Aggregation

Announcements

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

```
[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

```
[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

```
          [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
```

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
```

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
```

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
```

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
```

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;
```

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;
```

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

max(legs)
4

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

max(legs)
4

(Demo)

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

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

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

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

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

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

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

```
select max(legs), kind from animals;
```

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

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

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

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

```
select max(legs), kind from animals;
```

```
select avg(weight), kind from animals;
```

```
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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Mixing Aggregate Functions and Single Values

An aggregate function also selects some row in the table to supply the values of columns that are not aggregated. In the case of max or min, this row is that of the max or min value. Otherwise, it is arbitrary.

```
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;
```

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Discussion Question

What are all the kinds of animals that have the maximal number of legs?

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

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

```
                [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

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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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];
```

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

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

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

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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];
```

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

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

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

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

legs=4

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];
```

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

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

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

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

legs=4

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];
```

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

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

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

animals:

	kind	legs	weight
legs=4	dog	4	20
	cat	4	10
	ferret	4	10
legs=2	parrot	2	6
	penguin	2	10
	t-rex	2	12000

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];
```

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

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

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

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

legs	max(weight)
4	20
2	12000

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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

legs	max(weight)
4	20
2	12000

legs=4

legs=2

(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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5

weight/legs=2

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5

weight/legs=2

weight/legs=2

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5

weight/legs=2

weight/legs=2

weight/legs=3

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5

weight/legs=2

weight/legs=2

weight/legs=3

weight/legs=5

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs=5
weight/legs=2
weight/legs=2
weight/legs=3
weight/legs=5
weight/legs=6000

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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];
```

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

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

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

weight/legs	count(*)
5	2
2	2

weight/legs=5
weight/legs=2
weight/legs=2
weight/legs=3
weight/legs=5
weight/legs=6000

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs	count(*)
5	2
2	2

weight/legs=5
weight/legs=2
weight/legs=2
weight/legs=3
weight/legs=5
weight/legs=6000

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

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;
```

weight/legs	count(*)
5	2
2	2

animals:

kind	legs	weight
dog	4	20
cat	4	10
ferret	4	10
parrot	2	6
penguin	2	10
t-rex	2	12000

Discussion Question

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

Example: Big Game

Example: Big Game

(Demo)