Hive Analytics

UDF, UDAF, UDTF

Hive functions

Hive functions

1. Operators

```
=, !=, <, >, IS NULL, ...
+, -, *, /, ...
AND, OR, IN, ...
```

Hive functions

1. Operators
2. Functions (UDFs = User Defined Functions)

math: round, floor, ceil, exp, log, …

date: to_date, from_unixtimestamp, year, …

conditional: if, isnull, case, coalesce, …

string: char, concat, lower, trim, repeat, …

Hive functions

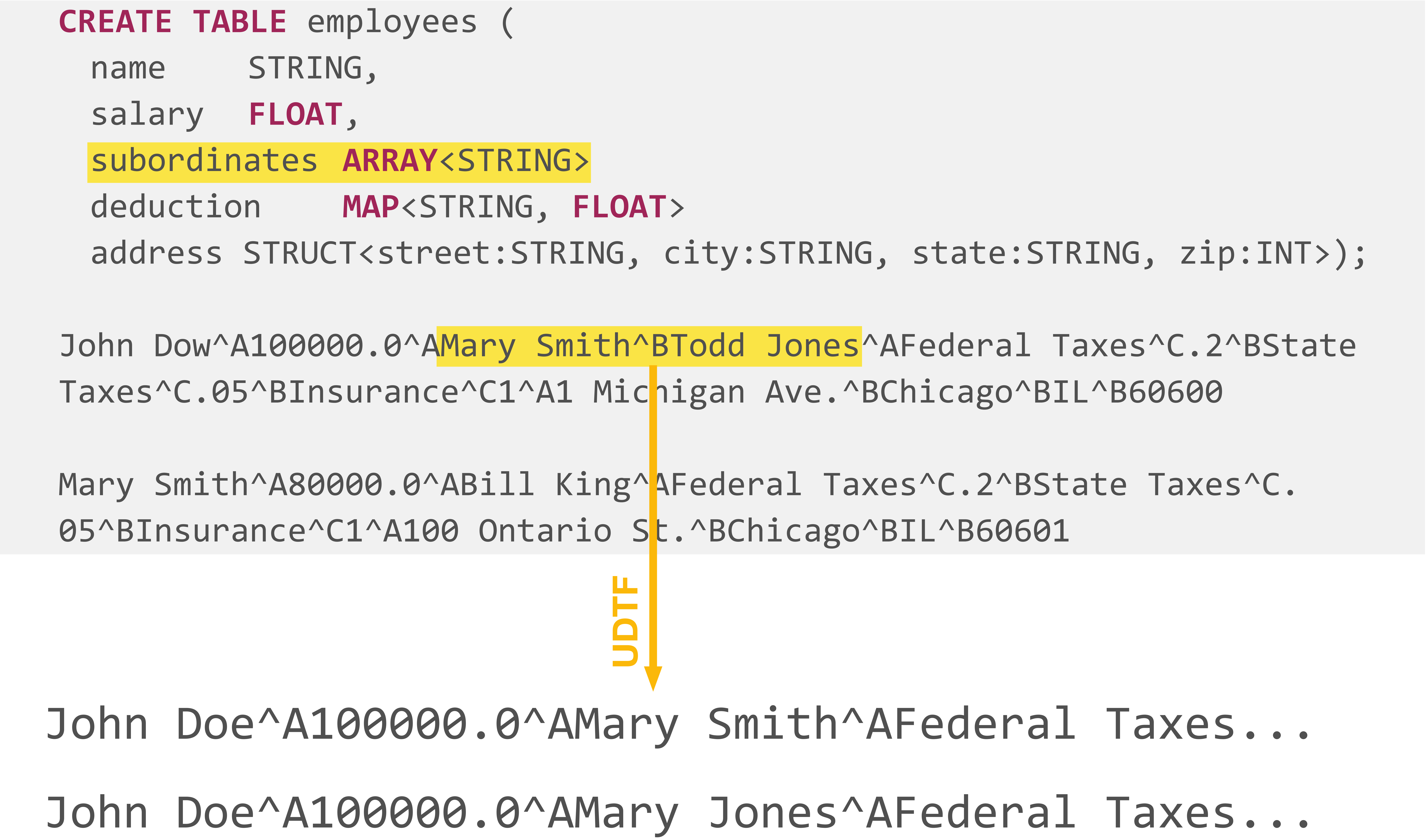
1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)

```
count, sum, min, max, corr, ...
```

Hive functions

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

```
explode, posexplode, parse_url_tuple, …
```



```
CREATE TABLE employees (
  name     STRING,
  salary   FLOAT,
  subordinates ARRAY<STRING>
  deduction    MAP<STRING, FLOAT>
  address STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>);

John Dow^A100000.0^AMary Smith^BTodd Jones^AFederal Taxes^C.2^BState
Taxes^C.05^BInsurance^C1^A1 Michigan Ave.^BChicago^BIL^B60600

Mary Smith^A80000.0^ABill King^AFederal Taxes^C.2^BState Taxes^C.
05^BInsurance^C1^A100 Ontario St.^BChicago^BIL^B60601
```

```
John Doe^A100000.0^AMary Smith^AFederal Taxes...

John Doe^A100000.0^AMary Jones^AFederal Taxes...
```

Hive functions

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

```
hive> show functions;
!
!=
%
*
...
abs
acos
add_months
and
...
```

200+functions

Hive functions

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

200+functions

```
hive> show functions;
!
!=
%
*
...
abs
acos
add_months
and
...
```

```
hive> describe function acos;
acos(x) - returns the arc cosine of x if
-1<=x<=1 or NULL otherwise
```

Hive functions

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

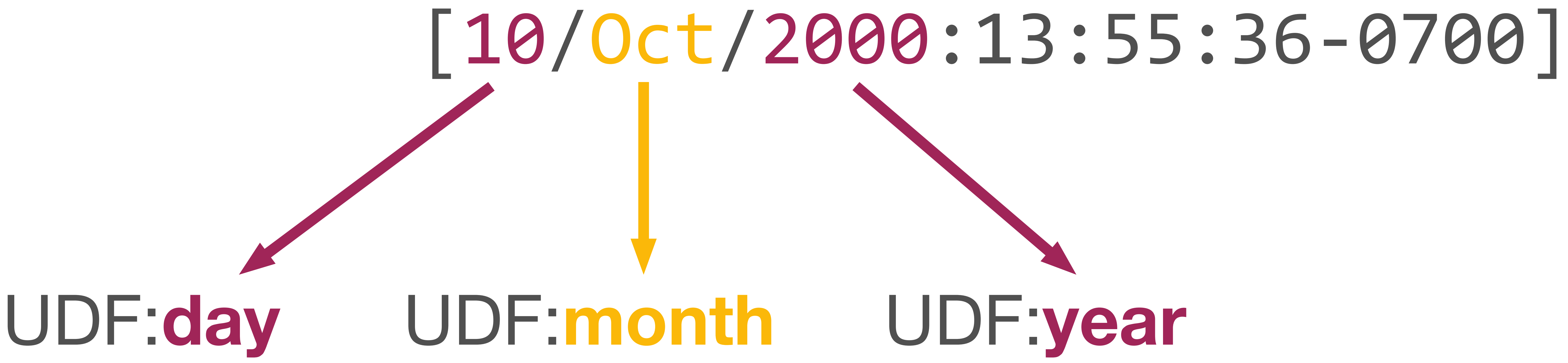
200+functions

```
hive> show functions;
!
!=
%
*
...
abs
acos
add_months
and
...
```

```
hive> describe function acos;
acos(x) - returns the arc cosine of x if
-1<=x<=1 or NULL otherwise

hive> describe function extended acos;
OK
acos(x) - returns the arc cosine of x if
-1<=x<=1 or NULL otherwise
Example:
  > SELECT acos(1) FROM src LIMIT 1;
  0
  > SELECT acos(2) FROM src LIMIT 1;
  NULL
```

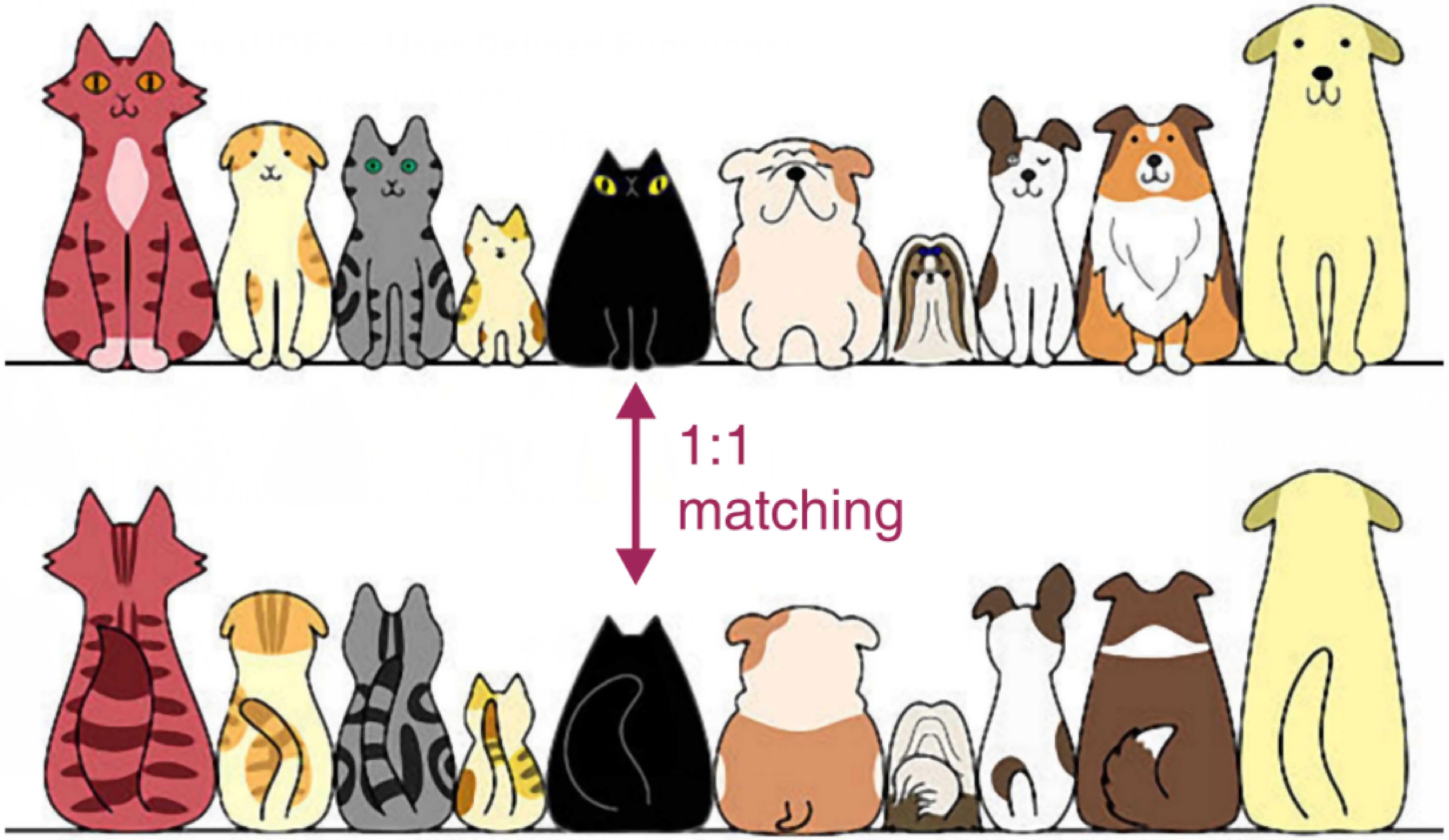
1. Operators
2. **Functions (UDFs = User Defined Functions)**
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)



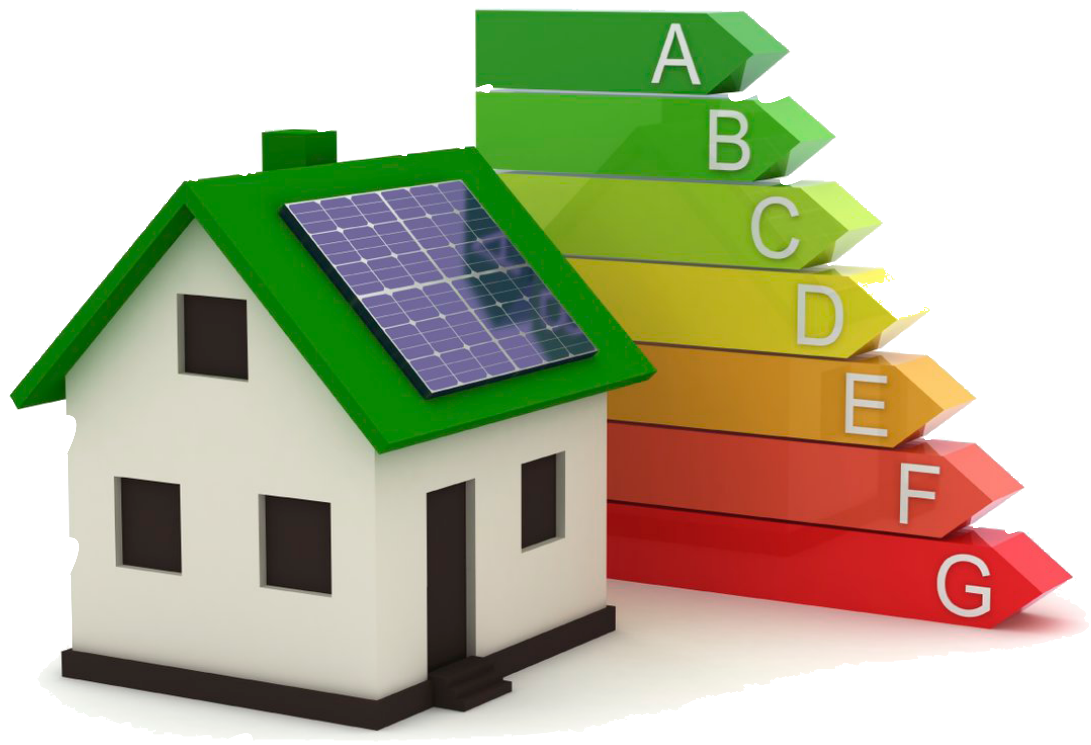
[10/Oct/2000:13:55:36-0700]

UDF:**day** UDF:**month** UDF:**year**

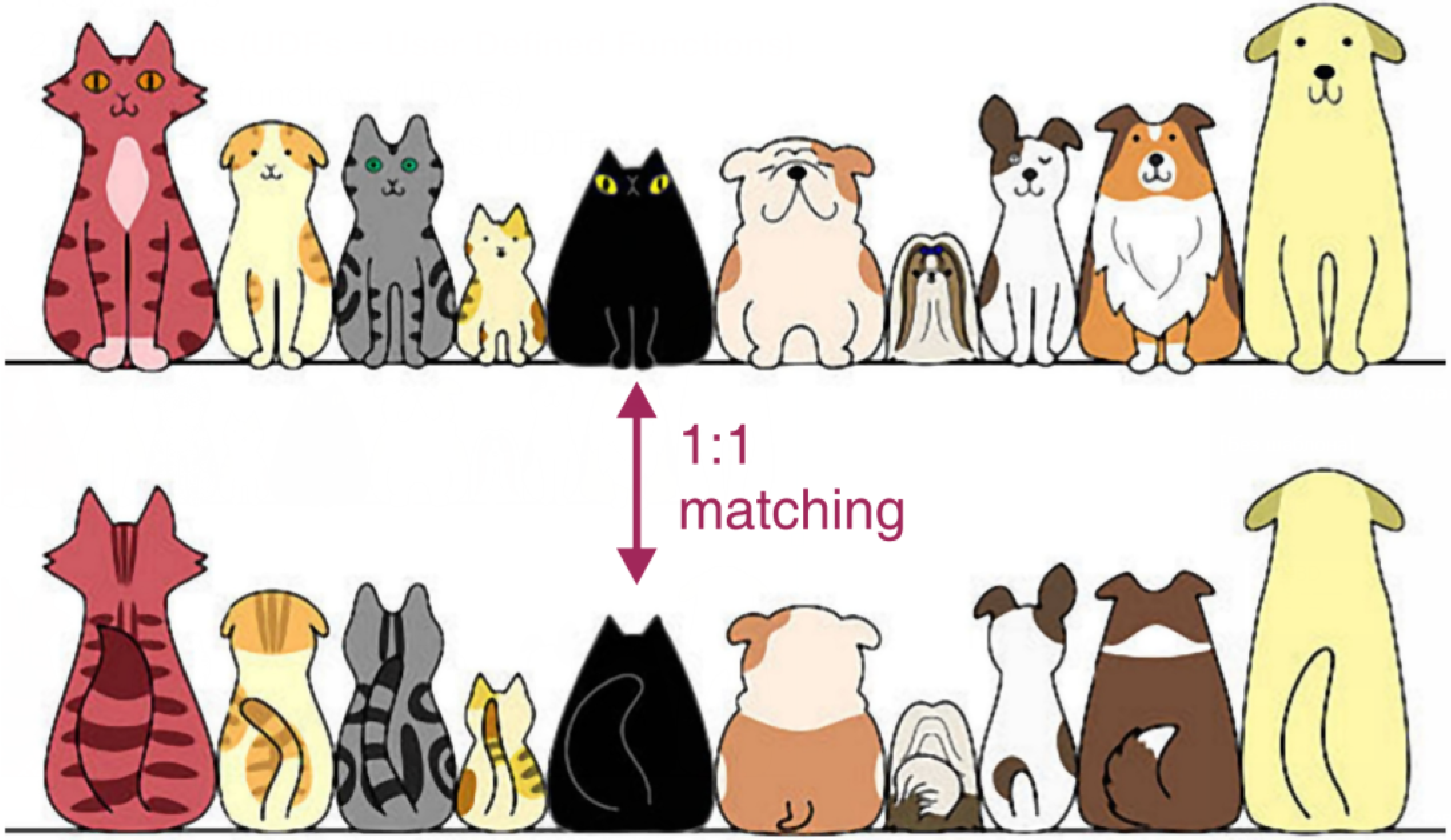
1. Operators
2. **Functions (UDFs = User Defined Functions)**
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

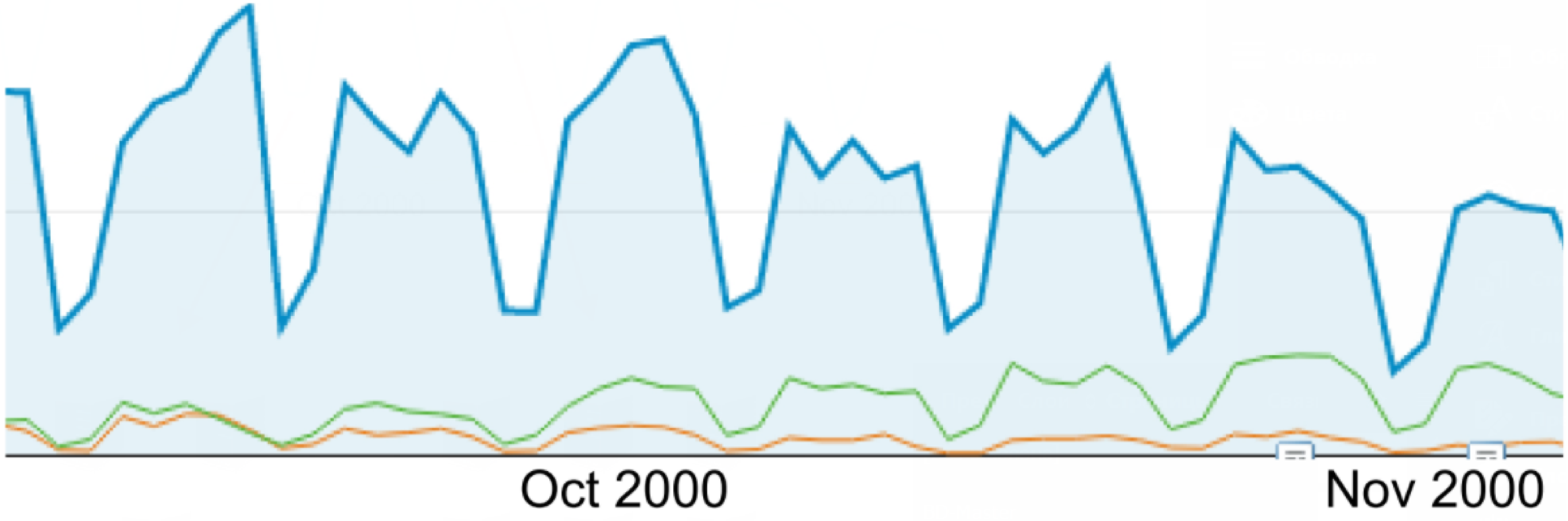


1. Operators
2. **Functions (UDFs = User Defined Functions)**
3. Aggregate functions (UDAFs)
4. Table-generating functions (UDTFs)

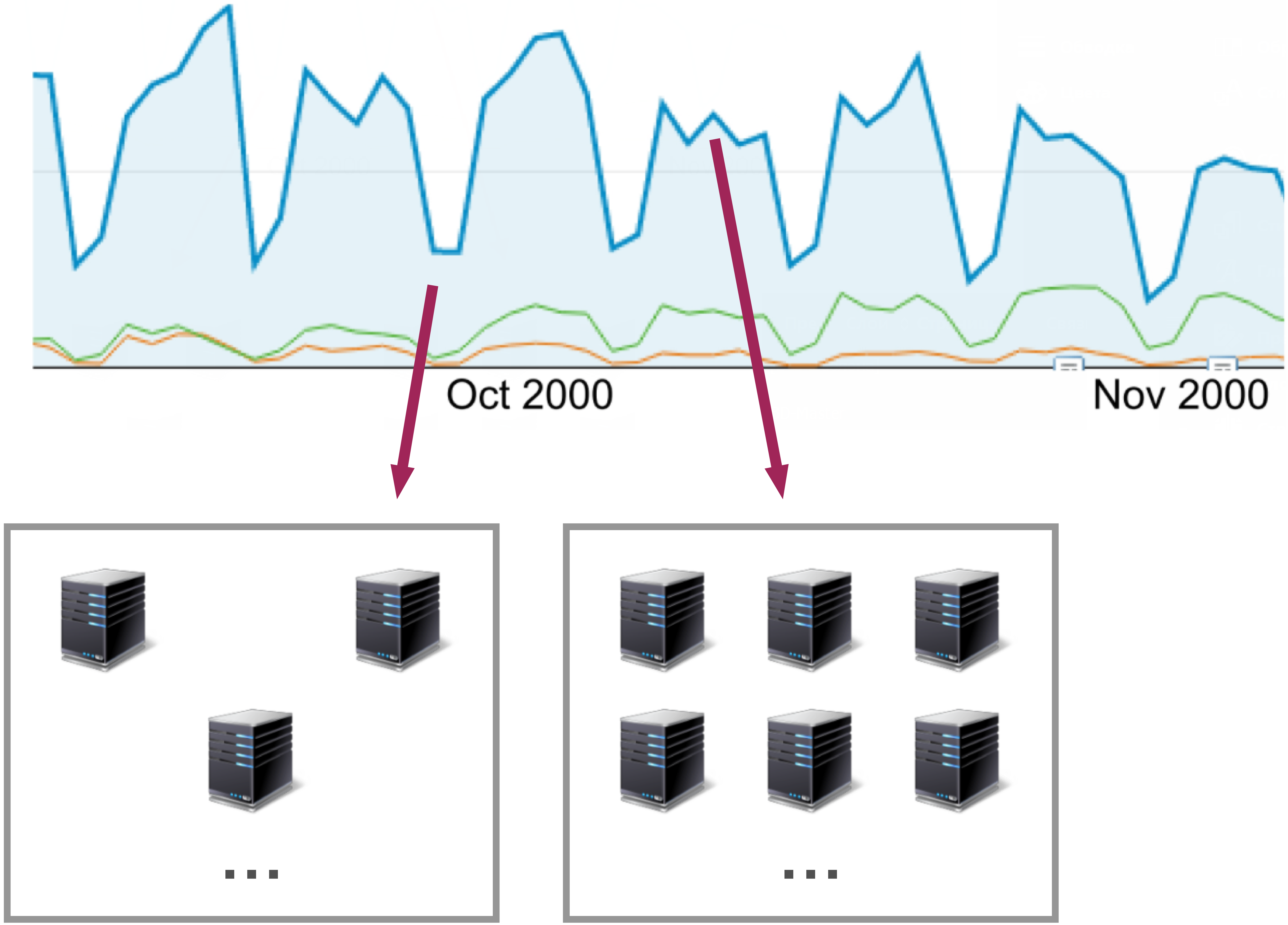


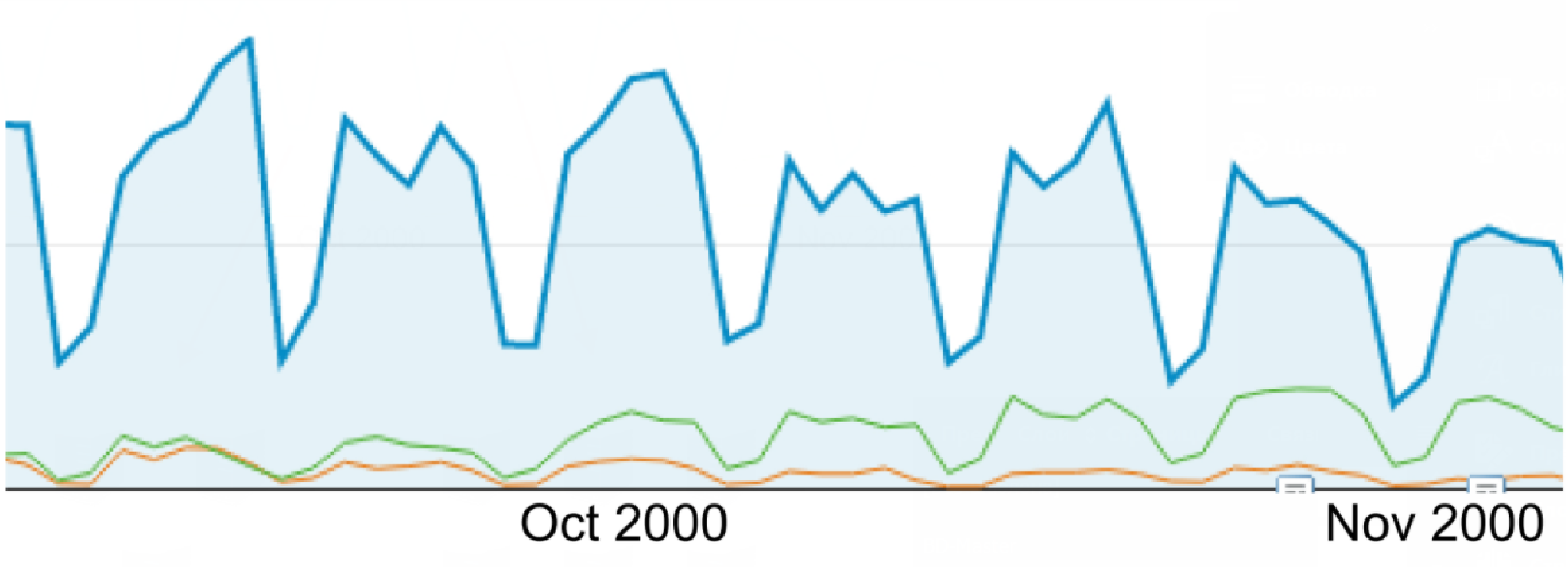
efficiency:
Map Phase




Oct 2000
Nov 2000

...


Oct 2000
Nov 2000
...
...



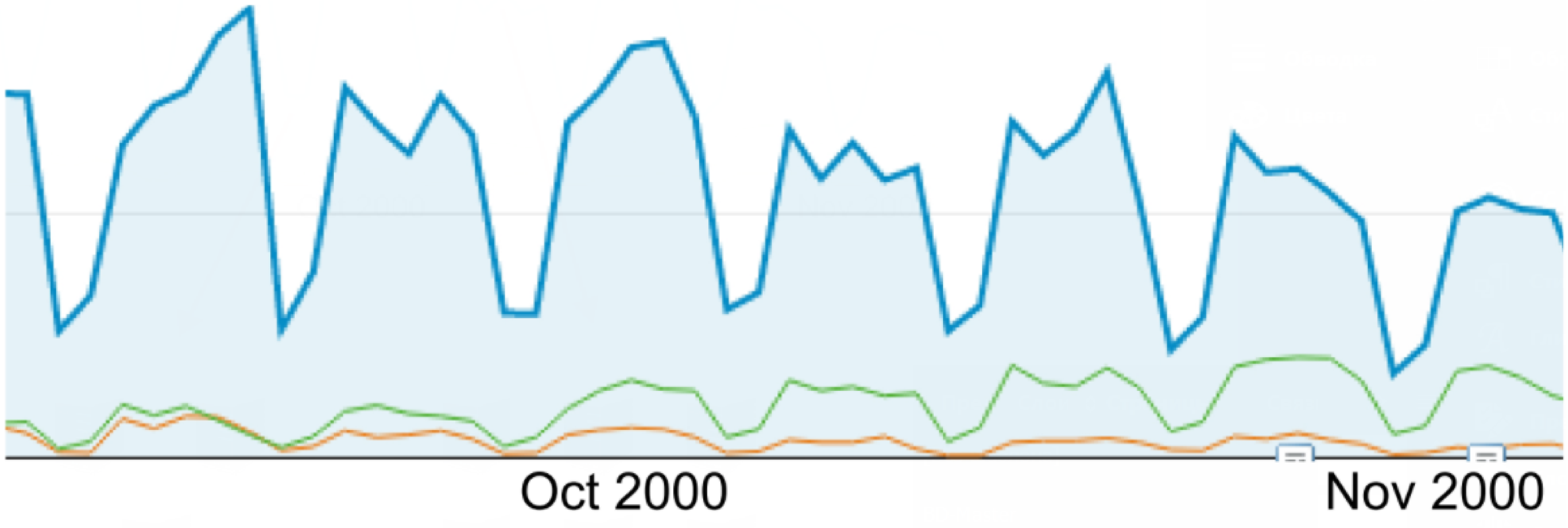
...

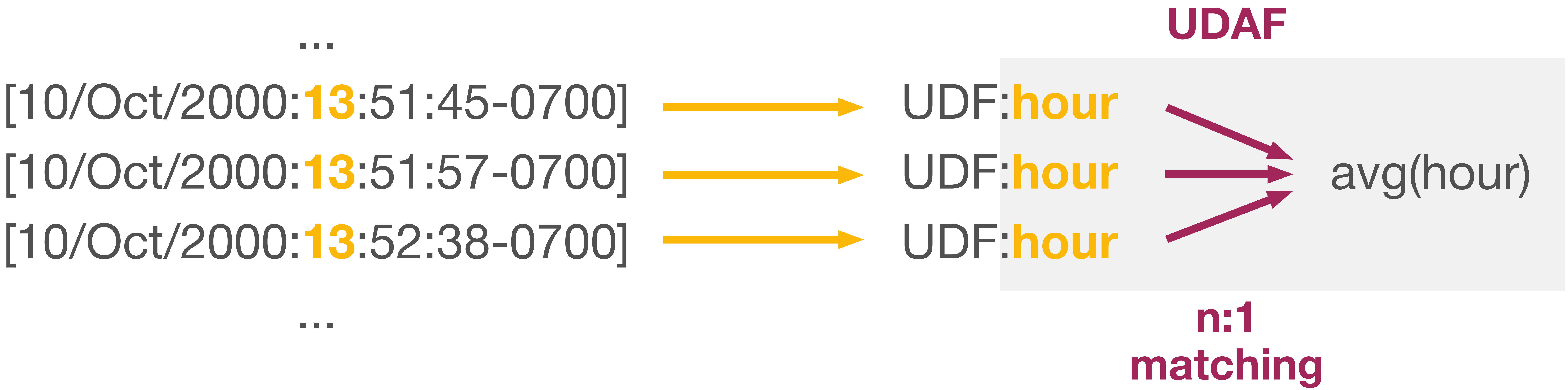
[10/Oct/2000:**13**:51:45-0700] → UDF:**hour**

[10/Oct/2000:**13**:51:57-0700] → UDF:**hour**

[10/Oct/2000:**13**:52:38-0700] → UDF:**hour**

...





UDAF

...

[10/Oct/2000:13:51:45-0700] → UDF:hour

[10/Oct/2000:13:51:57-0700] → UDF:hour

[10/Oct/2000:13:52:38-0700] → UDF:hour

...

avg(hour)

n:1 matching

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

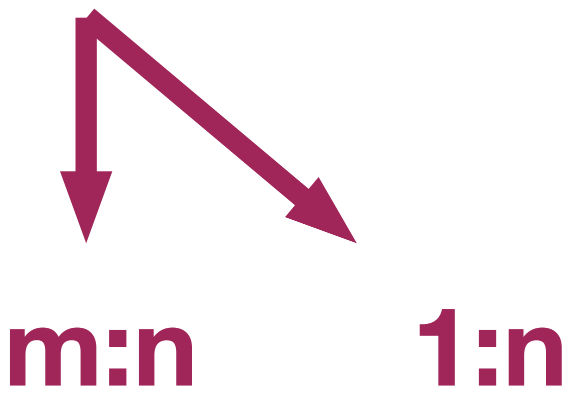
??? 4. Table-generating functions (UDTFs)

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

??? 4. Table-generating functions (UDTFs)



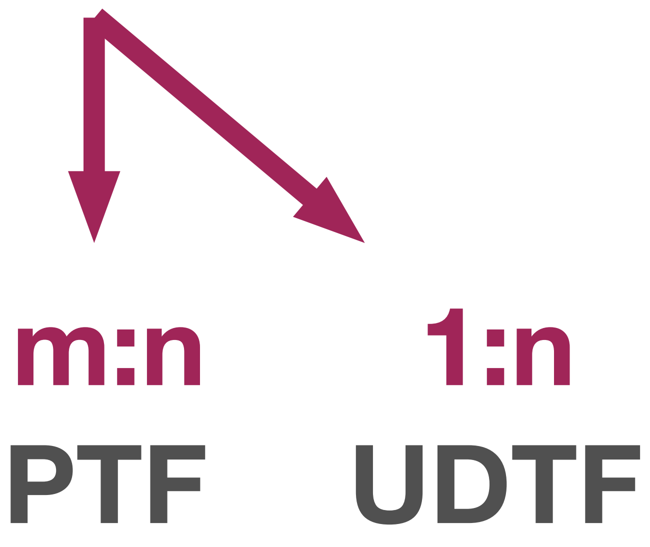
m:n **1:n**

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

??? 4. Table-generating functions (UDTFs)



m:n
PTF

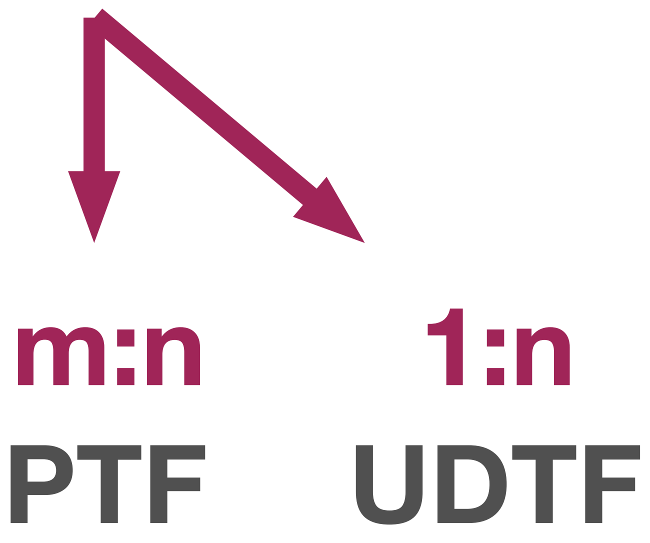
1:n
UDTF

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

??? 4. Table-generating functions (UDTFs)



m:n 1:n

PTF **UDTF**

- explode
- json_tuple
- parse_url_tuple
- posexplode
- stack

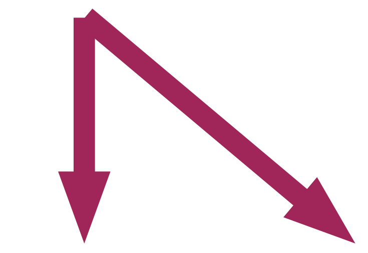
JAR third-party libraries

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

??? 4. Table-generating functions (UDTFs)

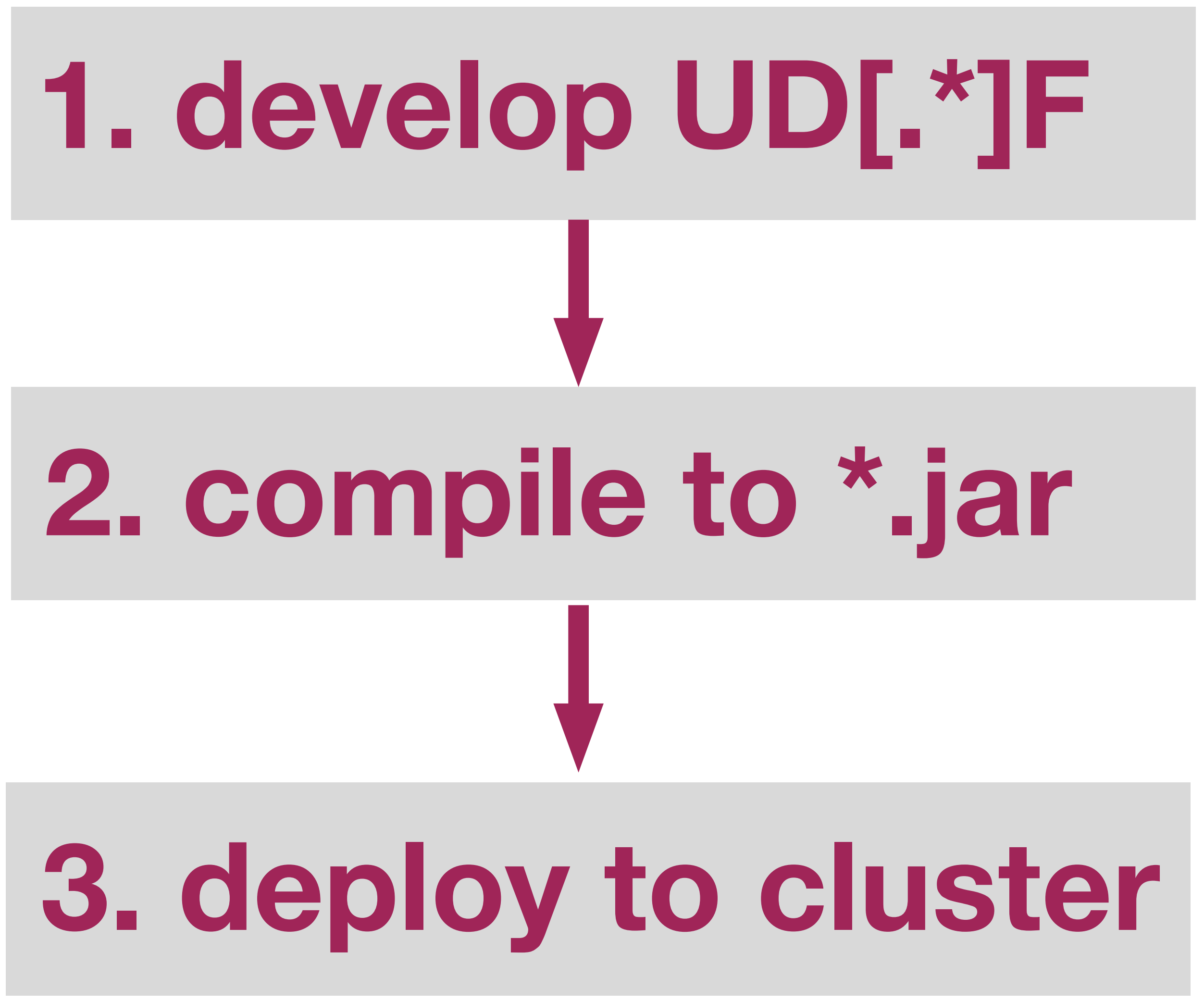


m:n **1:n**

PTF **UDTF**

- explode
- json_tuple
- parse_url_tuple
- posexplode
- stack

third-party libraries



1. develop UD[.*]F

2. compile to *.jar

3. deploy to cluster

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

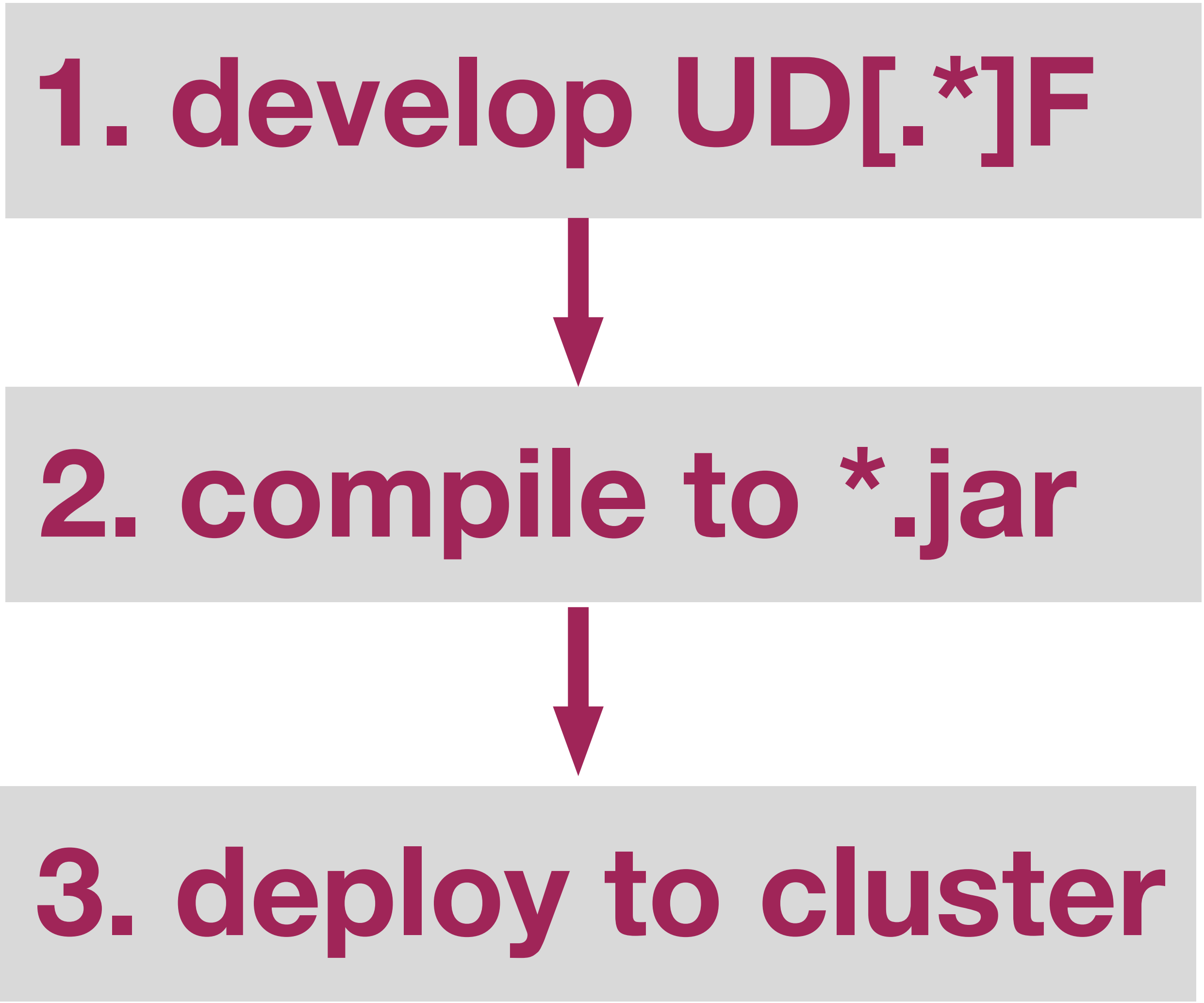
??? 4. Table-generating functions (UDTFs)

m:n **PTF**

1:n **UDTF**

- explode
- json_tuple
- parse_url_tuple
- posexplode
- stack

third-party libraries


1. develop UD[.*]F

2. compile to *.jar

3. deploy to cluster

```
hive> add jar /path/to/lib.jar;
```

place into Distributed Cache

Temporary Functions

```
hive> add jar /path/to/lib.jar;
hive> create temporary function func_name as "java.class.name";
hive> select func_name(...) ...;
...
hive> drop temporary function func_name;
```

Temporary Functions

```
hive> add jar /path/to/lib.jar;
hive> create temporary function func_name as "java.class.name";
hive> select func_name(...) ...;
...
hive> drop temporary function func_name;
```

Permanent Functions

```
hive> create function [db_name.]func_name as "java.class.name"
[USING JAR "/path/to/lib.jar"];
hive> select func_name(...) ...;
...
hive> drop function func_name;
```

1. Operators

n:1 2. Functions (UDFs = User Defined Functions)

1:1 3. Aggregate functions (UDAFs)

??? 4. Table-generating functions (UDTFs)

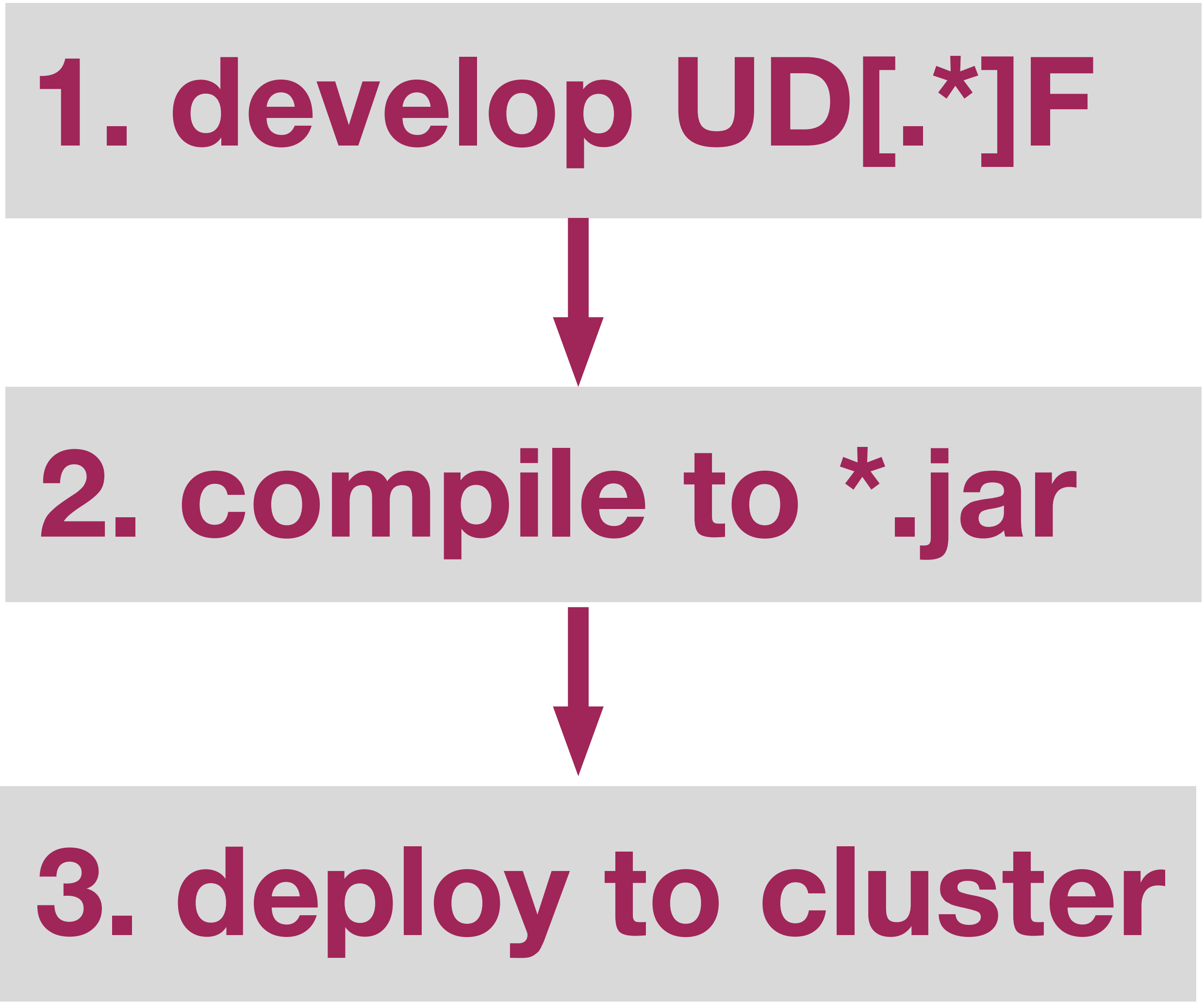
Map
Reduce
Map / Reduce

m:n **1:n**

PTF **UDTF**

- explode
- json_tuple
- parse_url_tuple
- posexplode
- stack

JAR third-party libraries



1. develop UD[.*]F

2. compile to *.jar

3. deploy to cluster

```
hive> add jar /path/to/lib.jar;
```

place into Distributed Cache

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. **Table-generating functions (UDTFs)**

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. **Table-generating functions (UDTFs)**

Table "Management":

- manager_name (STRING)
- direct_reports (**ARRAY**<STRING>)

Join ?

Table "Employees":

- name (STRING)
- surname (STRING)
- email (STRING)
- ...

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. **Table-generating functions (UDTFs)**

```
Table "Management":
  •  manager_name (STRING)
  •  direct_reports (ARRAY<STRING>)
```

→

```
SELECT explode(direct_reports)
  as employee
FROM Management;
```

Join!

```
Table "Employees":
  •  name (STRING)
  •  surname (STRING)
  •  email (STRING)
  •  ...
```

1. Operators
2. Functions (UDFs = User Defined Functions)
3. Aggregate functions (UDAFs)
4. **Table-generating functions (UDTFs)**

```
Table "Management":
  •  manager_name (STRING)
  •  direct_reports (ARRAY<STRING>)
```

→

```
SELECT explode(direct_reports)
  as employee
FROM Management;
```

Join!

```
Table "Employees":
  •  name (STRING)
  •  surname (STRING)
  •  email (STRING)
  •  ...
```

SELECT
  ~~manager_name~~,
  explode(direct_reports)
    as employee
FROM Management;

```
SELECT
 ~~manager_name~~,
 explode(direct_reports)
   as employee
FROM Management;
```

```
SELECT manager_name, employee
FROM Management
 LATERAL VIEW explode(direct_reports) lateral_table
 AS employee
```

```
EXPLAIN SELECT manager_name, employee
FROM Management
   LATERAL VIEW explode(direct_reports) lateral_table
   AS employee

STAGE PLANS:
...
                Select Operator
                ...
                 UDTF Operator
                   Statistics: ...
                   function name: explode
                   Lateral View Join Operator
                    outputColumnNames: ...
                    Statistics: ...
                    ...
```

Summary

Summary

- You can **explain** what UDF, UDAF and UDTF are and **how to use** them

Summary

- You can **explain** what UDF, UDAF and UDTF are and **how to use** them
- You know how to **use** SHOW statement list functions and DESCRIBE statement to get their docstrings

Summary

- You can **explain** what UDF, UDAF and UDTF are and **how to use** them
- You know how to **use** SHOW statement list functions and DESCRIBE statement to get their docstrings
- You can **use** LATERAL VIEW statement to merge output from UDTF

Summary

- You can **explain** what UDF, UDAF and UDTF are and **how to use** them
- You know how to **use** SHOW statement list functions and DESCRIBE statement to get their docstrings
- You can **use** LATERAL VIEW statement to merge output from UDTF
- You can **use** third-party UD[.*]F libraries in Hive and **explain** how they are distributed over the cluster

Summary

- You can **explain** what UDF, UDAF and UDTF are and **how to use** them
- You know how to **use** SHOW statement list functions and DESCRIBE statement to get their docstrings
- You can **use** LATERAL VIEW statement to merge output from UDTF
- You can **use** third-party UD[.*]F libraries in Hive and **explain** how they are distributed over the cluster

See: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF
See: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView