Declarative Languages

Database Management Systems

Declarative Programming

Structured Query Language (SQL)

SQL Overview

Getting Started with SQL
Selecting Value Literals
A select statement always includes a comma-separated list of column descriptions. A column description is an expression, optionally followed by as and a column name:

\[
\text{select} \ [\text{expression}] \ \text{as} \ \text{[name]}, \ \text{[expression]} \ \text{as} \ \text{[name]} \ldots
\]

Selecting literals creates a one-row table. The union of two select statements is a table containing the rows of both of their results:

\[
\text{select} \ \text{"abraham" as "parent", "barack" as "child"};
\]

Joining Tables
Two tables A & B are joined by a comma to yield all combos of a row from A & a row from B:

\[
\text{create table dogs as select "abraham" as name, "long" as fur union select "clinton" as name, "short" as fur union select "delano" as name, "short" as fur union select "fillmore" as name, "short" as fur union select "grover" as name, "short" as fur union select "herbert" as name, "curly" as fur};
\]

Joining Tables
Joining a table with itself:
Two tables may share a column name; dot expressions and aliases disambiguate column values:

\[
\text{select columns from \{table\} where \{condition\} order by \{order\}};
\]

Select all pairs of siblings:

\[
\text{select child from parents where parent = "abraham";}
\]

Projecting Tables
A subset of the rows of the input table can be selected using a where clause:

\[
\text{select [columns] from \{table\} where \{condition\} order by \{order\}};
\]

Select statements project existing tables:
An ordering over the remaining rows can be declared using an order by clause:

\[
\text{order by \{expression\}};
\]

Select Statements Project Existing Tables
A select statement can specify an input table using a from clause:

\[
\text{from \{table\}};
\]

Naming Tables
SQL is often used as an interactive language:
A create table statement gives the result a name:

\[
\text{create table \{name\} as \{select statement\};}
\]

Select the parents of curly-furred dogs:

\[
\text{select parent from parents where parent > "abraham";}\]

Naming Tables
The result of a select statement is displayed to the user, but not stored:

\[
\text{of \{select statement\};}
\]

FirstSecond

<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>clinton</td>
</tr>
<tr>
<td>fillmore</td>
<td>barack</td>
</tr>
<tr>
<td>eisenhower</td>
<td>fillmore</td>
</tr>
</tbody>
</table>

Aliases and Dot Expressions
A column description is an expression, optionally followed by as and a column name:

\[
\text{select \text{col} \text{as \{name\}}
\]

Joining Two Tables
Two tables may share a column name; dot expressions and aliases disambiguate column values:

\[
\text{select [columns] from \{table\} where [condition] order by [order]};
\]

Select all pairs of siblings:

\[
\text{where \{a, parent = \{b, parent\} and \{a, child = \{b, child\};}
\]

Joining Two Tables
An ordering over the remaining rows can be declared using an order by clause:

\[
\text{order by \{expression\}};
\]

FirstSecond

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

Selecting Value Literals
The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \"fillmore\" as \"fillmore\";}
\]

Selecting literals creates a one-row table. The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \"abraham\" as \"parent\", \"barack\" as \"child\" union select \"delano\" as \"parent\", \"herbert\" as \"child\" union select \"fillmore\" as \"parent\", \"abraham\" as \"child\" union select \"fillmore\" as \"parent\", \"delano\" as \"child\" union select \"fillmore\" as \"parent\", \"grover\" as \"child\" union select \"eisenhower\" as \"parent\", \"fillmore\" as \"child\";}
\]

Selecting Value Literals
The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \text{abraham} as \text{parent}, \text{barack} as \text{child};}
\]

Selecting literals creates a one-row table. The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \text{abraham} as \text{parent}, \text{barack} as \text{child} union select \text{delano} as \text{parent}, \text{herbert} as \text{child} union select \text{fillmore} as \text{parent}, \text{abraham} as \text{child} union select \text{fillmore} as \text{parent}, \text{delano} as \text{child} union select \text{fillmore} as \text{parent}, \text{grover} as \text{child} union select \text{eisenhower} as \text{parent}, \text{fillmore} as \text{child};}
\]

Selecting Value Literals
The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \text{abraham} as \text{parent}, \text{barack} as \text{child} union select \text{delano} as \text{parent}, \text{herbert} as \text{child} union select \text{fillmore} as \text{parent}, \text{abraham} as \text{child} union select \text{fillmore} as \text{parent}, \text{delano} as \text{child} union select \text{fillmore} as \text{parent}, \text{grover} as \text{child} union select \text{eisenhower} as \text{parent}, \text{fillmore} as \text{child};}
\]

Selecting Value Literals
The union of two select statements is a table containing the rows of both of their results:

\[
\text{select \text{abraham} as \text{parent}, \text{barack} as \text{child} union select \text{delano} as \text{parent}, \text{herbert} as \text{child} union select \text{fillmore} as \text{parent}, \text{abraham} as \text{child} union select \text{fillmore} as \text{parent}, \text{delano} as \text{child} union select \text{fillmore} as \text{parent}, \text{grover} as \text{child} union select \text{eisenhower} as \text{parent}, \text{fillmore} as \text{child};}
\]
Joining Multiple Tables

Multiple tables can be joined to yield all combinations of rows from each

```sql
create table grandparents as
    select a.parent as grandog, b.child as granpup
    from parents as a, parents as b
    where b.parent = a.child;
```

Select all grandparents with the same fur as their grandchildren

```sql
select grandog from grandparents, dogs as c, dogs as d
    where grandog = c.name and
        granpup = d.name and
        c.fur = d.fur;
```

Which tables need to be joined together?