**Joining Tables**

- **Reminder: John the Patriotic Dog Breeder**
- **Announcements**

**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.

```
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;
```

Select the parents of curly-furred dogs
```
select child from parents, dogs
where fur = "curly";
```

**(Demo)**

**Aliases and Dot Expressions**

- **Joining a Table with Itself**
  - Two tables may share a column name; dot expressions and aliases disambiguate column values.
  - `SELECT [columns] FROM [table] WHERE [condition] ORDER BY [order];`
  - `[table]` is a comma-separated list of table names with optional aliases.
  - **Select all pairs of siblings**
  ```
  select a.child as first, b.child as second
  from parents as a, parents as b
  where a.parent = b.parent and a.child < b.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 a.child = b.parent;
`
4. `select a.grandparent, b.child from parents as a, parents as b
   where a.parent = b.child;
`
5. None of the above
Joining Multiple Tables

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

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

Select all grandparents with the same fur as their grandchildren.

```sql
select grandparent, dog

as c, dog

as d

where grandparent = c.name 

and grandchild = d.name 

and c.fur = d.fur;
```

Which tables need to be joined together?

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

Numerical Expressions

Expressions can contain function calls and arithmetic operators.

```sql
select [columns] from [table] where [expression] order by [expression];

[expression] as [name], [expression] as [name], ...
```

Combine values: +, -, *, /, %, and, or,

Transform values: abs, round, not, -

Compare values: <, <=, >, >=, <>, !=, =

(Demo)

Example: Dog Triples

Fall 2014 Quiz Question

Write a SQL query that selects all possible combinations of three dogs with the same fur and lists them in order of increasing height.

```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" union
    select "delano", "jackson";
```

```sql
create table dogs as
    select "abraham" as name, "long" as fur, 26 as height union
    select "barack", "short", 52 union
    select "clinton", "long", 47 union
    select "delano", "long", 46 union
    select "fillmore", "curly", 35 union
    select "grover", "short", 34 union
    select "eisenhower", "curly", 31 union
    select "jackson", "long", 43;
```

String Expressions

String values can be combined to form longer strings:

```sql
sqlite> select "hello, world"
```

Basic string manipulation is built into SQL, but differs from Python.

```sql
sqlite> create table phrase as select "Hello, world" as s;
```

Strings can be used to represent structured values, but doing so is rarely a good idea.

```sql
sqlite> create table lists as select "one" as car, "two,three,four" as cdr;
```

(Demo)

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

Select the parents of curly-furred dogs:

```sql
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`