

Topic 1: SQL

L02: SQL basics

Please ask questions as we go along 😊

Wolfgang Gatterbauer

CS3200 Database design (fa22)

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

9/12/2022

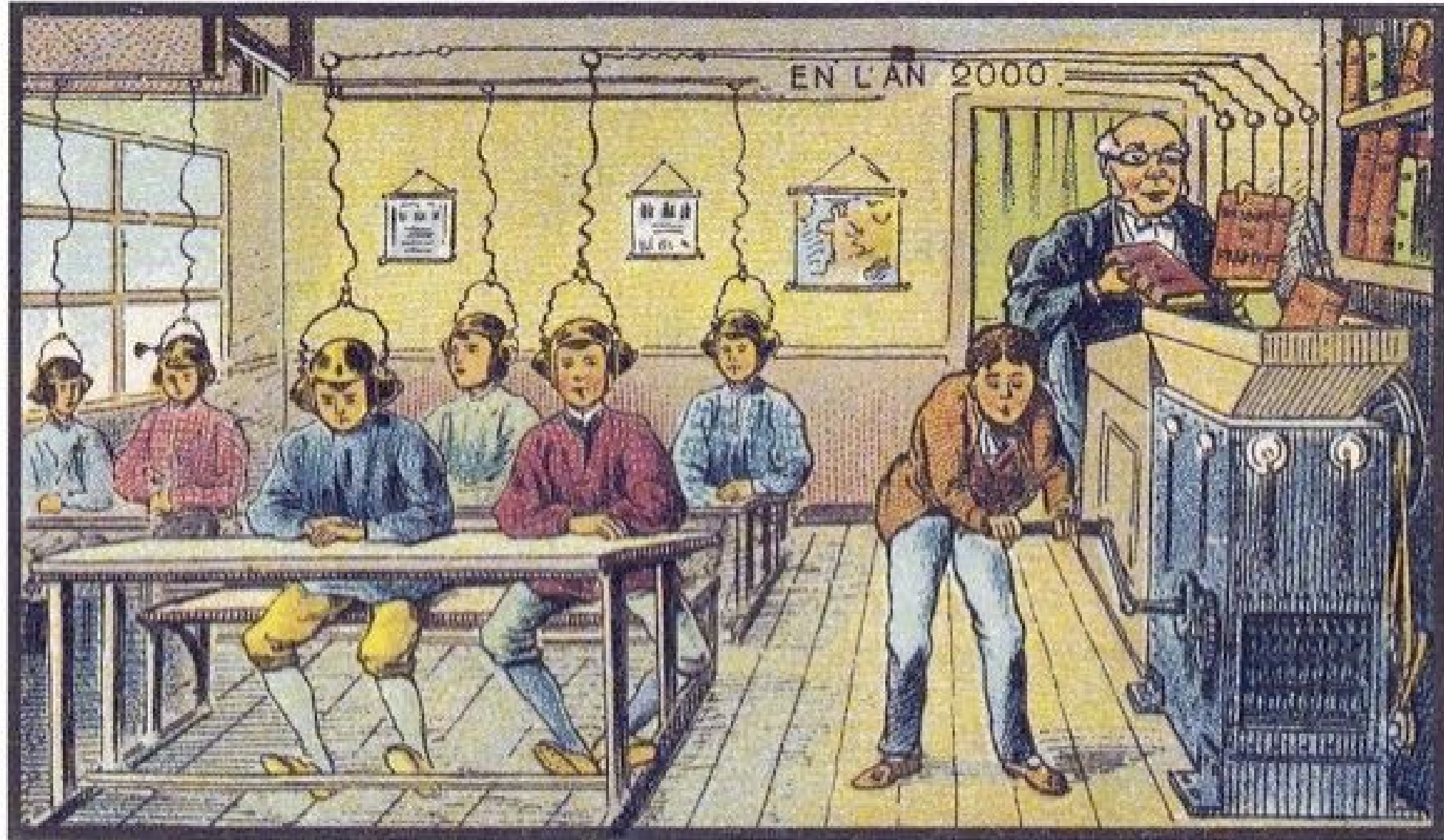
Class warm-up

- Last class recapitulation
- Status of Postgres setup?
- Due Dates are Thursdays
- Updated Honor codes are now online (differences in red)
- Why not flipped classroom?
- FM slide projection presenter view
- Game of life: parallel vs. serial computation
- more SQL (Please keep up the great questions!)

may tell you). Lecture slides will be posted after each class, usually by **the end of the day following class**. Also notice that due dates are usually Thursdays each week (shown in the Wednesday row below).

#	Date	Approximate Topics	Slides & further readings	Thursdays
SQL				
1	W Sept 7	course overview, databases, client-server architecture, tables, database schema, basic SQL (SELECT FROM WHERE)	Setup PostgreSQL L01-Introduction L01-SQL	
2	M Sept 12	selection, projection, distinct, ordering, in, like, schemas and key constraints and foreign key constraints (referential integrity), joins, table alias	Setup Gradiance, SAMS Ch 1-3, SDK 3.1-3.3, 3.4.4, 3.4.5, 3.9, 4.1, 4.4.5, 4.5.1	
3	W Sept 14	column alias, conceptual evaluation strategy of SQL, table aliases for self-joins, cross join, equi-joins, alternative join syntax (join on), some history, create tables and insert values	SAMS 4 & 12	G1, HW0

The year 2000 imagined in 1900



At School

Source: <https://publicdomainreview.org/collection/a-19th-century-vision-of-the-year-2000>

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

Predicting the role of IT in Education



◁ **Learning** by computer in the future will be fun. This computer is displaying a chemistry experiment for the older child and arithmetic problems for the younger one. The computer controls include light pens to draw on the screens. The chemistry student has done something wrong and has caused an explosion!

One reason why I don't post slides *before* lecture

From the preamble of one of the best physics books ever: „How to read this book“

The best way to use this book is NOT to simply read it or study it, but to read a question and STOP. Even close the book. Even put it away and THINK about the question. Only after you have formed a reasoned opinion should you read the solution. Why torture yourself thinking? Why jog? Why do push-ups?

If you are given a hammer with which to drive nails at the age of three you may think to yourself, “OK, nice.” But if you are given a hard rock with which to drive nails at the age of three, and at the age of four you are given a hammer, you think to yourself, “What a marvelous invention!” You see, you can't really appreciate the solution until you first appreciate the problem.

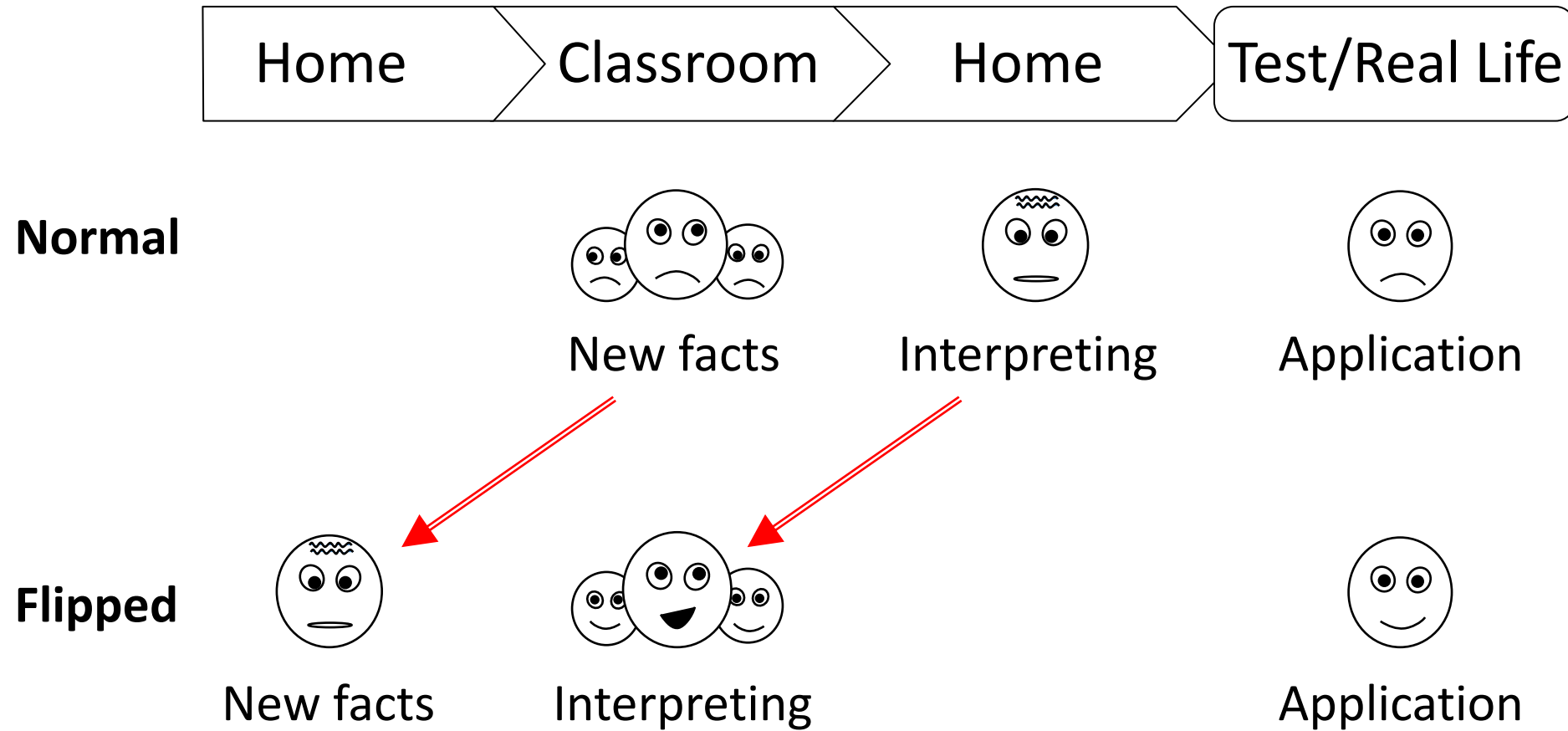
...

...

Let this book, then, be your guide to mental push-ups. Think carefully about the questions and their answers *before* you read the answers offered by the author. **You will find many answers don't turn out as you first expect. Does this mean you have no sense for physics? Not at all. Most questions were deliberately chosen to illustrate those aspects of physics which seem contrary to casual surmise. Revising ideas, even in the privacy of your own mind, is not painless work.** But in doing so you will revisit some of the problems that haunted the minds of Archimedes, Galileo, Newton, Maxwell, and Einstein.* The physics you cover here in hours took them centuries to master. Your hours of thinking will be a rewarding experience. Enjoy!

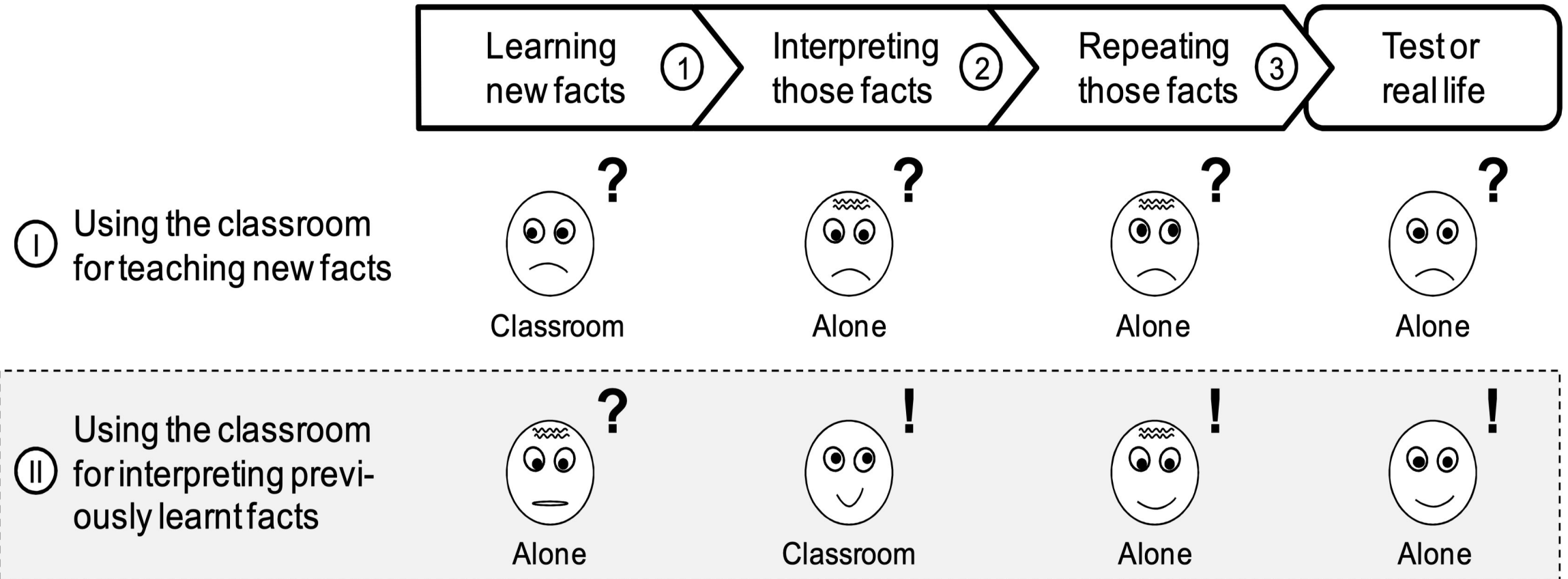
Lewis Epstein

Former Classroom Philosophy

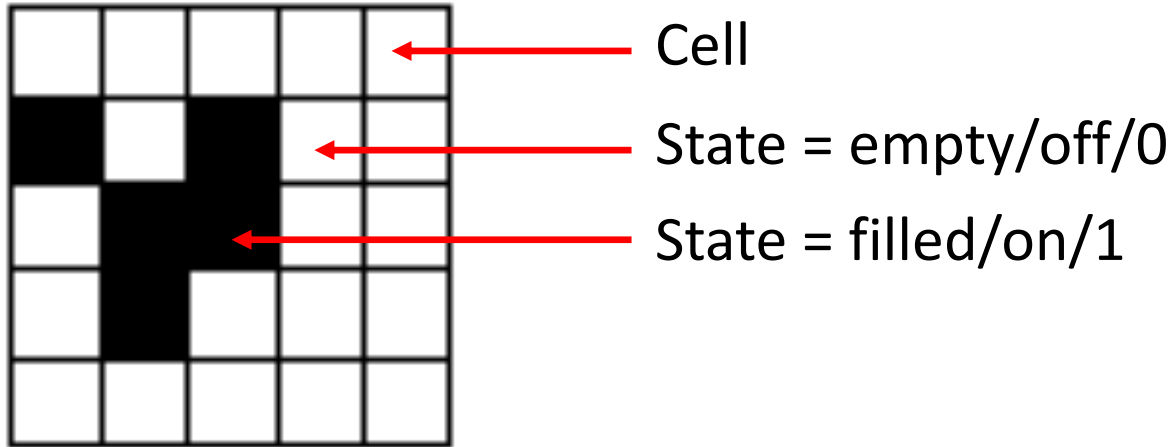


"Flipped Classroom" requires preparation before coming to class.
In my experience works better for graduate than undergraduate classes.

Former Classroom Philosophy



What is the optimal use of time in the classroom?

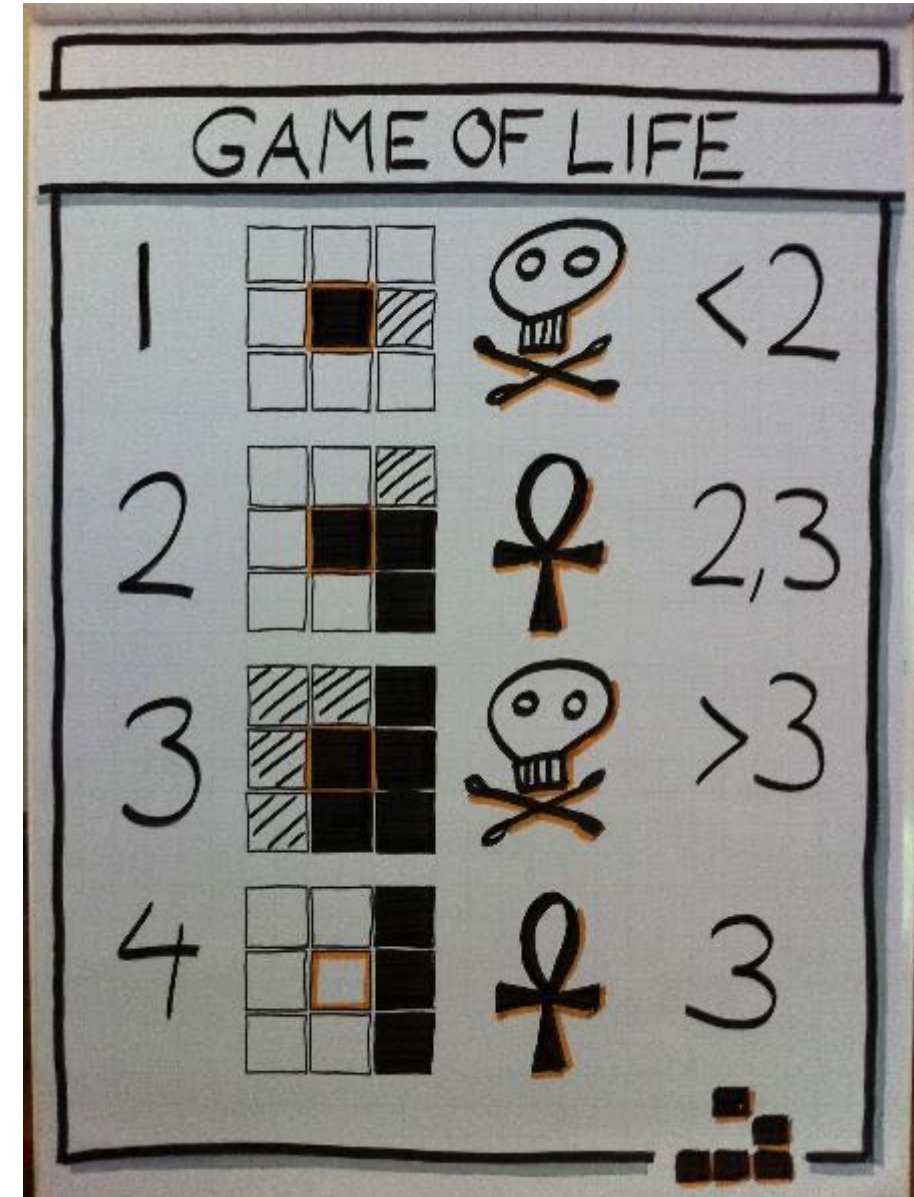


- An automaton consists of a grid/lattice of cells each of which can be in a (normally small and finite) number of states
- Concept was originally discovered in the 1940s by Stanislaw Ulam and John von Neumann

Conway's Game of Life

Side-topic

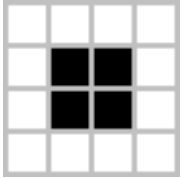
- Devised ~1970
by John Horton Conway
- Simple rules, but still Turing Complete, i.e., with a suitable initial pattern, one can do any computation that can be done on any computer
- Start with different (often random) initial conditions



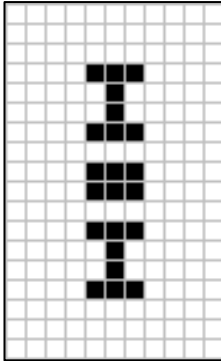
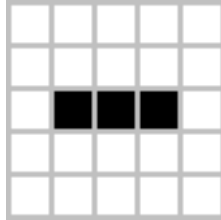
Example patterns

Side-topic

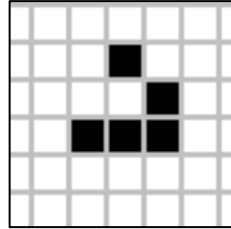
Still life



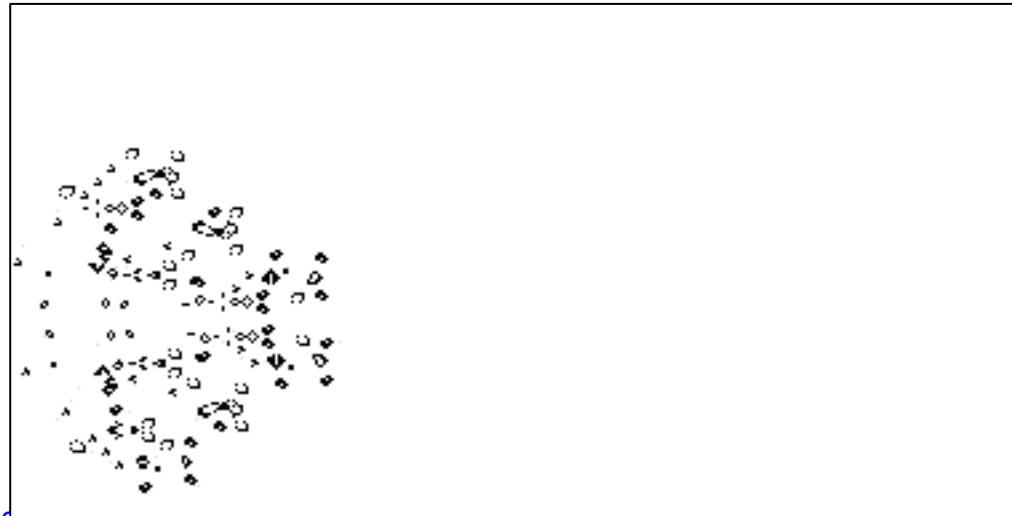
Oscillators



Spaceships



Guns



Product

PName	Price	Category	Manufacturer
Gizmo	19.99	Gadgets	GizmoWorks
PowerGizmo	29.99	Gadgets	GizmoWorks
SingleTouch	149.99	Photography	Canon
MultiTouch	203.99	Household	Hitachi



```
SELECT DISTINCT category  
FROM Product  
ORDER BY category
```



?

```
SELECT category  
FROM Product  
ORDER BY pName
```



?

```
SELECT DISTINCT category  
FROM Product  
ORDER BY pName
```



?

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT DISTINCT category
FROM Product
ORDER BY category
```



Category
Gadgets
Household
Photography

```
SELECT category
FROM Product
ORDER BY pName
```



?

```
SELECT DISTINCT category
FROM Product
ORDER BY pName
```



?

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT DISTINCT category
FROM Product
ORDER BY category
```



Category
Gadgets
Household
Photography

```
SELECT category
FROM Product
ORDER BY pName
```



Category
Gadgets
Household
Gadgets
Photography

```
SELECT DISTINCT category
FROM Product
ORDER BY pName
```



?

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT DISTINCT category
FROM Product
ORDER BY category
```



Category
Gadgets
Household
Photography

```
SELECT category
FROM Product
ORDER BY pName
```



Category
Gadgets
Household
Gadgets
Photography

```
SELECT DISTINCT category
FROM Product
ORDER BY pName
```



Syntax error on large more "principled" DBMSs
(Oracle, PostgreSQL, SQL server) /
unpredictable results on others (MySQL, SQLite)

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

If you like to see what MySQL returns,, visit:

<https://bit.ly/3QGeoL8>

(for <http://sqlfiddle.com/#!9/cec7edc/1>)



```
SELECT DISTINCT category
FROM Product
ORDER BY pName
```



category
Gadgets
Household
Photography

✓ Record Count: 3; Execution Time: 24ms

Simple SQL Query

If you like to run the query for now, visit:

<https://bit.ly/3QGeoL8>

(for <http://sqlfiddle.com/#!9/cec7edc/1>)

Our friend here shows that you can follow along in Postgres. Just install the database from the text file "302 - ..." available in our sql folder on Canvas

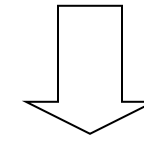


302

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT *  
FROM Product  
WHERE category='Gadgets'
```



Selection

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks

SQL overview

Key constraints

A key is a minimal subset of attributes that acts as a unique identifier for tuples in a relation

- A key is an implicit constraint on which tuples can be in the relation
 - i.e. if two tuples agree on the values of the key, then they must be the same tuple!

```
Students(sid:string, name:string, gpa: float)
```

1. Which would you select as a key?
2. Is a key always guaranteed to exist?
3. Can we have more than one key?

*Think of your NUid as
your surrogate key!*

NULL and NOT NULL

- To say “don’t know the value” we use NULL
 - NULL has (sometimes painful) semantics, more details later

Students(sid:string, name:string, gpa: float)

sid	name	gpa
123	Alice	3.9
143	Bob	NULL

Say, Bob just enrolled in his first class.

In SQL, we may constrain a column to be NOT NULL, e.g., “name” in this table

Can a column with NULLs be a key?

General Constraints

- We can actually specify arbitrary assertions
 - E.g. “There cannot be 25 people in the DB class”
- In practice, we don’t specify many such constraints. Why?
 - Performance!

Whenever we do something ugly (or avoid doing something convenient) it’s for the sake of performance

Summary of Schema Information

- Schema and Constraints are how databases understand the semantics (meaning) of data
- They are also useful for optimization
- SQL supports general constraints:
 - Keys and foreign keys are most important
 - We'll give you a chance to write the others

Basic SQL

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM Product
WHERE manufacturer in ('Canon','Hitachi')
```

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM   Product
WHERE  manufacturer in ('Canon','Hitachi')
```

**Selection
& Projection**

PName
SingleTouch
MultiTouch

WHERE ... IN (...) cp. to Excel

Side-topic

	A	B	C	D	E	F
1	Source: Walkenbach, Excel 2010 Formulas, p396, 2010.					
2						
3	Is this name contained in the range?					
4	Name:	Barbara	TRUE			
5		Betty	1			
6		Betty	TRUE			
7						
8	NameRange					
9						
10	Barbara	Karen	Nancy			
11	Betty	Kimberly	Patricia			
12	Carol	Laura	Ruth			
13	Deborah	Linda	Sandra			
14	Donna	Lisa	Sarah			
15	Dorothy	Margaret	Sharon			
16	Elizabeth	Maria	Susan			
17	Helen	Mary				
18	Jennifer	Michelle				
19						
20						

Assume that there is a range defined for A10:C18 called "NameRange"

Assume that there is
a range defined for
A10:C18 called
"NameRange"

WHERE ... IN (...) cp. to Excel

Side-topic

	A	B	C	D	E	F
1	Source: Walkenbach, Excel 2010 Formulas, p396, 2010.					
2						
3	Is this name contained in the range?					
4	Name:	Barbara	TRUE	{=OR(NameRange=B4)}		
5		Betty	1	=COUNTIF(NameRange,B5)		
6		Betty	TRUE	=COUNTIF(NameRange,B6)>0		
7						
8	NameRange					
9						
10	Barbara	Karen	Nancy			
11	Betty	Kimberly	Patricia			
12	Carol	Laura	Ruth			
13	Deborah	Linda	Sandra			
14	Donna	Lisa	Sarah			
15	Dorothy	Margaret	Sharon			
16	Elizabeth	Maria	Susan			
17	Helen	Mary				
18	Jennifer	Michelle				
19						
20						

Excel's formula bar (Ctrl + F)

Assume that there is a range defined for A10:C18 called "NameRange"

Excel's powerful "array formulas"
(Ctrl + Shift + Enter)

Assume that there is
a range defined for
A10:C18 called
"NameRange"

LIKE

LIKE: Simple String Pattern Matching

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM   Product
WHERE  pname LIKE '%izmo'
```



LIKE: Simple String Pattern Matching

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM   Product
WHERE  pname LIKE '%izmo'
```

PName
Gizmo
Powergizmo

% is a wildcard for any sequence of zero or more characters.

LIKE: Simple String Pattern Matching

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM   Product
WHERE  pname LIKE '_izmo'
```



LIKE: Simple String Pattern Matching

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT pName
FROM   Product
WHERE  pname LIKE '_izmo'
```

PName
Gizmo

'%_kill %'
↑
≥ 1

_ is a wildcard for exactly one character.

Table selection using comparison predicates

	Numbers	Text / Strings
Simple comparators	= = equal to < < smaler than <= <= smaller or equal to > > greater than >= >= greater or equal to <> <> unequal to	= = equal to (exact string)
 Complex comparators	BETWEEN value1 AND value2 any values within the range	LIKE equal to (pattern) 'S%' string starting with S '%S' string ending with S '%S%' string containing an S 'S_S' string with S at both ends and any character in the middle
Comparators that work across types	IN (value1, value2, ...) IS NULL IS NOT NULL	any values within the given set has no value has a value

Note: Combinations of multiple predicates with **AND** & **OR** (use brackets)

- LIKE

- *string* LIKE *pattern*
- *string* NOT LIKE *pattern*

```
'abc' LIKE 'abc'      true
'abc' LIKE 'a%'       true
'abc' LIKE '_b_'      true
'abc' LIKE 'c'        false
```

- SIMILAR TO

- *string* SIMILAR TO *pattern*
- *string* NOT SIMILAR TO *pattern*

```
'abc' SIMILAR TO 'abc'      true
'abc' SIMILAR TO 'a'        false
'abc' SIMILAR TO '%(b|d)%'  true
'abc' SIMILAR TO '(b|c)%'   false
```

- Posix regular expressions

- `substring()`
- `regex_replace()`
- ...

```
'abc' ~ 'abc'      true
'abc' ~ '^a'       true
'abc' ~ '(b|d)'    true
'abc' ~ '^(b|c)'   false
```

You should only know
this exists. You can
always look up details

Schemas and Keys

Product

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

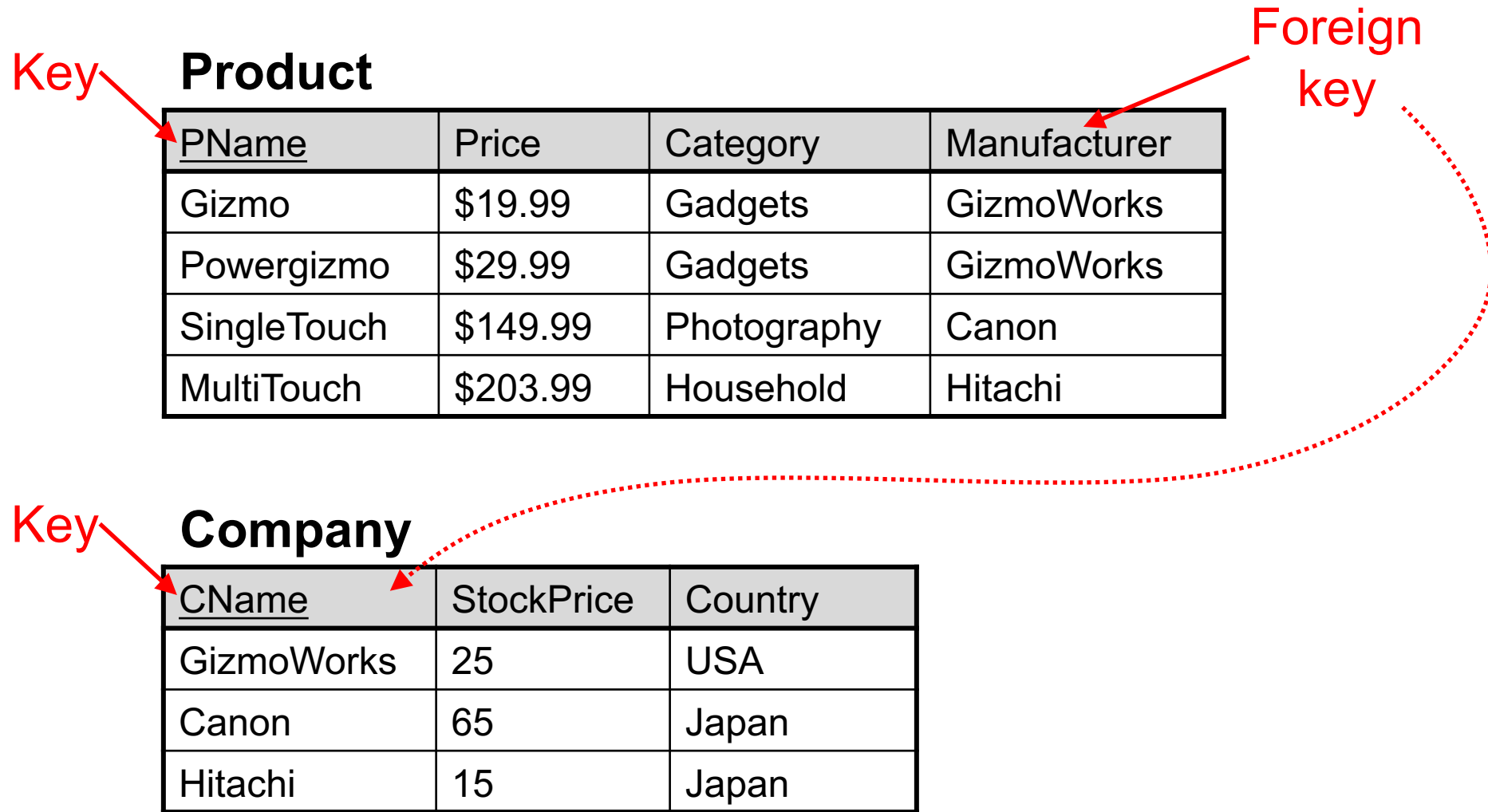
Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What is here
a key vs.
a foreign key?

?

Keys and Foreign Keys



Referential Integrity

Product

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```

graph LR
    subgraph Product
        direction TB
        P1[Gizmo]
        P2[Powergizmo]
        P3[SingleTouch]
        P4[MultiTouch]
    end
    subgraph Company
        direction TB
        C1[GizmoWorks]
        C2[Canon]
        C3[Hitachi]
    end
    P1 --> C1
    P2 --> C2
    P3 --> C2
    P4 --> C3
  
```

Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Foreign key: attribute in a relational table that matches a candidate key of another table

Referential Integrity

Product

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```

graph LR
    subgraph Product
        direction TB
        P1[Gizmo]
        P2[Powergizmo]
        P3[SingleTouch]
        P4[MultiTouch]
    end
    subgraph Company
        direction TB
        C1[GizmoWorks]
        C2[Canon]
        C3[Hitachi]
    end
    P1 --> C1
    P2 --> C2
    P3 --> C2
    P4 --> C3
  
```

Key constraint: minimal subset of the fields of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------



Foreign key: must match field in a relational table that matches a candidate key of another table

Referential Integrity

Product				Company		
<u>PName</u>	Price	Category	Manufacturer	<u>CName</u>	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
SingleTouch	\$149.99	Photography	Canon	Hitachi	15	Japan
MultiTouch	\$203.99	Household	Hitachi			

Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------

tuple violates key constraint

Foreign key: attribute in a relational table that matches a candidate key of another table

Referential Integrity

Product				Company		
<u>PName</u>	Price	Category	Manufacturer	<u>CName</u>	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
SingleTouch	\$149.99	Photography	Canon	Hitachi	15	Japan
MultiTouch	\$203.99	Household	Hitachi			

Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------

tuple violates key constraint

Foreign key: attribute in a relational table that matches a candidate key of another table

Insert into Product values ('SuperTouch', 249.99, 'Computer', 'NewCom');

SuperTouch	\$249.99	Computer	NewCom
------------	----------	----------	--------



Referential Integrity

Product				Company		
<u>PName</u>	Price	Category	Manufacturer	<u>CName</u>	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
SingleTouch	\$149.99	Photography	Canon	Hitachi	15	Japan
MultiTouch	\$203.99	Household	Hitachi			

Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------

tuple violates key constraint

Foreign key: attribute in a relational table that matches a candidate key of another table

Insert into Product values ('SuperTouch', 249.99, 'Computer', 'NewCom');

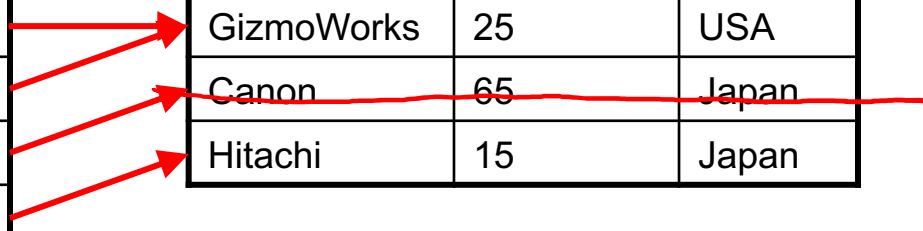
SuperTouch	\$249.99	Computer	NewCom
------------	----------	----------	--------

tuple violates
foreign key constraint

Referential Integrity

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan



Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------

tuple violates key constraint

Foreign key: attribute in a relational table that matches a candidate key of another table

Insert into Product values ('SuperTouch', 249.99, 'Computer', 'NewCom');

SuperTouch	\$249.99	Computer	NewCom
------------	----------	----------	--------

tuple violates foreign key constraint

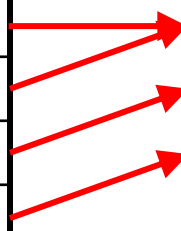
Delete from Company
where CName = 'Canon';



Referential Integrity

<u>PName</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan



Key constraint: minimal subset of the attributes of a relation is a unique identifier for a tuple.

Insert into Product values ('Gizmo', 14.99, 'Gadgets', 'Hitachi');

Gizmo	\$14.99	Gadgets	Hitachi
-------	---------	---------	---------

tuple violates key constraint

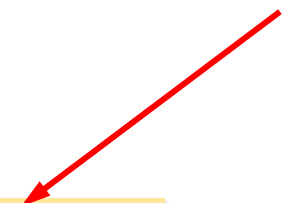


Foreign key: attribute in a relational table that matches a candidate key of another table

Insert into Product values ('SuperTouch', 249.99, 'Computer', 'NewCom');

SuperTouch	\$249.99	Computer	NewCom
------------	----------	----------	--------

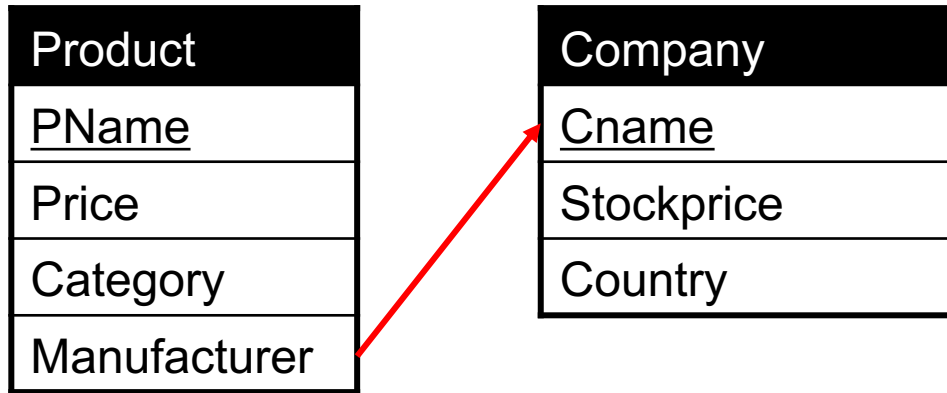
tuple violates foreign key constraint



Delete from Company
where CName = 'Canon';

Joins

(Relational Database) Schema



"Schema": describes the structure of data in terms of the relational data model.

A schema includes tables, columns, PKs, FKs, and other constraints

Product(pname, price, category, manufacturer)

Company(cname, stockprice, country)

Product.manufacturer is FK to Company

Schema specification in SQL



[northeastern-datalab](#) / [cs3200-activities](#) Public

[Code](#) [Issues](#) [Pull requests](#) 1 [Actions](#) [Projects](#) [...](#)

master ▾

[cs3200-activities](#) / [sql](#) /

wolfandthegang ...	on May 23 🕒
..	
300-SmallIMDB.txt	4 months ago
302-Simpleproducts.txt	4 months ago
304-Worker.txt	4 months ago
305-Conceptualevaluationstrategy.txt	4 months ago
306-NestedLoopJoin.py	10 months ago
308-Purchase.txt	4 months ago

```
-----  
-- Create the tables  
-----  
  
create table Company (  
    CName char(20) PRIMARY KEY,  
    StockPrice int,  
    Country char(20) );  
  
create table Product (  
    PName char(20),  
    Price decimal(9, 2),  
    Category char(20),  
    Manufacturer char(20),  
    PRIMARY KEY (PName),  
    FOREIGN KEY (Manufacturer) REFERENCES Company(CName) );  
  
-----  
-- Populate the tables  
-----  
  
insert into Company values ('GizmoWorks', 25, 'USA');  
insert into Company values ('Canon', 65, 'Japan');  
insert into Company values ('Hitachi', 15, 'Japan');  
  
insert into Product values ('Gizmo', 19.99, 'Gadgets', 'GizmoWorks');  
insert into Product values ('PowerGizmo', 29.99, 'Gadgets', 'GizmoWorks');
```

Example SQL code and database: <https://github.com/northeastern-datalab/cs3200-activities/tree/master/sql>

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

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

*Q: Find all products under \$200 manufactured in Japan;
return their names and prices!*

?

Product				Company		
PName	Price	Category	Manufacturer	CName	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
SingleTouch	\$149.99	Photography	Canon	Hitachi	15	Japan
MultiTouch	\$203.99	Household	Hitachi			

*Q: Find all products under \$200 manufactured in Japan;
return their names and prices!*

```

SELECT pName, price
FROM Product, Company
WHERE manufacturer=cName
      and country='Japan'
      and price <= 200
    
```

Join b/w Product
and Company

PName	Price
SingleTouch	\$149.99

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What does the query below return?

```
SELECT pName, StockPrice
FROM Product, Company
WHERE manufacturer=cName
      and country = 'USA'
```



?

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What does the query below return?

```
SELECT pName, StockPrice
FROM Product, Company
WHERE manufacturer=cName
      and country = 'USA'
```



PName	StockPrice
Gizmo	25
Powergizmo	25

Table Alias (Tuple Variables)

Person (pName, address, works_for)
University (uName, address)

```
SELECT DISTINCT pName, address  
FROM Person, University  
WHERE works_for = uName
```

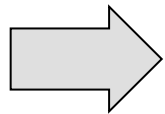
What will this
query return ?

Table Alias (Tuple Variables)

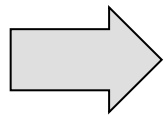
Person (pName, address, works_for)
University (uName, address)

```
SELECT DISTINCT pName, address
FROM    Person, University
WHERE   works_for = uName
```

*which address?
Error!*



```
SELECT DISTINCT Person.pName, University.address
FROM    Person, University
WHERE   Person.works_for = University.uName
```

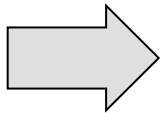


```
SELECT DISTINCT X.pName, Y.address
FROM    Person as X, University Y
WHERE   X.works_for = Y.uName
```

Notice that the use of "as" is not necessary, it is optional !!

Column Alias (rename attributes)

Person (pName, address, works_for)
University (uName, address)



```
SELECT DISTINCT X.pName as name, Y.address adr
FROM      Person as X, University Y
WHERE     X.works_for = Y.uName
```

```
SELECT DISTINCT X.pName, Y.address
FROM      Person as X, University Y
WHERE     X.works_for = Y.uName
```

Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Q: Find all US companies that manufacture products in the 'Gadgets' category!

```
SELECT cName  
FROM  
WHERE
```



?

Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Q: Find all US companies that manufacture products in the 'Gadgets' category!

```
SELECT cName
FROM Product P, Company
WHERE country = 'USA'
      and P.category = 'Gadgets'
      and P.manufacturer = cName
```



Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Q: Find all US companies that manufacture products in the 'Gadgets' category!

```
SELECT cName
FROM Product P, Company
WHERE country = 'USA'
      and P.category = 'Gadgets'
      and P.manufacturer = cName
```



Cname
GizmoWorks
GizmoWorks

Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Q: Find all US companies that manufacture products in the 'Gadgets' category!

```
SELECT DISTINCT cName
FROM   Product P, Company C
WHERE  C.country = 'USA'
      and P.category = 'Gadgets'
      and P.manufacturer = cName
```



Cname
GizmoWorks

Joins can introduce duplicates → remember to use DISTINCT!

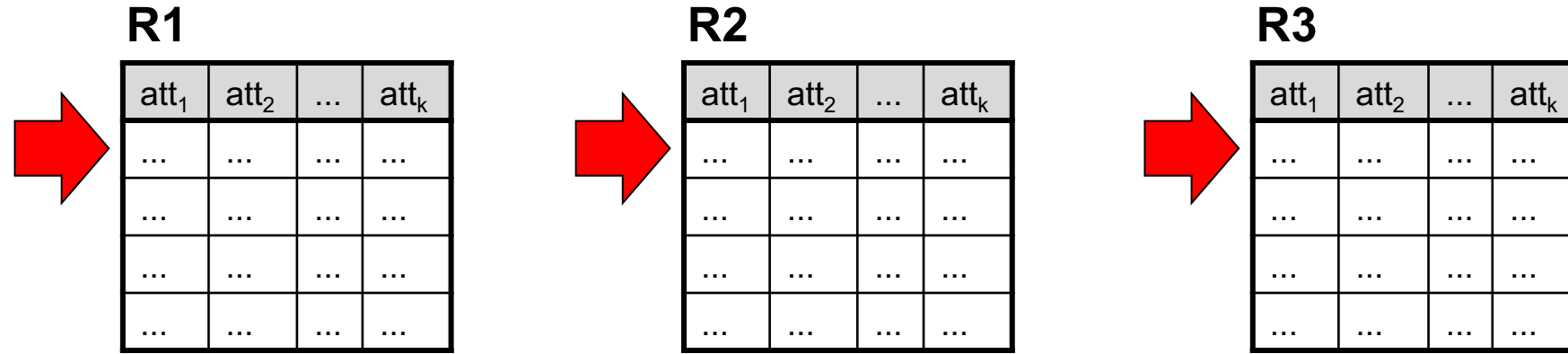
Meaning (Semantics) of conjunctive SQL Queries

```
SELECT a1, a2, ..., ak  
FROM   R1 as x1, R2 as x2, ..., Rn as xn  
WHERE  Conditions
```

Conceptual evaluation strategy (nested for loops):

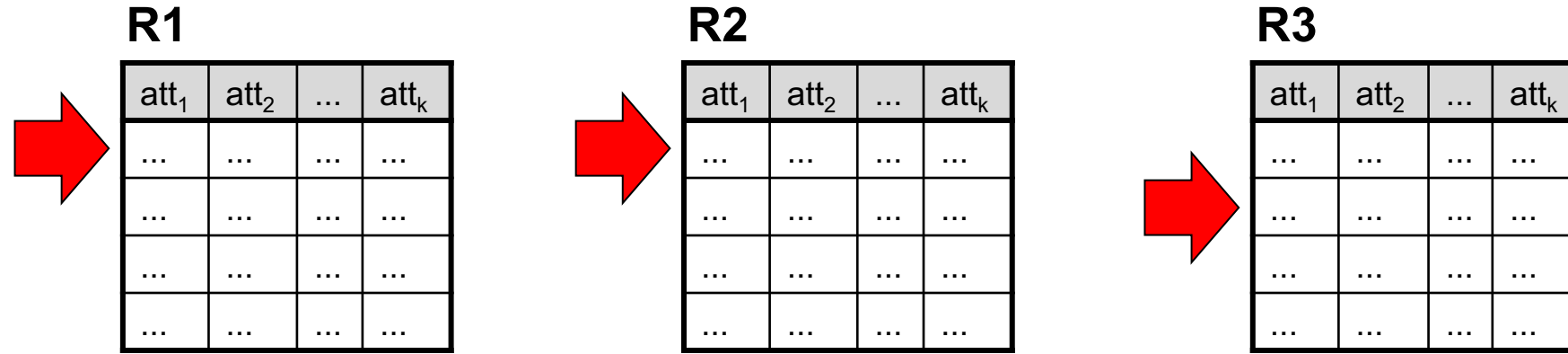
```
Answer = {}  
for x1 in R1 do  
  for x2 in R2 do  
    ....  
    for xn in Rn do  
      if Conditions  
        then Answer = Answer ∪ {(a1, ..., ak)}  
return Answer
```

Meaning (Semantics) of conjunctive SQL Queries



```
Answer = {}  
for x1 in R1 do  
  for x2 in R2 do  
    .....  
    for xn in Rn do  
      if Conditions  
        then Answer = Answer ∪ {(a1, ..., ak)}  
return Answer
```

Meaning (Semantics) of conjunctive SQL Queries



```
Answer = {}  
for x1 in R1 do  
  for x2 in R2 do  
    .....  
    for xn in Rn do  
      if Conditions  
        then Answer = Answer ∪ {(a1, ..., ak)}  
return Answer
```

Conceptual Evaluation Strategy

- Semantics of an SQL query defined in terms of the following conceptual evaluation strategy:
 - **FROM**: Compute the cross-product of relation-list.
 - **WHERE**: Discard resulting tuples if they fail qualifications ("select" the rest)
 - **SELECT**: Delete attributes that are not in target-list.
 - If **DISTINCT** is specified, eliminate duplicate rows.
- This strategy is probably the least efficient way to compute a query! An optimizer will find more efficient strategies to compute the same answers.
- Quiz: we say “semantics” not “execution order”. Why?



Conceptual Evaluation Strategy

- Semantics of an SQL query defined in terms of the following conceptual evaluation strategy:
 - **FROM**: Compute the cross-product of relation-list.
 - **WHERE**: Discard resulting tuples if they fail qualifications ("select" the rest)
 - **SELECT**: Delete attributes that are not in target-list.
 - If **DISTINCT** is specified, eliminate duplicate rows.
- This strategy is probably the least efficient way to compute a query! An optimizer will find more efficient strategies to compute the same answers.
- Quiz: we say “semantics” not “execution order”. Why?
 - The preceding slides show **what** a join means (**semantics = meaning**): "the logic"
 - Not actually **how** the DBMS actually executes it (separation of concerns): algebra

Our colorful hands represent "team exercises"
If we are online, please make a screenshot!



Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Q: Find all US companies that manufacture both a product below \$20 and a product above \$25.

```
SELECT DISTINCT cName  
FROM  
WHERE
```