. The attribute domains must be *compatible*

i.e, 2nd column of r deals with the same type of values as does the 2nd column of s.

Q. Did you expect there to be 4 rows?

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cup s$:

A	B
α	1
α	2
β	1
β	3

Union possible when same columns

Difference

remove every single item in table S

What would you expect them to be?

- Relations r, s:

r

A	B
α	1
~~α~~	~~2~~
β	1

s

A	B
α	2
β	3

- $r - s$:

A	B
α	1
β	1

January 14, 2019 10:26 AM

Intersection

- Relation *r*, *s*:

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

- $r \cap s$

A	B
α	2

Note: $r \cap s = r - (r - s)$ ← relation algebra dawg

Printout

January 14, 2019 10:30 AM

CSCB20
UTSC
Worksheet 1

January 14, 2019

1. In the figure below, are instances of two relations that might constitute part of a banking database. Indicate the following:

acctNo	*type*	*balance*
12345	savings	12000
23456	checking	1000
34567	savings	25

The relation Accounts

firstName	*lastName*	*idNo*	*account*
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

The relation Customers

Figure 1: *
For question1

(a) The attributes of each relation
(b) The tuples of each relation
(c) The components of one tuple from each relation
(d) The relation schema for each relation → set of relation schema
(e) The database schema
(f) A suitable domain for each attribute
(g) Another equivalent way to present each relation.

d) Account (acctNo, Type, balance)
Customers (FN, LN, IdNo, Acc Num)

e) box is database schema

f) Account Num: integer
type: string
id-Num: string

g)

AcctNum	balance	type

equivalent to table shown

2. Assume that the database schema consists of four relations, whose schemas are:

Table →
```
Product(maker, model, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

We also give you a sample snapshot of these relations see below in figure ??. Write expressions of relational algebra to answer the following queries. Your answer should work for arbitrary data, not just the data of these figures. We also assume for convenience purposes that the model numbers are unique across all the different manufacturers and across all the product types.

(a) What PC models have a speed of atleast 3.00? $\Pi_{model}(\sigma_{speed \geq 3.0}(PC))$ $R_2 = \Pi_{model}(R_1)$

(b) Which manufactures make laptops with a hard disk of atleast 100GB? ① $R_1 = (\sigma_{hd \geq 100}(laptops))$ → model, speed, ram, HD, price, scan

(c) Find the model number and price of all products (of any type) made by manufacturer B?

\# Natural join

(d) Find the model numbers of all color laser printers?

$Product \bowtie \sigma_{hd \geq 100}(Laptops)$

\# common attribute is model.

(e) Find those manufacturers that sell Laptops, but not PCs?

or

(f) Find those hard-disk sizes that occur in two or more PCs ③ $R_2 = Product \bowtie R_1$ ④ $\Pi_{maker}(R_2)$

(g) Find those pairs of PC models that have both the same speed and RAM. A pair should be listed only once; e.g. list (i,j) but not (j,i).

(h) Find those manufacturers of at-least two different computers (PC's or laptops) with speeds of at-least 2.80

(i) Find the manufacturer(s) of the computer (PC or laptop) with the highest available speed.

(j) Find the manufacturer of PC's with at-least three different speed.

(k) Find the manufacturers who sell exactly three different models of PC.

c) $(PC \bowtie Product) = R_1$

$(Laptop \bowtie Product) = R_2$

$(Printer \bowtie Product) = R_3$

$\Pi_{model, price}\ \sigma_{maker=B}(R_1) = R_4$

$\Pi_{model, price}\ \sigma_{maker=B}(R_2) = R_5$

$\Pi_{model, price}\ \sigma_{maker=C}(R_3) = R_6$

$R_4 \cup R_5 \cup R_6$ = Result

maker	*model*	*type*
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
D	1008	pc
D	1009	pc
D	1010	pc
D	3004	printer
D	3005	printer
E	1011	pc
E	1012	pc
E	1013	pc
E	2001	laptop
E	2002	laptop
E	2003	laptop
E	3001	printer
E	3002	printer
E	3003	printer
F	2008	laptop
F	2009	laptop
G	2010	laptop
H	3006	printer
H	3007	printer

Figure 2: *
Sample data for Product

model	*speed*	*ram*	*hd*	*price*
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649
1005	3.20	512	250	630
1006	3.20	1024	320	1049
1007	2.20	1024	200	510
1008	2.20	2048	250	770
1009	2.00	1024	250	650
1010	2.80	2048	300	770
1011	1.86	2048	160	959
1012	2.80	1024	160	649
1013	3.06	512	80	529

Figure 3: *
Sample data for relation PC

model	*speed*	*ram*	*hd*	*screen*	*price*
2001	2.00	2048	240	20.1	3673
2002	1.73	1024	80	17.0	949
2003	1.80	512	60	15.4	549
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500
2006	2.00	2048	80	15.4	1700
2007	1.83	1024	120	13.3	1429
2008	1.60	1024	100	15.4	900
2009	1.60	512	80	14.1	680
2010	2.00	2048	160	15.4	2300

Figure 4: *
Sample data for relation Laptop

model	*color*	*type*	*price*
3001	true	ink-jet	99
3002	false	laser	239
3003	true	laser	899
3004	true	ink-jet	120
3005	false	laser	120
3006	true	ink-jet	100
3007	true	laser	200

Figure 5: *
Sample data for relation Printer