Joining Tables

Joining Two Tables

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

```sql
create table dogs as
    select "abraham" as name, "long" as fur union
    select "barack" as name, "short" as fur union
    select "clinton" as name, "long" as fur union
    select "delano" as name, "short" as fur union
    select "eisenhower" as name, "short" as fur union
    select "fillmore" as name, "curly" as fur union
    select "grover" as name, "short" as fur union
    select "herbert" as name, "curly" as fur union

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

Select the parents of curly-furred dogs

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

Joining a Table with Itself

Two tables may share a column name; dot expressions and aliases disambiguate column values.

```sql
select a.grandparent, b.child
    from parents as a, parents as b
    where b.parent = a.child;
```

Example: Grandparents

Which select statement evaluates to all grandparent, grandchild pairs?

1. `select a.grandparent, b.child from parents as a, parents as b where b.parent = a.child;`
2. `select a.parent, b.child from parents as a, parents as b where a.parent = b.child;`
3. `select a.parent, b.child from parents as a, parents as b where b.parent = a.child;`
4. `select a.grandparent, b.child from parents as a, parents as b where a.parent = b.child;`
5. None of the above

Aliases and Dot Expressions

Two tables may share a column name; dot expressions and aliases disambiguate column values.

```
create table dogs as
    select "abraham" as name, "long" as fur union
    select "barack" as name, "short" as fur union
    select "clinton" as name, "long" as fur union
    select "delano" as name, "long" as fur union
    select "eisenhower" as name, "short" as fur union
    select "fillmore" as name, "curly" as fur union
    select "grover" as name, "short" as fur union
    select "herbert" as name, "curly" as fur union

create table parents as
    select "abraham" as parent, "barack" as child union
    select "abraham" as parent, "clinton" 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 union;
```
Joining Multiple Tables

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

Example: Dog Triples

Fall 2014 Quiz Question (Slightly Modified)

Write a SQL query that selects all possible combinations of three different dogs with the same fur and lists each triple in reverse alphabetical order.

Numerical Expressions

Expressions can contain function calls and arithmetic operators.

String Expressions

String values can be combined to form longer strings.

Database Management Systems
Database Management System Architecture

Query Planning

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

Select the parents of curly-furred dogs:

\[
\text{select parent from parents, dogs;}
\]

\[
\text{where child = name and fur = "curly";}
\]

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