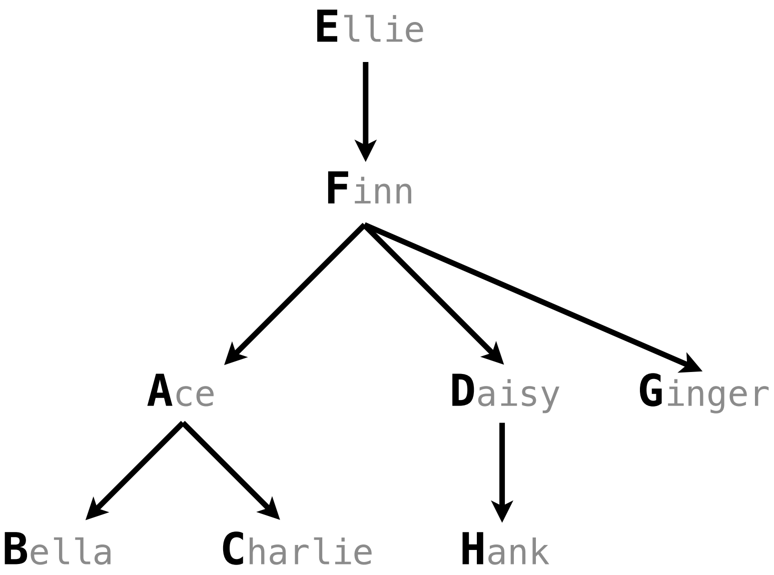
Tables

Announcements

Joining Tables

Dog Family Tree

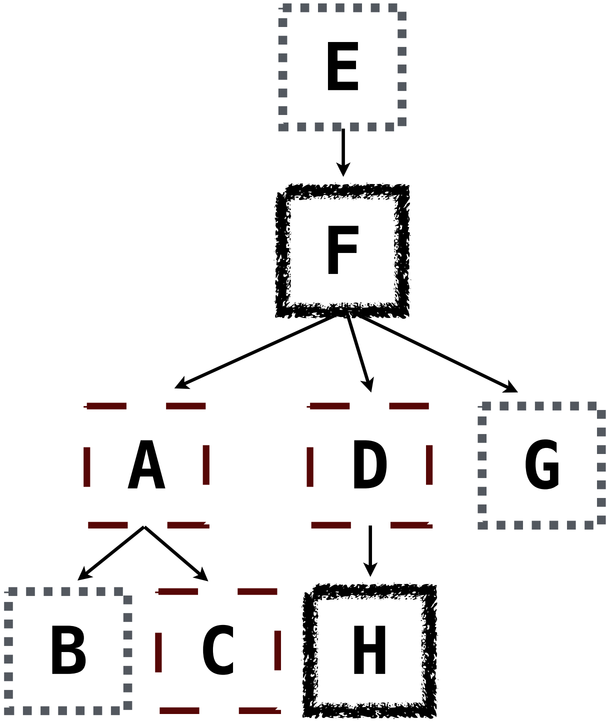
```
CREATE TABLE parents AS
  SELECT "ace" AS parent, "bella" AS child UNION
  SELECT "ace"          , "charlie"        UNION
  SELECT "daisy"        , "hank"           UNION
  SELECT "finn"         , "ace"            UNION
  SELECT "finn"         , "daisy"          UNION
  SELECT "finn"         , "ginger"         UNION
  SELECT "ellie"        , "finn";
```



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 "ace" AS name, "long" AS fur UNION
  SELECT "bella"      , "short"       UNION
  SELECT "charlie"    , "long"        UNION
  SELECT "daisy"      , "long"        UNION
  SELECT "ellie"      , "short"       UNION
  SELECT "finn"       , "curly"       UNION
  SELECT "ginger"     , "short"       UNION
  SELECT "hank"       , "curly";

CREATE TABLE parents AS
  SELECT "ace" AS parent, "bella" AS child UNION
  SELECT "ace"          , "charlie"        UNION
  ...;
```



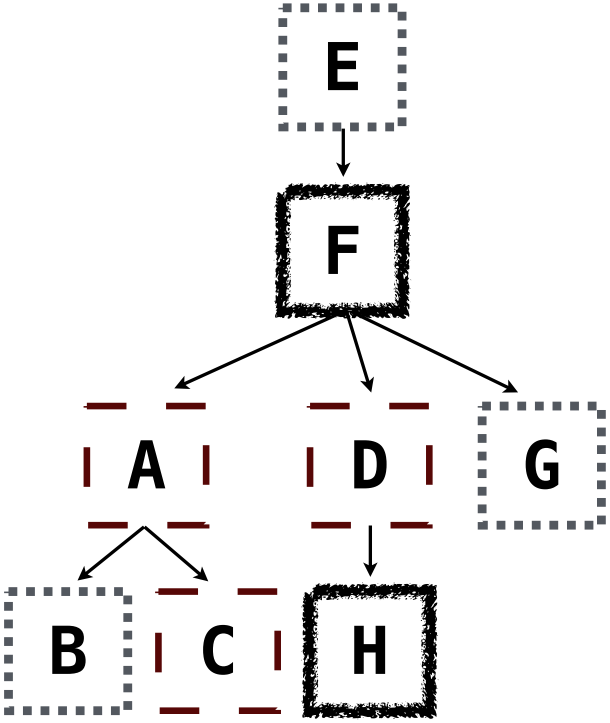
Select the parents of curly-furred dogs

```
SELECT parent FROM parents, dogs
              WHERE child = name AND fur = "curly";

SELECT parent FROM parents JOIN dogs
              ON child = name WHERE fur = "curly";
```

(Demo)

Discussion Question

```
CREATE TABLE dogs AS
  SELECT "ace" AS name, "long" AS fur UNION
  SELECT "bella"      , "short"       UNION
  SELECT "charlie"    , "long"        UNION
  SELECT "daisy"      , "long"        UNION
  SELECT "ellie"      , "short"       UNION
  SELECT "finn"       , "curly"       UNION
  SELECT "ginger"     , "short"       UNION
  SELECT "hank"       , "curly";

CREATE TABLE parents AS
  SELECT "ace" AS parent, "bella" AS child UNION
  SELECT "ace"          , "charlie"        UNION
  ...;
```



Show the name and fur of the parents of Daisy and Bella

```
SELECT name, fur FROM parents JOIN dogs ON  parent=name

                    WHERE  child="daisy" or child="bella" ;
```

Aliases and Dot Expressions

Joining a Table with Itself

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

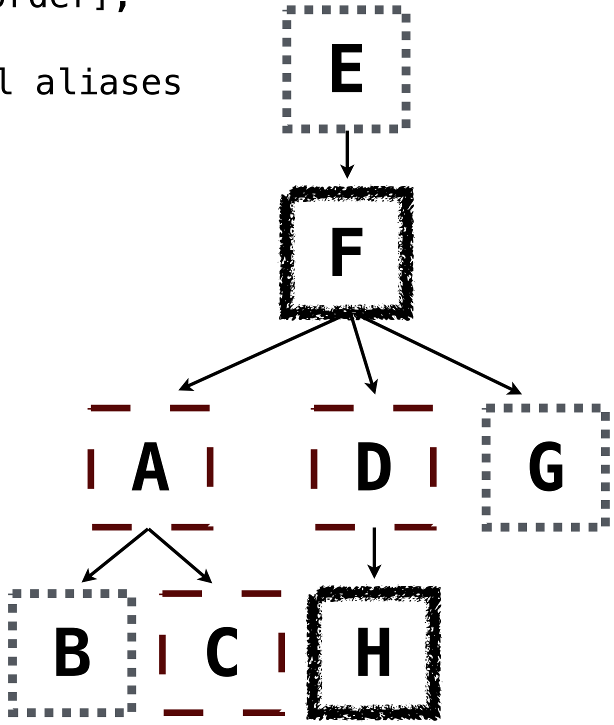
```
SELECT [columns] FROM [table] WHERE [condition] ORDER BY [order];
```

[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;
```

first	second
bella	charlie
ace	daisy
ace	ginger
daisy	ginger



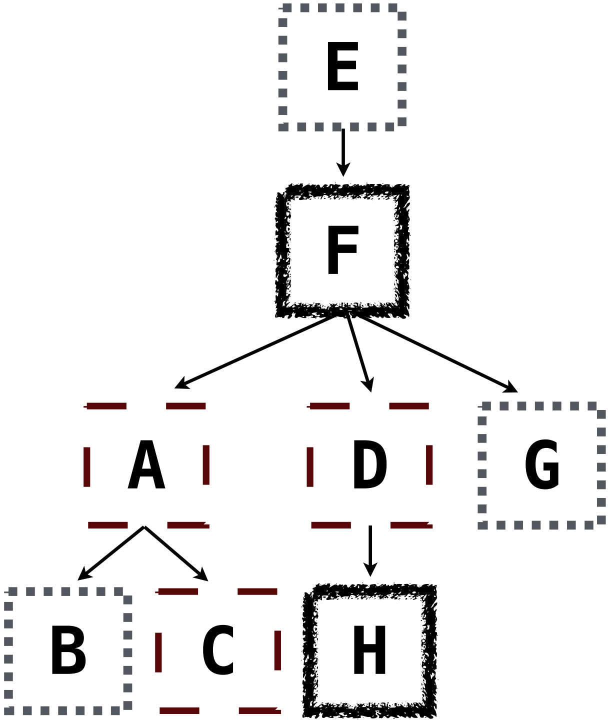
(Demo)

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 *inverse* alphabetical order

```
  CREATE TABLE dogs AS
    SELECT "ace" AS name, "long" AS fur UNION
    SELECT "bella"      , "short"       UNION
    ...;

  CREATE TABLE parents AS
    SELECT "ace" AS parent, "bella" AS child UNION
    SELECT "ace"          , "charlie"        UNION
    ...;
```



Expected output:

```
delano|clinton|abraham
grover|eisenhower|barack
```

(Demo)

String Expressions

(Demo)