

Window Functions

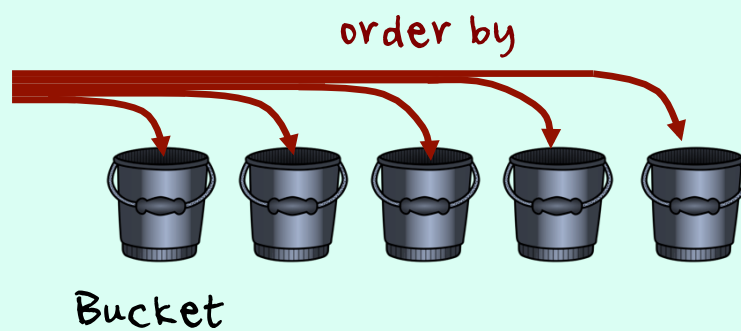
Stored Functions

Country/territory with the most films
per continent and per year since 2000
- beware of ties!

Country/territory with the most films
per continent and per year since 2000
- with `string_agg()` for ties

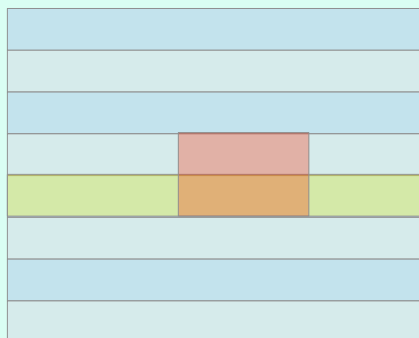
All information for Chinese films since
2010 on one line per film (one column
for actors, one for directors, one for
"alternate titles")

Top 20% of countries/territories
producing films since 2000 (ntile())



Year on year percent change in US film
production since 2000 (lag())

lag() returns a value from a previous row



lead() to look at the "next row"

Stored Functions

Syntax

```
create function funcname(param_name type,  
                           ...)  
returns type  
as $$  
declare  
    var_name type := initial_value;  
    ...  
begin  
    ...  
    return ..;  
end;  
$$ language plpgsql;
```

Beware of statement separators in
Squirrel SQL (see notes in
lab7_sample_queries.sql)

Assignment

```
var_name := value;
```

Some products use SET or SELECT INTO for assignment.

Test

```
if condition then
```

```
...
```

```
else
```

Anything that you can use in a WHERE can be used in an IF (is null, between, and so forth)

```
...
```

```
end if;
```

Loop

```
while condition loop
```

```
...
```

```
end loop;
```

```
for var in from .. to loop
```

```
...
```

```
end loop;
```

You can also say "in reverse" to go from higher value to lower value.

```
for var in (val1, ...) loop
```

```
...
```

```
end loop;
```

You could also have a subquery but avoid it as much as you can in a function.

Case

```
case var  
  when ... then ...;  
  ...  
  else ...;  
end case;
```

Beware of CASE because the syntax in a stored procedure is quite different from the syntax in an SQL statement; however, you can also say
case
 when condition ... as in SQL

Function that shows a number of minutes as a "hours:minutes" string

TODAY'S ASSIGNMENT

Write a function called `arrival_time()` that takes departure time (string, HH:MI) , flight time in minutes (int), time zone difference in minutes (int) and returns the arrival time at destination (string). It should be followed by +1 if it's the next day, for instance:

07:35+1

(it can also be -1 and the previous day between some places)

```
-- Test query
select ad.city || ' (' || ad.code || ')' as departure,
       aa.city || ' (' || aa.code || ')' as arrival,
       f.dep_time as departure_time,
       arrival_time(f.dep_time,
                   f.duration,
                   cast((aa.utc_offset - ad.utc_offset) * 60
                       as int)) as arrival_time
from flights f
  join airports ad
    on ad.code = f.departure
  join airports aa
    on aa.code = f.arrival
where ad.code in ('LHR', 'SFO', 'PEK', 'JFK', 'HKG', 'HNL')
   and aa.code in ('LHR', 'SFO', 'PEK', 'JFK', 'HKG', 'HNL')
order by random()
limit 20
;
```