61A Lecture 33

Monday, April 20

Announcements

Announcements

- Course survey due Monday 4/20 @ 11:59pm

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

- Project 4 due Thursday 4/23 @ 11:59pm

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

- Project 4 due Thursday 4/23 @ 11:59pm
  - Early point #2: All questions (including Extra Credit) by Wednesday 4/22 @ 11:59pm

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

- Project 4 due Thursday 4/23 @ 11:59pm
  - Early point #2: All questions (including Extra Credit) by Wednesday 4/22 @ 11:59pm
- Recursive Art Contest Entries due Monday 4/27 @ 11:59pm

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

- Project 4 due Thursday 4/23 @ 11:59pm
  - Early point #2: All questions (including Extra Credit) by Wednesday 4/22 @ 11:59pm
- Recursive Art Contest Entries due Monday 4/27 @ 11:59pm
  - Email your code & a screenshot of your art to cs61a-tae@imail.eecs.berkeley.edu (Albert)

Announcements

- Course survey due Monday 4/20 @ 11:59pm
- If 85% of students complete the course survey on resources, everyone gets 1 bonus point!

http://goo.gl/ajEBkT

- Project 4 due Thursday 4/23 @ 11:59pm
  - Early point #2: All questions (including Extra Credit) by Wednesday 4/22 @ 11:59pm
- Recursive Art Contest Entries due Monday 4/27 @ 11:59pm
  - Email your code & a screenshot of your art to cs61a-tae@imail.eecs.berkeley.edu (Albert)
- Homework 9 merged with Homework 10; both are due Wednesday 4/29 @ 11:59pm

Local Tables

Local Tables

A `create table` statement names a table globally

Local Tables

A `create table` statement names a table globally

```
create table parents as
  select "abraham" as parent, "barack" as child union
  select "abraham"          , "clinton"         union
  select "delano"           , "herbert"         union
  select "fillmore"         , "abraham"         union
  select "fillmore"         , "delano"          union
  select "fillmore"         , "grover"          union
  select "eisenhower"       , "fillmore";
```

Local Tables

A `create table` statement names a table globally

```
create table parents as
  select "abraham" as parent, "barack" as child union
  select "abraham"          , "clinton"         union
  select "delano"           , "herbert"         union
  select "fillmore"         , "abraham"         union
  select "fillmore"         , "delano"          union
  select "fillmore"         , "grover"          union
  select "eisenhower"       , "fillmore";
```

parents:

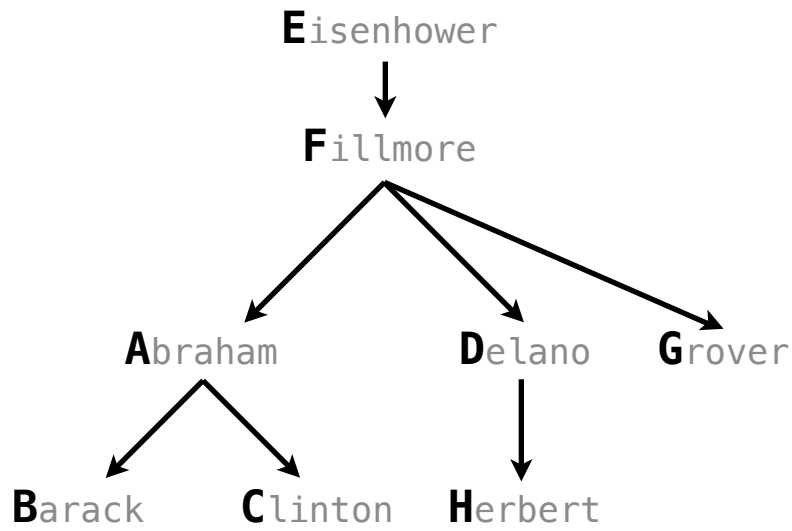
Parent	Child
abraham	barack
abraham	clinton
delano	herbert
fillmore	abraham
fillmore	delano
fillmore	grover
eisenhower	fillmore

Local Tables

A `create table` statement names a table globally

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:



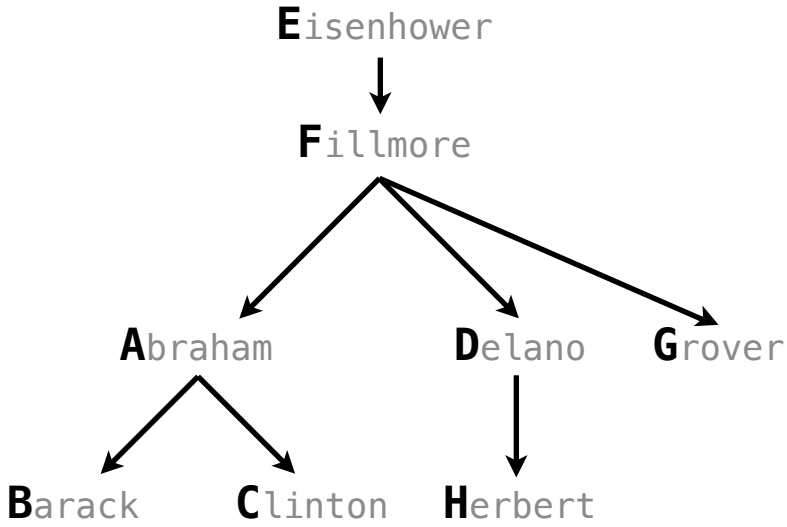
Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:



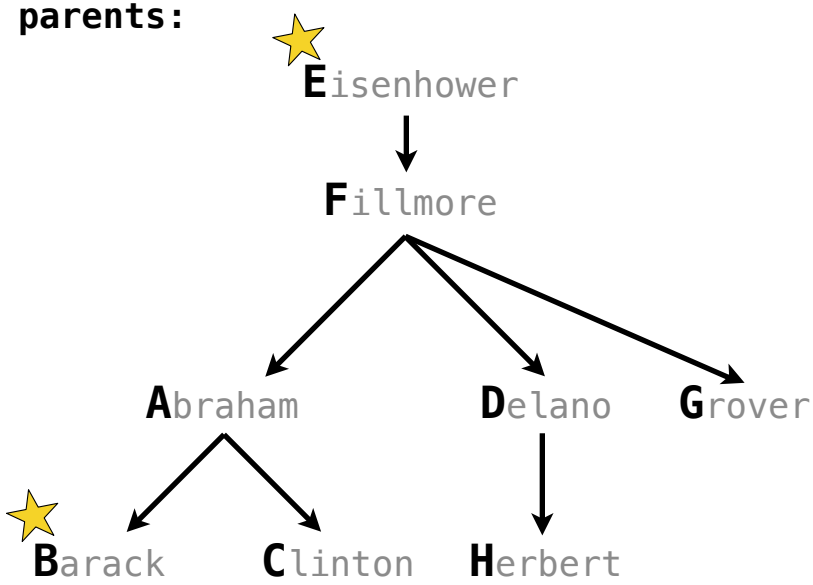
Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:



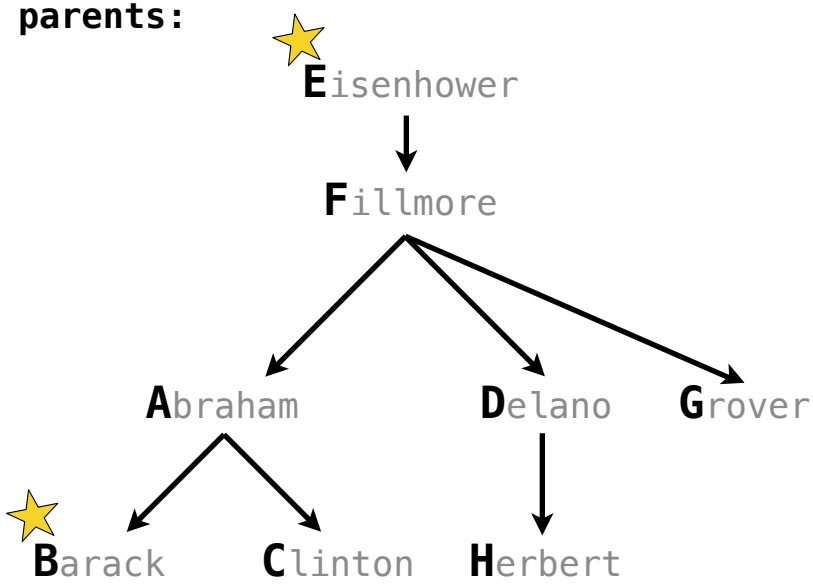
Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:




```
select parent from ...
```

Local Tables

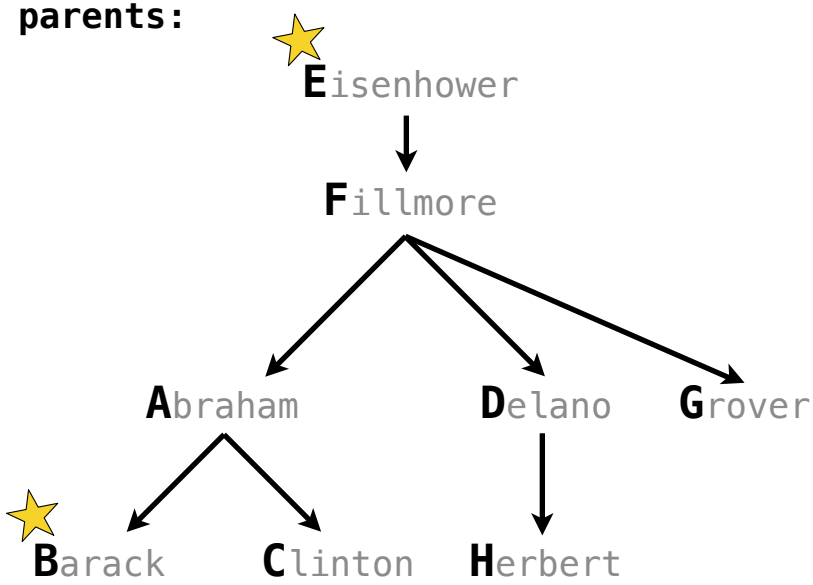
A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
with

select parent from ...
```

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

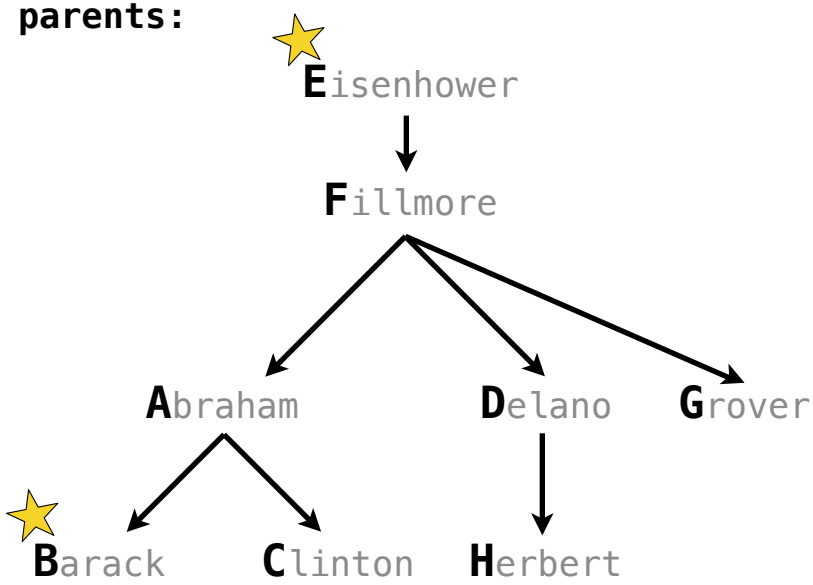
```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (



select parent from ...
```

parents:



Local Tables

A `create table` statement names a table globally

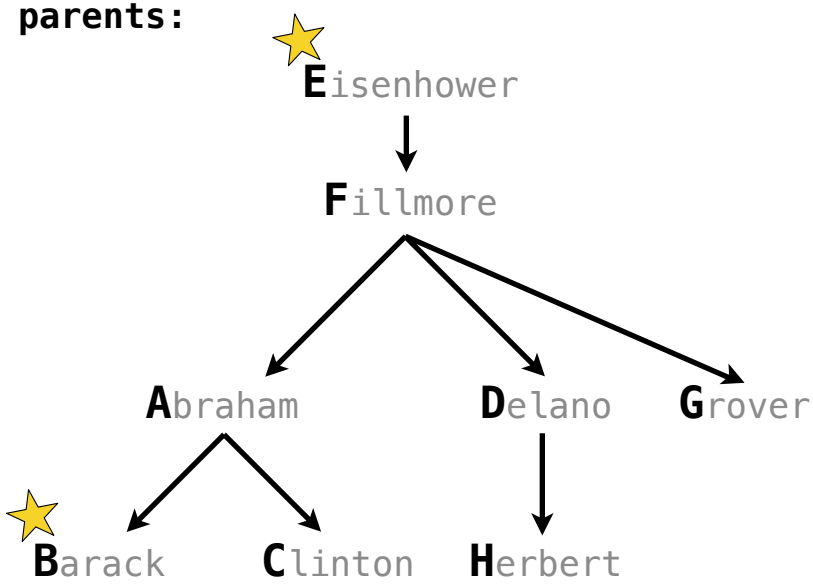
A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (
    select "eisenhower" union


select parent from ...
```

parents:



Local Tables

A `create table` statement names a table globally

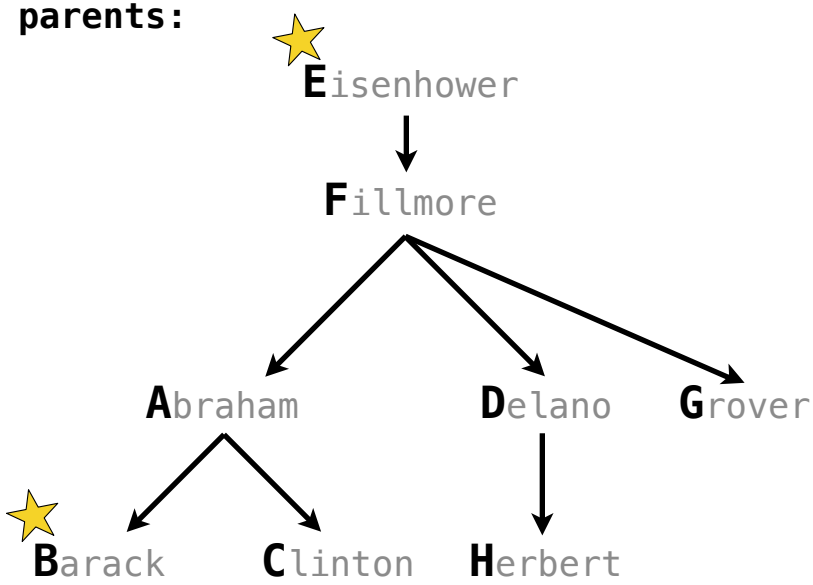
A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (
    select "eisenhower" union
    select "barack"

select parent from ...
```

parents:



Local Tables

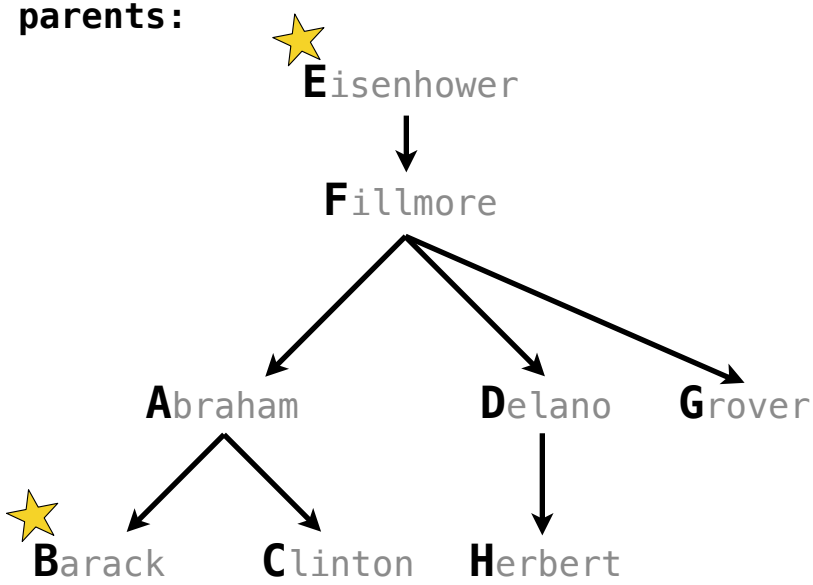
A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from ...
```

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

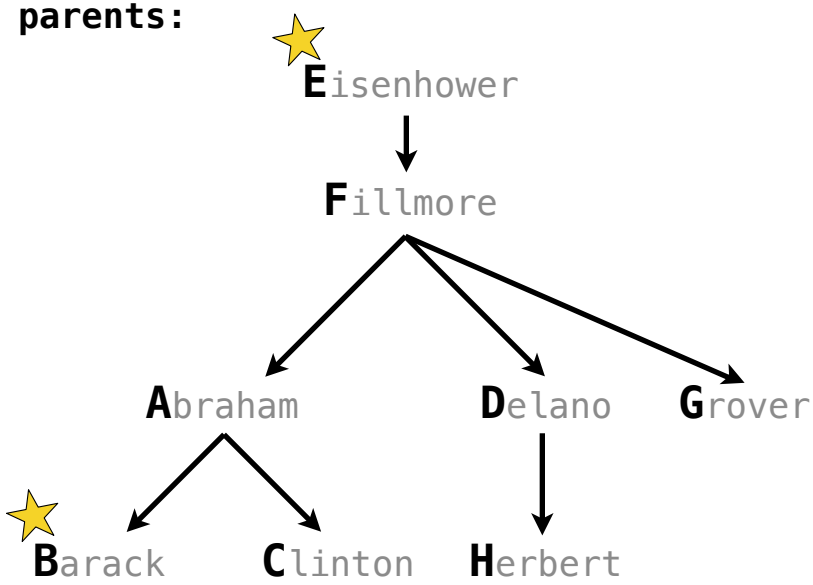
```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from ...
```

best:

dog
eisenhower
barack

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

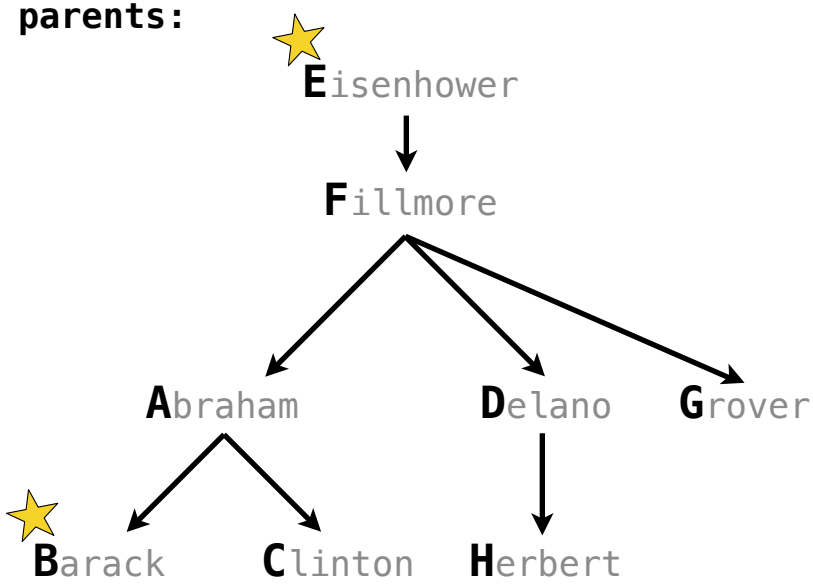
```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from parents, best where child=dog;
```

best:

dog
eisenhower
barack

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

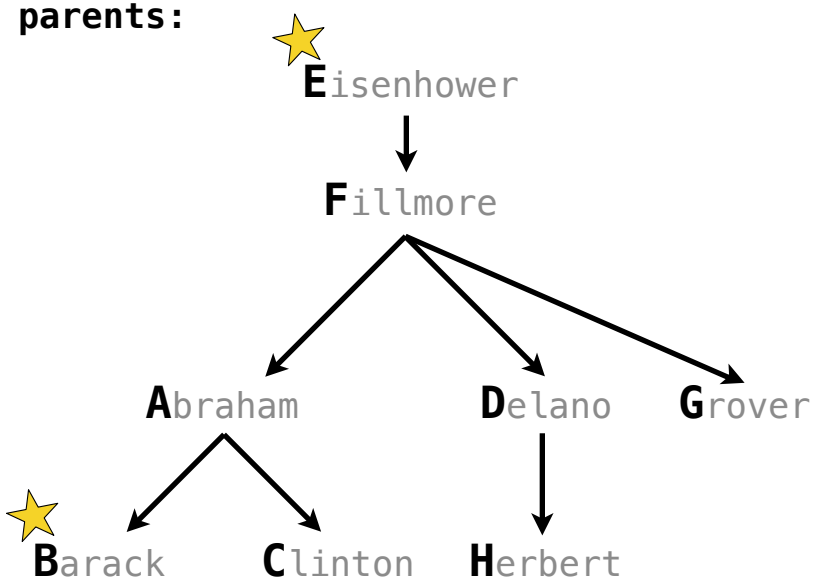
```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from parents, best where child=dog;
```

best:

dog
eisenhower
barack

parent
abraham

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from parents, best where child=dog;
```

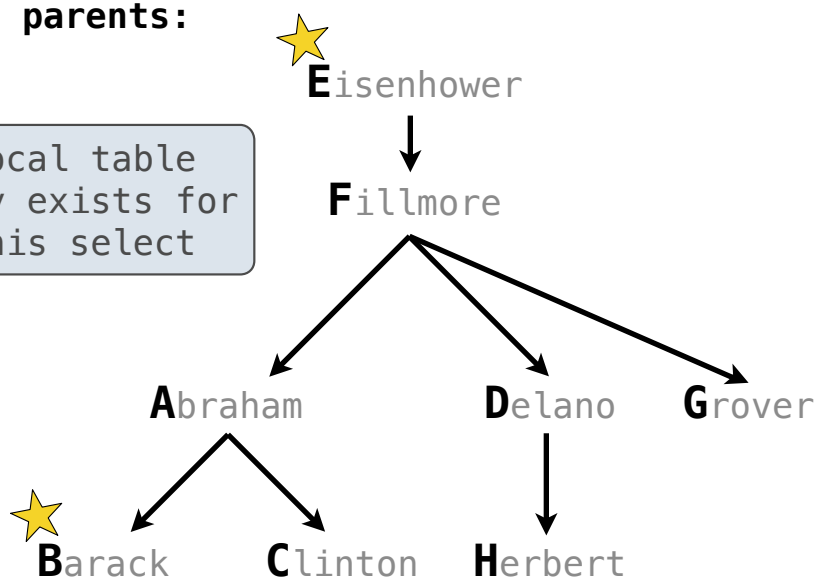
best:

dog
eisenhower
barack

Local table only exists for this select

parent
abraham

parents:



Local Tables

A create table statement names a table globally

A with clause of a select statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from parents, best where child=dog;
```

Part of the select statement

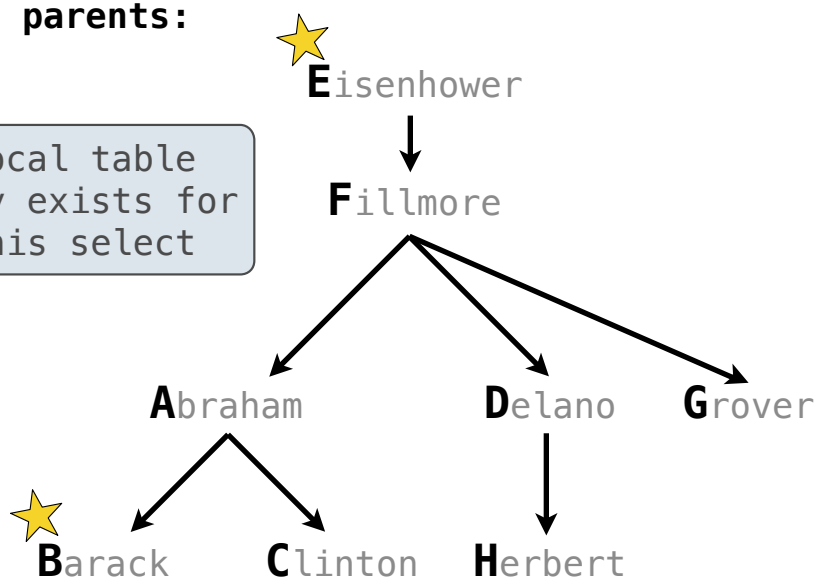
Local table only exists for this select

best:

dog
eisenhower
barack

parent
abraham

parents:



Local Tables

A `create table` statement names a table globally

A `with` clause of a `select` statement names a table that is local to the statement

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
with
  best(dog) as (
    select "eisenhower" union
    select "barack"
  )
select parent from parents, best where child=dog;
```

Part of the select statement

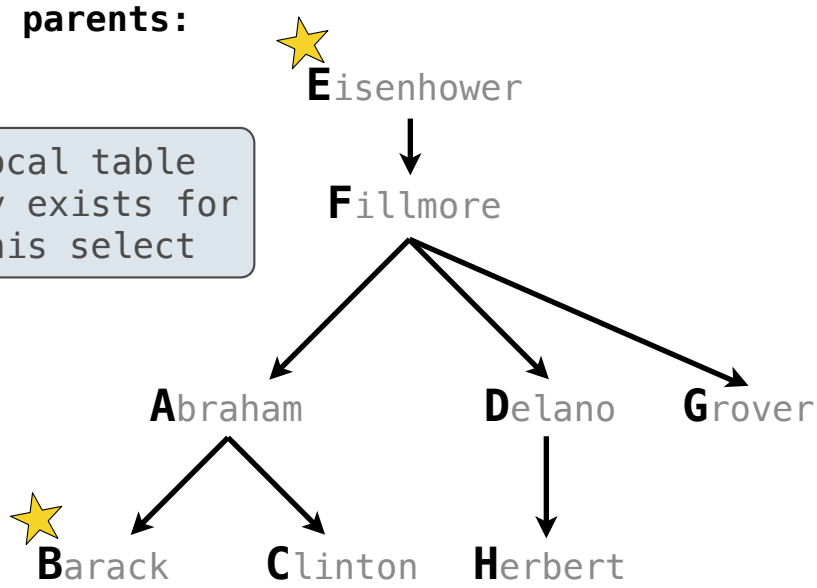
Local table only exists for this select

best:

dog
eisenhower
barack

parent
abraham

parents:

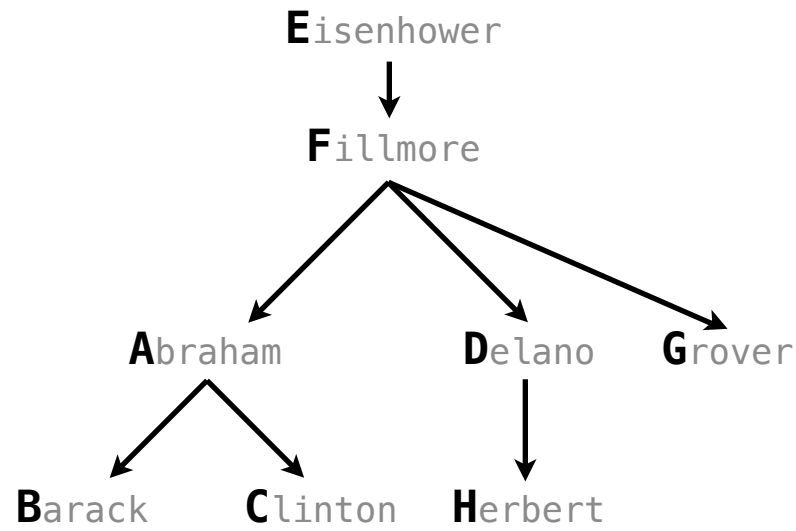


(Demo)

Example: Relationships

```
with
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

parents:

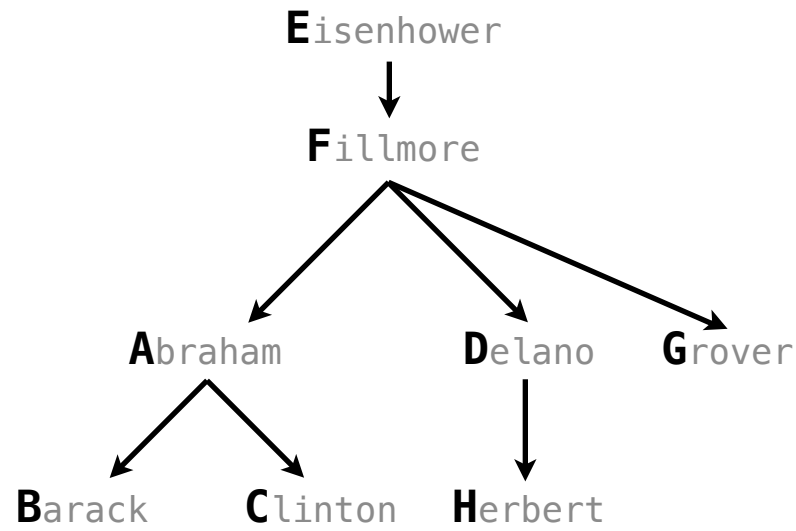


Example: Relationships

(A) What are appropriate names for the columns in this result?

```
with
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

parents:



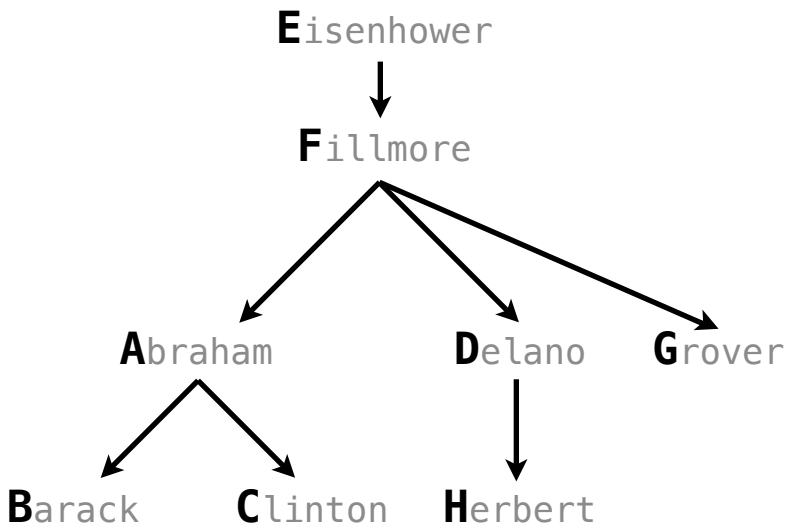
Example: Relationships

(A) What are appropriate names for the columns in this result?

(B) How many rows will result?

```
with
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

parents:



Example: Relationships

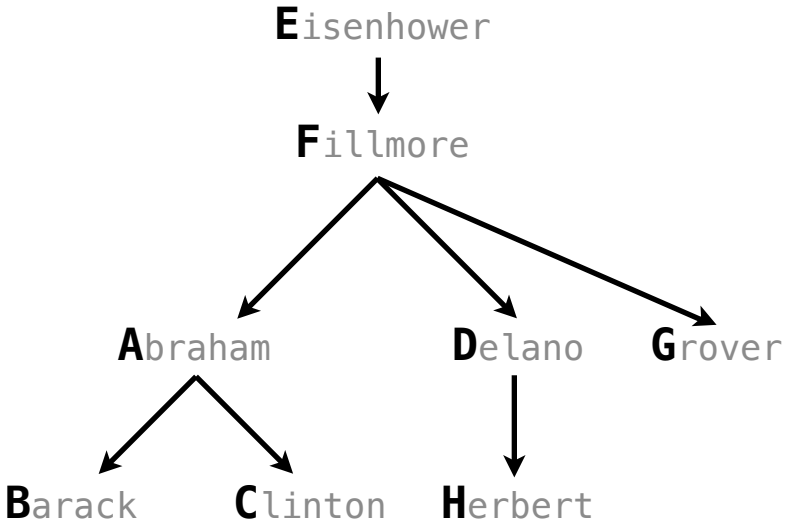
(A) What are appropriate names for the columns in this result?

(B) How many rows will result?

```
with
 siblings
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

siblings

parents:



Example: Relationships

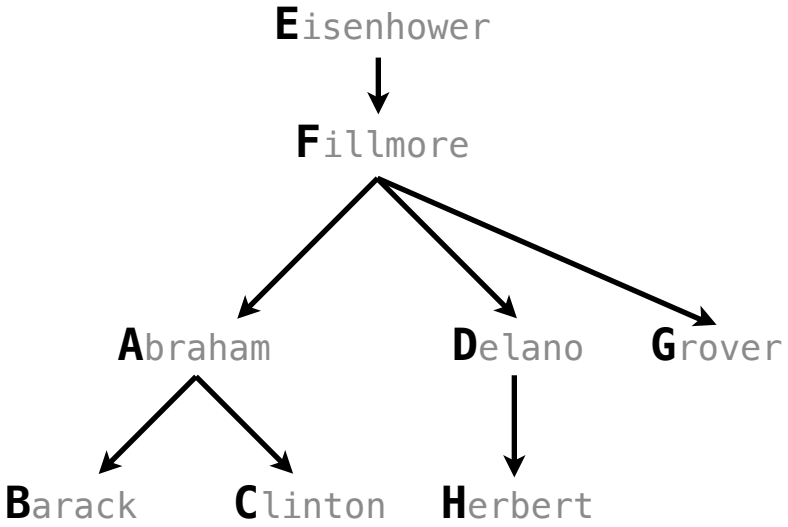
(A) What are appropriate names for the columns in this result?

(B) How many rows will result?

```
with
 siblings
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

siblings

parents:



parent	child	first	second
abraham	barack	abraham	delano

Example: Relationships

(A) What are appropriate names for the columns in this result?

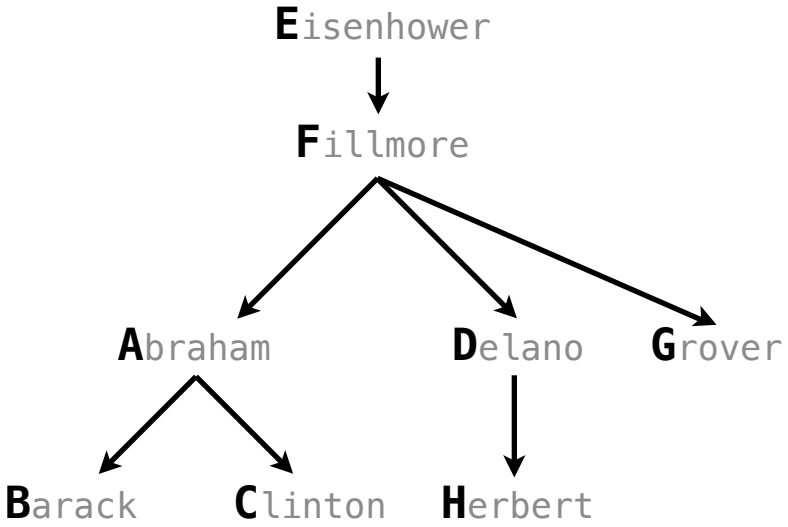
(B) How many rows will result?

```
with
 siblings
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____________, second as _____________
       from parents, what where parent=first;
```

siblings

parent	child	first	second
abraham	barack	abraham	delano

parents:



Example: Relationships

(A) What are appropriate names for the columns in this result?

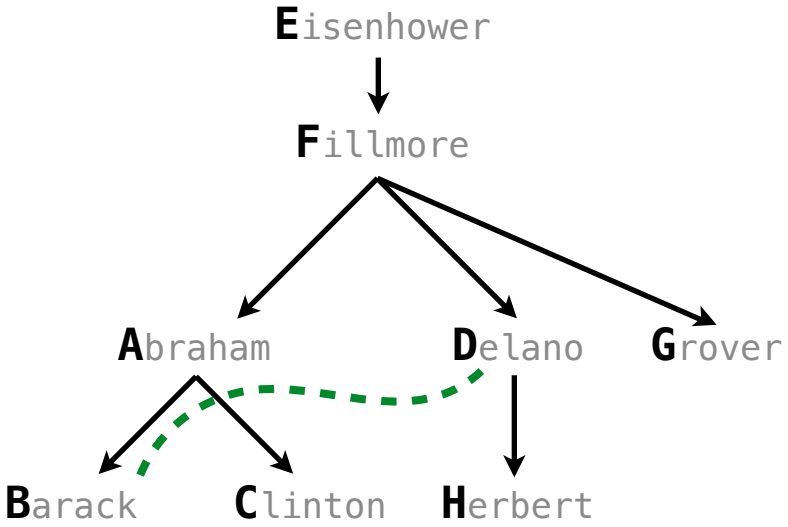
(B) How many rows will result?

```
with
 siblings
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____, second as _____
       from parents, what where parent=first;
```

siblings

parent	child	first	second
abraham	barack	abraham	delano

parents:



Example: Relationships

(A) What are appropriate names for the columns in this result?

(B) How many rows will result?

```
with
 siblings
  ~~what~~(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____, second as _____
       from parents, ~~what~~ where parent=first;
```

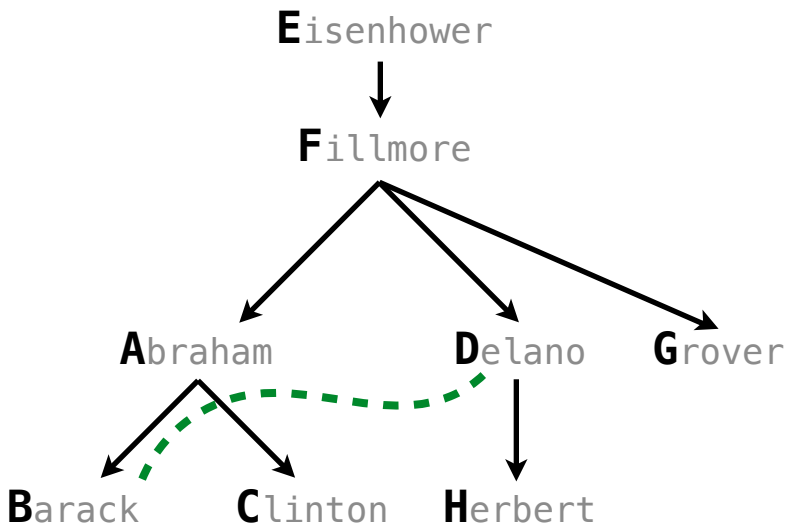
nephew

siblings

nephew

parent	~~child~~	first	second
abraham	barack	abraham	delano

parents:



Example: Relationships

(A) What are appropriate names for the columns in this result?

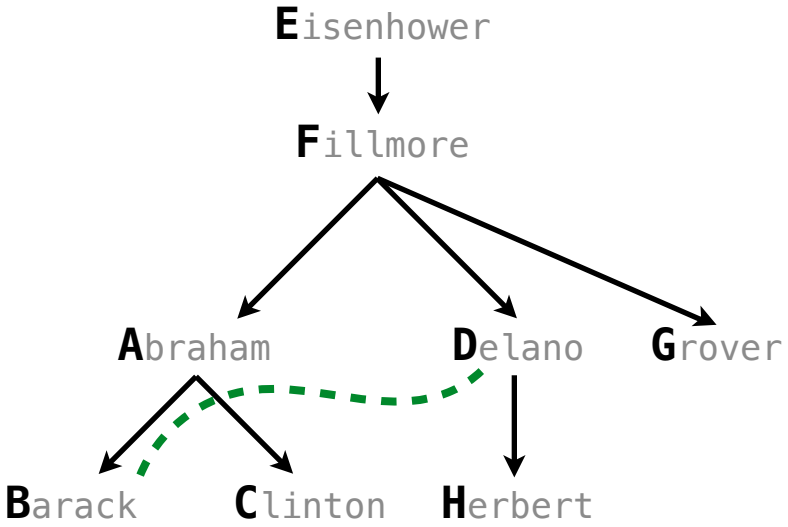
(B) How many rows will result?

```
with
 siblings
  what(first, second) as (
    select a.child, b.child
           from parents as a, parents as b
           where a.parent = b.parent and
                 a.child != b.child
  )
select child as _____, second as _____
       from parents, what where parent=first;
```

nephew (siblings) uncle

parent	~~child~~	first	~~second~~
abraham	barack	abraham	delano

parents:



Recursive Local Tables

Local Tables can be Declared Recursively

Local Tables can be Declared Recursively

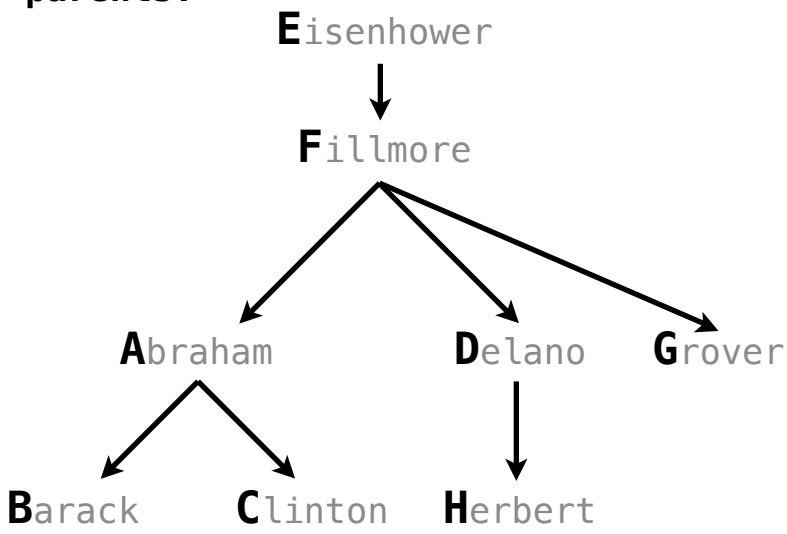
```
An ancestor is your parent or an ancestor of your parent
```

Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:

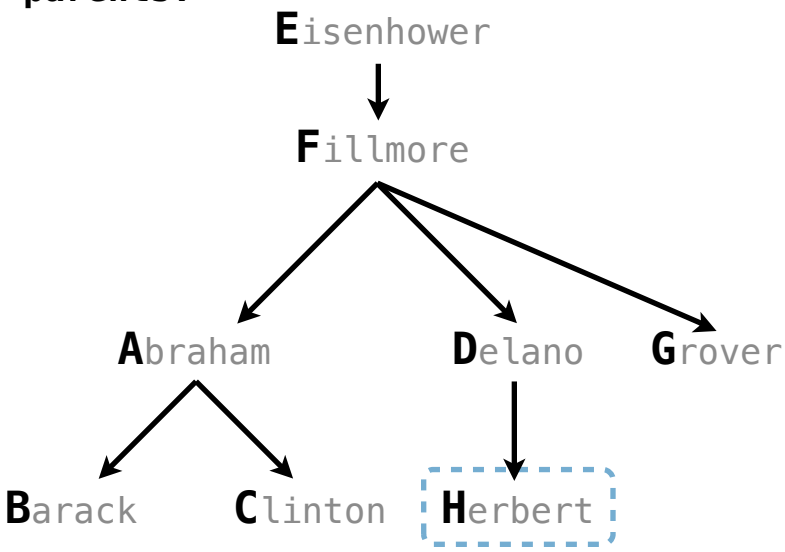


Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

parents:



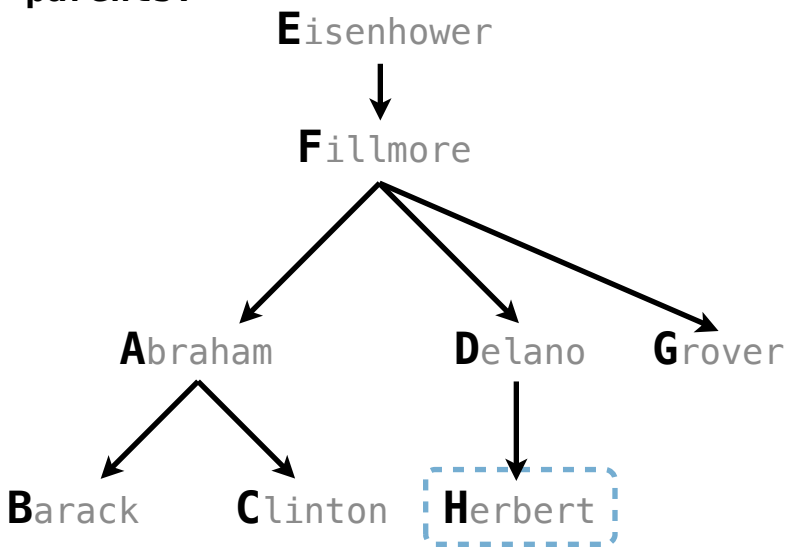
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...
```

```
  ancestors(ancestor, descendent)
```

parents:



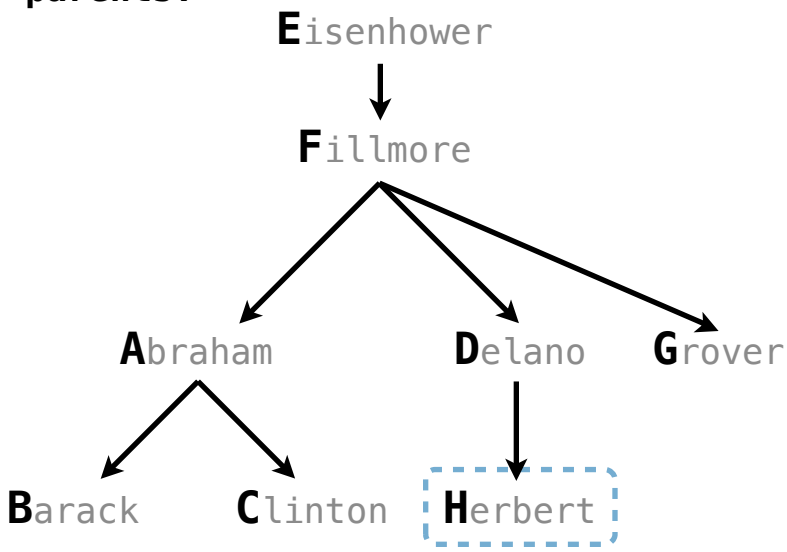
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...


  ancestors(ancestor, descendent) as (
    select parent, child from parents union
    select ancestor, child
           from ancestors, parents
           where parent = descendent
  )
```

parents:



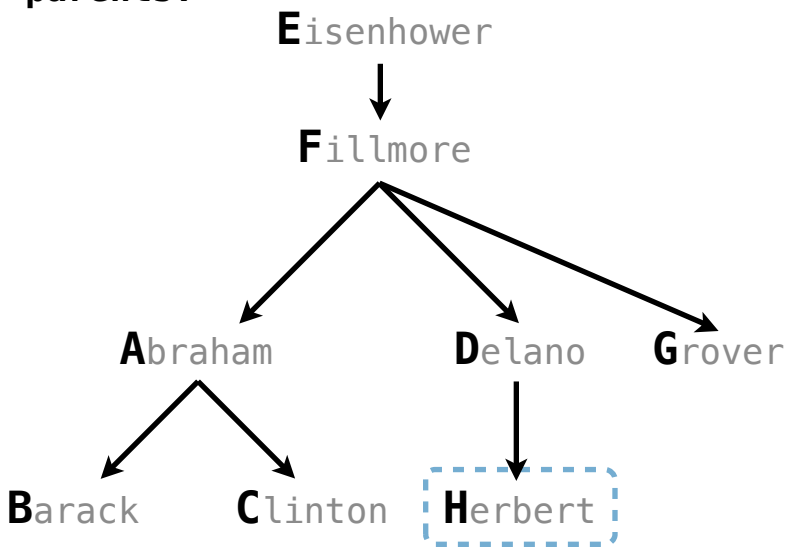
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  ancestors(ancestor, descendent) as (
    select parent, child from parents union
    select ancestor, child
           from ancestors, parents
           where parent = descendent
  )
select ancestor from ancestors where descendent="herbert";
```

parents:



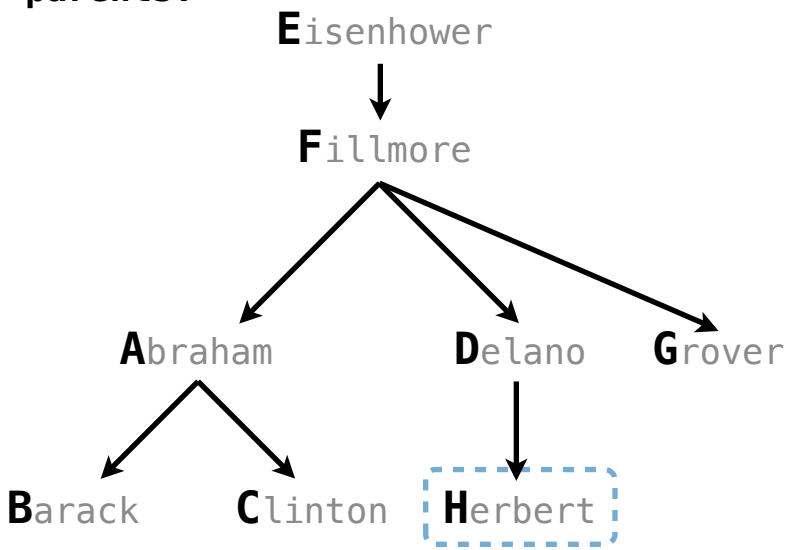
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```
create table parents as
  select "abraham" as parent, "barack" as child union
  ...

with
  ancestors(ancestor, descendent) as (
    select parent, child from parents union
    select ancestor, child
           from ancestors, parents
           where parent = descendent
  )
select ancestor from ancestors where descendent="herbert";
```

parents:



ancestor
delano
fillmore
eisenhower

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

```
create table odds as
  with
    odds(n) as (
      select 1 union
      select n+2 from odds where n < 15
    )
  select n from odds;
```

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

```
create table odds as
  with
    odds(n) as (
      select 1 union
      select n+2 from odds where n < 15
    )
  select n from odds;
```

odds:

n
1
3
5
7
9
11
13
15

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

```
create table odds as
  with
    odds(n) as (
      select 1 union
      select n+2 from odds where n < 15
    )
  select n from odds;
```

odds:

n
1
3
5
7
9
11
13
15

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

```
create table odds as
  with
    odds(n) as (
      select 1 union
      select n+2 from odds where n < 15
    )
  select n from odds;
```

odds:

n
1
3
5
7
9
11
13
15

Which names above can change without affecting the result?

Global Names for Recursive Tables

To create a table with a global name, you need to select the contents of the local table

```
create table odds as
  with
    odds(n) as (
      select 1 union
      select n+2 from odds where n < 15
    )
  select n from odds;
```

odds:

n
1
3
5
7
9
11
13
15

Which names above can change without affecting the result?

Limits on Recursive Select Statements

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

No tree recursion: the table being defined can only appear once in a from clause

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

No tree recursion: the table being defined can only appear once in a from clause

```
with
  ints(x) as (
    select 1 union
    select x-1 from ints union
    select x+1 from ints
  )
select x from ints;
```

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

No tree recursion: the table being defined can only appear once in a from clause

Nope!

```
with
  ints(x) as (
    select 1 union
    select x-1 from ints union
    select x+1 from ints
  )
select x from ints;
```

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

No tree recursion: the table being defined can only appear once in a from clause

Nope!

```
with
  ints(x) as (
    select 1 union
    select x-1 from ints union
    select x+1 from ints
  )
select x from ints;
```

```
with
  ints(x) as (
    select 1 union
    select a.x + b.x
           from ints as a, ints as b
  )
select x from ints;
```

Limits on Recursive Select Statements

Recursive table definitions are only possible within a with clause

No mutual recursion: two or more tables cannot be defined in terms of each other

Nope!

```
with
  odds(x) as (
    select 1 union select x+1 from evens
  ),
  evens(x) as (
    select x+1 from odds
  )
select x from odds
```

No tree recursion: the table being defined can only appear once in a from clause

Nope!

```
with
  ints(x) as (
    select 1 union
    select x-1 from ints union
    select x+1 from ints
  )
select x from ints;
```

Nope!

```
with
  ints(x) as (
    select 1 union
    select a.x + b.x
           from ints as a, ints as b
  )
select x from ints;
```

String Examples

Language is Recursive

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

The dog chased the cat

that the bird chased

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

The dog chased the cat

that the bird chased

The dog chased the cat

the bird chased

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

The dog chased the cat

that the bird chased

The dog chased the cat

the bird chased

The dog the bird the cat chased chased chased me

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

The dog chased the cat

that the bird chased

The dog chased the cat

the bird chased

The dog the bird the cat chased chased chased me

Bulldogs bulldogs bulldogs fight fight fight

Language is Recursive

Noun phrases can contain relative pronouns that introduce relative clauses

The dog chased the cat

that chased the bird

The dog chased the cat

that the bird chased

The dog chased the cat

the bird chased

The dog the bird the cat chased chased chased me

Bulldogs bulldogs bulldogs fight fight fight

(Demo)

Integer Examples

Input-Output Tables

A table containing the inputs to a function can be used to map from output to input

Input-Output Tables

A table containing the inputs to a function can be used to map from output to input

```
create table pairs as
  with
    i(n) as (
      select 1 union
      select n+1 from i where n < 50
    )
  select a.n as x, b.n as y from i as a, i as b where a.n <= b.n;
```

Input-Output Tables

A table containing the inputs to a function can be used to map from output to input

```
create table pairs as
  with
    i(n) as (
      select 1 union
      select n+1 from i where n < 50
    )
  select a.n as x, b.n as y from i as a, i as b where a.n <= b.n;
```

What integers can I add/multiply together to get 24?

Input-Output Tables

A table containing the inputs to a function can be used to map from output to input

```
create table pairs as
  with
    i(n) as (
      select 1 union
      select n+1 from i where n < 50
    )
  select a.n as x, b.n as y from i as a, i as b where a.n <= b.n;
```

What integers can I add/multiply together to get 24?

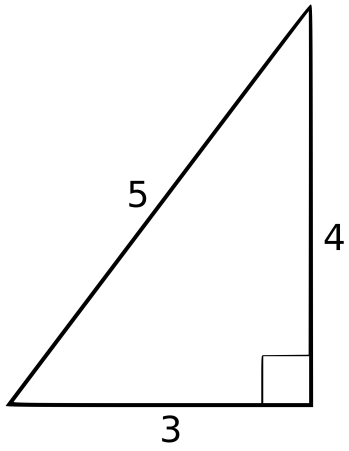
(Demo)

Example: Pythagorean Triples

All triples a, b, c such that $a^2 + b^2 = c^2$

Example: Pythagorean Triples

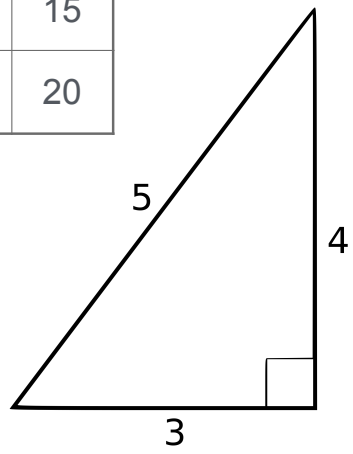
All triples a, b, c such that $a^2 + b^2 = c^2$



Example: Pythagorean Triples

All triples a, b, c such that $a^2 + b^2 = c^2$

a	b	c
3	4	5
5	12	13
6	8	10
8	15	17
9	12	15
12	16	20

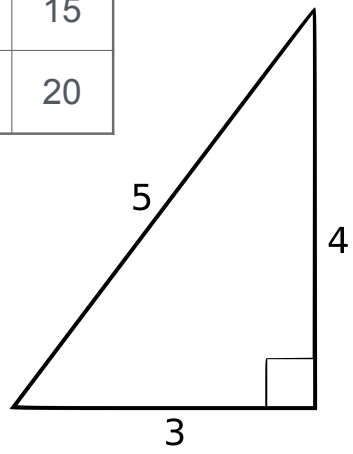


Example: Pythagorean Triples

All triples a, b, c such that $a^2 + b^2 = c^2$

```
with
  i(n) as (
    select 1 union select n+1 from i where n < 20
  )
select a.n as a, b.n as b, c.n as c
       from ____________________________________________
       where _______________ and a.n*a.n + b.n*b.n = c.n*c.n;
```

a	b	c
3	4	5
5	12	13
6	8	10
8	15	17
9	12	15
12	16	20



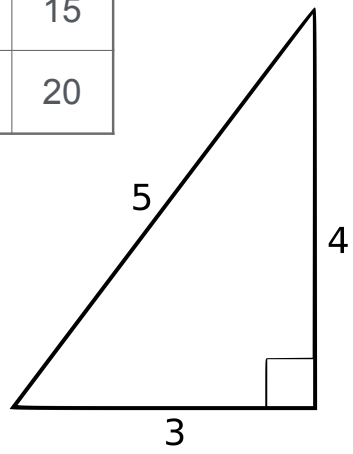
Example: Pythagorean Triples

All triples a, b, c such that $a^2 + b^2 = c^2$

```
with
  i(n) as (
    select 1 union select n+1 from i where n < 20
  )
select a.n as a, b.n as b, c.n as c
       from ____________________________________________
       where _______________ and a.n*a.n + b.n*b.n = c.n*c.n;
```

i as a, i as b, i as c

a	b	c
3	4	5
5	12	13
6	8	10
8	15	17
9	12	15
12	16	20

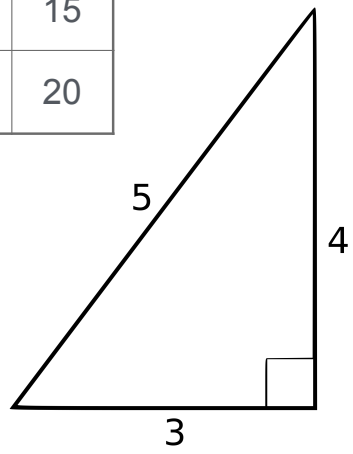


Example: Pythagorean Triples

All triples a, b, c such that $a^2 + b^2 = c^2$

```
with
  i(n) as (
    select 1 union select n+1 from i where n < 20
  )
select a.n as a, b.n as b, c.n as c
       from ____i as a, i as b, i as c____________________
       where ____a.n < b.n_____ and a.n*a.n + b.n*b.n = c.n*c.n;
```

a	b	c
3	4	5
5	12	13
6	8	10
8	15	17
9	12	15
12	16	20



Example: Fibonacci Sequence

Example: Fibonacci Sequence

Computing the next Fibonacci number requires both the previous and current numbers

Example: Fibonacci Sequence

Computing the next Fibonacci number requires both the previous and current numbers

fibs:

n
0
1
1
2
3
5
8
13

Example: Fibonacci Sequence

Computing the next Fibonacci number requires both the previous and current numbers

```
create table fibs as
  with
    fib(previous, current) as (
      select 0, 1 union
      select current, previous+current from fib
      where current <= ____________________________
    )
  select ____________________________ as n from fib;
```

fibs:

n
0
1
1
2
3
5
8
13

Example: Fibonacci Sequence

Computing the next Fibonacci number requires both the previous and current numbers

```
create table fibs as
  with
    fib(previous, current) as (
      select 0, 1 union
      select current, previous+current from fib
      where current <= ____________________________
    )
  select ____previous____ as n from fib;
```

fibs:

n
0
1
1
2
3
5
8
13

Example: Fibonacci Sequence

Computing the next Fibonacci number requires both the previous and current numbers

```
create table fibs as
  with
    fib(previous, current) as (
      select 0, 1 union
      select current, previous+current from fib
      where current <= ____14.15926535____
    )
  select ____previous____ as n from fib;
```

fibs:

n
0
1
1
2
3
5
8
13

A Very Interesting Number

```
The mathematician G. H. Hardy once remarked to the mathematician Srinivasa Ramanujan...
```

A Very Interesting Number

The mathematician G. H. Hardy once remarked to the mathematician Srinivasa Ramanujan...

(Demo)