Announcements

• Project 4 due Friday 11/21 @ 11:59pm
Announcements

• Project 4 due Friday 11/21 @ 11:59pm

• Please submit project 4 in two different ways:
Announcements

• Project 4 due Friday 11/21 @ 11:59pm
• Please submit project 4 in two different ways:
  • `python3 ok --submit` and set your group on ok.cs61a.org
Announcements

• Project 4 due Friday 11/21 @ 11:59pm
• Please submit project 4 in two different ways:
  ▪ `python3 ok --submit` and set your group on ok.cs61a.org
  ▪ Copy files to class account and `submit proj4`
Announcements

• Project 4 due Friday 11/21 @ 11:59pm

• Please submit project 4 in two different ways:
  
  ▪ **python3 ok --submit** and set your group on ok.cs61a.org
  
  ▪ Copy files to class account and **submit proj4**
  
  ▪ A correction to tests/q20.py was released yesterday afternoon
Announcements

• Project 4 due Friday 11/21 @ 11:59pm

• Please submit project 4 in two different ways:
  
  ▪ `python3 ok --submit` and set your group on ok.cs61a.org
  
  ▪ Copy files to class account and `submit proj4`

  ▪ A correction to tests/q20.py was released yesterday afternoon

• Homework 9 (6 pts) due Wednesday 11/26 @ 11:59pm
Announcements

• Project 4 due Friday 11/21 @ 11:59pm
• Please submit project 4 in two different ways:
  ▪ **python3 ok --submit** and set your group on ok.cs61a.org
  ▪ Copy files to class account and **submit proj4**
  ▪ A correction to tests/q20.py was released yesterday afternoon
• Homework 9 (6 pts) due Wednesday 11/26 @ 11:59pm
  ▪ Homework Party Monday evening, location TBD
Announcements

• Project 4 due Friday 11/21 @ 11:59pm

• Please submit project 4 in two different ways:
  
  ▪ **python3 ok --submit** and set your group on ok.cs61a.org
  
  ▪ Copy files to class account and **submit proj4**
  
  ▪ A correction to tests/q20.py was released yesterday afternoon

• Homework 9 (6 pts) due Wednesday 11/26 @ 11:59pm
  
  ▪ Homework Party Monday evening, location TBD

• Guest in live lecture, TA Soumya Basu, on Monday 11/24 (videos by John)
**Announcements**

- Project 4 due Friday 11/21 @ 11:59pm
- Please submit project 4 in two different ways:
  - `python3 ok --submit` and set your group on ok.cs61a.org
  - Copy files to class account and `submit proj4`
  - A correction to tests/q20.py was released yesterday afternoon
- Homework 9 (6 pts) due Wednesday 11/26 @ 11:59pm
  - Homework Party Monday evening, location TBD
- Guest in live lecture, TA Soumya Basu, on Monday 11/24 (videos by John)
- No lecture on Wednesday 11/26 (turkey)
Announcements

• Project 4 due Friday 11/21 @ 11:59pm
• Please submit project 4 in two different ways:
  - `python3 ok --submit` and set your group on ok.cs61a.org
  - Copy files to class account and submit `proj4`
  - A correction to tests/q20.py was released yesterday afternoon
• Homework 9 (6 pts) due Wednesday 11/26 @ 11:59pm
  - Homework Party Monday evening, location TBD
• Guest in live lecture, TA Soumya Basu, on Monday 11/24 (videos by John)
• No lecture on Wednesday 11/26 (turkey)
• No lab next Tuesday 11/25 & Wednesday 11/26
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

```sql
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";
```

<table>
<thead>
<tr>
<th>Parent</th>
<th>Child</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
</tr>
<tr>
<td>abraham</td>
<td>clinton</td>
</tr>
<tr>
<td>delano</td>
<td>herbert</td>
</tr>
<tr>
<td>fillmore</td>
<td>abraham</td>
</tr>
<tr>
<td>fillmore</td>
<td>delano</td>
</tr>
<tr>
<td>fillmore</td>
<td>grover</td>
</tr>
<tr>
<td>eisenhower</td>
<td>fillmore</td>
</tr>
</tbody>
</table>
Local Tables

A create table statement names a table globally

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

parents:
  Eisenhower
    Fillmore
      Abraham
        Barack
      Delano
      Grover
  Clinton
  Herbert
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

...
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:
    Eisenhower
    Fillmore
    Abraham
    Delano
    Grover
    Barack
    Clinton
    Herbert
```
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 ...
```

```
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 ...
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 ...
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 ...
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:

- Eisenhower
- Fillmore

- Abraham
- Delano
- Grover

- Barack
- Clinton
- Herbert
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"
  )
sel ect 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
  best(dog) as (  
    select "eisenhower" union
    select "barack"
  )
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
    best(dog) as ( 
        select "eisenhower" union 
        select "barack"
    )
select parent from parents, best where child=dog;
```
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;
```

<table>
<thead>
<tr>
<th>parent</th>
<th>dog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>eisenhower</td>
</tr>
<tr>
<td></td>
<td>barack</td>
</tr>
</tbody>
</table>

Diagram:
- `parents`: Abraham → Eisenhower → Fillmore
- `best`: dog
- `parent`: abraham
- `child`: barack
- `rel`: parent → child

- Abraham
- Clinton
- Herbert
- Delano
- Grover
- Barack
- Fillmore
- Eisenhower
**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.

```sql
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;
```

<table>
<thead>
<tr>
<th>parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
</tr>
</tbody>
</table>
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

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

... Part of the select statement

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

<table>
<thead>
<tr>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
</tr>
<tr>
<td>eisenhower</td>
</tr>
<tr>
<td>barack</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
</tr>
</tbody>
</table>

Local table only exists for this select
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;
```

(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;
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;
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;
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;
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;

<table>
<thead>
<tr>
<th>parent</th>
<th>child</th>
<th>first</th>
<th>second</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
<td>abraham</td>
<td>delano</td>
</tr>
</tbody>
</table>
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;
```

<table>
<thead>
<tr>
<th>parent</th>
<th>child</th>
<th>first</th>
<th>second</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
<td>abraham</td>
<td>delano</td>
</tr>
</tbody>
</table>
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;

<table>
<thead>
<tr>
<th>parent</th>
<th>child</th>
<th>first</th>
<th>second</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
<td>abraham</td>
<td>delano</td>
</tr>
</tbody>
</table>
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;
```

```
<table>
<thead>
<tr>
<th>parent</th>
<th>child</th>
<th>first</th>
<th>second</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
<td>abraham</td>
<td>delano</td>
</tr>
</tbody>
</table>

parents:

- Eisenhower
  - Fillmore
    - Abraham
    - Barack
    - Delano
    - Clinton
    - Herbert
  - Grover

siblings

- nephew
- parent
- child
- first
- second
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;

<table>
<thead>
<tr>
<th>parent</th>
<th>child</th>
<th>first</th>
<th>second</th>
</tr>
</thead>
<tbody>
<tr>
<td>abraham</td>
<td>barack</td>
<td>abraham</td>
<td>delano</td>
</tr>
</tbody>
</table>
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

```sql
create table parents as
    select "abraham" as parent, "barack" as child union
    ...
```
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```sql
create table parents as
    select "abraham" as parent, "barack" as child union
    ...
```
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)
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

```sql
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
)
```
Local Tables can be Declared Recursively

An ancestor is your parent or an ancestor of your parent

\[
\begin{align*}
\text{create table parents as} & \quad \text{(}\mathbf{\text{parents:}}) \\
& \quad \begin{align*}
& \quad \text{select "abraham" as parent, "barack" as child union} \\
& \quad \quad \ldots \\
& \quad \text{with} \\
& \quad \quad \text{ancestors(ancestor, descendent) as (} \\
& \quad \quad \quad \begin{align*}
& \quad \quad \text{select parent, child from parents union} \\
& \quad \quad \quad \text{select ancestor, child} \\
& \quad \quad \quad \quad \text{from ancestors, parents} \\
& \quad \quad \quad \quad \text{where parent = descendent} \\
& \quad \quad \)} \\
& \quad \quad \text{select ancestor from ancestors where descendent="herbert";}
\end{align*}
\end{align*}
\end{align*}
\]
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";
```
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;
```

<table>
<thead>
<tr>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>
Global Names for Recursive Tables

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

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

<table>
<thead>
<tr>
<th>odds</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

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

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

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

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

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

```sql
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

```sql
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

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

Nope!

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

```sql
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

```sql
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

```sql
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

```sql
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

```
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;
```

```
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.

```sql
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.

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

```sql
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
	hat 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.

```sql
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.

```sql
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.

```sql
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 $a^2 + b^2 = c^2$

<table>
<thead>
<tr>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>20</td>
</tr>
</tbody>
</table>
Example: Pythagorean Triples

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

with

$$i(n) \text{ as } (\begin{array}{c}
\text{select 1 union select } n+1 \text{ from } i \text{ where } n < 20
\end{array})$$

$$\text{select } a.n \text{ as } a, b.n \text{ as } b, c.n \text{ as } c$$

$$\text{from } \begin{array}{c}
\end{array}$$

$$\text{where } \begin{array}{c}
\end{array} \text{ and } a.n*a.n + b.n*b.n = c.n*c.n;$$
Example: Pythagorean Triples

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

with

\[
i(n) \text{ as (}
\begin{align*}
& \text{select 1 union select } n+1 \text{ from } i \text{ where } n < 20 \\
& )
\end{align*}
\]

select a.n as a, b.n as b, c.n as c

\[
\begin{align*}
& \text{from } i \text{ as a, i as b, i as c} \\
& \text{where } \\
& \text{and } a.n*a.n + b.n*b.n = c.n*c.n;
\end{align*}
\]
Example: Pythagorean Triples

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

with

\[
i(n) \text{ as } (\text{select 1 union select } n+1 \text{ from } i \text{ where } n < 20)\]

\[
\text{select } a.n \text{ as } a, b.n \text{ as } b, c.n \text{ as } c
\]

\[
\text{from } \quad i \text{ as } a, i \text{ as } b, i \text{ as } c
\]

\[
\text{where } a.n < b.n \text{ and } a.n*a.n + b.n*b.n = c.n*c.n;
\]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>20</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>fibs:</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
</tbody>
</table>
Example: Fibonacci Sequence

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

```sql
create table fibs as
    with
    fib(previous, current) as ( 
        select 0, 1 union
        select current, previous+current from fib
        where current <= 13
    )
    select ____________ as n from fib;
```

<table>
<thead>
<tr>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>13</td>
</tr>
</tbody>
</table>
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;
```

<table>
<thead>
<tr>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>13</td>
</tr>
</tbody>
</table>
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;
```
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)