```
-- generating indiv_dist and related (tragedy index)

-- 1) ////////////////////////////////////////////////////////////////////////
delete from trag_text;
insert into trag_text(indiv_id,deathtext,eventext)
  SELECT indiv.indiv_id,
  string_agg(DISTINCT COALESCE(event.place_text,' ')||COALESCE(event.notes,' ')||COALESCE(event.cause,' ')||
    COALESCE(event.label,' '),' '), -- as deathtext,
  replace(string_agg(DISTINCT COALESCE(event2.place_text,' ')||COALESCE(event2.notes,'
')||COALESCE(event2.cause,' ')||
      COALESCE(event2.label,' '),' '),'null','' ) -- as eventext
  FROM indiv
  LEFT JOIN event ON event.indiv_id = indiv.indiv_id AND event.type_ = 'DEAT'
  LEFT JOIN event AS event2 ON event2.indiv_id = indiv.indiv_id AND event2.type_ = 'EVEN'
  GROUP BY indiv.indiv_id;

-- 2) ////////////////////////////////////////////////////////////////////////
with z as (
select i.indiv_id, p.event_id, e.year, e.year_abt, e.year_est from indiv i
   join particip p on i.indiv_id = p.actor_id
   join event e on p.event_id = e.recno
   where p.role = 'child'
) update indiv i set
  birthyear = z.year,
  birth_abt = z.year_abt,
  best = z.year_est from z
  where z.indiv_id = i.indiv_id
  and birthyear is null and birth_abt is null and best is null;
-- and death data
with z as (
select i.indiv_id, p.event_id, e.year, e.year_abt, e.year_est from indiv i
   join particip p on i.indiv_id = p.actor_id
   join event e on p.event_id = e.recno
   where p.role = 'deceased'
) update indiv i set
  deathyear = z.year,
  death_abt = z.year_abt,
  dest = z.year_est from z
  where z.indiv_id = i.indiv_id
  and deathyear is null and death_abt is null and dest is null;

-- set dest if not dead
UPDATE indiv SET dest = 2017 WHERE deathyear is null and death_abt is null and dest is null;

-- now do tragic computation
UPDATE indiv SET dest = 2099 WHERE dest = 2017;

DROP TABLE tragic;

CREATE TABLE tragic AS
  SELECT indiv.indiv_id,
  SUM(
  CASE
  when target.indiv_id IS NULL AND COALESCE(indiv.deathyear,indiv.death_abt,indiv.dest) -
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) <= 45  then 1
  else 0
  END
  ) as diedyoung,


  SUM(
  (
  CASE
  when edges.relation = 'spouseOf' AND COALESCE(indiv.deathyear,indiv.death_abt,indiv.dest) -
COALESCE(target.deathyear,target.death_abt,target.dest) >= 20 then 1
  when edges.relation = 'siblingOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) >
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(target.birthyear,target.birth_abt,target.best) <= 12 then 1
  when edges.relation = 'childOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) >
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(target.birthyear,target.birth_abt,target.best) <= 12 then 1
  when edges.relation = 'childOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) <
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) <= 12 then 1
  else 0
  END
  )
  ) as total,
  ''||


  SUM(
  (
  CASE
  when edges.relation = 'spouseOf' AND COALESCE(indiv.deathyear,indiv.death_abt,indiv.dest) -
COALESCE(target.deathyear,target.death_abt,target.dest) >= 20 then 1
  else 0
  END
  )
  )||','||


  SUM(
  (
  CASE
  when edges.relation = 'siblingOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) >
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(target.birthyear,target.birth_abt,target.best) <= 12 then 1
  else 0
  END
  )
  )||','||
  SUM(
  (
  CASE
  when edges.relation = 'childOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) >
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(target.birthyear,target.birth_abt,target.best) <= 12 then 1
  else 0
  END
  )
  )||','||
  SUM(
  (
  CASE
  when edges.relation = 'childOf' AND COALESCE(target.birthyear,target.birth_abt,target.best) <
COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) AND COALESCE(target.deathyear,target.death_abt,target.dest)
- COALESCE(indiv.birthyear,indiv.birth_abt,indiv.best) <= 12 then 1
  else 0
  END
  )
  ) as trarray
  FROM indiv
  LEFT JOIN edges ON indiv.indiv_id IN (edges.source, edges.target)
  LEFT JOIN indiv as target ON target.indiv_id IN (edges.source, edges.target) AND target.indiv_ID <>
indiv.indiv_id
  GROUP BY indiv.indiv_id;


UPDATE indiv SET dest = 2017 WHERE dest = 2099;


-- 3) /////////////////////////////////////////////////////////////////////////
-- remove ghost indiv_dist rows (no public.indiv row)
delete from indiv_dist where indiv_id not in (select indiv_id from indiv);
-- need to add records to indiv_dist for new indiv records
insert into indiv_dist(indiv_id,odnb_id)
  select indiv_id, odnb from indiv i where i.indiv_id not in (select indiv_id from indiv_dist);
-- no nulls allowed, even though we're not computing these
update indiv_dist set centrality = 0 where centrality is null;
update indiv_dist set inbred = 0 where inbred is null;
-- 3a) ///////////////////////////////////////////////////////////////
UPDATE indiv_dist
SET trarray =
'['||(
CASE
when LOWER(deathtext) LIKE '%wounds%' OR LOWER(deathtext) LIKE '%battle%' OR LOWER(deathtext) LIKE '%killed in
action%' OR LOWER(deathtext) LIKE '%cwgc.org%' then 1
when LOWER(deathtext) LIKE '%hanged%' OR LOWER(deathtext) LIKE '%shot%' OR LOWER(deathtext) LIKE '%executed%' OR
LOWER(deathtext) LIKE '%beheaded%' OR LOWER(deathtext) LIKE '%tower hill%' OR LOWER(deathtext) LIKE '%tyburn%'
then 1
when LOWER(deathtext) LIKE '%murdered%' OR LOWER(deathtext) LIKE '%stabbed%' OR LOWER(deathtext) LIKE
'%suicide%' OR LOWER(deathtext) LIKE '%killed herself%' OR LOWER(deathtext) LIKE '%killed himself%' then 1
else diedyoung
END
)||
','||tragic.trarray||','||
(
CASE
when LOWER(eventext) LIKE '%insane%' OR LOWER(eventext) LIKE '%breakdown%' OR LOWER(eventext) LIKE '%lunatic%'
then 1
else 0
END
)||']',


tragedy = total +
(
CASE
```

```
when LOWER(deathtext) LIKE '%wounds%' OR LOWER(deathtext) LIKE '%battle%' OR LOWER(deathtext) LIKE '%killed in
action%' OR LOWER(deathtext) LIKE '%cwgc.org%' then 1
when LOWER(deathtext) LIKE '%hanged%' OR LOWER(deathtext) LIKE '%shot%' OR LOWER(deathtext) LIKE '%executed%' OR
LOWER(deathtext) LIKE '%beheaded%' OR LOWER(deathtext) LIKE '%tower hill%' OR LOWER(deathtext) LIKE '%tyburn%'
then 1
when LOWER(deathtext) LIKE '%murdered%' OR LOWER(deathtext) LIKE '%stabbed%' OR LOWER(deathtext) LIKE
'%suicide%' OR LOWER(deathtext) LIKE '%killed herself%' OR LOWER(deathtext) LIKE '%killed himself%' then 1
else diedyoung
END
)
+
(
CASE
when LOWER(eventext) LIKE '%insane%' OR LOWER(eventext) LIKE '%breakdown%' OR LOWER(eventext) LIKE '%lunatic%'
then 1
else 0
END
)

FROM tragic, trag_text
WHERE trag_text.indiv_id = tragic.indiv_id
AND tragic.indiv_id = indiv_dist.indiv_id;

-- 3b) ///////////////////////////////////////////////////////////////
update indiv_dist id set
  children = coalesce(array_length(ef.children,1),0),
  marriage = coalesce(array_length(ef.spouses,1),0)
  from extfamily ef
  where ef.indiv_id = id.indiv_id;

-- 3c) ///////////////////////////////////////////////////////////////
-- create indiv_dist.odnb_id value for new indivs with odnb_id !!!
update indiv_dist id set odnb_id = i.odnb_id from indiv i where i.indiv_id = id.indiv_id and i.odnb_id is not
null;
-- get word count for all odnbers
update indiv_dist id set
  odnb_wordcount = o.words
  from odnbers o
  where id.odnb_id is not null
  and o.odnb_id = id.odnb_id;

-- 4) //////////////////////////////////////////////////////////////
delete from indiv_text;
insert into indiv_text(indiv_id, occutext)
  select e.indiv_id, string_agg(e.label, '. ')
  from event e
  where e.type_ in('OCCU', 'EVEN')
  group by indiv_id;

-- 5) //////////////////////////////////////////////////////////////
DROP table bak.indiv_events;
select * into bak.indiv_events from indiv_events;
delete from indiv_events;
insert into indiv_events(indiv_id, particip_array)
```

```
  with y as (
  with z as ( select e.recno AS event_id,
      array_accum(p.actor_id) as particip_array
      from event e, particip p where e.recno = p.event_id group by e.recno
      order by event_id )
    select  e.type_ as eventtype, e.label as eventlabel, e.place_text as eventplace,
    period_text as eventdate, e.place_id AS place,
    coalesce(year,year_abt,year_est,-1) as Year,
    case when(year_abt is not null OR year_est is not null) then 'roughly'
         when year > 0 then 'known'
         else ''
    end as accuracy,
    z.particip_array as actor
    from z join event e on z.event_id = e.recno
    )
    select i.indiv_id,
    '['||
    array_to_string(
    array_agg('{"eventtype":"'||y.eventtype||'","eventlabel":"'||y.eventlabel||
      '","eventplace":"'||coalesce(y.eventplace,'')||'","eventdate":"'||coalesce(y.eventdate,'')||
      case when y.place IS null then '"'
           ELSE '","place":'||coalesce(y.place::text,'""')
           end ||
      case when i.birthyear IS null then ''
           ELSE ',"year":'||(y.year-i.birthyear) -- age at event if birth known
           end ||

      ',"accuracy":"'||coalesce(y.accuracy,'')||
      '","actor":'||'["'||array_to_string(y.actor,'","')||'"]}'),',')||
    ']' -- as particip_array
    from indiv i join y on i.indiv_id = any(y.actor)
    group by i.indiv_id order by indiv_id -- limit 300

-- 6) ///////////////////////////////////////////////////////////////
-- create extfamily
-- given current 'indiv', 'event', and 'particip' tables
-- ///////////////////////////////////////////////////////////
-- uses helpers:
  -- p_parent(indiv_id,['mother' | 'father'])
  -- p_spouses(indiv_id)
  -- p_children(indiv_id)
  -- p_siblings()
drop table bak.extfamily;
select * into bak.extfamily from extfamily;
delete from extfamily;
-- [run 03May2016, 29952 rows 14+ min.; 01Jun2016 wo/spouse,children, ~14 min]
-- ERROR: null value in column "recno" violates not-null constraint
-- cause: moving/copying table between schemas changed serial sequence to plain integer
-- select * from event where recno is null
insert into extfamily(indiv_id,sex,mother,father,-- spouses,children, -- siblings,
    birthyear,birth_abt,birth_est,deathyear,death_abt,death_est)
  select i.indiv_id, i.sex,
  p_parent(i.indiv_id,'mother'),
  p_parent(i.indiv_id,'father'),
--  p_spouses(indiv_id),
-- p_children(indiv_id),
-- p_siblings(indiv_id), -- depends on children
  i.birthyear,i.birth_abt,i.best,i.deathyear,i.death_abt,i.dest
  from indiv i order by indiv_id; -- limit 120;

-- ///////// run helper functions all run 01Jun2016 ////////////////////////
-- spouses
update extfamily set spouses = p_spouses(indiv_id); -- 10 min.
-- ///////////////////////////////////////////////////////////////
-- children
update extfamily set children = p_children(indiv_id); -- 10 min.
-- ///////////////////////////////////////////////////////////////
-- siblings; depends on children (all rows)
update extfamily set siblings = p_siblings(indiv_id); -- 3.6 min.

-- 7) ////////////////////////////////////////////////////////////////////////
-- edges (recno,target,source,relation)
-- > depends on new extfamily
-- relation [spouseOf, childOf, siblingOf, selfLoop ]
-- [run 03May2016, before: 90270 rows, after: 97914]
drop table bak.edges;
select * into bak.edges from edges;
delete from edges;

-- selfLoop [run 03May2016, +29952; 01Jun2016]
insert into edges(source,target,relation)
  select i.indiv_id, i.indiv_id, 'selfLoop' from indiv i;
-- spouseOf [run 03May2016, +15191; 01Jun2016]
insert into edges(source,target,relation)
  ( with tbl as (
    select ef.indiv_id as source, unnest(ef.spouses) as target, 'spouseOf' as relation
    from extfamily ef order by source
  ) select a.source, a.target, a.relation from tbl a, tbl b
    WHERE  (a.source, a.target) = (b.target, b.source)
    AND    a.target > a.source
  );
-- childOf [run 03May2016, +34086; 01Jun2016]
insert into edges(source,target,relation)
  ( select ef.indiv_id as source, unnest(children) as target, 'childOf' as relation
  from extfamily ef order by source );
-- siblingOf [run 03May2016, +18685; 01Jun2016]
insert into edges(source,target,relation)
  ( with tbl as (
    select ef.indiv_id as source, unnest(ef.siblings) as target, 'siblingOf' as relation
    from extfamily ef order by source
  ) select a.source, a.target, a.relation from tbl a, tbl b
    WHERE  (a.source, a.target) = (b.target, b.source)
    AND    a.target > a.source
  );


-- 8) //////////////////////////////////////////////////////////////
UPDATE indiv_dist SET trarray =
  '['||(
  CASE
  when LOWER(deathtext) LIKE '%wounds%' OR LOWER(deathtext) LIKE '%battle%' OR LOWER(deathtext) LIKE '%killed in
action%' OR LOWER(deathtext) LIKE '%cwgc.org%' then 1
  when LOWER(deathtext) LIKE '%hanged%' OR LOWER(deathtext) LIKE '%shot%' OR LOWER(deathtext) LIKE '%executed%'
OR LOWER(deathtext) LIKE '%beheaded%' OR LOWER(deathtext) LIKE '%tower hill%' OR LOWER(deathtext) LIKE
'%tyburn%' then 1
  when LOWER(deathtext) LIKE '%murdered%' OR LOWER(deathtext) LIKE '%stabbed%' OR LOWER(deathtext) LIKE
'%suicide%' OR LOWER(deathtext) LIKE '%killed herself%' OR LOWER(deathtext) LIKE '%killed himself%' then 1
  else diedyoung
  END
  )||
  ','||tragic.trarray||','||
  (
  CASE
  when LOWER(eventext) LIKE '%insane%' OR LOWER(eventext) LIKE '%breakdown%' OR LOWER(eventext) LIKE '%lunatic%'
then 1
  else 0
  END
  )||']',


  tragedy = total +
  (
  CASE
  when LOWER(deathtext) LIKE '%wounds%' OR LOWER(deathtext) LIKE '%battle%' OR LOWER(deathtext) LIKE '%killed in
action%' OR LOWER(deathtext) LIKE '%cwgc.org%' then 1
  when LOWER(deathtext) LIKE '%hanged%' OR LOWER(deathtext) LIKE '%shot%' OR LOWER(deathtext) LIKE '%executed%'
OR LOWER(deathtext) LIKE '%beheaded%' OR LOWER(deathtext) LIKE '%tower hill%' OR LOWER(deathtext) LIKE
'%tyburn%' then 1
  when LOWER(deathtext) LIKE '%murdered%' OR LOWER(deathtext) LIKE '%stabbed%' OR LOWER(deathtext) LIKE
'%suicide%' OR LOWER(deathtext) LIKE '%killed herself%' OR LOWER(deathtext) LIKE '%killed himself%' then 1
  else diedyoung
  END
  )
  +
  (
  CASE
  when LOWER(eventext) LIKE '%insane%' OR LOWER(eventext) LIKE '%breakdown%' OR LOWER(eventext) LIKE '%lunatic%'
then 1
  else 0
  END
  )


  FROM tragic, trag_text
  WHERE trag_text.indiv_id = tragic.indiv_id
  AND tragic.indiv_id = indiv_dist.indiv_id;


-- 8a) //////////////////////////////////////////////////////////////
-- requires updated extfamily
-- put number of children, marriages into indiv_dist
update indiv_dist id set
```

```
  children = coalesce(array_length(ef.children,1),0),
  marriage = coalesce(array_length(ef.spouses,1),0)
  from extfamily ef
  where ef.indiv_id = id.indiv_id;

-- 8b) //////////////////////////////////////////////////////////////
-- odnb_wordcount
-- !!! relies on indiv.odnb value for new indiv records !!!
-- NOTE: json.odnb in code refers to indiv_dist.odnb, NOT indiv.odnb_id
update indiv_dist id set
--   odnb_id = o.odnb_id,
  odnb_wordcount = o.words
  from odnbers o
  where id.odnb_id is not null
  and o.odnb_id = id.odnb_id;

-- 9) //////////////////////////////////////////////////////////////
-- select p_odnb();
begin
        FOR _id IN SELECT indiv_id FROM indiv_dist LOOP
           begin
         with y as (
         with z as (
         SELECT seq, id1 AS source, id2 AS target, cost FROM pgr_kdijkstraCost(
             'SELECT recno::int4 as id, right(source,-1)::int4 as source,
            right(target,-1)::int4 as target, 1::float8 as cost FROM edges where relation != ''selfLoop''',
             right(_id,-1)::int4, (select array_agg(right(indiv_id,-1)::int) AS arr from indiv where odnb_id is
not null) , false, false)
         ) select _id as indiv_id, min(cost) as foo from z
         ) update indiv_dist id set odnb = y.foo from y where y.indiv_id = id.indiv_id;
         exception when others then
            -- no problem, these people are not in public.indiv
            RAISE NOTICE 'error on (%)', _id;
            update indiv_dist id set odnb = -1 where _id = id.indiv_id;
           END;
        END LOOP;

RAISE NOTICE 'Done';
END;

-- 10) //////////////////////////////////////////////////////////////
-- select p_parentless();
DECLARE _id varchar;
begin
        -- update z_edges for this calculation
        delete from z_edges;
        insert into z_edges(source,target,relation)
               select right(target,-1)::int4, right(source,-1)::int4, relation from edges where relation in
('childOf');
        FOR _id IN SELECT indiv_id FROM indiv_dist where recno < 1001 LOOP
      begin
         with y as (
         with z as (
         SELECT seq, id1 AS source, id2 AS target, cost FROM pgr_kdijkstraCost(
             'SELECT id::int4, source::int4 as source, target::int4 as target, 1::float8 as cost
            FROM z_edges',
            right(_id,-1)::int4, (
            select array_agg(distinct(alledges.source))::int4[] as arr from
            (select source from z_edges union select target from z_edges ) as alledges
            where source not in (select distinct(source) from z_edges)
            )
            , true, FALSE)
         ) select max(cost) as foo from z
         ) update indiv_dist id set parentless = y.foo from y where _id = id.indiv_id;
          exception when others then
            RAISE NOTICE 'error on (%)', _id;
            -- these are people with no parents or children & get a 0
            update indiv_dist id set parentless = 0 where _id = id.indiv_id;
      END;
        END LOOP;

RAISE NOTICE 'Done';
END;

-- 11a) /////////////////////////////////////////////////////////////////
delete from similarity;
insert into similarity(indiv_id,byear,dyear,children,siblings)
  select e.indiv_id, coalesce(i.birthyear,i.birth_abt,i.best),coalesce(i.deathyear,i.death_abt,i.dest),
  coalesce(array_length(children,1),0), coalesce(array_length(siblings,1),0)
  from extfamily e
  join indiv i on e.indiv_id = i.indiv_id;

-- occ 11,019
with z as (
select indiv_id, to_tsvector(array_agg(occu_text)::text) as occ from indiv_occu io
  group by indiv_id
) update similarity s set occ = z.occ from z where s.indiv_id = z.indiv_id;

-- event 11,930
with z as (
select i.indiv_id, to_tsvector(array_agg(e.label)::text) as event from indiv i
  join particip p on i.indiv_id = p.actor_id
  join event e on p.event_id = e.recno
  where e.type_ in ('OCCU','EDUC','EVEN','IMMI','GRAD')
  --and i.indiv_id not in (select indiv_id from similarity)
  group by i.indiv_id
) update similarity s set event = z.event from z where s.indiv_id = z.indiv_id;

-- loc (alternate) 21,762
with z as (
select i.indiv_id, to_tsvector(array_to_string(array_agg(coalesce(pl.admin2,'')||'
'||coalesce(pl.admin1,'')||coalesce(pl.ccode,'')),' ')) as loc
  from indiv i
  join particip p on i.indiv_id = p.actor_id
```

```
  join event e on p.event_id = e.recno
  join place pl on e.place_id = pl.placeid
  --where i.indiv_id not in (select indiv_id from similarity)
  group by i.indiv_id
) update similarity s set loc = z.loc from z where s.indiv_id = z.indiv_id;

-- 11b) ////////////////////////////////////////////////////////////////////
-- select p_simmy();

DECLARE
    _id varchar;
    _ids VARCHAR[];
BEGIN
        delete from test_sims;
        FOR _id IN SELECT indiv_id FROM similarity limit 10 LOOP
        with w as (
        with x as(
        with y as (
        with z as (select indiv_id, ARRAY[byear,dyear,children,siblings] as vitals,
                string_to_array(replace(array_to_string(tsvector2textarray(occ),','),',unknown',''),',') as occ,
                tsvector2textarray(event) as ev, tsvector2textarray(loc) as loc from similarity
        )
        SELECT z.indiv_id,
                smlar(z.vitals, original.vitals) AS sim_vitals,
                smlar(z.occ, original.occ) AS sim_occ,
                smlar(z.ev, original.ev) AS sim_ev,
                smlar(z.loc, original.loc) AS sim_loc
                FROM z,
                (SELECT vitals, occ, ev, loc, indiv_id FROM z WHERE indiv_id = _id LIMIT 1) AS original
                WHERE z.indiv_id != original.indiv_id
                and z.occ is not null
        --      ORDER BY sim_vitals DESC
        --      LIMIT 15;
        ) select indiv_id, unnest(ARRAY[sim_vitals,sim_occ,sim_ev,sim_loc]) from y
        ) select indiv_id, sum(unnest) as sim from x group by indiv_id
                order by sim desc limit 15
        ) insert into test_sims(indiv_id, sim_id) select _id, array_agg(indiv_id) from w;
        --insert into test_sims(indiv_id) select id;
    END LOOP;

    RAISE NOTICE 'Done';
--     RETURN 1;
END;

update indiv set marnm = null where marnm = '';
update indiv set search_names = to_tsvector('english', fullname||' '||coalesce(marnm,'') );
```