Announcements
Database Management Systems
Database Management System Architecture

Architecture of a Database System by Hellerstein, Stonebreaker, and Hamilton
Query Planning

The manner in which tables are filtered, sorted, and joined affects execution time

Select the parents of curly-furred dogs:

```
select parent from parents, dogs
where child = name and fur = "curly";
```

Join all rows of parents to all rows of dogs, filter by `child = name` and `fur = "curly"`

Join only rows of parents and dogs where `child = name`, filter by `fur = "curly"`

Filter dogs by `fur = "curly"`, join result with all rows of parents, filter by `child = name`

Filter dogs by `fur = "curly"`, join only rows of result and parents where `child = name`
Local Tables
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.

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;
```

(Demo)

Local table only exists for this select

Part of the select statement

Parent table names a table that is globally visible.

Best table only exists in the context of the with clause.

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

<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>

Abraham

Eisenhower

Fillmore

Delano

Grover

Clinton

Herbert

Barack
Example: Relationships

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

(B) How many rows and columns 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, 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>
Recursive Local Tables
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 ...

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

```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

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

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

```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
```

```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

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

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

with

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

$$\text{select a.n as a, b.n as b, c.n as c}
\begin{align*}
\text{from }
\end{align*}
\begin{align*}
\text{i as a, i as b, i as c}
\end{align*}
$$

where $a.n < b.n$ and $a.n*a.n + b.n*b.n = c.n*c.n$;
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>

Local table

<table>
<thead>
<tr>
<th>previous</th>
<th>current</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>21</td>
</tr>
</tbody>
</table>
A Very Interesting Number

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