```
-- set seed to 0.6;

set search_path to hbi_analytics;
drop table if exists drug_master;
create temp table drug_master as
   select
   gcn,
   ndc,
   label_name,
   active_ingrd as active_ingredient,
   strength,
   measure,
   drug_narx_type_one,
   drug_dea_schedule,
--    bupren_type,
   drug_mat_yn
from hbi_analytics.dim_drug_master
;


----------------------------------------------------------------------------------------------------
-- the following is for adding in synthetic dispensation dates for scoring ORSv2 that might
not be dispensation dates
--
drop table if exists arbitrary_refdates;
create temp table arbitrary_refdates (filled_at date);
insert into arbitrary_refdates values

                    ('2014-01-31'::date),
                    ('2014-02-28'::date),
                    ('2014-03-31'::date),
                    ('2014-04-30'::date),
                    ('2014-05-31'::date),
                    ('2014-06-30'::date),
                    ('2014-07-31'::date),
                    ('2014-08-31'::date),
                    ('2014-09-30'::date),
                    ('2014-10-31'::date),
('2014-11-30'::date),
('2014-12-31'::date),

('2015-01-31'::date),
('2015-02-28'::date),
('2015-03-31'::date),
('2015-04-30'::date),
('2015-05-31'::date),
('2015-06-30'::date),
('2015-07-31'::date),
('2015-08-31'::date),
('2015-09-30'::date),
('2015-10-31'::date),
('2015-11-30'::date),
('2015-12-31'::date),

('2016-01-31'::date),
('2016-02-29'::date),
('2016-03-31'::date),
('2016-04-30'::date),
('2016-05-31'::date),
('2016-06-30'::date),
('2016-07-31'::date),
('2016-08-31'::date),
('2016-09-30'::date),
('2016-10-31'::date),
('2016-11-30'::date),
('2016-12-31'::date),

('2017-01-31'::date),
('2017-02-28'::date),
('2017-03-31'::date),
('2017-04-30'::date),
('2017-05-31'::date),
('2017-06-30'::date),
('2017-07-31'::date),
('2017-08-31'::date),
('2017-09-30'::date),
('2017-10-31'::date),
('2017-11-30'::date),
('2017-12-31'::date),

('2018-01-31'::date),
('2018-02-28'::date),
('2018-03-31'::date),
('2018-04-30'::date),
('2018-05-31'::date),
('2018-06-30'::date),
('2018-07-31'::date),
('2018-08-31'::date),
('2018-09-30'::date),
('2018-10-31'::date),
('2018-11-30'::date),
('2018-12-31'::date),

('2019-01-31'::date),
('2019-02-28'::date),
('2019-03-31'::date),
('2019-04-30'::date),
('2019-05-31'::date),
('2019-06-30'::date),
('2019-07-31'::date),
('2019-08-31'::date),
('2019-09-30'::date),
('2019-10-31'::date),
('2019-11-30'::date),
('2019-12-31'::date),

('2020-01-31'::date),
('2020-02-29'::date),
('2020-03-31'::date),
('2020-04-30'::date),
('2020-05-31'::date),
('2020-06-30'::date),
('2020-07-31'::date),
('2020-08-31'::date),
('2020-09-30'::date),
('2020-10-31'::date),
('2020-11-30'::date),
('2020-12-31'::date),

('2021-01-31'::date),
('2021-02-28'::date),
('2021-03-31'::date),
('2021-04-30'::date),
('2021-05-31'::date),
('2021-06-30'::date),
('2021-07-31'::date),
('2021-08-31'::date),
('2021-09-30'::date),
('2021-10-31'::date),
('2021-11-30'::date),
('2021-12-31'::date),

('2022-01-31'::date),
('2022-02-28'::date),
('2022-03-31'::date),
('2022-04-30'::date),
('2022-05-31'::date),
('2022-06-30'::date),
('2022-07-31'::date),
('2022-08-31'::date),
('2022-09-30'::date),
('2022-10-31'::date),
('2022-11-30'::date),
('2022-12-31'::date),

('2023-01-31'::date),
('2023-02-28'::date),
('2023-03-31'::date),
('2023-04-30'::date),
('2023-05-31'::date),
('2023-06-30'::date),
('2023-07-31'::date),
                    ('2023-08-31'::date),
                    ('2023-09-30'::date),
                    ('2023-10-31'::date),
                    ('2023-11-30'::date),
                    ('2023-12-31'::date),

                    ('2024-01-31'::date),
                    ('2024-02-29'::date),
                    ('2024-03-31'::date),
                    ('2024-04-30'::date),
                    ('2024-05-31'::date),
                    ('2024-06-30'::date),
                    ('2024-07-31'::date),
                    ('2024-08-31'::date),
                    ('2024-09-30'::date)

-- where filled_at > '2016-12-31' and filled_at <= current_date
;



------------------------------------
-- full query (abridged query below)
------------------------------------

-- set search_path to farm;
-- drop table if exists ors_pdmp_data_pull;
-- create temp table ors_pdmp_data_pull sortkey(consolidation_identifier) as
--    select
--    a.*
-- --    quantity / days_supply AS Dose_Per_Day,
-- --    (quantity / days_supply) * strength_numeric AS Total_Dose_Per_Day
-- from (
-- SELECT distinct
--    a.consolidation_identifier,
--    a.patient_id,
--    b.first_name,
--    b.middle_name,
--    b.last_name,
--    b.birthdate,
--    b.gender,
--    k.address_one,
--    k.city,
--    k.state,
--    k.postal_code,
--    c.id as dispensation_id,
--    CAST(c.filled_at AS DATE) AS filled_at,
--    c.refill_number,
--    c.payment_type,
--    c.sold_at,
--    d.authorized_refill_count,
--    dd.gcn,
--    dd.ndc,
--    dd.label_name,
--    dd.active_ingredient,
--    dd.drug_narx_type_one as drug_type,
--    dd.drug_dea_schedule,
-- --    dd.bupren_type,
--    dd.drug_mat_yn,
--    UPPER(e.first_name) AS provider_first_name,
--    UPPER(e.middle_name) AS provider_middle_name,
--    UPPER(e.last_name) AS provider_last_name,
--    UPPER(e.suffix) AS provider_suffix,
--    UPPER(e.address_one) AS provider_address_one,
--    e.address_two AS provider_address_two,
--    e.city AS provider_city,
--    e.state AS provider_state,
--    e.postal_code AS provider_postal_code,
--    e.national_provider_id AS provider_npi,
--    e.dea_number AS provider_dea,
--    f.dea_number AS pharmacy_dea,
--    f.name AS pharmacy_name,
--    f.address_one AS pharmacy_address_one,
--    f.address_two AS pharmacy_address_two,
--    f.city AS pharmacy_city,
--    f.state AS pharmacy_state,
--    f.postal_code AS pharmacy_postal_code,
--    cast(g.quantity as real) as quantity,
--    cast(c.days_supply AS REAL) AS days_supply,
--    dd.strength,
--    CAST(REGEXP_REPLACE(dd.strength, '[^0-9.]', '') AS REAL) AS strength_numeric,
--    dd.measure,
--    g.product_id_type,
--    g.product_id,
--    g.units,
--    g.narx_unit_eq_one AS mme_conversion
--
-- FROM pmp_farm_de.farm.consolidation_identifiers a
--   LEFT JOIN pmp_farm_de.farm.patients b ON a.patient_id = b.id
--   LEFT JOIN pmp_farm_de.farm.patient_addresses k ON a.patient_id = k.patient_id
--   LEFT JOIN pmp_farm_de.farm.dispensations c ON b.id = c.patient_id
--   LEFT JOIN pmp_farm_de.farm.prescriptions d ON d.dispensation_id = c.id
--   LEFT JOIN pmp_farm_de.farm.prescribers e ON e.dispensation_id = c.id
--   LEFT JOIN pmp_farm_de.farm.dispensaries f ON f.dispensation_id = c.id
--   LEFT JOIN pmp_farm_de.farm.drugs g ON g.dispensation_id = c.id
--   LEFT JOIN drug_master dd ON dd.ndc = g.narx_national_drug_code
--
-- where a.consolidation_identifier is not null or a.patient_id is not null
-- -- where (filled_at > '2021-12-12' and filled_at <= '2023-12-12')
-- and (filled_at > '2022-07-01' and filled_at <= current_date)
-- -- WHERE filled_at <= current_date - INTERVAL '6 months'
-- -- and b.birthdate = date '2006-07-18'
-- -- and b.first_name ilike ('%JO%')
-- -- and b.last_name ilike ('%BUCKSAR%')
-- and k.state in ('DE')
-- -- and consolidation_identifier in ('20331863768')
-- -- and active_ingredient in ('HYDROCODONE')
-- -- and active_ingredient in ('TRAMADOL')
-- -- and days_supply = 1
-- -- and quantity <= 1.0
-- -- and (quantity >= 1 and quantity <= 100)
-- -- and (quantity >= 100)
-- ) a
-- ;
```

```
--------------------------------------
-- abridged query for all patients
--------------------------------------

SET search_path TO farm;
DROP TABLE IF EXISTS ors_pdmp_data_pull;
CREATE TEMP TABLE ors_pdmp_data_pull AS
SELECT
   a.consolidation_identifier,
   b.birthdate,
   b.gender,
   c.id AS dispensation_id,
   CAST(c.filled_at AS DATE) AS filled_at,
   cast(c.sold_at as date) as sold_at,
   dd.active_ingredient,
   dd.label_name,
   dd.drug_narx_type_one AS drug_type,
   dd.drug_mat_yn,
   e.dea_number AS provider_dea,
   f.dea_number AS pharmacy_dea,
   CAST(g.quantity AS REAL) AS quantity,
   CAST(c.days_supply AS REAL) AS days_supply,
   g.narx_unit_eq_one AS mme_conversion
FROM pmp_farm_tx.farm.consolidation_identifiers a
LEFT JOIN pmp_farm_tx.farm.patients b ON a.patient_id = b.id
LEFT JOIN pmp_farm_tx.farm.patient_addresses k ON a.patient_id = k.patient_id
LEFT JOIN pmp_farm_tx.farm.dispensations c ON b.id = c.patient_id
LEFT JOIN pmp_farm_tx.farm.prescriptions d ON d.dispensation_id = c.id
LEFT JOIN pmp_farm_tx.farm.prescribers e ON e.dispensation_id = c.id
LEFT JOIN pmp_farm_tx.farm.dispensaries f ON f.dispensation_id = c.id
LEFT JOIN pmp_farm_tx.farm.drugs g ON g.dispensation_id = c.id
LEFT JOIN drug_master dd ON dd.ndc = g.narx_national_drug_code
WHERE filled_at >= '2022-07-01' AND filled_at <= CURRENT_DATE
and b.birthdate = date '1948-05-11'
and b.first_name ilike ('%KENNETH%')
and b.last_name ilike ('%ELLIS%')
and k.state in ('TX')
```

```
;


------------------------------------
-- BUP patients only
------------------------------------

-- SET search_path TO farm;
-- DROP TABLE IF EXISTS ors_pdmp_data_pull;
-- CREATE TEMP TABLE ors_pdmp_data_pull AS
-- WITH patients_with_drug_mat AS (
--     SELECT DISTINCT a.consolidation_identifier
--     FROM pmp_farm_va.farm.consolidation_identifiers a
--     inner JOIN pmp_farm_va.farm.patients b ON a.patient_id = b.id
--     inner JOIN pmp_farm_va.farm.dispensations c ON b.id = c.patient_id
--     inner JOIN pmp_farm_va.farm.drugs g ON g.dispensation_id = c.id
--     inner JOIN drug_master dd ON dd.ndc = g.narx_national_drug_code
--     WHERE dd.drug_mat_yn = 'Y'
-- )
-- SELECT distinct
--     a.consolidation_identifier,
--     b.birthdate,
--     b.gender,
--     c.id AS dispensation_id,
--     CAST(c.filled_at AS DATE) AS filled_at,
--     cast(c.sold_at as date) as sold_at,
--     dd.active_ingredient,
--     dd.label_name,
--     dd.drug_narx_type_one AS drug_type,
--     dd.drug_mat_yn,
--     e.dea_number AS provider_dea,
--     f.dea_number AS pharmacy_dea,
--     CAST(g.quantity AS REAL) AS quantity,
--     CAST(c.days_supply AS REAL) AS days_supply,
--     g.narx_unit_eq_one AS mme_conversion
-- FROM pmp_farm_va.farm.consolidation_identifiers a
-- LEFT JOIN pmp_farm_va.farm.patients b ON a.patient_id = b.id
-- LEFT JOIN pmp_farm_va.farm.patient_addresses k ON a.patient_id = k.patient_id
-- LEFT JOIN pmp_farm_va.farm.dispensations c ON b.id = c.patient_id
-- LEFT JOIN pmp_farm_va.farm.prescriptions d ON d.dispensation_id = c.id
-- LEFT JOIN pmp_farm_va.farm.prescribers e ON e.dispensation_id = c.id
-- LEFT JOIN pmp_farm_va.farm.dispensaries f ON f.dispensation_id = c.id
-- LEFT JOIN pmp_farm_va.farm.drugs g ON g.dispensation_id = c.id
-- LEFT JOIN drug_master dd ON dd.ndc = g.narx_national_drug_code
-- WHERE filled_at >= '2022-07-01'
-- AND filled_at <= CURRENT_DATE
-- AND k.state = 'VA'
-- AND a.consolidation_identifier IN (SELECT distinct consolidation_identifier FROM
patients_with_drug_mat)
-- ;


select
   count(*) as row_count,
   count(distinct consolidation_identifier) as patient_count
from ors_pdmp_data_pull
;
--1488454
--21881


select * from ors_pdmp_data_pull
;


select count(*) as count
from ors_pdmp_data_pull
where sold_at is null
;


SELECT distinct
   date_category,
   COUNT(distinct consolidation_identifier) AS patient_count
FROM (
   SELECT distinct
```

```
    c.consolidation_identifier,
    COALESCE(CAST(c.filled_at AS DATE), CAST(c.sold_at AS DATE)) AS effective_filled_at,
    COALESCE(CAST(c.sold_at AS DATE), CAST(c.filled_at AS DATE)) AS effective_sold_at,
    GREATEST(0, DATEDIFF(day, COALESCE(CAST(c.filled_at AS DATE), CAST(c.sold_at AS DATE)), COALESCE(CAST(c.sold_at AS DATE), CAST(c.filled_at AS DATE)))) AS date_difference,
    CASE
      WHEN GREATEST(0, DATEDIFF(day, COALESCE(CAST(c.filled_at AS DATE), CAST(c.sold_at AS DATE)), COALESCE(CAST(c.sold_at AS DATE), CAST(c.filled_at AS DATE)))) = 0 THEN 'same_day'
      WHEN GREATEST(0, DATEDIFF(day, COALESCE(CAST(c.filled_at AS DATE), CAST(c.sold_at AS DATE)), COALESCE(CAST(c.sold_at AS DATE), CAST(c.filled_at AS DATE)))) BETWEEN 1 AND 7 THEN 'within_seven_days'
      WHEN GREATEST(0, DATEDIFF(day, COALESCE(CAST(c.filled_at AS DATE), CAST(c.sold_at AS DATE)), COALESCE(CAST(c.sold_at AS DATE), CAST(c.filled_at AS DATE)))) > 7 THEN 'more_than_seven'
    END AS date_category
  FROM
    ors_pdmp_data_pull c
) AS sub
GROUP BY
  1
;

----------------------------------------------------------------------------------------------------------------------

select
  count(*) as row_count,
  count(distinct consolidation_identifier) as patient_count
from ors_pdmp_data_pull
;
--1488454
--21881


select * from ors_pdmp_data_pull
-- where days_supply = 900
-- where consolidation_identifier in ('20331863768')
order by consolidation_identifier
limit 1000
;


select distinct
   min(days_supply) as min_s,
   max(days_supply) as max_s,
   min(quantity) as min_q,
   max(quantity) as max_q
from ors_pdmp_data_pull
order by 1
;

------------------------------------------------------------------------------------------------------------
------

-- creates source table that is time bounded from 01/01/2016 to today's date
drop table if exists orsv2_source_data;
create temp table orsv2_source_data distkey(consolidation_identifier)
sortkey(consolidation_identifier, filled_at) as
   (select * from ors_pdmp_data_pull)
          --where filled_at > '2020-04-01' and filled_at <= current_date)
--           and consolidation_identifier in ('10000135670'))
--          where consolidation_identifier in (select consolidation_identifier
--                                  from health_raw_pdmp.raw_pdmp_va
--                                  group by consolidation_identifier
--                                  order by consolidation_identifier desc
--                                  limit 100)
--      )
;

-- completed in 13:59.584

--simulating a patient with only sedative dispensations
-- select * from orsv2_source_data
-- where upper(drug_type) in ('SEDATIVE')
-- and upper(drug_type) not in ('NARCOTIC')
-- limit 15
-- ;
-- consolidation_identifier = 10001455220

-- select * from orsv2_source_data
-- where consolidation_identifier IN ('10001455220')
-- ;


SELECT
    consolidation_identifier,
    gender,
    COUNT(*)
FROM orsv2_source_data
GROUP BY 1, 2
limit 25
;

-- select * from orsv2_source_data
-- where consolidation_identifier in ('100004120')
-- order by filled_at
-- ;


select
    count(*) as row_count,
    count(distinct consolidation_identifier) as patient_count
from orsv2_source_data
;
--1664437
--30942

select count(*) as row_count
from arbitrary_refdates
;
--113
```

```
-----------------------------------------------------------------------------------------------
-- the following section corrects for the gender and age of each consolidation_identifier by making sure the most
-- often reported gender and age is populated for each consolidation_identifier.
drop table if exists gender_mode;
create temp table gender_mode as
select * from
(
select consolidation_identifier,
    gender as gender_mode,
    row_number() over (partition by consolidation_identifier order by cnt desc) as rownum
from
  (
     select
          consolidation_identifier,
          gender,
          count(1) as cnt
     from orsv2_source_data
     group by 1, 2
    )
order by consolidation_identifier, cnt desc, gender)
where rownum = 1
;

select
  count(*) as row_count,
  count(distinct consolidation_identifier) as patient_count
from gender_mode;
;
--30942
--30942


drop table if exists age_mode;
create temp table age_mode as
select * from
(
select consolidation_identifier,
      age as age_mode,
      row_number() over (partition by consolidation_identifier order by cnt desc) as rownum
from
    (
          select
                consolidation_identifier,
                datediff(years, birthdate::date, filled_at::date) as age,
                count(1) as cnt
          from orsv2_source_data
          group by 1, 2
       )
order by consolidation_identifier, cnt desc, age)
where rownum = 1
;

select
   count(*) as row_count,
   count(distinct consolidation_identifier) as patient_count
from age_mode
;
--30942
--30942

drop table if exists orsv2_source_data_w_gender;
create temp table orsv2_source_data_w_gender as
   (select a.*,
          b.gender_mode,
          c.age_mode as age
       from orsv2_source_data a
          left join gender_mode b on a.consolidation_identifier = b.consolidation_identifier
          left join age_mode c on a.consolidation_identifier = c.consolidation_identifier
)
;

alter table orsv2_source_data_w_gender drop column gender;
alter table orsv2_source_data_w_gender add column gender varchar(255);
update orsv2_source_data_w_gender set gender = gender_mode;
```

```
alter table orsv2_source_data_w_gender drop column gender_mode;

drop table orsv2_source_data;
alter table orsv2_source_data_w_gender rename to orsv2_source_data;


select
   count(*) as row_count,
   count(distinct consolidation_identifier) as patient_count
from orsv2_source_data
;
--1664437
--30942

----------------------------------------------------------------------------------------------------

drop table if exists orsv2_intro;
create temp table orsv2_intro distkey(master_num) sortkey(master_num, filled_at) as
select distinct
      consolidation_identifier as master_num,
      filled_at,
      gender as gender_raw,
      label_name,
      age,
      birthdate,
      cast((CASE WHEN drug_mat_yn = 'Y' THEN 'mat'
               WHEN drug_mat_yn = 'N' THEN 'pm'
               ELSE 'na'
           END) as varchar(20)) as bupren_type,
      drug_type as drug_type,
      days_supply,
      quantity,
      mme_conversion::real as narx_unit_eq_one,
      narx_unit_eq_one::real*quantity::real as narc_mg,
      active_ingredient as active_ingredient,
--      document_id as dispensation_id,
      dispensation_id::varchar as dispensation_id,
      pharmacy_dea,
      provider_dea
from orsv2_source_data t
order by
   master_num,
   filled_at desc
-- limit 100
;

---------------------------------------------------------

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_intro
;
--1664437
--30942

select
   bupren_type,
   count(distinct master_num) as patient_count
from orsv2_intro
group by 1
;

-- cases:
-- 85 moud
-- 345 pain management
-- no nulls

-- controls:
-- 1720 moud
-- 30424 pain management
-- no nulls

---------------------------------------------------------

drop table if exists synthetic_dispensations;
```

```
create temp table synthetic_dispensations distkey(master_num) sortkey(master_num,
filled_at) as
   select distinct
        a.master_num,
        b.filled_at::date as filled_at,
        a.gender_raw,
        null as narx_label_name,
        a.age,
        null as bupren_type,
        null as drug_type,
        null::double precision as day_supply,
        null::double precision as quantity,
        null::real as narx_unit_eq_one,
        null::real as narc_mg,
        null as active_ingredient,
        'ARB' as dispensation_id,
        null as pharmacy_dea,
        null as provider_dea
   from orsv2_intro  a
   inner join arbitrary_refdates b
      on (datediff(days, a.filled_at::date, b.filled_at::date) <= 365)
      and (datediff(days, a.filled_at::date, b.filled_at::date) >= 0)
   group by
      master_num,
      a.filled_at,
      b.filled_at,
      a.gender_raw,
      a.age
   having
      b.filled_at != max(a.filled_at)
order by master_num, filled_at desc
-- limit 100
;

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from synthetic_dispensations
```

```
;
--1614459
--30942

select
  min(filled_at) as min,
  max(filled_at) as max
from synthetic_dispensations
;
--2017-04-30
--2023-05-31
--------------------------------------------------------------------------------------------------------------------
------

drop table if exists _all_rows1;
create temp table _all_rows1 distkey(master_num) sortkey(master_num, filled_at) as
 select distinct
    master_num,
    filled_at::date as filled_at,
    gender_raw,
    label_name,
    age,
    bupren_type,
    drug_type as drug_type,
    least(days_supply::double precision, 365) as days_supply,
    quantity::double precision as quantity,
    narx_unit_eq_one,
    narc_mg,
    active_ingredient,
    dispensation_id,
    pharmacy_dea,
    provider_dea
 from orsv2_intro
 UNION
 select distinct * from synthetic_dispensations
order by master_num, filled_at desc
-- limit 100
;
```

```
select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from _all_rows1
;
-- 3278896
-- 30942

select * from _all_rows1
order by
  master_num,
  filled_at DESC
limit 1000
;

--------------------------------------------------------------------------------------------------------------------
------

drop table if exists _all_rows2;
create temp table _all_rows2 distkey(master_num) sortkey(master_num, filled_at) as
select t.*,
    row_number() over (partition by master_num order by filled_at desc) as disp_num
from _all_rows1 t
-- where master_num in ('10031392158')
order by master_num, disp_num
-- limit 100
;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from _all_rows2
;
--3278896
--30942
```

```
drop table if exists orsv2_part1;
create temp table orsv2_part1 distkey(master_num) sortkey(master_num, filled_at) as
select t.*,

    lead(filled_at,1) over (partition by master_num order by filled_at, disp_num desc) as next_filled_at,

    -- The DATEDIFF function below is used to calculate the difference between two dates in terms of the specified date part,
    -- which in this case is "day".
    -- The "::date" syntax is used to convert the "filled_at" and "next_filled_at" columns to the date data type, which is required by the DATEDIFF function.
    -- The result of the DATEDIFF function is the number of days between the two dates, which is assigned to the "gap" column in the output table.
    datediff(day, filled_at::date, next_filled_at::date) as gap

from _all_rows2 t
where t.filled_at <= current_date
order by master_num, filled_at desc, disp_num desc;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_part1
;
--3278896
--30942

-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------

drop table _all_rows1;
drop table _all_rows2;


-- this is a source table for the original variables
drop table if exists orsv2_input_vars_with_idx;
```

```
create temp table orsv2_input_vars_with_idx distkey(master_num) sortkey(master_num,
filled_at) as
  select
    t.*,

    -- The SQL below calculates a running total of the "gap" column over a window
partitioned by the "master_num" column, and ordered by the "filled_at" and "disp_num"
columns in descending order.
    -- The running total is assigned to a new column called "filled_at_idx".
    -- The SUM function is used to calculate the running total of the "gap" column, which is
wrapped inside a COALESCE
    -- function with a default value of 0 to handle NULL values.
    -- The OVER clause is used to define the window within which the running total is
calculated.
    -- The window is partitioned by "master_num", meaning that the running total is
calculated separately for each distinct value of "master_num".
    -- The window is also ordered by the "filled_at" and "disp_num" columns in descending
order, meaning that the
    -- running total includes the sum of all previous "gap" values for the same
"master_num" value and all earlier "filled_at" and "disp_num" values.
    -- The "rows unbounded preceding" syntax in the window definition means that the
window starts from the beginning
    -- of the partition, and includes all rows up to and including the current row.
   sum(coalesce(gap,0)) over (partition by master_num order by filled_at desc, disp_num
rows unbounded preceding) as filled_at_idx
from orsv2_part1 t
-- where master_num in ('10031392158')
order by master_num, filled_at desc, disp_num
;


select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_input_vars_with_idx
;
--3278896
--30942
```

```
drop table if exists orsv2_script_classification;
create temp table orsv2_script_classification distkey(master_num) sortkey(master_num,
filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
     t.dispensation_id,
     t.drug_type,
     t.pharmacy_dea,
     t.provider_dea,
     t.days_supply,
     t.narc_mg,
     t.quantity,
     case when upper(bupren_type) = 'MAT' then 1 else 0 end as mat_script,

     case when upper(drug_type) = 'SEDATIVE' then 1 else 0 end as sed_script,

     case when lower(active_ingredient) in ('alprazolam', 'chlordiazepoxide', 'clobazam',
'clonazepam', 'clorazepate', 'diazepam',
                              'estazolam', 'flurazepam', 'lorazepam', 'midazolam', 'oxazepam',
'quazepam','temazepam', 'triazolam') then 1 else 0 end as benzo_script,

     case when upper(drug_type) = 'NARCOTIC' and upper(bupren_type) != 'MAT' then 1 else
0 end as narc_script,

     case when upper(drug_type) = 'NARCOTIC' and upper(bupren_type) != 'MAT' and
days_supply >= 15 then 1 else 0 end as supply15plus_script,

     case when (upper(label_name) like '%FENTANYL%' and upper(label_name) like
'%PATCH%')
            or upper(label_name) like '%METHADONE%' or upper(label_name) like
'%SUBOXONE%'
            --or bupren_type in ('Buprenorphine for Pain Management','Buprenorphine for
Narcotic Withdrawal Therapy')
            --or upper(bupren_type) = 'MAT'
```

```
           or upper(bupren_type) = 'PM'
           or  (upper(label_name) like '%MORPHINE%' and upper(label_name) like '%ER%')
           or  (upper(label_name) like '%OXYCONTIN%')
       then 1
       else 0
    end as hi_risk_script,

    case when UPPER(bupren_type) = 'NA'
         and (
            (upper(label_name) like '%ER%'
              or upper(label_name) like '%MS CONTIN%'
              or upper(label_name) like '%OXYCONTIN%'
              or (upper(label_name) like '%FENTANYL%' and upper(label_name) like
'%PATCH%')
              or (upper(label_name) like '%DURAGESIC%' and upper(label_name) like
'%DOLOPHINE%')
            )
           and upper(drug_type) = 'NARCOTIC'
          )
         then 1
         else 0
    end as long_acting_script,
    case when UPPER(bupren_type) = 'NA' and long_acting_script = 0 and upper(drug_type)
= 'NARCOTIC' then 1 else 0 end as short_acting_script

from orsv2_input_vars_with_idx t
order by master_num, filled_at desc, disp_num
;



select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_script_classification
;
--3278896
--30942
```

```
select * from orsv2_script_classification
;


drop table if exists orsv2_provns_pharmns_calcs;
create temp table orsv2_provns_pharmns_calcs distkey(master_num)
sortkey(master_num, filled_at) as
select t.master_num,
   t.filled_at,
   t.filled_at_idx,
   t.disp_num,
  (select count(distinct provider_dea) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +59
                             and master_num = t.master_num
                             and upper(drug_type) in ('SEDATIVE', 'NARCOTIC')) as
Prov_NS60,
   (select count(distinct pharmacy_dea) from orsv2_script_classification where
filled_at_idx between t.filled_at_idx and  t.filled_at_idx +59
                            and master_num = t.master_num
                            and narc_script = 1 ) as Pharm_N60,
   (select count(distinct pharmacy_dea) from orsv2_script_classification where
filled_at_idx between t.filled_at_idx and  t.filled_at_idx +59
                            and master_num = t.master_num
                            and upper(drug_type) in ('SEDATIVE', 'NARCOTIC')) as
Pharm_NS60,

   (select count(distinct pharmacy_dea) from orsv2_script_classification where
filled_at_idx between t.filled_at_idx and  t.filled_at_idx +179
                            and master_num = t.master_num
                            and upper(drug_type) in ('SEDATIVE', 'NARCOTIC')) as
Pharm_NS180
from orsv2_script_classification t
order by master_num, filled_at desc, disp_num;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_provns_pharmns_calcs
```

```
;
--3278896
--30942


drop table if exists orsv2_surrounded_dispensations_calcs;
create temp table orsv2_surrounded_dispensations_calcs distkey(master_num)
sortkey(master_num, filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
     t.days_supply,
     t.sed_script,
     t.narc_script,
     t.benzo_script,
     t.mat_script,
     t.hi_risk_script,
     t.long_acting_script,
     t.supply15plus_script,
    coalesce((select count(disp_num) from orsv2_script_classification where  filled_at_idx
>= t.filled_at_idx
                                    and filled_at_idx - days_supply + 1 <= t.filled_at_idx -
t.days_supply + 1
                                    and master_num = t.master_num
                                    and mat_script=1
                                    and t.mat_script =1
                                    and disp_num > t.disp_num ),0) as surrounded_mat_fill,
    coalesce((select count(disp_num) from orsv2_script_classification where  filled_at_idx
>= t.filled_at_idx
                                    and filled_at_idx - days_supply + 1 <= t.filled_at_idx -
t.days_supply + 1
                                    and master_num = t.master_num
                                    and narc_script=1
                                    and t.narc_script =1
                                    and disp_num > t.disp_num ),0) as surrounded_narc_fill,
    coalesce((select count(disp_num) from orsv2_script_classification where  filled_at_idx
>= t.filled_at_idx
```

```
                              and filled_at_idx - days_supply + 1 <= t.filled_at_idx -
t.days_supply + 1
                              and master_num = t.master_num
                              and benzo_script=1
                              and t.benzo_script =1
                              and disp_num > t.disp_num ),0) as surrounded_benzo_fill,
    coalesce((select count(disp_num) from orsv2_script_classification where  filled_at_idx
>= t.filled_at_idx
                              and filled_at_idx - days_supply + 1 <= t.filled_at_idx -
t.days_supply + 1
                              and master_num = t.master_num
                              and sed_script=1
                              and t.sed_script =1
                              and disp_num > t.disp_num ),0) as surrounded_sed_fill
from orsv2_script_classification t
order by master_num, filled_at desc, disp_num;


select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_surrounded_dispensations_calcs
;
--3278896
--30942


drop table if exists orsv2_next_fills_calcs;
create temp table orsv2_next_fills_calcs distkey(master_num) sortkey(master_num,
filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
     t.days_supply,
     t.narc_script,
     t.sed_script,
     t.mat_script,
    t.benzo_script,
    t.surrounded_mat_fill,
    t.surrounded_narc_fill,
    t.surrounded_sed_fill,
    t.surrounded_benzo_fill,
    coalesce((select max(filled_at_idx) from orsv2_surrounded_dispensations_calcs where
(filled_at_idx <= t.filled_at_idx)
                                        and mat_script = 1
                                        and surrounded_mat_fill = 0
                                        and t.disp_num > disp_num
                                        and master_num = t.master_num ), -1)
          as next_mat_fill,

    coalesce((select max(filled_at_idx) from orsv2_surrounded_dispensations_calcs where
(filled_at_idx <= t.filled_at_idx)
                                        and narc_script = 1
                                        and surrounded_narc_fill = 0
                                        and t.disp_num > disp_num
                                        and master_num = t.master_num ), -1)
          as next_narc_fill,

    coalesce((select max(filled_at_idx) from orsv2_surrounded_dispensations_calcs where
(filled_at_idx <= t.filled_at_idx)
                                        and benzo_script = 1
                                        and surrounded_benzo_fill = 0
                                        and t.disp_num > disp_num
                                        and master_num = t.master_num ), -1)
          as next_benzo_fill,

    coalesce((select max(filled_at_idx) from orsv2_surrounded_dispensations_calcs where
(filled_at_idx <= t.filled_at_idx)
                                        and sed_script = 1
                                        and surrounded_sed_fill = 0
                                        and t.disp_num > disp_num
                                        and master_num = t.master_num ), -1)
          as next_sed_fill

from orsv2_surrounded_dispensations_calcs t
```

```
order by master_num, filled_at desc, disp_num;

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_next_fills_calcs
;
--3278896
--30942


drop table if exists orsv2_drug_days_classifications;
create temp table orsv2_drug_days_classifications distkey(master_num)
sortkey(master_num, filled_at) as
select t.master_num,
     t.filled_at,
     t.filled_at_idx,
     t.disp_num,
      t.days_supply,
      t.narc_script,
      t.sed_script,
      t.mat_script,
      t.benzo_script,
      t.surrounded_mat_fill,
      t.surrounded_narc_fill,
      t.surrounded_sed_fill,
      t.surrounded_benzo_fill,
      t.next_narc_fill,
      t.next_benzo_fill,
      t.next_sed_fill,
      case when mat_script = 1 and surrounded_mat_fill = 0 then least(days_supply,
filled_at_idx - next_mat_fill) else 0 end as mat_days,

      -- The SQL below creates a new column called "narc_days" using a conditional
statement that evaluates two conditions and returns different values based on the results.
      -- The first condition checks whether the value of the "narc_script" column is 1 and the
value of the "surrounded_narc_fill" column is 0.
      -- If both conditions are true, then the LEAST function is used to calculate the smaller
```

```
value between the "days_supply" column and
    -- the difference between the "filled_at_idx" and "next_narc_fill" columns.
    -- The "days_supply" column is the number of days supplied for the medication, and the
"next_narc_fill" column is
    -- the value of the "filled_at_idx" column for the next row where the "narc_script"
column is 1.
    -- The difference between the "filled_at_idx" column of the current row and
    -- the "next_narc_fill" column of the next row represents the number of days between
the two prescriptions for narcotic medication.
    -- If the first condition is false, or if the result of the LEAST function is greater than the
"days_supply" column, then the value 0 is returned.

    -- The resulting values for the "narc_days" column represent the number of days that the
current prescription
    -- for narcotic medication overlaps with the next prescription for narcotic medication,
up to a maximum of the number of days supplied for the current prescription.
    case when narc_script = 1 and surrounded_narc_fill = 0 then least(days_supply,
filled_at_idx - next_narc_fill) else 0 end as narc_days

from orsv2_next_fills_calcs t
order by master_num, filled_at desc, disp_num;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_drug_days_classifications
;
--3278896
--30942


-------------------------------------------------------------------------------------------------------------------
------
select * from orsv2_drug_days_classifications
-- where master_num in ('10031392158')
-- order by 1, 3
;
```

```
drop table if exists orsv2_drug_days_calcs;
create temp table orsv2_drug_days_calcs distkey(master_num) sortkey(master_num,
filled_at) as
select
     t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
     t.days_supply,
     t.narc_script,
     t.sed_script,
     t.mat_script,
     t.benzo_script,
     t.surrounded_mat_fill,
     t.surrounded_narc_fill,
     t.surrounded_sed_fill,
     t.surrounded_benzo_fill,
     t.next_narc_fill,
     t.next_benzo_fill,
     t.next_sed_fill,
    coalesce( (select sum(mat_days)
              from orsv2_drug_days_classifications
              where ( filled_at_idx between t.filled_at_idx + 1 and  t.filled_at_idx +364)
                   and master_num = t.master_num
         ), 0) as mat_days365_p1,
    coalesce( (select sum(t.filled_at_idx - (filled_at_idx - mat_days + 1) )
              from orsv2_drug_days_classifications
              where ( filled_at_idx - mat_days +1 <= t.filled_at_idx)
                   and filled_at_idx > t.filled_at_idx
                   and master_num = t.master_num
                   and disp_num != t.disp_num
         ), 0) as mat_days365_p2,
    coalesce( (select sum(t.filled_at_idx + 365 - (filled_at_idx - mat_days + 1) )
              from orsv2_drug_days_classifications
              where ( filled_at_idx - mat_days +1 <= t.filled_at_idx + 364)
                   and filled_at_idx > t.filled_at_idx + 364
```

```
                 and master_num = t.master_num
                 and disp_num != t.disp_num
        ), 0) as mat_days365_p3,

  -- "narc_days90_p1" represents the number of days the patient had a narcotic medication on hand during a given 90 day window.
  -- The column value below is determined by a subquery that uses the SUM function to add up the "narc_days" column for each
  -- row in the table "orsv2_drug_days_classifications".
  -- The subquery is filtered to include only those rows where the "filled_at_idx" column is between the "filled_at_idx" column of the current row plus 1
  -- and the "filled_at_idx" column of the current row plus 89, and where the "master_num" is the same as that of the current row.
  -- If the subquery returns no rows, the COALESCE function replaces the NULL value with 0 for the "narc_days90_p1" column in the output table.

   coalesce( (select sum(narc_days)
            from orsv2_drug_days_classifications
            where ( filled_at_idx between t.filled_at_idx + 1 and  t.filled_at_idx + 89)
                 and master_num = t.master_num
        ), 0) as narc_days90_p1,

  -- "narc_days90_p2" represents the number of days of overlapping narcotic medication prescribed during the same 90-day period as the current prescription but by different dispensing events.
  -- The calculated value below is determined by subtracting the "filled_at_idx" column of each row from the "narc_days" column
  -- and then subtracting the result from the "filled_at_idx" column of the current row.
  -- If the result is less than or equal to the "filled_at_idx" column of the other rows with the same "master_num",
  -- and if it is greater than the "filled_at_idx" column of the current row, the calculated value is added to the sum.
  -- If the subquery returns no rows, the COALESCE function replaces the NULL value with 0 for the "narc_days90_p2" column in the output table.

   coalesce( (select sum(t.filled_at_idx - (filled_at_idx - narc_days + 1) )
            from orsv2_drug_days_classifications
            where ( filled_at_idx - narc_days + 1 <= t.filled_at_idx)
```

```
                and filled_at_idx > t.filled_at_idx
                and master_num = t.master_num
                and disp_num != t.disp_num
        ), 0) as narc_days90_p2,

   -- "narc_days90_p3" represents the number of days of overlapping narcotic medication
prescribed during the subsequent 90-day period as the current prescription but by different
dispensing events.
   -- The calculated value below is determined by subtracting the "filled_at_idx" column of
each row from the "narc_days" column and adding 90.
   -- If the result is less than or equal to the "filled_at_idx" column of the current row plus 89,
   -- and if the "filled_at_idx" column of the current row is less than the "filled_at_idx"
column of the other rows with the same "master_num", the calculated value is added to the
sum.
   -- If the subquery returns no rows, the COALESCE function replaces the NULL value with
0 for the "narc_days90_p3" column in the output table.

    coalesce( (select sum(t.filled_at_idx + 90 - (filled_at_idx - narc_days + 1) )
            from orsv2_drug_days_classifications
            where ( filled_at_idx - narc_days + 1 <= t.filled_at_idx + 89)
                and filled_at_idx > t.filled_at_idx + 89
                and master_num = t.master_num
                and disp_num != t.disp_num
        ), 0) as narc_days90_p3

from orsv2_drug_days_classifications t
-- where master_num in ('10000135670')
-- where master_num in ('10031392158')
order by master_num, filled_at desc, disp_num
-- limit 10000
;

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_drug_days_calcs
;
--3337109
```

```
--30942


drop table if exists orsv2_overlap_counts_p1_and_p2;
create temp table orsv2_overlap_counts_p1_and_p2 distkey(master_num)
sortkey(master_num, filled_at) as
select t.master_num,
     t.filled_at,
     t.filled_at_idx,
     t.disp_num,
      t.mat_script,
      t.mat_days365_p1,
      t.mat_days365_p2,
      t.mat_days365_p3,
      t.narc_script,
      t.narc_days90_p1,
      t.narc_days90_p2,
      t.narc_days90_p3,
     coalesce((select least(coalesce(max(t.filled_at_idx - (filled_at_idx - days_supply +1) +
1),0), t.days_supply, t.filled_at_idx -t.next_narc_fill, t.filled_at_idx - t.next_benzo_fill )
                                                      from orsv2_drug_days_calcs where (
(filled_at_idx - days_supply + 1) <= t.filled_at_idx)
                                                                      and (filled_at_idx >= t.filled_at_idx)
                                                                      and benzo_script = 1
                                                                      and t.narc_script = 1
                                                                      and surrounded_benzo_fill = 0
                                                                      and t.surrounded_narc_fill = 0
                                                                      and disp_num != t.disp_num
                                                                      and master_num = t.master_num ),
0)
            as benzna_overlap_day_count_p1,

     (select least(coalesce(max(t.filled_at_idx - (filled_at_idx - days_supply +1) + 1),0),
t.days_supply, t.filled_at_idx -t.next_narc_fill, t.filled_at_idx - t.next_benzo_fill )
                                                      from orsv2_drug_days_calcs where (
(filled_at_idx - days_supply + 1) <= t.filled_at_idx)
                                                                      and (filled_at_idx >= t.filled_at_idx)
                                                                      and narc_script = 1
```

```
                                                                and t.benzo_script = 1
                                                                and surrounded_narc_fill = 0
                                                                and t.surrounded_benzo_fill = 0
                                                                and disp_num > t.disp_num
                                                                and master_num = t.master_num )
            as benzna_overlap_day_count_p2,

    coalesce((select least(coalesce(max(t.filled_at_idx - (filled_at_idx - days_supply +1) +
1),0), t.days_supply, t.filled_at_idx -t.next_narc_fill, t.filled_at_idx - t.next_sed_fill )
                                                from orsv2_drug_days_calcs where (
(filled_at_idx - days_supply + 1) <= t.filled_at_idx)
                                                                and (filled_at_idx >= t.filled_at_idx)
                                                                and sed_script = 1
                                                                and t.narc_script = 1
                                                                and surrounded_sed_fill = 0
                                                                and t.surrounded_narc_fill = 0
                                                                and disp_num != t.disp_num
                                                                and master_num = t.master_num ),
0)
            as sedna_overlap_day_count_p1,

    (select least(coalesce(max(t.filled_at_idx - (filled_at_idx - days_supply +1) + 1),0),
t.days_supply, t.filled_at_idx -t.next_narc_fill, t.filled_at_idx - t.next_sed_fill )
                                                from orsv2_drug_days_calcs where (
(filled_at_idx - days_supply + 1) <= t.filled_at_idx)
                                                                and (filled_at_idx >= t.filled_at_idx)
                                                                and narc_script = 1
                                                                and t.sed_script = 1
                                                                and surrounded_narc_fill = 0
                                                                and t.surrounded_sed_fill = 0
                                                                and disp_num > t.disp_num
                                                                and master_num = t.master_num )
            as sedna_overlap_day_count_p2
from orsv2_drug_days_calcs t
order by master_num, filled_at desc, disp_num
;

select
```

```
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_overlap_counts_p1_and_p2
;
--3337109
--30942


drop table if exists orsv2_overlap_counts_p3_and_p4;
create temp table orsv2_overlap_counts_p3_and_p4 distkey(master_num)
sortkey(master_num, filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
     t.mat_script,
     t.mat_days365_p1,
     t.mat_days365_p2,
     t.mat_days365_p3,
     t.narc_script,
     t.narc_days90_p1,
     t.narc_days90_p2,
     t.narc_days90_p3,
    coalesce((select sum(benzna_overlap_day_count_p1) +
sum(benzna_overlap_day_count_p2)
              from orsv2_overlap_counts_p1_and_p2
              where ( filled_at_idx between t.filled_at_idx + 1 and  t.filled_at_idx +89)
                   and master_num = t.master_num
         ), 0) as benzna_overlap_days90_p1,

    coalesce((select count(disp_num)
           from orsv2_overlap_counts_p1_and_p2
           where filled_at_idx = t.filled_at_idx
            and master_num = t.master_num
            and benzna_overlap_day_count_p1 + benzna_overlap_day_count_p2 > 0
          ), 0) as benzna_same_day_addition,

    coalesce((select sum(t.filled_at_idx + 90 - (filled_at_idx -
(benzna_overlap_day_count_p2 + benzna_overlap_day_count_p1) + 1 ))
           from orsv2_overlap_counts_p1_and_p2
           where filled_at_idx > t.filled_at_idx + 89
            and filled_at_idx - (benzna_overlap_day_count_p1 +
benzna_overlap_day_count_p2) + 1 <= t.filled_at_idx + 89
            and master_num = t.master_num
            and benzna_overlap_day_count_p1 + benzna_overlap_day_count_p2 > 0
          ), 0) as benzna_overlap_days90_p3,

    coalesce((select sum(t.filled_at_idx - (filled_at_idx - (benzna_overlap_day_count_p2 +
benzna_overlap_day_count_p1) + 1 ))
           from orsv2_overlap_counts_p1_and_p2
           where filled_at_idx > t.filled_at_idx
            and filled_at_idx - (benzna_overlap_day_count_p1 +
benzna_overlap_day_count_p2) + 1 <= t.filled_at_idx
            and master_num = t.master_num
            and benzna_overlap_day_count_p1 + benzna_overlap_day_count_p2 > 0
          ), 0) as benzna_overlap_days90_p4,

    coalesce((select sum(sedna_overlap_day_count_p1) +
sum(sedna_overlap_day_count_p2)
              from orsv2_overlap_counts_p1_and_p2
              where ( filled_at_idx between t.filled_at_idx + 1 and  t.filled_at_idx +89)
                   and master_num = t.master_num
         ), 0) as sedna_overlap_days90_p1,

    coalesce((select count(disp_num)
           from orsv2_overlap_counts_p1_and_p2
           where filled_at_idx = t.filled_at_idx
            and master_num = t.master_num
            and sedna_overlap_day_count_p1 + sedna_overlap_day_count_p2 > 0
          ), 0) as sedna_same_day_addition,

    coalesce((select sum(t.filled_at_idx + 90 - (filled_at_idx - (sedna_overlap_day_count_p2
+ sedna_overlap_day_count_p1) + 1 ))
           from orsv2_overlap_counts_p1_and_p2
           where filled_at_idx > t.filled_at_idx + 89
            and filled_at_idx - (sedna_overlap_day_count_p1 + sedna_overlap_day_count_p2)
```

```
+ 1 <= t.filled_at_idx + 89
          and master_num = t.master_num
          and sedna_overlap_day_count_p1 + sedna_overlap_day_count_p2 > 0
        ), 0) as sedna_overlap_days90_p3,

   coalesce((select sum(t.filled_at_idx - (filled_at_idx - (sedna_overlap_day_count_p2 +
sedna_overlap_day_count_p1) + 1 ))
         from orsv2_overlap_counts_p1_and_p2
         where filled_at_idx > t.filled_at_idx
          and filled_at_idx - (sedna_overlap_day_count_p1 + sedna_overlap_day_count_p2)
+ 1 <= t.filled_at_idx
          and master_num = t.master_num
          and sedna_overlap_day_count_p1 + sedna_overlap_day_count_p2 > 0
        ), 0) as sedna_overlap_days90_p4
from orsv2_overlap_counts_p1_and_p2 t
order by master_num, filled_at desc, disp_num
;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_overlap_counts_p3_and_p4
;
--3337109
--30942


drop table if exists orsv2_benzna_sedna_overlap_days;
create temp table orsv2_benzna_sedna_overlap_days as
select t.master_num,
   t.filled_at,
   t.filled_at_idx,
   t.disp_num,
    mat_days365_p1 - mat_days365_p2 + mat_days365_p3 + mat_script as mat_days365,

    -- The SQL below creates a new column called "na_days90" by performing a calculation
using three previously defined columns:
    -- "narc_days90_p1", "narc_days90_p2", "narc_days90_p3", and the "narc_script"
```

```
column.
    -- The calculation subtracts the value of "narc_days90_p2" from "narc_days90_p1", adds the value of "narc_days90_p3", and adds the value of "narc_script".
    -- The resulting value represents the number of days that a patient was prescribed narcotic medication during a 90-day period,
    -- taking into account overlapping prescription periods as well as whether the current prescription is for a narcotic medication.

    -- as mentioned above starting on line 571:
    -- "narc_days90_p1" represents the number of days out of the next 90 days a narcotic medication was prescribed.
    -- "narc_days90_p2" represents the number of days of overlapping narcotic medication prescribed during the same 90-day period as the current prescription but by different dispensing events.
        -- we subtract this off, since we're trying to account for the total number of distinct days a patient had a narcotic in their possession.
    -- "narc_days90_p3" represents the number of days of overlapping narcotic medication prescribed during the subsequent 90-day period overlaps with the current 90 day window defined by the days supply of the current prescription but by different dispensing events.
        -- we include these prescriptions but do not count the number of days that it overlapped with any other narcotic prescriptions.
    -- visual representation
    --
    -- --------------------p1---------------------------
    --        -----p2-----
    --                              -----------------------p3---------------------
    --
    -- Finally, the "narc_script" column indicates whether the current prescription is for a narcotic medication, with a value of 1 for narcotic prescriptions and 0 for non-narcotic prescriptions.
    narc_days90_p1 - narc_days90_p2 + narc_days90_p3 + narc_script as na_days90,

    benzna_overlap_days90_p1  + (case when benzna_same_day_addition > 0 then 1 else 0 end) + benzna_overlap_days90_p3 - benzna_overlap_days90_p4 as benzna_overlap_days90,
    sedna_overlap_days90_p1  + (case when sedna_same_day_addition > 0 then 1 else 0 end) + sedna_overlap_days90_p3 - sedna_overlap_days90_p4 as sedna_overlap_days90
from orsv2_overlap_counts_p3_and_p4 t
```

```
order by master_num, filled_at desc, disp_num
;

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_benzna_sedna_overlap_days
;
--3337109
--30942


drop table if exists orsv2_scripts_counts;
create temp table orsv2_scripts_counts distkey(master_num) sortkey(master_num,
filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
    (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +89
                       and master_num = t.master_num
                       and dispensation_id != 'ARB')
                            as tot_dispensations90,
    (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +179
                       and master_num = t.master_num
                       and dispensation_id != 'ARB')
                            as tot_dispensations180,
    (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +364
                       and master_num = t.master_num
                       and dispensation_id != 'ARB')
                            as tot_dispensations365,

    (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +179
                       and benzo_script = 1
```

```
                                      and master_num = t.master_num)
                                               as totbenz_scripts180,
   (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +364
                                      and benzo_script = 1
                                      and master_num = t.master_num)
                                               as totbenz_scripts365,

   (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +179
                                      and hi_risk_script = 1
                                      and master_num = t.master_num)
                                               as hi_risk_scripts180,
   (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +364
                                      and hi_risk_script = 1
                                      and master_num = t.master_num)
                                               as hi_risk_scripts365,

   (select count(distinct disp_num) from orsv2_script_classification where filled_at_idx
between t.filled_at_idx and  t.filled_at_idx +89
                                      and supply15plus_script = 1
                                      and master_num = t.master_num)
                                               as supply15plus_scripts90
from orsv2_script_classification t
order by master_num, filled_at desc, disp_num
;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_scripts_counts
;
--3337109
--30942


drop table if exists orsv2_mg_and_mat_counts;
```

```
create temp table orsv2_mg_and_mat_counts distkey(master_num) sortkey(master_num,
filled_at) as
select t.master_num,
    t.filled_at,
    t.filled_at_idx,
    t.disp_num,
    coalesce((select sum(narc_mg) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +2
              and master_num = t.master_num
              and narc_script = 1),0)
                        as narcotic_mg3,

    coalesce((select sum(narc_mg) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +86
              and master_num = t.master_num
              and narc_script = 1),0)
                        as narcotic_mg87,

    coalesce((select sum(narc_mg) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +89
              and master_num = t.master_num
              and narc_script = 1),0)
                        as narcotic_mg90,

    coalesce((select sum(days_supply) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +89
              and master_num = t.master_num
              and long_acting_script = 1),0)
                        as totlong_ds90,

    coalesce((select sum(days_supply) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +89
              and master_num = t.master_num
              and short_acting_script = 1),0)
                        as totshort_ds90,

    coalesce((select sum(quantity) from orsv2_script_classification
         where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +89
```

```
            and master_num = t.master_num
            and long_acting_script = 1),0)
                        as totlong_quant90,

    coalesce((select sum(quantity) from orsv2_script_classification
        where filled_at_idx between t.filled_at_idx and  t.filled_at_idx +89
            and master_num = t.master_num
            and short_acting_script = 1),0)
                        as totshort_quant90,

    coalesce((select min(filled_at_idx - t.filled_at_idx) from orsv2_script_classification
where filled_at_idx >=  t.filled_at_idx
                                      and mat_script = 1
                                      and master_num = t.master_num),-1)
                                               as last_mat
from orsv2_script_classification t
order by master_num, filled_at desc, disp_num
;

select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_mg_and_mat_counts
;
--3337109
--30942


drop table if exists orsv2_variables_calcs;
create temp table orsv2_variables_calcs distkey ( master_num ) sortkey (master_num,
filled_at) as
select  a.master_num,
     a.filled_at,
    a.filled_at_idx,
     a.disp_num,
     a.benzna_overlap_days90::real as benzna_overlap_days90,
     a.na_days90::real as na_days90,
     (case when na_days90::real >= 80 then 1 else 0 end)::real as chronic_opioid,
     a.mat_days365::real as mat_days365,

     (case when (b.hi_risk_scripts180 = b.hi_risk_scripts365 ) then 0 else
b.hi_risk_scripts180::real/b.hi_risk_scripts365::real end)::real as hi_risk_scripts_180_365,
     case when b.totbenz_scripts365::real = 0.0 then 1 else 0 end as
totbenz_scripts_180_365_group0,

     -- creating buckets for hi_risk_script 180 365 ratios...
    b.hi_risk_scripts180::real as hi_risk_scripts180,
     b.hi_risk_scripts365::real as hi_risk_scripts365,

     case when   (case when hi_risk_scripts365 <= 0.00001 then 0 else
b.hi_risk_scripts180::real/(b.hi_risk_scripts365::real ) end) >= 1.0
      then 1
      else 0 end as hi_risk_scripts_180_365_group6,

     (case when c.narcotic_mg3::real/3.0 >= 90 and (c.narcotic_mg90::real -
c.narcotic_mg87::real)/3.0 = 0 then 1 else 0 end)::real as new_patient_hi_dose,
     c.last_mat::real as last_mat,
     c.totshort_ds90::real as totshort_ds90,
     (case when c.totshort_ds90::real >= 365.0 then 365.0 else c.totshort_ds90::real
end)::real as totshort_ds90_cap,
     (case when c.totlong_ds90::real >= 91.0 then 91.0 else c.totlong_ds90::real end)::real
as totlong_ds90_cap,
     (case when last_mat::real = -1 then 1 else 0 end)::real as no_mat_history,
     (case when no_mat_history::real = 1 then 0 else 1 end)::real as mat_history,

     d.pharm_ns180::real as pharm_ns180,

     case when (age::real >=100) then 100
      when (age::real < 0) then 0
      when (age is null) then 0
      else age::real
     end as age,
     case when age::real >= 71 and age::real <=100 then 1 else 0 end as "71_130",
     case when age::real < 18 then 1 else 0 end as "under_18",
     case when age::real >= 18 and age::real <=25 then 1 else 0 end as "18_25",
     case when age::real >= 26 and age::real <=30 then 1 else 0 end as "26_30",
     case when age::real >= 31 and age::real <=35 then 1 else 0 end as "31_35",
     case when age::real >= 36 and age::real <=40 then 1 else 0 end as "36_40",
     case when age::real >= 41 and age::real <=45 then 1 else 0 end as "41_45",
     case when age::real >= 46 and age::real <=50 then 1 else 0 end as "46_50",
     case when age::real >= 51 and age::real <=55 then 1 else 0 end as "51_55",
     case when age::real >= 56 and age::real <=60 then 1 else 0 end as "56_60",
     case when age::real >= 61 and age::real <=65 then 1 else 0 end as "61_65",
     case when age::real >= 66 and age::real <=70 then 1 else 0 end as "66_70",

     -- this assumes gender comes with the examples.
     (case when upper(e.gender_raw) like 'M%' then 1.0
       when upper(e.gender_raw) like 'F%' then 2.0
       else 3.0 end)::decimal(4,1) as gender,
     (case when gender = 1.0 then 1 else 0 end)::real as male,

     case when "56_60"*b.tot_dispensations90 >=6 then 1 else 0 end as
"56_60_x_tot_dispensations90_group4",
     case when "51_55"*b.tot_dispensations90 >=6 then 1 else 0 end as
"51_55_x_tot_dispensations90_group4",
     case when "46_50"*b.tot_dispensations90 >=6 then 1 else 0 end as
"46_50_x_tot_dispensations90_group4",
     case when "41_45"*b.tot_dispensations90 >=6 then 1 else 0 end as
"41_45_x_tot_dispensations90_group4",
     case when "36_40"*b.tot_dispensations90 < 0.00001 and ("36_40" = 1) then 1 else 0
end as "36_40_x_tot_dispensations90_group0",
     case when "31_35"*b.tot_dispensations90 >=6 then 1 else 0 end as
"31_35_x_tot_dispensations90_group4"

from orsv2_benzna_sedna_overlap_days a
  left join orsv2_scripts_counts b on a.master_num = b.master_num and a.filled_at_idx =
b.filled_at_idx and a.disp_num = b.disp_num
  left join orsv2_mg_and_mat_counts c on a.master_num = c.master_num and
a.filled_at_idx = c.filled_at_idx and a.disp_num = c.disp_num
  left join orsv2_provns_pharmns_calcs d on a.master_num = d.master_num and
a.filled_at_idx = d.filled_at_idx and a.disp_num = d.disp_num
  left join orsv2_input_vars_with_idx e on a.master_num = e.master_num and
a.filled_at_idx = e.filled_at_idx and a.disp_num = e.disp_num
order by master_num, filled_at desc
```

```
;

select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_variables_calcs
;
--3337109
--30942

-- select * from orsv2_variables_calcs
-- ;


-- select *, row_number() over (partition by master_num, filled_at order by master_num) as
dup_num
-- from orsv2_variables_calcs
-- ;


create temp table orsv2_variables_calcs_deduped as
  select * from
   (select t.*,
         row_number() over (partition by master_num, filled_at order by master_num) as
dup_num
     from orsv2_variables_calcs t)
where dup_num = 1;

drop table orsv2_variables_calcs;
alter table orsv2_variables_calcs_deduped rename to orsv2_variables_calcs;


select
   count(*) as row_count,
   count(distinct master_num) as patient_count
from orsv2_variables_calcs
;
--3100112
```

```
--30942


-- select * from orsv2_variables_calcs
-- ;


drop table if exists orsv2_score_calcs;
create temp table orsv2_score_calcs distkey(master_num) sortkey(master_num, filled_at)
as
  select
      t.*,
      "31_35_x_tot_dispensations90_group4"*(0.4657)+
      "36_40_x_tot_dispensations90_group0"*(-0.3056)+
      "41_45_x_tot_dispensations90_group4"*(0.5402)+
      "46_50_x_tot_dispensations90_group4"*(0.3587)+
      "51_55_x_tot_dispensations90_group4"*(0.2566)+
      "56_60_x_tot_dispensations90_group4"*(0.0489)+
      "61_65"*(-1.0344)+
      "66_70"*(-2.3516)+
      "71_130"*(-2.8774)+
      age*(-0.0078)+
      benzna_overlap_days90*(0.0108)+
      chronic_opioid*(-0.145)+
      hi_risk_scripts_180_365_group6*(0.8747)+
      male*(0.7312)+
      mat_history*(1.406)+
      pharm_ns180*(0.4667)+
      totshort_ds90_cap*(0.0052)+
      under_18*(-4.2161)+
      (-3.5831)
            as logit_13,

      least(990, greatest(0, floor(floor(logit_13*64/ln(2) + 760.09  - 127.43)/10.0)*10)) as
orsv2_score,
      logit_13*64/ln(2) as raw_score,

      ("31_35_x_tot_dispensations90_group4" - 0)*(0.4657)*64/ln(2)  as
```

```
"31_35_x_tot_dispensations90_group4_contrib",
    ("36_40_x_tot_dispensations90_group0" - 0)*(-0.3056)*64/ln(2)  as
"36_40_x_tot_dispensations90_group0_contrib",
    ("41_45_x_tot_dispensations90_group4" - 0)*(0.5402)*64/ln(2)  as
"41_45_x_tot_dispensations90_group4_contrib",
    ("46_50_x_tot_dispensations90_group4" - 0)*(0.3587)*64/ln(2)  as
"46_50_x_tot_dispensations90_group4_contrib",
    ("51_55_x_tot_dispensations90_group4" - 0)*(0.2566)*64/ln(2)  as
"51_55_x_tot_dispensations90_group4_contrib",
    ("56_60_x_tot_dispensations90_group4" - 0)*(0.0489)*64/ln(2)  as
"56_60_x_tot_dispensations90_group4_contrib",
    ("61_65" - 0)*(-1.0344)*64/ln(2)  as   "61_65_contrib",
    ("66_70" - 0)*(-2.3516)*64/ln(2)  as   "66_70_contrib",
    ("71_130" - 0)*(-2.8774)*64/ln(2)  as   "71_130_contrib",
    (age - 48.07)*(-0.0078)*64/ln(2)  as   "age_contrib",
    (benzna_overlap_days90 - 1.744980537)*(0.0108)*64/ln(2)  as
"benzna_overlap_days90_contrib",
    (chronic_opioid - 0.043037894)*(-0.145)*64/ln(2)  as   "chronic_opioid_contrib",
    (hi_risk_scripts_180_365_group6 - 0)*(0.8747)*64/ln(2)  as
"hi_risk_scripts_180_365_group6_contrib",
    (male - 0.445178572)*(0.7312)*64/ln(2)  as   "male_contrib",
    (mat_history - 0.015105017)*(1.406)*64/ln(2)  as   "mat_history_contrib",
    (pharm_ns180 - 1.046453629)*(0.4667)*64/ln(2)  as   "pharm_ns180_contrib",
    (totshort_ds90_cap - 44.0683959)*(0.0052)*64/ln(2)  as   "totshort_ds90_cap_contrib",
    (under_18 - 0)*(-4.2161)*64/ln(2)  as   "under_18_contrib"
from orsv2_variables_calcs t
order by master_num, filled_at desc
;


select
  count(*) as row_count,
  count(distinct master_num) as patient_count
from orsv2_score_calcs
;
--3100112
--30942
```

```
-- run this to determine the answer to the following really common question:
-- why is my patient's score so high given the scarcity of dispensation data?
select * from orsv2_score_calcs
;



drop table if exists orsv2_scores_contribs;
create temp table orsv2_scores_contribs distkey(master_num) sortkey(master_num,
filled_at) as
 select master_num,
        filled_at,
        orsv2_score,
        logit_13 as orsv2_logit,
        "31_35_x_tot_dispensations90_group4" as "31_35_x_gt6_dispensations90",
        "36_40_x_tot_dispensations90_group0" as "36_40_x_no_dispensations90",
        "41_45_x_tot_dispensations90_group4" as "41_45_x_gt6_dispensations90",
        "46_50_x_tot_dispensations90_group4" as "46_50_x_gt6_dispensations90",
        "51_55_x_tot_dispensations90_group4" as "51_55_x_gt6_dispensations90",
        "56_60_x_tot_dispensations90_group4" as "56_60_x_gt6_dispensations90",
        "61_65",
        "66_70",
        "71_130",
        age,
        benzna_overlap_days90,
        chronic_opioid,
        hi_risk_scripts_180_365_group6 as hi_risk_scripts_last_365_all_in_last_180,
        male,
        mat_history,
        pharm_ns180,
        totshort_ds90_cap,
        under_18
from orsv2_score_calcs t
order by master_num, filled_at desc
;



select
   count(*) as row_count,
   count(distinct master_num) as patient_count
-- from orsv2_scores_contribs_fnl_nondecedents
from orsv2_scores_contribs
;
--25811
--353


select * from orsv2_scores_contribs
;


drop table if exists orsv2_scores_contribs_most_recent_score;
create temp table orsv2_scores_contribs_most_recent_score as
SELECT master_num, filled_at, orsv2_score
FROM (
   SELECT master_num, filled_at, orsv2_score,
        ROW_NUMBER() OVER (PARTITION BY master_num ORDER BY filled_at DESC) AS
row_num
   FROM orsv2_scores_contribs
) ranked
WHERE row_num = 1
;


select * from orsv2_scores_contribs_most_recent_score
;


-- stats here ----------------------------------------------------------------------------------------------------
--------

SELECT
  MIN(orsv2_score) AS minimum,
  MAX(orsv2_score) AS maximum,
  percentile_cont(0.5) WITHIN GROUP (ORDER BY orsv2_score) AS median,
  AVG(orsv2_score) AS mean,
  percentile_cont(0.25) WITHIN GROUP (ORDER BY orsv2_score) AS percentile_25,
 percentile_cont(0.75) WITHIN GROUP (ORDER BY orsv2_score) AS percentile_75
FROM
 orsv2_scores_contribs_most_recent_score
;
--0
--860
--260
--221
--70
--340

--30
--900
--470
--457.54
--430
--510




select
   count(*) as row_count,
   count(distinct master_num) as patient_count
-- from orsv2_scores_contribs_fnl_nondecedents
from orsv2_scores_contribs_most_recent_score
;
--1488454


--21881


--32226


SELECT
   CASE
```

```
      WHEN orsv2_score >= 0 AND orsv2_score <= 379 THEN 'Low Score (0-379)'
      WHEN orsv2_score > 379 THEN 'High Score (>379)'
      ELSE 'No Score'
   END AS ors_score_category,
   COUNT(distinct master_num) AS patient_count
FROM orsv2_scores_contribs_most_recent_score
GROUP BY
   CASE
      WHEN orsv2_score >= 0 AND orsv2_score <= 379 THEN 'Low Score (0-379)'
      WHEN orsv2_score > 379 THEN 'High Score (>379)'
      ELSE 'No Score'
   END
ORDER BY ors_score_category
;

-- 213350 high score
-- 1275104 low score

--18938
--2943

--21738
--10488
----------------------------------------------------------------------------------------------------------
------
-- create final table of features and orsv1 score for decedents
drop table if exists orsv2_scores_and_pdmp_features_decedents;
create temp table orsv2_scores_and_pdmp_features_decedents as
SELECT
   t1.*,
   1 as decedent
FROM orsv2_scores_contribs_fnl_decedents t1
JOIN (
  SELECT master_num, MAX(filled_at) AS max_date
  FROM orsv2_scores_contribs_fnl_decedents
  GROUP BY master_num
) t2 ON t1.master_num = t2.master_num AND t1.filled_at = t2.max_date
;
```

```
-- create final table of features and orsv1 score for nondecedents
drop table if exists orsv2_scores_and_pdmp_features_nondecedents;
create temp table orsv2_scores_and_pdmp_features_nondecedents as
SELECT
   t1.*,
   0 as decedent
FROM orsv2_scores_contribs_fnl_nondecedents t1
JOIN (
  SELECT master_num, MAX(filled_at) AS max_date
  FROM orsv2_scores_contribs_fnl_nondecedents
  GROUP BY master_num
) t2 ON t1.master_num = t2.master_num AND t1.filled_at = t2.max_date
;


----------------------------------------------------------------------------------------------------------------
------
-- step 7: merge dfs
-- join the two above tables
drop table if exists orsv2_scores_and_pdmp_features_final_cohort_ct_set4;
create temp table orsv2_scores_and_pdmp_features_final_cohort_ct_set4 as

select * from orsv2_scores_and_pdmp_features_decedents
   union
select * from orsv2_scores_and_pdmp_features_nondecedents
;



--QA counts
select
   decedent,
   count(distinct master_num) as patient_count
from orsv2_scores_and_pdmp_features_final_cohort_ct_set4
group by 1
```

```
;
-- 353
-- 30589

-- for the time being, this is what is unloaded to S3 and copied to my work schema so that we can grab model performance stats
-- using a Jupyter notebook that Emily and I have already built for data in the data science Redshift cluster
select * from orsv2_scores_and_pdmp_features_final_cohort_ct_set4
;

------------------------------------------------------------------------------------------------------------
------



drop table if exists orsv2_score_categories_ct;
create temp table orsv2_score_categories_ct as
select
   *,
   case when orsv2_score >= 0 and orsv2_score <= 199 then 1
       when orsv2_score > 199 and orsv2_score <= 299 then 2
       when orsv2_score > 299 and orsv2_score <= 399 then 3
       when orsv2_score > 399 and orsv2_score <= 499 then 4
       when orsv2_score > 499 and orsv2_score <= 599 then 5
       when orsv2_score > 599 and orsv2_score <= 699 then 6
       when orsv2_score > 699 and orsv2_score <= 799 then 7
       when orsv2_score > 799 and orsv2_score <= 899 then 8
       when orsv2_score > 899 and orsv2_score <= 990 then 9
   else 0
   end as index
from orsv2_scores_and_pdmp_features_final_cohort_ct_set4
;




select
   min(orsv2_score),
   max(orsv2_score),
  PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY orsv2_score) AS percentile_25,
  AVG(orsv2_score) AS average,
  PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY orsv2_score) AS percentile_50,
  PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY orsv2_score) AS percentile_75
from orsv2_score_categories_ct
;
--0
--990


select
  decedent,
  count(distinct master_num) as count
from orsv2_score_categories_ct
group by 1
order by 1
;
-- 658
-- 275

-- use as input to R code when calculating ORs
select
  decedent,
  index,
  count(distinct master_num) as count
from orsv2_score_categories_ct
group by 1,2
order by 1,2
;
```

```
--------------------------------------------------------------
--
-- drop table if exists numbers;
-- create  table numbers as
-- select row_number() over (order by 1) as i
-- from work_agillette.va_orsv2_scores_eom_only
-- limit 18;
--
-- drop table if exists orsv2_contributions_part1;
-- create  table orsv2_contributions_part1 distkey(master_num) sortkey(master_num, filled_at) as
--    select * from (select * from work_agillette.va_orsv2_scores_eom_only where filled_at > '2018-12-31')
--       join numbers
--          on true;
--
--
-- drop table if exists orsv2_contributions_part2;
-- create  table orsv2_contributions_part2 distkey(master_num) sortkey(master_num, filled_at) as
-- select master_num,
--      filled_at,
--      i,
--      case when i = 1 then 'In the age range of 31 to 35 and greater than 6 dispensations in the last 90 days'
--          when i = 2 then 'In the age range of  of 36 to 40 and no dispensations in the last 90 days'
--          when i = 3 then 'In the age range of 41 to 45 and greater than 6 dispensations in the last 90 days'
--          when i = 4 then 'In the age range of  46 to 50 and greater than 6 dispensations in the last 90 days'
--          when i = 5 then 'In the age range of  51 to 55 and greater than 6 dispensations in the last 90 days'
--          when i = 6 then 'In the age range of  56 to 60 and greater than 6 dispensations in the last 90 days'
--          when i = 7 then 'In the age range of 61 to 65'
--          when i = 8 then 'In the age range of 66 to 70'
--          when i = 9 then 'In the age range of 71+'
--          when i = 10 then 'Age'
--          when i = 11 then 'Benzo-narcotic dispensation overlaps in the last 90 days'
--          when i = 12 then 'Chronic opioid usage'
--          when i = 13 then 'All high-risk scripts in the last year occurred in the last 180 days'
--          when i = 14 then 'Gender'
--          when i = 15 then 'MAT history'
--          when i = 16 then 'Number of pharmacies with narcotic or sedative dispensations in
the last 80 days'
--          when i = 17 then 'Total short-acting-drug days supply in last 90 days, capped at 365'
--          when i = 18 then 'Under the age of 18'
--       end as varname,
--      case when i = 1 then "31_35_x_gt6_dispensations90"
--          when i = 2 then "36_40_x_no_dispensations90"
--          when i = 3 then "41_45_x_gt6_dispensations90"
--          when i = 4 then "46_50_x_gt6_dispensations90"
--          when i = 5 then "51_55_x_gt6_dispensations90"
--          when i = 6 then "56_60_x_gt6_dispensations90"
--          when i = 7 then "61_65"
--          when i = 8 then "66_70"
--          when i = 9 then "71_130"
--          when i = 10 then age
--          when i = 11 then benzna_overlap_days90
--          when i = 12 then chronic_opioid
--          when i = 13 then hi_risk_scripts_last_365_all_in_last_180
--          when i = 14 then male
--          when i = 15 then mat_history
--          when i = 16 then pharm_ns180
--          when i = 17 then totshort_ds90_cap
--          when i = 18 then under_18
--       end as varval,
--      case when i = 1 then ("31_35_x_gt6_dispensations90" - 0)*(0.4657)*64/ln(2)
--          when i = 2 then ("36_40_x_no_dispensations90" - 0)*(-0.3056)*64/ln(2)
--          when i = 3 then ("41_45_x_gt6_dispensations90" - 0)*(0.5402)*64/ln(2)
--          when i = 4 then ("46_50_x_gt6_dispensations90" - 0)*(0.3587)*64/ln(2)
--          when i = 5 then ("51_55_x_gt6_dispensations90" - 0)*(0.2566)*64/ln(2)
--          when i = 6 then ("56_60_x_gt6_dispensations90" - 0)*(0.0489)*64/ln(2)
--          when i = 7 then ("61_65" - 0)*(-1.0344)*64/ln(2)
--          when i = 8 then ("66_70" - 0)*(-2.3516)*64/ln(2)
--          when i = 9 then ("71_130" - 0)*(-2.8774)*64/ln(2)
--          when i = 10 then (age - 48.07)*(-0.0078)*64/ln(2)
--          when i = 11 then (benzna_overlap_days90 - 1.744980537)*(0.0108)*64/ln(2)
--          when i = 12 then (chronic_opioid - 0.043037894)*(-0.145)*64/ln(2)
--          when i = 13 then (hi_risk_scripts_last_365_all_in_last_180 - 0)*(0.8747)*64/ln(2)
--          when i = 14 then (male - 0.445178572)*(0.7312)*64/ln(2)
--          when i = 15 then (mat_history - 0.015105017)*(1.406)*64/ln(2)
--          when i = 16 then (pharm_ns180 - 1.046453629)*(0.4667)*64/ln(2)
--          when i = 17 then (totshort_ds90_cap - 44.0683959)*(0.0052)*64/ln(2)
--          when i = 18 then (under_18 - 0)*(-4.2161)*64/ln(2)
--       end as contribution,
--      --dense_rank() over (partition by master_num, filled_at order by abs(contribution) desc) as contribution_rank,
--      orsv2_score
-- from orsv2_contributions_part1
-- order by master_num, filled_at --, contribution_rank desc
-- ;
--
-- drop table if exists orsv2_contributions_part3;
-- create  table orsv2_contributions_part3 distkey(master_num) sortkey(master_num, filled_at) as
-- select a.master_num,
--      a.filled_at,
--      a.i,
--      nth_value(a.varname, 1) over (partition by a.master_num, a.filled_at
--                              order by abs(a.contribution) desc, a.varname
--                              rows between unbounded preceding and unbounded following) as contribution1,
--      nth_value(abs(a.contribution), 1) over (partition by a.master_num, a.filled_at
--                              order by abs(a.contribution) desc, a.varname
--                              rows between unbounded preceding and unbounded following) as contribution1_val,
--      nth_value(a.varname, 2) over (partition by a.master_num, a.filled_at
--                              order by abs(a.contribution) desc, a.varname
--                              rows between unbounded preceding and unbounded following) as contribution2,
--      nth_value(a.contribution, 2) over (partition by a.master_num, a.filled_at
--                              order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution2_val,
--        nth_value(a.varname, 3) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution3,
--        nth_value(a.contribution, 3) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution3_val,
--        nth_value(a.varname, 4) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution4,
--        nth_value(a.contribution, 4) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution4_val,
--        nth_value(a.varname, 5) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution5,
--        nth_value(a.contribution, 5) over (partition by a.master_num, a.filled_at
--                                  order by abs(a.contribution) desc, a.varname
--                                  rows between unbounded preceding and unbounded following) as
contribution5_val
-- from
-- orsv2_contributions_part2 a
-- order by a.master_num, a.filled_at;
--
-- drop table if exists orsv2_contributions_final;
-- create  table orsv2_contributions_final as
--     (select * from orsv2_contributions_part3
--     where i = 1
--     order by master_num, filled_at);
--
--
-- ---------------------------------------------------
-- create temp table work_agillette.va_orsv2_contribs_eom_only distkey(master_num)
sortkey(master_num, filled_at) as
--     select * from orsv2_contributions_final;
```