

OpenAtlas - Feature #420

SQL for date values (Data file attached)

2015-05-16 14:33 - Alexander Watzinger

Status:	Closed	Start date:	2015-05-16
Priority:	Low	Estimated time:	0.00 hour
Assignee:	Stefan Eichert		
Category:			
Target version:	0.7.0		
Description			
Daniel testet 0.6.0 and was so kind to enter data with datevalues for testing. Data file attached.			
Format for overviews: earliest begin and latest end year			
SQL for:			
- actor overview, list with two date columns: http://openatlas.local/admin/actor			
- actor view, dates in relation list e.g. http://medcon-dev.craws.net/admin/actor/view/id/107			

History

#1 - 2015-05-18 13:43 - Alexander Watzinger

- Description updated

#2 - 2015-05-22 11:08 - Stefan Eichert

Query for actors and birth or first appearance (begin of the timespan):

```
SELECT DISTINCT ON (name)
  actor.id,
  actor.name,
  actor.description,
  timeprimitive.value_timestamp as firstdate
FROM
  crm.entity actor,
  crm.link,
  crm.entity timeprimitive
WHERE
  actor.id = link.domain_id AND
  link.range_id = timeprimitive.id AND
  link.property_id IN (141,143) AND --birth or first appearance
  actor.class_ID IN (21, 39, 68) -- person, group or legal body
ORDER BY actor.name, timeprimitive.value_timestamp
```

Query for actors and death or last appearance (end of the timespan):

```
SELECT DISTINCT ON (name)
  actor.id,
  actor.name,
  actor.description,
  timeprimitive.value_timestamp as lastdate
FROM
  crm.entity actor,
  crm.link,
  crm.entity timeprimitive
WHERE
  actor.id = link.domain_id AND
  link.range_id = timeprimitive.id AND
  link.property_id IN (142,144) AND --birth or first appearance
```

```
actor.class_ID IN (21, 39, 68) -- person, group or legal body
ORDER BY actor.name, timeprimitive.value_timestamp desc
```

#3 - 2015-05-22 12:41 - Stefan Eichert

This query returns one list that contains one record for each actor with the following information:

id, name, firstdate and lastdate

it shows one record for each actor even if the chronological information is not recorded. If only one (eg. only begin or only end) is known, the other field is left empty. If neither begin nor end are known both are left empty.

For the begin date the first value is shown, for the end date the last value

```
WITH sortbyname AS(
WITH alldates AS(
--alldates returns one or more record for each actors with or without begin/end dates with id, name, firstdate
and lastdate.
--the number of rows per actor depends on the various possibilites (e.g. no dates at all, only one begin, from
-to values for end ...)
```

```
WITH datetable AS(
WITH actortimelist AS (
```

```
SELECT DISTINCT ON (name) -- this query returns a row for each actor that has a begin/birth date and shows th
e first value for begin/birth
```

```
actor.id,
actor.name,
actor.description,
timeprimitive.value_timestamp,
timeprimitive.class_id,
link.property_id
```

```
FROM
crm.entity actor,
crm.link,
crm.entity timeprimitive
```

```
WHERE
actor.id = link.domain_id AND
link.range_id = timeprimitive.id AND
link.property_id IN (141,143) --birth or first appearance
ORDER BY actor.name, timeprimitive.value_timestamp)
```

```
SELECT Distinct on (name) -- this select statement uses the previous query and adds the last value from the en
d/death date as an additional row
```

```
crm.entity.id,
crm.entity.name,
actortimelist.value_timestamp AS firstdate,
timeprimitive.value_timestamp AS lastdate
```

```
FROM
crm.entity actor,
crm.link,
crm.entity timeprimitive,
actortimelist RIGHT JOIN crm.entity ON crm.entity.id = actortimelist.id
```

```
WHERE
actor.id = link.domain_id AND
link.range_id = timeprimitive.id AND
link.property_id IN (142,144) AND --death or last appearance
actortimelist.id = actor.id
ORDER BY name, lastdate desc)
```

```
select * from datetable -- datetable now shows all actors that have a begin/birth AND end/death date (first
value of begin and last value of end timespan)
```

```
UNION ALL
```

```
SELECT -- this query returns the first values for begin/birth. Necessary if for example only begin and no end
dates are recorded, such cases would not show up in datetable (see above)
```

```
actor.id,
actor.name,
timeprimitive.value_timestamp as firstdate,
NULL AS lastdate
```

```
FROM
crm.entity actor,
crm.link,
```

```

    crm.entity timeprimitive
WHERE
    actor.id = link.domain_id AND
    link.range_id = timeprimitive.id AND
    link.property_id IN (141,143) --birth or first appearance

UNION ALL

SELECT -- this query returns the last values for end/death. Necessary if for example only end and no begin dates are recorded, such cases would not show up in datetable (see above)
    actor.id,
    actor.name,
    NULL AS firstdate,
    timeprimitive.value_timestamp as lastdate
FROM
    crm.entity actor,
    crm.link,
    crm.entity timeprimitive
WHERE
    actor.id = link.domain_id AND
    link.range_id = timeprimitive.id AND
    link.property_id IN (142,144) --death or last appearance

UNION ALL

SELECT -- this query returns the NULL values for begin/birth and death/end. Necessary if for example neither begin nor end dates are recorded, such cases would not show up in datetable (see above)
    id,
    name,
    NULL AS firstdate,
    NULL AS lastdate
FROM
    crm.entity
WHERE
    class_id IN (21, 39, 68))

SELECT DISTINCT ON (name, id) -- select only the record with the most complete information on dates and leave other possibilities aside
    id,
    name,
    firstdate,
    lastdate
FROM
    alldates
    ORDER BY id, name, firstdate NULLS LAST, lastdate desc NULLS LAST) -- sort the result so the most complete information is in the first row

SELECT * FROM sortbyname
ORDER BY name -- sort results by name or whatever sorting you want (exchange sort expression if desired)

```

#4 - 2015-05-25 17:49 - Alexander Watzinger

- Status changed from Assigned to Closed

And that was the easy part ;) dates in relationships are even more complicated because they are linked to a property ...

Thanks for your efforts, it gave me some interesting ideas. Here the approach I implemented (with changes for performance, readability and consistency):

```
SELECT min(date_part('year', d.value_timestamp)) AS first FROM crm.entity e
JOIN crm.link l ON e.id = l.domain_id
JOIN crm.entity d ON l.range_id = d.id
JOIN crm.property p ON l.property_id = p.id
WHERE p.code IN ('OA1', 'OA3') AND e.id = :entity_id";
```

In this case:

- an entity_id is passed as param (returns only dates for one entity but could be transformed easily)
- joins instead of "where one_table.id = second_table.id" (much faster)
- used "p.code IN ('OA1', 'OA2', 'OA3', 'OA4')" instead of ids because we can't trust ids
- for the last date replace min with max and OA1/OA3 with OA2/OA4

It's implemented for actors and places, we will improve/adapt as we go, closing ticket.

#5 - 2015-05-25 21:37 - Alexander Watzinger

fixed above sql to get different values for first and last, otherwise the result could be misleading

#6 - 2015-06-15 14:01 - Alexander Watzinger

- Project changed from 19 to OpenAtlas
- Category deleted (37)
- Target version deleted (33)

#7 - 2015-06-23 20:43 - Alexander Watzinger

- Target version set to 0.7.0

Files

medcon_dev.sql.zip	169 KB	2015-05-16	Alexander Watzinger
--------------------	--------	------------	---------------------