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@mail.eecs.berkeley.edu (Albert)
• Homework 9 merged with Homework 10; both are due Wednesday 4/29 @ 11:59pm
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:

<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
  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
   ...

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.

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

Diagram:

```
global

parents:

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

```
local
```
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 ...
```

![Diagram](image)
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
```

```sql
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 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

```sql
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 ...
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.

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

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

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

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

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

**Table: parent**

<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
    ...
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

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

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

Local table only exists for this select

(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;
(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

```sql
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:

- Eisenhower
- Fillmore
- Abraham
- Delano
- Grover
- Barack
- Clinton
- Herbert
Example: Relationships

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

(B) How many rows will result?

```sql
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?

```sql
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?

```sql
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?

```sql
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?

```sql
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, siblings
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>
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
...
```
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:

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

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

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";
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:

```sql
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

```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>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>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
</tr>
<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;
```

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
```

Nope!
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.

```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;
```
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;
```

Nope!
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 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 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

```
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

$$i(n) = \begin{cases} 
1 & \text{if } n = 1 \\
1 + n & \text{if } n > 1 
\end{cases}$$

$$i(n) = i(n-1) + n$$

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 $\cdots$ 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

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

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

$$\text{from} \ \text{__________________________________________}$$

$$\text{where} \ ____________ \ \text{and} \ a.n*a.n + b.n*b.n = c.n*c.n;$$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<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

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

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

from __________________________

\[
\text{where } a.n < b.n
\]

where ____________ and $a.n*a.n + b.n*b.n = c.n*c.n$;
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 <= _________________
    )
select _________________ as n from fib;
```

fibs:  

<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 <= _______________________
  )
previous
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

```sql
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;
```

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
</tr>
</tbody>
</table>
The mathematician G. H. Hardy once remarked to the mathematician Srinivasa Ramanujan...
The mathematician G. H. Hardy once remarked to the mathematician Srinivasa Ramanujan...