Announcements
A Very Interesting Number

The mathematician G. H. Hardy once remarked to the mathematician Srinivasa Ramanujan...
A Very Interesting Number

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

So far, all SQL expressions have referred to the values in a single row at a time
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

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

So far, all SQL expressions have referred to the values in a single row at a time

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

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

An aggregate function in the [columns] clause computes a value from a group of rows
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

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

An aggregate function in the [columns] clause computes a value from a group of rows

create table animals as
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

An aggregate function in the [columns] clause computes a value from a group of rows

```sql
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

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

An aggregate function in the [columns] clause computes a value from a group of rows

```sql
create table animals as
    select "dog" as kind, 4 as legs, 20 as weight union
select "cat" , 4 , 10 union
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

An aggregate function in the [columns] clause computes a value from a group of rows

```sql
create table animals as
  select "dog" as kind, 4 as legs, 20 as weight union
select "cat", 4, 10 union
select "ferret", 4, 10 union
```
**Aggregate Functions**

So far, all SQL expressions have referred to the values in a single row at a time

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

An aggregate function in the [columns] clause computes a value from a group of rows

```sql
create table animals as
    select "dog" as kind, 4 as legs, 20 as weight union
    select "cat"  , 4     , 10     union
    select "ferret" , 4    , 10     union
    select "parrot" , 2    , 6      union
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

An aggregate function in the [columns] clause computes a value from a group of rows

```
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat", 4, 10 union
select "ferret", 4, 10 union
select "parrot", 2, 6 union
select "penguin", 2, 10 union
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time.

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

An aggregate function in the `[columns]` clause computes a value from a group of rows.

```
create table animals as
  select "dog" as kind, 4 as legs, 20 as weight union
  select "cat" , 4   , 10    union
  select "ferret" , 4  , 10    union
  select "parrot" , 2   , 6     union
  select "penguin", 2   , 10    union
  select "t-rex"  , 2    , 12000;
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

An aggregate function in the [columns] clause computes a value from a group of rows

```
create table animals as
    select "dog" as kind, 4 as legs, 20 as weight union
    select "cat"    , 4  , 10          union
    select "ferret" , 4  , 10          union
    select "parrot" , 2  ,  6          union
    select "penguin", 2  , 10          union
    select "t-rex"  , 2  , 12000;

animals:
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>kind</td>
<td>legs</td>
<td>weight</td>
</tr>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time

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

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

An aggregate function in the [columns] clause computes a value from a group of rows

```
create table animals as
  select "dog" as kind, 4 as legs, 20 as weight union
  select "cat" , 4 , 10 union
  select "ferret" , 4 , 10 union
  select "parrot" , 2 , 6 union
  select "penguin" , 2 , 10 union
  select "t-rex" , 2 , 12000;

select max(legs) from animals;
```
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time.

An aggregate function in the [columns] clause computes a value from a group of rows.

```sql
create table animals as
    select "dog" as kind, 4 as legs, 20 as weight union
    select "cat", 4, 10 union
    select "ferret", 4, 10 union
    select "parrot", 2, 6 union
    select "penguin", 2, 10 union
    select "t-rex", 2, 12000;

select max(legs) from animals;
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

max(legs)

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
</tr>
</tbody>
</table>
Aggregate Functions

So far, all SQL expressions have referred to the values in a single row at a time.

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

An aggregate function in the [columns] clause computes a value from a group of rows.

```
create table animals as
    select "dog" as kind, 4 as legs, 20 as weight union
    select "cat" , 4 , 10 union
    select "ferret" , 4 , 10 union
    select "parrot" , 2 , 6 union
    select "penguin" , 2 , 10 union
    select "t-rex" , 2 , 12000;

select max(legs) from animals;  
```

```
animals:
<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
```

(Demo)
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat", 4, 10 union
select "ferret", 4, 10 union
select "parrot", 2, 6 union
select "penguin", 2, 10 union
select "t-rex", 2, 12000;

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful.

```sql
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat" , 4 , 10 union
select "ferret" , 4 , 10 union
select "parrot" , 2 , 6 union
select "penguin" , 2 , 10 union
select "t-rex" , 2 , 12000;
```

<table>
<thead>
<tr>
<th>animals:</th>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
<td></td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful

```sql
select max(weight), kind from animals;
```

create table animals as
```sql
select "dog" as kind, 4 as legs, 20 as weight union
select "cat" , 4 , 10 union
select "ferret" , 4 , 10 union
select "parrot" , 2 , 6 union
select "penguin" , 2 , 10 union
select "t-rex" , 2 , 12000;
```

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful:

```sql
select max(weight), kind from animals;

select min(kind), kind from animals;
```

```sql
create table animals as
  select "dog" as kind, 4 as legs, 20 as weight union
  select "cat" , 4 , 10 union
  select "ferret" , 4 , 10 union
  select "parrot" , 2 , 6 union
  select "penguin" , 2 , 10 union
  select "t-rex"  , 2 , 12000;
```

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful

```sql
select max(weight), kind from animals;
select min(kind), kind from animals;
select max(legs), kind from animals;
```

```sql
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat" , 4 , 10 union
select "ferret" , 4 , 10 union
select "parrot" , 2 , 6 union
select "penguin" , 2 , 10 union
select "t-rex" , 2 , 12000;
```

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful:

```sql
select max(weight), kind from animals;
select min(kind), kind from animals;
select max(legs), kind from animals;
select avg(weight), kind from animals;
```

```sql
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat"    , 4    , 10    union
select "ferret" , 4    , 10    union
select "parrot" , 2    , 6     union
select "penguin", 2    , 10    union
select "t-rex"  , 2    , 12000 union;
```

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Mixing Aggregate Functions and Single Values

An aggregate function also selects a row in the table, which may be meaningful

```
select max(weight), kind from animals;
select min(kind), kind from animals;
select max(legs), kind from animals;
select avg(weight), kind from animals;
```

(Demo)

```
create table animals as
select "dog" as kind, 4 as legs, 20 as weight union
select "cat" , 4 , 10 union
select "ferret" , 4 , 10 union
select "parrot" , 2 , 6 union
select "penguin" , 2 , 10 union
select "t-rex" , 2 , 12000;
```

```
<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
```
Groups
Grouping Rows
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group.
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

```sql
select [columns] from [table] group by [expression] having [expression];
```
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

```
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression

---

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group:

```sql
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression:

```sql
select legs, max(weight) from animals group by legs;
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

```
SELECT [expression] AS [name], [expression] AS [name], ...
FROM [table]
GROUP BY [expression]
HAVING [expression];
```

The number of groups is the number of unique values of an expression

```
SELECT legs, MAX(weight) FROM animals GROUP BY legs;
```

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
**Grouping Rows**

Rows in a table can be grouped, and aggregation is performed on each group

\[
\text{select [columns] from [table] group by [expression] having [expression]};
\]

The number of groups is the number of unique values of an expression

\[
\text{select legs, max(weight) from animals group by legs};
\]
**Grouping Rows**

Rows in a table can be grouped, and aggregation is performed on each group

```sql
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression

```sql
select legs, max(weight) from animals group by legs;
```

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

legs=4
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

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

```sql
select [columns] from [table] group by [expression] having [expression];
```

The number of groups is the number of unique values of an expression

```sql
select legs, max(weight) from animals group by legs;
```

<table>
<thead>
<tr>
<th>animals:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>kind</td>
<td>legs</td>
<td>weight</td>
</tr>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

legs=4

legs=2
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

\[
\text{select} \ [\text{columns}] \ \text{from} \ [\text{table}] \ \text{group by} \ [\text{expression}] \ \text{having} \ [\text{expression}];
\]

The number of groups is the number of unique values of an expression

\[
\text{select} \ \text{legs}, \ \text{max}(\text{weight}) \ \text{from} \ \text{animals} \ \text{group by} \ \text{legs};
\]

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>legs</th>
<th>max(weight)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

legs=4

legs=2
Grouping Rows

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by (expression) having (expression);
```

The number of groups is the number of unique values of an expression

```
select legs, max(weight) from animals group by legs;
```

---

### animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

(Demo)
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

select [columns] from [table] group by [expression] having [expression];
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```
select [columns] from [table] group by [expression] having [expression];
```

A `having` clause filters the set of groups that are aggregated

### animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

```sql
select [columns] from [table] group by [expression] having [expression];
```

A having clause filters the set of groups that are aggregated

```sql
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

### animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by [expression] having [expression];
```

A having clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

weight/legs=5
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by [expression] having [expression];
```

A `having` clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

```
animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
```

weight/legs=5

weight/legs=2
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```
expression as name, expression as name, ...

select [columns] from [table] group by [expression] having [expression];
```

A having clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```
select [columns] from [table] group by [expression] having [expression];
```

A `having` clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```
select [columns] from [table] group by [expression] having [expression];
```

A having clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by [expression] having [expression];
```

A having clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

```sql
select [columns] from [table] group by [expression] having [expression];
```

A `having` clause filters the set of groups that are aggregated

```sql
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

<table>
<thead>
<tr>
<th>weight/legs</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**animals:**

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

select [columns] from [table] group by [expression] having [expression];
```

A `having` clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>weight/legs</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Selecting Groups

Rows in a table can be grouped, and aggregation is performed on each group

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

```
select [columns] from [table] group by [expression] having [expression];
```

A **having** clause filters the set of groups that are aggregated

```
select weight/legs, count(*) from animals group by weight/legs having count(*)>1;
```

<table>
<thead>
<tr>
<th>weight/legs</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

animals:

<table>
<thead>
<tr>
<th>kind</th>
<th>legs</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>cat</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ferret</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>parrot</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>penguin</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>t-rex</td>
<td>2</td>
<td>12000</td>
</tr>
</tbody>
</table>
Discussion Question

What's the maximum difference between leg count for two animals with the same weight?
Select Grammar
SQL Select Statements
SQL Select Statements

WITH

RECURSIVE

common-table-expression

, 

SELECT

DISTINCT

ALL

result-column

, 

FROM

table-or-subquery

, 

join-clause

WHERE

expr

GROUP BY

expr

HAVING

expr
SQL Select Statements

WITH
  RECURSIVE
    common-table-expression
      ,
        
SELECT
  DISTINCT
    result-column
      ,
        
FROM
  table-or-subquery
      ,
        
WHERE
  expr

GROUP BY
  expr
      ,
        
HAVING
  expr

ORDER BY
  ordering-term
      ,
        
LIMIT
  expr
      ,
        
OFFSET
  expr

VALUES
  ( expr )
      ,
        
compound-operator

http://www.sqlite.org/lang_select.html
SQL Select Statements

http://www.sqlite.org/lang_select.html
SQL Select Statements

http://www.sqlite.org/lang_select.html
SQL Select Statements

http://www.sqlite.org/lang_select.html
SQL Select Statements

http://www.sqlite.org/lang_select.html
SQL Select Statements

http://www.sqlite.org/lang_select.html