

Columnar Table plus Shadow Table Solution for Ad-Hoc Queries

- Issues:
- Part One: Use Shadow Table to Fix the Immutable Columnar Table Defect
 - Construct Demo Tables
 - Simulate Optimized CDC apply operation
 - Check The Report View Anytime
- Part Two: Tested in HUAT with Real Data
 - Construct Shadow Table and Report View
 - Simulate CDC Apply Operation
 - Base Table Rebuild and Switchover
- Appendix
 - Jira Tickets:
 - The Query for Base Datamart Table.

Issues:

The largest datamart table we created contains more than 51 million records, the table size is more than 510GB.

Ad-hoc query are pretty slow for the view based on this table. Almost all of the jasper filter option does not work based on the query on low cardinality column in this table.

To fix it, we introduced the columnar storage extension, it greatly improved the ad-hoc query performance, the same query speed got improved more than 10x times, the columnar table size got dropped down to 40GB compare with normal table 510GB. It help fixed almost all of the jasper filter construction issues.

Then, the new issues come in like

- The citus columnar table does not support insert/delete, which block the cdc (change data capture) flow;
- The hydra columnar extension supports insert/delete on columnar table, but, the problem of hydra columnar extension are:
 - The latest version of hydra extension is only for postgresql 15. hydra 15 does not support the postgresql 17 which we will upgrade to.
 - After turn on existing cdc code, we found the error after apply change to hydra columnar table. The unique constraint does not work, we start to see duplicate key issue after apply cdc change.

We have to keep going to do research on market to find the better columnar storage provider, but could not find the best.

This paper is addressing above issue and provide new design that use citus columnar table plus shadow table to fix above issues.

Part One: Use Shadow Table to Fix the Immutable Columnar Table Defect

Based on what we already built

- Ability to construct large complicate datamart table
- The absolute ability of columnar table for analytic job
- The limit change data volume (2M transaction per day)

We do have chance to keep changed data in shadow table and using view to mix immutable columnar base table with shadow table to support end user ad-hoc queries.

Here is the demo

Construct Demo Tables

We will create the base columnar table A and its normal shadow table A_shadow, and the relevant view.

```

create table A(id int primary key, name varchar) using columnar;

create table A_shadow (like A including all);

alter table A_shadow add column action char(1);

create or replace view report as
select A.* from A
where not exists ( select 1 from a_shadow where a_shadow.id=A.id)
union all
select id,name from a_shadow where action='i'
order by id;

INSERT INTO A (id, name)
SELECT generate_series, 'data_' || generate_series
FROM generate_series(1, 100);

select * from A where id>90

  id |      name
-----+-----
  91 | data_91
  92 | data_91
  93 | data_93
  94 | data_94
  95 | data_95
  96 | data_96
  97 | data_97
  98 | data_98
  99 | data_99
 100 | data_100

```

Simulate Optimized CDC apply operation

- Instead of directly apply change to large immutable datamart table A, we apply the change to shadow table. Remember the A_shadow table also has primary key constraint, it is not like dynamic cdc history table which wrote by trigger on source table, which will keep multiple action for same PK row, in A_shadow table, any record will be unique, it will reply the final state of that record
- The action field is for the desired change on base table, 'd' is for delete, 'i' is for insert, the update action required by cdc will be implement in A_shadow table as delete then insert. So, any record got changed, will be finally kept in A_shadow table with action value is either 'd' or 'i'
- At the begin, there is no record in A_shadow table, so, all of the CDC apply will use insert on conflict update statement, to make sure
 - create the change for certain record
 - keep the change to latest status.

```

-- Per CDC required insert : the new value should be directly added into A_shadow table;
insert into A_shadow values(101,'pure insert','i') on conflict(id) do update set id=101,name='pure insert',
action='i';

-- Per CDC required update: The update should be converted to delete first and on conflict update
-- 1. There is already insert records before, apply change flag to 'd'

insert into A_shadow values(92,'update required delete','d') on conflict(id) do update set id=92, name='update
required delete',action='d';
insert into A_shadow values(92,'updated 92 new values','i') on conflict(id) do update set id=92,name='updated
92 new values',action='i';

-- 2. There is no that records, even the A table also does not contain that records, but, if not missing any
cdc request, must be insert already run before. So, here, just use same logic.

insert into A_shadow values(102,'update required delete','d') on conflict(id) do update set id=102,name='update
required delete',action='d';
insert into A_shadow values(102,'updated 102 new values','i') on conflict(id) do update set id=102,
name='updated 102 new values',action='i';

-- delete: Just add the delete operation using same statements. Suppose the original table should contain this
record, if not, it will still keep data integrity
-- because it is pure delete.

insert into A_shadow values(103,'pure delete','d') on conflict(id) do update set id=103,name='pure delete',
action='d';
insert into A_shadow values(93,'pure delete','d') on conflict(id) do update set id=93,name='pure delete',
action='d';

test=# select * from a_shadow;
 id |          name          | action
----+-----+-----+
 101 | pure insert          | i
   92 | updated 92 new values | i
 102 | updated 102 new values | i
 103 | pure delete          | d
   93 | pure delete          | d
(5 rows)

```

Check The Report View Anytime

Then, just check the report view, we can see it looks like all of the CDC change got applied to the base table correctly.

```

test=# select * from report where id>90;
 id |          name
----+-----+
  91 | data_91
  92 | updated 92 new values
  94 | data_94
  95 | data_95
  96 | data_96
  97 | data_97
  98 | data_98
  99 | data_99
 100 | data_100
 101 | pure insert
 102 | updated 102 new values
(11 rows)

```

id 93: Got "deleted"

id 92: Got "updated"

....

id 101, 102: Got "inserted" and "updated"

So, the solution works.

Part Two: Tested in HUAT with Real Data

Here is the test in huat with the real data

Construct Shadow Table and Report View

```
set search_path to datamart_humana;

-- The claim table is columnar table with real data
ods_domaini=# select count(1) from claim;
   count
-----
 51411921
(1 row)

create table claim_shadow (like claim including all);

create or replace view report as
select A.*,'' from datamart_humana.claim A
where not exists ( select 1 from datamart_humana.claim_shadow s where s.customer_id=A.customer_id and s.
claim_id=A.claim_id)
union all
select * from datamart_humana.claim_shadow where action='i';
```

Simulate CDC Apply Operation

It is not easy to manually construct the cdc apply operation. Also, the test purpose is confirm the performance acceptable, So, instead of one by one apply CDC change, we directly move data into claim_shadow table, to check view performance for certain critical queries

```
insert into claim_shadow select *,'i' from claim limit 10;

ods_domaini=# select customer_id,claim_id from claim_shadow;
customer_id |   claim_id
-----+-----
 319 | 187442723521
 319 | 187477782111
 319 | 187545650331
 319 | 194033525671
 319 | 194234171021
 319 | 194275176151
 319 | 194357502411
 319 | 194426388441
 319 | 194561607151
 319 | 194614296811
(10 rows)

update claim_shadow set action='d' where customer_id=319 and claim_id='194614296811';

ods_domaini=# select count(1) from report where customer_id=319 and claim_id='194614296811';
   count
-----
 0
(1 row)

Time: 514.529 ms

select "report"."claim_id" as "report_claim_id",
"report"."client_id" as "report_client_id",
"customer_info"."customer_id" as "customer_info_customer_id"
from datamart_humana."report" "report"
inner join (select customr0.customer_no customer_id
```

```

, customr0.full_title customer_nm
, customr0.cust strt_dte customer_effective_begin_dt
, customr0.customer_end_date customer_effective_end_dt
, e21.customer_set_id
, e21b.customer_set_id AS benefit_customer_set_id
from customer.customr0
inner join customer.e21
on customr0.customer_no = e21.customer_id
and customer_set_type_id = 2
left outer JOIN customer.e21 e21b
ON customr0.customer_no = e21b.customer_id
AND e21b.customer_set_type_id = 3
) "customer_info" on ("report"."customer_id" = "customer_info"."customer_id")
where ((report"."client_id" in (1)) and ("customer_info"."customer_id" in (319)))
limit 1000;

```

report_claim_id | report_client_id | customer_info_customer_id

report_claim_id	report_client_id	customer_info_customer_id
266090038531	1	319
252302000011	1	319
252302000021	1	319
252302000031	1	319
266090038561	1	319
266090038771	1	319
266090038781	1	319
266090038791	1	319
266090038801	1	319

(9 rows)

Time: 1014.192 ms (00:01.014)

```
ods_domani=# select postpay_ind_3 from report group by postpay_ind_3;
postpay_ind_3
```

postpay_ind_3
1
A
B
C
E
O

(8 rows)

Time: 34542.137 ms (00:34.542)

```
ods_domani=# update claim_shadow set postpay_ind_3='F' where customer_id=319 and claim_id='194357502411';
UPDATE 1
Time: 86.177 ms
```

```
ods_domani=# select postpay_ind_3 from report group by postpay_ind_3;
postpay_ind_3
```

postpay_ind_3
1
A
B
C
E
F
O

(9 rows)

Time: 34531.300 ms (00:34.531)

The performance is almost same as query columnar table itself.

Base Table Rebuild and Switchover

If the growth of shadow table greatly drop down the view report query performance, we can rebuild the base table and switchover them.

Two approaches for it

- Parallel maintain existing CDC apply to normal datamart table (The original writable table before we convert) when we turn on the columnar table and shadow table, no meaningful, suppose will not go this way
 - Keep running with shadow table and view solution, until we decide to rebuild, directly construct new base using data from report view. The logic like below, need to add the the detail later to prevent the data missing or override issue, etc.

```
create table claim_new (like claim including all) using columnar;
create table claim_shadow_new (like claim_shadow including all);
insert into claim_new select * from report;
```

Then, mark the latest applied CDC record, switch over table and rebuild the report view.

```
alter table claim rename to claim_old;
alter table claim_shadow rename to claim_shadow_old;
alter table claim_new rename to claim;
alter table claim_shadow_new rename to claim_shadow;
```

```
create or replace view report as
select A.*,'' from datamart_humana.claim A
where not exists ( select 1 from datamart_humana.claim_shadow s where s.customer_id=A.customer_id and s.
claim_id=A.claim_id)
union all
select * from datamart_humana.claim_shadow where action='i';
```

Appendix

Jira Tickets:

<https://domainrx.atlassian.net/browse/DEVOPS-11718>

The Query for Base Datamart Table.

Very complex query which joins about 50 tables from different schema, and the output is more than 1100 fields.

<https://domainrx.atlassian.net/browse/DEVOPS-7297>

```
--DROP VIEW misc.claim_price_detail_vw;

CREATE OR REPLACE VIEW misc.claim_price_detail_vw AS (
WITH global_indicators AS (SELECT gcgri.customer_id,
                                    gcgri.claim_id,
                                    gcgri.global_position_cnt,
                                    jsonb_array_elements_text(value) AS global_generic_reporting_ind
                           FROM (SELECT claim_generic_reporting_indicators.customer_id,
                                       claim_generic_reporting_indicators.claim_id,
                                       key::INTEGER AS global_position_cnt,
                                       value::jsonb
                                  FROM claimsprocess.claim_generic_reporting_indicators,
                                       jsonb_each_text(claim_generic_reporting_indicators.
claim_global_reporting_information_json) AS x(key, value)
                                 WHERE jsonb_typeof(claim_generic_reporting_indicators.
claim_global_reporting_information_json) = 'object'
                                         AND value IS NOT NULL) AS gcgri),
customer_indicators AS (SELECT ccgri.customer_id,
                                    ccgri.claim_id,
                                    ccgri.customer_position_cnt,
                                    jsonb_array_elements_text(value) AS customer_generic_reporting_in
                           FROM (SELECT claim_generic_reporting_indicators.customer_id,
                                       claim_generic_reporting_indicators.claim_id,
                                       key::INTEGER AS customer_position_cnt,
                                       value::jsonb
                                  FROM claimsprocess.claim_generic_reporting_indicators,
                                       jsonb_each_text(customer_generic_reporting_indicators.
customer_global_reporting_information_json) AS y(key, value)
                                 WHERE jsonb_typeof(customer_generic_reporting_indicators.
customer_global_reporting_information_json) = 'object'
                                         AND value IS NOT NULL) AS ccgri)
SELECT gcgri.global_position_cnt,
        gcgri.global_generic_reporting_ind,
        gcgri.claim_id,
        gcgri.customer_id,
        gcgri.customer_position_cnt,
        gcgri.value,
        ccgri.global_position_cnt,
        ccgri.global_generic_reporting_ind,
        ccgri.claim_id,
        ccgri.customer_id,
        ccgri.customer_position_cnt,
        ccgri.value
   FROM global_indicators
   JOIN customer_indicators
     ON gcgri.customer_id = ccgri.customer_id
    AND gcgri.global_position_cnt = ccgri.global_position_cnt
    AND gcgri.global_generic_reporting_ind = ccgri.global_generic_reporting_ind
    AND gcgri.claim_id = ccgri.claim_id
    AND gcgri.customer_position_cnt = ccgri.customer_position_cnt
    AND gcgri.value = ccgri.value;
```

```

        FROM claimsprocess.claim_generic_reporting_indicators,
              jsonb_each_text(claim_generic_reporting_indicators.
claim_customer_reporting_information_json) AS x(key, value)
        WHERE jsonb_typeof(claim_generic_reporting_indicators.
claim_customer_reporting_information_json) = 'object'
              AND value IS NOT NULL) AS ccgri)

SELECT e21.customer_set_id,
       c76.customer_set_long_nm AS customer_set_nm,
       e21b.customer_set_id AS benefit_customer_set_id,
       claim_core.customer_id,
       customr0.full_title AS customer_nm,
       claim_core.client_id,
       clients0.full_title AS client_nm,
       claim_core.group_id,
       claim_core.used_external_group AS used_external_group_id,
       groups1.full_title AS group_nm,
       COALESCE(a14.accum_comp_short_nm, '') AS external_accumulator_component_id,
       COALESCE(claim_medicare_partd.coverage_gap_attributed_amt, 0.00) AS attributed_to_coverage_gap_amt,
       COALESCE(claim_medicare_partd.coverage_gap_discount_amt, 0.00) AS coverage_gap_discount_amt,
       COALESCE(claim_medicare_partd.mic_total_partd_cost_amt, 0.00) AS mic_total_partd_cost_amt,
       COALESCE(claim_medicare_partd.mic_total_partd_ingredient_cost_amt, 0.00) AS
mic_total_partd_ingredient_cost_amt,
       COALESCE(claim_medicare_partd.nx_other_payer_paid_amt, 0.00) AS nx_other_payer_paid_amt,
       COALESCE(claim_medicare_partd.coverage_gap_discount_amt, 0.00) AS reported_gap_discount_amt,
       (CASE WHEN claim_core.record_source_ind = 'S'
             THEN claim_core.total_paid_amt
             WHEN claim_extension_1.version_release_id = 'D0'
             THEN (claim_core.plan_ingredient_cost_amt + claim_core.plan_dispense_fee_amt + claim_core.
incentive_paid_amt +
                   COALESCE(m47_seq_1.claim_other_amount_paid_amt, 0) + COALESCE(m47_seq_2.
claim_other_amount_paid_amt, 0) + COALESCE(m47_seq_3.claim_other_amount_paid_amt, 0) +
                   claim_extension_1.percent_sales_tax_paid_amt)
             ELSE claim_core.plan_ingredient_cost_amt + claim_core.plan_dispense_fee_amt + claim_extension_1.
plan_sales_tax_amt + claim_core.incentive_paid_amt + claim_extension_1.generic_incentive_fee_amt
         END) - (CASE WHEN claim_extension_1.version_release_id = 'D0'
                     THEN (claim_core.plan_calculated_copay_plus_coins_amt - claim_core.
member_deductible_cost_ant)
                     ELSE claim_core.plan_calculated_copay_plus_coins_amt
                 END) - claim_core.lis_amt AS humana_paid_amt,
       claim_medicare_partd.process_as_cd AS claim_processed_as_ind,
       claim_medicare_partd.health_insurance_claim_id,
       claim_medicare_partd.other_payer_member_id,
       claim_medicare_partd.partd_drug_type_cd,
       COALESCE(mpddtc1.field_configuration_description_txt, 'Undetermined') AS partd_drug_type_description_txt,
       COALESCE(andasic.field_configuration_description_txt, '') AS abbreviated_new_drug_application_ind,
       COALESCE(andasi.field_configuration_description_txt, '') AS
abbreviated_new_drug_application_description_txt,
       COALESCE(ndasic.field_configuration_description_txt, '') AS new_drug_application_ind,
       COALESCE(ndasi.field_configuration_description_txt, '') AS new_drug_application_description_txt,
       e70.prcs_strt_tmstp AS awp_process_effective_begin_ts,
       e70.awp_disc_pctg AS drug_awp_discount_prct,
       e70.bgn_dte AS drug_awp_effective_begin_dt,
       e70.end_dte AS drug_awp_effective_end_dt,
       y13.max_daily_dose_strength_ct AS drug_adult_maximum_daily dosage_qty,
       y13.max_daily_dose_strength_cd AS drug_adult_maximum_daily_dose_unit_cd,
       y13.max_daily_dose_unit_form_cd AS drug_adult_maximum_daily_unit_form_cd,
       y13.max_daily_dose_unit_ct AS drug_adult_maximum_daily_unit_qty,
       y13.min_daily_dose_strength_ct AS drug_adult_minimum_daily dosage_qty,
       y13.min_daily_dose_strength_cd AS drug_adult_minimum_daily_dose_unit_cd,
       y13.min_daily_dose_unit_form_cd AS drug_adult_minimum_daily_unit_form_cd,
       y13.min_daily_dose_unit_ct AS drug_adult_minimum_daily_unit_qty,
       y16.max_daily_dose_strength_ct AS drug_geriatric_maximum_daily dosage_qty,
       y16.max_daily_dose_strength_cd AS drug_geriatric_maximum_daily_dose_unit_cd,
       y16.max_daily_dose_unit_form_cd AS drug_geriatric_maximum_daily_unit_form_cd,
       y16.max_daily_dose_unit_ct AS drug_geriatric_maximum_daily_unit_qty,
       y16.min_daily_dose_strength_ct AS drug_geriatric_minimum_daily dosage_qty,
       y16.min_daily_dose_strength_cd AS drug_geriatric_minimum_daily_dose_unit_cd,
       y16.min_daily_dose_unit_form_cd AS drug_geriatric_minimum_daily_unit_form_cd,
       y16.min_daily_dose_unit_ct AS drug_geriatric_minimum_daily_unit_qty,
       y00.medicare_reference_cd AS drug_hcfa_common_procedure_cd,
       x80.hcfa_shelf_life_termtn_dt AS drug_hcfa_shelf_life_effective_end_dt,

```

```

x85.ingredient_list_id AS drug_hicl_sequence_nbr,
COALESCE(diic.field_configuration_description_txt, '') AS drug_innovator_indicator_cd,
COALESCE(dii.field_configuration_description_txt, 'Unknown') AS drug_innovator_indicator_description_txt,
COALESCE(dipic.field_configuration_description_txt, 'Undetermined') AS
drug_institutional_product_description_txt,
x80.instnl_product_ind_cd AS drug_institutional_product_ind,
x80.label_nm AS drug_label_nm,
x80.bla_approved_cd AS drug_bla_approved_cd,
COALESCE(dbac.field_configuration_description_txt, 'Unknown') AS drug_bla_approved_description_txt,
x80.brand_nm AS drug_brand_nm,
x80.generic_name_cd AS drug_brand_name_cd,
CASE WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '1'
THEN 'Generic'
WHEN claim_core.generic_ind = '2' AND claim_core.generic_product_ind = '2'
THEN 'Brand'
WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '2'
THEN 'Brand'
WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '3'
THEN 'Brand'
ELSE 'Undetermined'
END AS drug_brand_or_generic_txt,
CASE WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '0'
THEN 'Supplies'
WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '1'
THEN 'Generic'
WHEN claim_core.generic_ind = '2' AND claim_core.generic_product_ind = '2'
THEN 'Single Source Brand'
WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '2'
THEN 'Multi Source Brand'
WHEN claim_core.generic_ind = '1' AND claim_core.generic_product_ind = '3'
THEN 'Cross Brand'
ELSE 'Unknown'
END AS drug_brand_type_txt,
x80.drug_class_cd AS drug_class_ind,
COALESCE(dcc.field_configuration_description_txt, 'Unknown') AS drug_class_indicator_description_txt,
x80.desi_2_status_dt AS drug_desi_2_change_effective_dt,
x80.ndc_add_dt AS drug_fdb_add_dt,
x80.obsolete_dt AS drug_fdb_obsolete_dt,
x80.ndc_update_dt AS drug_fdb_update_dt,
x80.hcfa_approval_dt AS drug_hcfa_approval_dt,
x80.hcfa_desi_status_dt AS drug_hcfa_desi_status_dt,
x80.hcfa_market_entry_dt AS drug_hcfa_market_entry_dt,
x80.generic_manufacturer_ind_cd AS drug_generic_manufacturer_ind,
y06.generic_nm AS drug_generic_nm,
x80.therapeutic_evalnnc_ind_cd AS drug_generic_therapy_class_ind,
x80.last_mnt_ts AS drug_last_maintenance_ts,
x80.unit_dos_only_cd AS drug_unit_dose_ind,
x80.repackaged_ind_cd AS drug_repackage_ind,
COALESCE(drlic.field_configuration_description_txt, 'Undetermined') AS drug_repackage_description_txt,
x80.core_9_sml_package_size_ct AS drug_core_9_smallest_package_size_cnt,
x80.gcn_sml_package_size_ct AS drug_gcn_smallest_package_size_cnt,
claim_core.member_id,
claim_core.claim_id,
claim_core.fill_dt,
DATE_PART('month', claim_core.fill_dt)::INTEGER AS fill_month_nbr,
TO_CHAR(claim_core.fill_dt, 'Mon') AS fill_month_nm,
DATE_PART('quarter', claim_core.fill_dt)::INTEGER AS fill_quarter_nbr,
DATE_PART('week', claim_core.fill_dt)::INTEGER AS fill_week_nbr,
DATE_PART('year', claim_core.fill_dt)::INTEGER AS fill_year_nbr,
claim_core.received_dt,
a_claim_core.process_adjust_created_ts AS claim_adjustment_ts,
claim_core.adjudication_cd,
claim_core.approved_amt,
claim_core.awp_ingredient_cost_amt,
claim_core.calculated_submitted_balance_amt,
claim_core.cics_claim_status_cd AS claim_status_detail_cd,
COALESCE(csdic.field_configuration_description_txt, '') AS claim_status_detail_description_txt,
claim_core.claim_status_cd,
COALESCE(csc.field_configuration_description_txt, '') AS claim_status_code_description_txt,
(SELECT CASE WHEN claim_core.cics_claim_status_cd IN ('HRV', 'HIS')
THEN 'Other Claim'

```

```

WHEN claim_core.client_id = 9999
THEN 'Other Claim'
WHEN claim_core.claim_status_cd IN (10, 11)
THEN 'Paid Claim'
WHEN claim_core.claim_status_cd IN (70, 71)
THEN CASE WHEN ce.claim_error_cd = '074'
        THEN 'Reversed Claim'
        WHEN ce.auto_deny_error = 'Y'
        THEN 'Denied Claim'
        ELSE 'Other Claim'
      END
ELSE 'Other Claim'
END AS "case"
FROM claimsprocess.claim_error ce
WHERE ce.customer_id = claim_core.customer_id
  AND ce.claim_id = claim_core.claim_id
LIMIT 1) AS claim_status_description_txt,
claim_error.claim_error_reason_subcategory_set,
(SELECT CASE WHEN claim_core.claim_status_cd IN (10, 11)
            THEN CASE WHEN ce.claim_error_cd > '' AND ce.claim_error_sequence_id = 1
                      THEN 'Soft Error'
                      ELSE ''
                    END
            WHEN ce.claim_error_cd = '074'
            THEN 'First Hard Error'
            WHEN ce.auto_deny_error = 'Y'
            THEN 'First Hard Error'
            ELSE CASE WHEN ce.claim_error_cd > '' AND ce.claim_error_sequence_id = 1
                      THEN 'Soft Error'
                      ELSE ''
                    END
          END
        END AS "case"
FROM claimsprocess.claim_error ce
WHERE ce.customer_id = claim_core.customer_id
  AND ce.claim_id = claim_core.claim_id
LIMIT 1) AS error_type_description_txt,
claim_core.adjudication_cd||claim_core.customized_formulary_ind||claim_core.lics_ind AS
claim_supplemental_ind,
claim_core.claim_type_cd,
claim_core.compound_cd,
COALESCE(cc.field_configuration_description_txt, 'Non-Compound') AS compound_code_description_txt,
claim_core.compound_dispense_form_ind,
claim_core.compound dosage_description_cd,
claim_core.compound_route_of_administration_cd,
SUBSTR(claim_core.ndc_id, 1, 9) AS drug_core_9_cd,
claim_core.coverage_cd_1,
claim_core.coverage_cd_2,
groups1.coverage_code_1 AS group_coverage_cd_1,
groups1.coverage_code_2 AS group_coverage_cd_2,
groups1.grp_begin_date_1 AS group_effective_begin_dt_1,
groups1.grp_begin_date_2 AS group_effective_begin_dt_2,
groups1.grp_begin_date_3 AS group_effective_begin_dt_3,
groups1.grp_end_date_1 AS group_effective_end_dt_1,
groups1.grp_end_date_2 AS group_effective_end_dt_2,
groups1.grp_end_date_3 AS group_effective_end_dt_3,
claim_core.customer_calculated_copay_amt,
claim_core.customer_calculated_dispense_fee_amt,
claim_core.customer_calculated_ingredient_cost_amt,
claim_core.customized_authorization_nbr,
claim_dmr_external.dmr_claim_external_id AS dcn_nbr,
claim_core.deductible_excluded_amt,
claim_core.deductible_specific_amt,
claim_core.deductible_specific_excluded_amt,
claim_core.deductible_status_ind AS deductible_status_ind_1,
claim_core.deductible_status_ind_2,
claim_core.drug_class_cd,
claim_core.drug_list_table_nm,
claim_core.member_history_created_ind,
claim_core.dur_conflict_cd_1,
claim_core.dur_conflict_cd_2,
claim_core.dur_conflict_cd_3,

```

```
claim_core.dur_conflict_cd_4,
claim_core.dur_conflict_cd_5,
claim_core.dur_conflict_cd_6,
claim_core.dur_conflict_cd_7,
claim_core.dur_conflict_cd_8,
claim_core.dur_conflict_cd_9,
claim_core.dur_intervention_cd_1,
claim_core.dur_intervention_cd_2,
claim_core.dur_intervention_cd_3,
claim_core.dur_intervention_cd_4,
claim_core.dur_intervention_cd_5,
claim_core.dur_intervention_cd_6,
claim_core.dur_intervention_cd_7,
claim_core.dur_intervention_cd_8,
claim_core.dur_intervention_cd_9,
claim_core.dur_outcome_cd_1,
claim_core.dur_outcome_cd_2,
claim_core.dur_outcome_cd_3,
claim_core.dur_outcome_cd_4,
claim_core.dur_outcome_cd_5,
claim_core.dur_outcome_cd_6,
claim_core.dur_outcome_cd_7,
claim_core.dur_outcome_cd_8,
claim_core.dur_outcome_cd_9,
claim_core.dur_pps_level_effort_1,
claim_core.dur_pps_level_effort_2,
claim_core.dur_pps_level_effort_3,
claim_core.dur_pps_level_effort_4,
claim_core.dur_pps_level_effort_5,
claim_core.dur_pps_level_effort_6,
claim_core.dur_pps_level_effort_7,
claim_core.dur_pps_level_effort_8,
claim_core.dur_pps_level_effort_9,
claim_core.formulary_ind,
claim_core.gcn_id AS claim_gcn_id,
x80.gcn_id AS drug_gcn_id,
claim_core.generic_ind,
COALESCE(gi.field_configuration_description_txt, 'Unspecified') AS generic_indicator_description_txt,
claim_core.generic_product_ind,
COALESCE(gpi.field_configuration_description_txt, 'Unspecified') AS generic_product_description_txt,
x80.gpi_cd AS current_drug_generic_price_ind,
claim_core.incentive_paid_amt,
claim_core.lis_amt AS cms_lics_amt,
claim_core.lis_cd AS cms_lics_cd,
claim_core.mac_ingredient_cost_amt,
claim_core.mac_unit_cost,
claim_core.member_deductible_applied_amt,
claim_core.member_deductible_cost_amt,
claim_core.member_deductible_plan_ind,
claim_core.member_pay_difference_amt AS mpd_amt,
claim_core.mop_specific_amt,
claim_core.mop_specific_excluded_amt,
claim_core.mpp_specific_excluded_amt,
claim_core.ndc_amt,
claim_core.ndc_drug_category_cd,
claim_core.ndc_id,
claim_core.ndc_label_nm,
claim_core.network_differential_amt,
claim_core.network_id,
claim_core.network_ind,
claim_core.npi_pharmacy_id,
claim_core.other_payer_recognized_amt,
claim_core.other_troop_amt,
claim_core.override_copay_ind,
claim_core.paid_professional_service_amt,
claim_core.plan_calculated_copay_plus_coins_amt,
claim_core.plan_client_administrative_expense_amt,
claim_core.plan_days_supply_cnt,
claim_core.plan_deductible_applied_amt,
claim_core.plan_dispense_fee_amt,
claim_core.plan_maximum_specific_amt,
```

```

claim_core.plan_metric_decimal_qty,
claim_core.plan_withhold_fee_amt,
claim_core.plro_amnt,
claim_core.postpay_ind_2,
claim_core.postpay_ind_3,
claim_core.postpay_ind_4,
claim_core.preauthorization_nbr,
claim_core.prescription_origin_cd,
claim_core.price_source_ind,
COALESCE(psi.field_configuration_description_txt, '') AS price_source_indicator_description_txt,
f96.pricing_source_abbrn_nm AS price_source_abbreviated_nm,
claim_core.process_create_ts,
claim_core.process_create_by_user_id,
claim_core.process_create_by_program_id,
claim_core.process_update_ts,
claim_medicare_partd.process_as_cd,
claim_core.provider_status_cd AS pharmacy_status_cd,
CASE WHEN SUBSTR(claim_core.provider_status_cd, 1, 1) = 'M'
    THEN 'Mail'
    ELSE 'Retail'
END AS claim_mail_retail_ind,
claim_core.refill_nbr,
CASE WHEN claim_core.refill_nbr = 0
    THEN 'First Fill'
    ELSE 'Refill'
END AS refill_status_description_txt,
claim_core.restricted_authorization_nbr,
claim_core.reversal_pos_id,
claim_core.specialty_drug_ind,
COALESCE(sd.field_configuration_description_txt, 'Unknown') AS specialty_description_txt,
claim_core.specific_therapy_class_cd,
y10.specific_tc_desc_tx AS specific_therapy_class_description_txt,
claim_core.standard_package_ind,
CASE WHEN claim_core.standard_package_ind = '1'
    THEN 'Maintenance'
    ELSE 'Non-Maintenance'
END AS drug_maintenance_description_txt,
x90.manufacturer_nm AS drug_manufacturer_nm,
x80.medication_id AS drug_medication_id,
y02.drug_strength_ct AS drug_strength_qty,
y02.drug_strength_unit_tx AS drug_strength_unit_size_txt,
y02.drug_strength_vol_ct AS drug_strength_volume_cnt,
claim_core.submitted_birth_dt,
claim_core.submitted_copay_amt,
claim_core.submitted_dispense_as_written_ind AS submitted_daw_ind,
COALESCE(sdi.field_configuration_description_txt, '') AS submitted_daw_indicator_description_txt,
claim_core.submitted_incentive_amt,
claim_core.submitted_ingredient_cost_amt,
claim_core.submitted_preauthorization_nbr,
claim_core.submitted_total_prescription_cost_amt,
claim_core.therapy_protocol_nm,
claim_core.tier_nbr,
claim_core.total_member_liability_amt,
claim_core.total_paid_amt,
claim_core.troop_applied_amt,
claim_core.troop_eligibility_ind,
claim_core.usual_and_customary_amt,
claim_core.wac_ingredient_cost_amt,
(SELECT x88.ndc_price
  FROM global.x88 x88
 WHERE x88.ndc_id = claim_core.ndc_id
   AND x88.ndc_price_type_cd = '09'
   AND x88.ndc_price_dt <= CURRENT_DATE
 ORDER BY x88.ndc_price_dt DESC
 LIMIT 1) AS current_wac_unit_price_amt,
claim_core.wc_injury_dt,
claim_core.member_gender,
CASE WHEN claim_core.member_gender = 'F'
    THEN 'Female'
    WHEN claim_core.member_gender = 'M'
    THEN 'Male'

```

```

        ELSE 'Unknown'
    END AS member_gender_description_txt,
    claim_extension_1.alternate_price_benchmark_cost_amt,
    claim_extension_1.assign_qualified_health_plan_id,
    claim_extension_1.authorization_id,
    claim_extension_1.base_price_source_cd,
    claim_extension_1.billing_entity_type_cd,
    claim_extension_1.claim_benefit_stage_amt_1,
    claim_extension_1.claim_benefit_stage_amt_2,
    claim_extension_1.claim_benefit_stage_amt_3,
    claim_extension_1.claim_benefit_stage_amt_4,
    claim_extension_1.claim_benefit_stage_type_cd_1,
    claim_extension_1.claim_benefit_stage_type_cd_2,
    claim_extension_1.claim_benefit_stage_type_cd_3,
    claim_extension_1.claim_benefit_stage_type_cd_4,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '04'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '04'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '04'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '04'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_catastrophic_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '63'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '63'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '63'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '63'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_coadm_medicaid_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '62'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '62'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '62'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '62'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_coadm_non_partd_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '61'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '61'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '61'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '61'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_coadm_partd_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '03'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '03'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '03'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '03'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_coverage_gap_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '80'
        THEN claim_benefit_stage_amt_1
        WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '80'
        THEN claim_benefit_stage_amt_2
        WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '80'
        THEN claim_benefit_stage_amt_3
        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '80'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_ddc_non_partd_amt,
    CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '70'
        THEN claim_benefit_stage_amt_1

```

```

WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '70'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '70'
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '70'
THEN claim_benefit_stage_amt_4
END AS benefit_stage_ddc_partd_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '01'
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '01'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '01'
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '01'
THEN claim_benefit_stage_amt_4
END AS benefit_stage_deductible_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '02'
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '02'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '02'
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '02'
THEN claim_benefit_stage_amt_4
END AS benefit_stage_initial_coverage_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '50'
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '50'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '50'
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '50'
THEN claim_benefit_stage_amt_4
END AS benefit_stage_mapd_plan_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '90'
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '90'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '90'
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '90'
THEN claim_benefit_stage_amt_4
END AS benefit_stage_non_partd_covered_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = ' '
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = ' '
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = ' '
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = ' '
THEN claim_benefit_stage_amt_4
END AS benefit_stage_not_specified_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 NOT IN ('01', '02', '03', '04', '50', '60',
'61', '62', '63', '70', '80', '90')
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 NOT IN ('01', '02', '03', '04', '50', '60',
'61', '62', '63', '70', '80', '90')
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 NOT IN ('01', '02', '03', '04', '50', '60',
'61', '62', '63', '70', '80', '90')
THEN claim_benefit_stage_amt_3
WHEN claim_extension_1.claim_benefit_stage_type_cd_4 NOT IN ('01', '02', '03', '04', '50', '60',
'61', '62', '63', '70', '80', '90')
THEN claim_benefit_stage_amt_4
END AS benefit_stage_other_amnt,
CASE WHEN claim_extension_1.claim_benefit_stage_type_cd_1 = '60'
THEN claim_benefit_stage_amt_1
WHEN claim_extension_1.claim_benefit_stage_type_cd_2 = '60'
THEN claim_benefit_stage_amt_2
WHEN claim_extension_1.claim_benefit_stage_type_cd_3 = '60'
THEN claim_benefit_stage_amt_3

```

```

        WHEN claim_extension_1.claim_benefit_stage_type_cd_4 = '60'
        THEN claim_benefit_stage_amt_4
    END AS benefit_stage_wrap_plan_amt,
p32.effective_dt AS benefit_coverage_effective_begin_dt,
p32.termination_dt AS benefit_coverage_effective_end_dt,
CASE WHEN p32.coverage_id IS NULL
    THEN 'Legacy'
    ELSE 'RxNova'
END AS benefit_processed_as_description_txt,
p32.coverage_description_tx AS coverage_code_description_txt,
claim_core.formatted_diagnosis_cd_1,
claim_core.formatted_diagnosis_cd_2,
claim_core.formatted_diagnosis_cd_3,
claim_core.formatted_diagnosis_cd_4,
claim_core.formatted_diagnosis_cd_5,
claim_core.formatted_diagnosis_cd_qualifier_1,
claim_core.formatted_diagnosis_cd_qualifier_2,
claim_core.formatted_diagnosis_cd_qualifier_3,
claim_core.formatted_diagnosis_cd_qualifier_4,
claim_core.formatted_diagnosis_cd_qualifier_5,
claim_extension_1.client_pricing_calculation_cd,
claim_extension_1.cost_share_reduction_cd,
CASE WHEN claim_extension_1.cost_share_reduction_cd IS NULL
    THEN 'No Cost Share Reduction'
    ELSE COALESCE(csrc.field_configuration_description_txt, 'Unknown')
END AS cost_share_reduction_code_description_txt,
claim_extension_1.cost_share_reduction_mop_variance_amt,
claim_extension_1.cost_share_reduction_variance_amt,
claim_extension_1.cost_share_reduction_variance_ded_amt,
claim_extension_1.ethnicity_cd,
claim_extension_1.generic_incentive_fee_amt,
claim_extension_1.grace_period_cd,
claim_extension_1.health_coverage_policy_id,
claim_extension_1.health_coverage_policy_qlfr_cd AS health_coverage_policy_id_qlfr,
claim_extension_1.health_plan_id,
claim_extension_1.hicl_external_sequence_id,
claim_extension_1.mandatory_fill_type_cd,
claim_extension_1.mandatory_plan_type_cd,
claim_extension_1.mandatory_specialty_edit_cd,
claim_extension_1.marketplace_plan_type_cd,
claim_extension_1.medication_id,
claim_extension_1.member_first_nm,
claim_extension_1.member_last_nm,
o58.middle_initial AS member_middle_initial,
o58.str_addr_line_1_tx AS member_street_address_line_txt_1,
o58.str_addr_line_2_tx AS member_street_address_line_txt_2,
CASE WHEN o58.birth_date <> 0
    THEN AGE(CURRENT_DATE::TIMESTAMP WITH TIME ZONE, o58.birth_date::TEXT::DATE::TIMESTAMP WITH TIME
ZONE)
        ELSE '00:00:00'::INTERVAL
    END::TEXT AS member_age,
CASE WHEN o58.birth_date <> 0
    THEN AGE(claim_core.fill_dt::TIMESTAMP WITH TIME ZONE, o58.birth_date::TEXT::DATE::TIMESTAMP WITH
TIME ZONE)
        ELSE '00:00:00'::interval
    END::TEXT AS member_age_claim,
o58.alt_mbr_id_2 AS alternate_member_id_2,
o58.alt_mbr_id_3 AS alternate_member_id_3,
o58.cardholder_first_name AS member_cardholder_first_nm,
o58.cardholder_last_name AS member_cardholder_last_nm,
o58.cardholder_middle_initial AS member_cardholder_middle_initial,
o58.service_city AS member_city,
o58.service_state AS member_state_cd,
o58.service_zip_code AS member_zip_cd,
o58.service_zip_code_4 AS member_zip_cd_4,
o58.home_phone AS member_phone_nbr,
CASE WHEN claim_extension_1.version_release_id = 'D0'
    THEN (claim_core.plan_calculated_copay_plus_coins_amt - claim_core.member_deductible_cost_amt)
    ELSE claim_core.plan_calculated_copay_plus_coins_amt
END AS member_copay_amt,
CASE WHEN claim_extension_1.version_release_id = 'D0'

```

```

THEN claim_core.plan_calculated_copay_plus_coins_amt
ELSE (claim_core.plan_calculated_copay_plus_coins_amt + claim_core.member_deductible_cost_amt)
END AS member_cost_amt,
claim_core.dual_coverage_ind AS member_dual_coverage_ind,
o58.esrd_ind AS member_esrd_ind,
o58.esrd_process_date AS member_esrd_process_dt,
o58.hlth_id_cde AS member_health_identification_cd,
o58.pref_lang_code AS member_preferred_communication_cd,
claim_extension_1.original_prescribed_product_id,
claim_extension_1.partd_qualified_facility_cd,
claim_extension_1.place_of_service_type_cd,
claim_extension_1.plan_sales_tax_amt,
claim_extension_1.pricing_list_nm_1,
claim_extension_1.processor_fee_attributed_amt,
claim_extension_1.reduce_copay_option_cd,
claim_extension_1.residence_type_cd,
claim_extension_1.standard_coinsurance_amt,
claim_extension_1.standard_copay_amt,
claim_extension_1.standard_deductible_amt,
claim_extension_1.standard_mop_amt,
claim_extension_1.submitted_prescription_id,
claim_vendor_drug.fdb_ahfs_therapeutic_class_cd,
claim_vendor_drug.fdb_core9_smallest_package_size_cnt,
claim_vendor_drug.fdb_dea_class_cd,
claim_vendor_drug.fdb_delete_dt,
claim_vendor_drug.fdb_gcn_smallest_package_size_cnt,
claim_vendor_drug.fdb_generic_available_cd,
claim_vendor_drug.fdb_generic_ind_cd AS fdb_generic_indicator_cd,
claim_vendor_drug.fdb_generic_price_ind_cd AS fdb_generic_price_indicator_cd,
claim_vendor_drug.fdb_maintenance_drug_cd,
claim_vendor_drug.fdb_obsolete_dt,
claim_vendor_drug.fdb_orange_book_available_cd,
claim_vendor_drug.fdb_otc_available_cd,
claim_vendor_drug.fdb_package_qty_cnt AS fdb_package_qty,
claim_vendor_drug.fdb_package_size_cnt,
claim_vendor_drug.fdb_package_unit_measure_cd,
claim_vendor_drug.fdb_partd_drug_typ_cd AS fdb_partd_drug_type_cd,
claim_vendor_drug.fdb_repackage_cd,
claim_vendor_drug.fdb_prescription_otc_cd,
claim_vendor_drug.fdb_unit dosage_cd,
claim_vendor_drug.fdb_unit dosage_only_cd,
x80.ndc_update_dt AS fdb_update_dt,
claim_vendor_drug.medispan_ahfs_therapy_class_cd,
claim_vendor_drug.medispan_core9_small_package_size_cnt,
claim_vendor_drug.medispan_dea_class_cd,
claim_vendor_drug.medispan_delete_dt,
claim_vendor_drug.medispan_dosage_form_description_txt,
claim_vendor_drug.medispan_extend_ahfs_therapy_class_cd,
claim_vendor_drug.medispan_generic_small_package_size_cnt AS medispan_gcn_small_package_size_cnt,
claim_vendor_drug.medispan_generic_available_cd,
claim_vendor_drug.medispan_generic_id,
claim_vendor_drug.medispan_generic_ind_cd AS medispan_generic_indicator_cd,
claim_vendor_drug.medispan_generic_price_ind_cd AS medispan_generic_price_indicator_cd,
claim_vendor_drug.medispan_generic_product_id,
claim_vendor_drug.medispan_ingredient_generic_id,
claim_vendor_drug.medispan_labeler_id,
claim_vendor_drug.medispan_maintenance_drug_cd,
claim_vendor_drug.medispan_multiple_source_cd,
claim_vendor_drug.medispanObsolete_dt,
claim_vendor_drug.medispan_orange_book_available_cd,
claim_vendor_drug.medispan_otc_available_cd,
claim_vendor_drug.medispan_package_cnt,
claim_vendor_drug.medispan_package_size_cnt,
claim_vendor_drug.medispan_package_unit_measure_cd,
claim_vendor_drug.medispan_partd_drug_typ_cd AS medispan_partd_drug_type_cd,
claim_vendor_drug.medispan_repackage_cd,
claim_vendor_drug.medispan_route_administration_cd,
claim_vendor_drug.medispan_prescription_otc_cd,
claim_vendor_drug.medispan_therapeutic_equivalence_eval_cd AS medispan_tee_cd,
claim_vendor_drug.medispan_third_party_restriction_cd,
claim_vendor_drug.medispan_unit dosage_cd,

```

```

claim_vendor_drug.medispan_unit_dosage_only_cd,
claim_core.case_id,
claim_core.pharmacy_id,
claim_core.member_birth_dt,
claim_core.member_relationship_cd,
claim_core.prescription_id,
claim_core.submitted_days_supply_cnt,
claim_core.plan_prescriber_id,
claim_core.submitted_dispense_fee_amt,
claim_core.submitted_sales_tax_amt,
claim_core.submitted_balance_amt,
claim_core.prescription_written_dt,
claim_core.plan_ingredient_cost_amt,
claim_core.adjudicated_amt,
claim_core.dea_cd,
CASE WHEN claim_core.dea_cd = '0'
      THEN 'Non-Controlled'
      ELSE 'Controlled'
END AS dea_description_txt,
COALESCE(dc.field_configuration_description_txt, 'Unknown') AS dea_code_description_txt,
x85.dosage_form_cd AS drug_dosage_form_cd,
y03.dosage_form_desc_tx AS drug_dosage_form_description_txt,
x85.drug_strength_desc_tx AS drug_dosage_strength_txt,
x80.part_d_drug_type_cd AS drug_fda_defined_status_cd,
COALESCE(mpddtc2.field_configuration_description_txt, 'Undetermined') AS
drug_fda_defined_status_description_txt,
x80.drug_form_cd AS drug_form_cd,
COALESCE(dfc.field_configuration_description_txt, 'Unknown') AS drug_form_description_txt,
y32.evd_ext_vocab_id AS drug_prescription_concept_unique_id,
claim_core.standard_class_cd,
claim_core.standard_therapy_class_cd,
y37.standard_tc_desc_tx AS standard_therapy_class_description_txt,
claim_core.generic_therapy_class_cd,
y12.generic_tc_desc_tx AS generic_therapy_class_description_txt,
claim_core.member_effective_begin_dt,
claim_core.member_effective_end_dt,
claim_core.postpay_ind_1,
claim_core.prepay_review_ind,
claim_core.reversal_pos_ts,
claim_core.member_miscellaneous_data_1,
claim_core.flex_report_cd,
claim_core.member_patient_nm,
claim_core.pharmacy_affiliation_id,
COALESCE(s50c_affiliate.pharmacy_nm, s50m_affiliate.pharmacy_nm, s50a_affiliate.pharmacy_nm) AS
pharmacy_affiliation_nm,
CASE WHEN COALESCE(claim_core.pharmacy_affiliation_id,0) = 0 AND COALESCE(claim_core.pharmacy_chain_id,
0) = 0
      THEN 'Independent'
      WHEN COALESCE(claim_core.pharmacy_affiliation_id,0) = 0 AND COALESCE(claim_core.pharmacy_chain_id,
0) > 0
      THEN 'Chain'
      WHEN COALESCE(claim_core.pharmacy_affiliation_id,0) > 0 AND COALESCE(claim_core.pharmacy_chain_id,
0) = 0
      THEN 'Affiliate'
      WHEN COALESCE(claim_core.pharmacy_affiliation_id,0) > 0 AND COALESCE(claim_core.pharmacy_chain_id,
0) > 0
      THEN 'Affiliate'
      ELSE 'Unknown'
END AS pharmacy_affiliation_description_txt,
claim_core.member_reimbursement_ind,
claim_core.record_source_ind,
COALESCE(rsi.field_configuration_description_txt, 'Unknown') AS record_source_description_txt,
claim_core.adjusted_dt,
claim_core.partial_fill_copay_ind,
claim_core.pricing_module_program,
claim_core.pricing_module_version,
claim_core.dosage_strength,
claim_core.package_size,
claim_core.ndc_format_ind,
claim_core.ahfs_therapy_class AS ahfs_therapy_class_cd,
y11.ahfs_tc_desc_tx AS ahfs_therapy_class_description_txt,

```

```
claim_core.prescription_reason_cd AS dmr_reason_cd,
claim_core.submitted_member_id,
claim_core.submitted_member_person_cd,
claim_core.pricing_type_ind,
claim_core.submitted_prescriber_id,
claim_core.chronic_medication_scripts,
claim_core.customized_formulary_ind,
COALESCE(cfi.field_configuration_description_txt, 'Unknown') AS customized_formulary_description_txt,
claim_core.lics_ind,
claim_core.lis_level AS lics_level,
claim_core.lis_effective_begin_dt AS lics_effective_begin_dt,
claim_core.lis_effective_end_dt AS lics_effective_end_dt,
claim_core.customer_calculated_sales_tax_amt,
claim_core.ndc_manufacturer_nm,
claim_core.deductible_applied_ingredient_cost_amt,
claim_core.deductible_applied_dispense_fee_amt,
claim_core.deductible_applied_sales_tax_amt,
claim_core.refillsAuthorized,
claim_core.prior_authorization_cd,
claim_core.primary_prescriber AS primary_prescriber_id,
claim_core.clinic_id,
claim_core.basis_of_cost AS submitted_basis_of_cost_cd,
claim_core.diagnosis_cd,
claim_core.medicaid_cost_amt,
claim_core.member_record_status_ind,
claim_core.member_miscellaneous_data_2,
claim_core.member_miscellaneous_data_3,
SUBSTR(claim_core.member_miscellaneous_data_3, 12, 3) AS plan_benefit_segment_id,
claim_core.member_cardholder_nm,
claim_core.quantity_rounding_cd,
claim_core.drug_interaction_ind,
claim_core.duplicate_therapy_ind,
claim_core.drug_of_preference_ind AS dop_ind,
claim_core.usage_range_check_ind,
claim_core.alternate_member_id,
claim_core.benefit_plan_id,
claim_core.override_copay_amt,
claim_core.override_copay_prct,
claim_core.member_network_ind,
claim_core.alternate_business AS alternate_business_cd,
claim_core.therapy_protocol_ind,
claim_core.prescriber_specialty_cd,
au0.prescriber_specialty_lng_nm AS prescriber_specialty_nm,
claim_core.report_for_rebate_ind,
claim_core.drug_age_conflict_ind,
claim_core.drug_disease_ind,
claim_core.drug_allergy_ind,
claim_core.refill_too_soon_ind,
claim_core.prescriber_plan_id,
claim_core.cost_plan_nm,
claim_core.submitted_metric_decimal_qty,
claim_core.override_module_nm,
claim_core.override_module_version,
claim_core.plan_group_cd,
claim_core.city_cd,
claim_core.copay_source_cd,
claim_core.plan_copay_amt,
claim_core.partial_fill_claim_id,
claim_core.dispensing_status_ind,
claim_core.associated_prescription_id,
claim_core.associated_prescription_service_dt,
claim_core.intended_to_be_dispensed_qty,
claim_core.intended_to_be_dispensed_days_supply_cnt,
claim_core.basis_calculated_dispense_fee_cd,
claim_core.basis_calculated_copay_cd,
claim_core.orange_book_cd,
x80.otc_available_cd AS drug_otc_available_ind,
x80.package_size_ct AS drug_package_size_cnt,
claim_core.pharmacy_chain_id,
COALESCE(s50c_chain.pharmacy_nm, s50m_chain.pharmacy_nm, s50a_chain.pharmacy_nm) AS pharmacy_chain_nm,
claim_core.partial_fill_dispense_fee_amt,
```

```
claim_core.savings_fee_amt,
claim_core.gpi_claim_status_ind,
claim_core.original_plan_allowed_amt,
claim_core.cob_ind,
claim_core.medicaid_agency_id,
claim_core.medicaid_claim_id,
claim_core.medicaid_patient_id_qualifier,
claim_core.medicaid_patient_id,
claim_core.alternate_total_paid_amt,
claim_core.alternate_patient_pay_amt,
claim_core.alternate_ingredient_cost_amt,
claim_core.alternate_sales_tax_amt,
claim_core.alternate_dispense_fee_amt,
claim_core.cob_applied_amt,
claim_core.pos_rebate_amt,
claim_core.cms_processed_ind,
claim_core.internal_processing_amt,
claim_core.customer_processing_amt,
claim_core.rebate_invoice_manufacturer_amt,
claim_core.administrative_fee_amt,
claim_core.rebate_ind,
claim_core.rebate_price_source_ind,
claim_core.manufacturer_administrative_fee_ind,
claim_core.rebate_used_pct,
claim_core.hra_applied_amt,
claim_core.hsa_applied_amt,
claim_core.fsa_applied_amt,
claim_core.deductible_system_ind AS deductible_system_ind_1,
claim_core.pos_rebate_contract_id,
claim_core.line_of_business_cd AS lob_cd,
claim_core.mpd_applied_to_deductible_amt,
claim_core.mpd_applied_to_mop_amt,
claim_core.drug_discount_ind,
claim_core.total_plan_liability_amt,
claim_core.post_processing_fee_amt,
claim_core.mtm_cd,
claim_core.mtm_effective_begin_dt,
claim_core.institutional_cd,
COALESCE(ic.field_configuration_description_txt, 'Unknown') AS claim_institutional_code_description_txt,
claim_core.institutional_effective_begin_dt,
claim_core.prescriber_idQualifier_cd,
claim_core.pharmacy_idQualifier AS pharmacy_id_qlfr,
claim_core.cms_transaction_reference_nbr,
claim_core.submitted_professional_service_fee_amt,
claim_core.total_cob_applied_amt,
claim_core.product_service_idQualifier AS product_service_id_qlfr,
claim_core.product_service_id,
claim_core.level_of_service_cd,
claim_core.lis_deductible_cost_amt AS lics_deductible_cost_amt,
claim_core.npi_prescriber_id,
claim_core.cms_transition_cd,
claim_core.adjudication_applied_amt,
claim_core.spap_applied_amt,
claim_core.claim_sequence_id,
claim_extension_1.medicaid_government_unit_id,
claim_extension_1.provider_id,
claim_extension_1.provider_id_type_cd,
claim_extension_1.service_pharmacy_id,
claim_extension_1.generic_equivalent_product_id,
claim_extension_1.submitted_customer_id,
claim_extension_1.submitted_client_id,
claim_extension_1.submitted_external_group_id,
claim_extension_1.submitted_cardholder_id,
claim_extension_1.submitted_bin_id,
claim_extension_1.submitted_pcn_id,
claim_extension_1.returned_client_id,
claim_extension_1.returned_external_group_id,
claim_extension_1.returned_cardholder_id,
claim_extension_1.returned_bin_id,
claim_extension_1.returned_pcn_id,
claim_extension_1.transaction_cd,
```

```
claim_extension_1.provider_accept_assignment_cd,
claim_extension_1.facility_id,
claim_extension_1.facility_long_nm,
claim_extension_1.medicaid_member_id,
claim_extension_1.member_benefit_assignment_cd,
claim_extension_1.route_of_administration_id,
claim_extension_1.compound_type_cd,
claim_extension_1.prescriber_first_nm,
claim_extension_1.prescriber_last_nm,
claim_extension_1.pay_to_id_type_cd,
claim_extension_1.pay_to_id,
claim_extension_1.pay_to_nm,
p.payment_dt AS check_dt,
p.payment_instrument_nbr AS check_nbr,
p.payment_dt AS paid_dt,
c.cycle_effective_end_dt AS financial_cutoff_dt,
claim_extension_1.member_sales_tax_amt,
claim_extension_1.coinsurance_amt,
claim_extension_1.copay_amt,
claim_extension_1.coinsurance_calculation_basis_cd,
claim_extension_1.estimated_generic_savings_amt,
claim_extension_1.spending_account_remaining_amt,
claim_extension_1.health_plan_funded_assistance_amt,
claim_extension_1.prescriber_network_selection_attributed_amt,
claim_extension_1.product_selection_brand_attributed_amt,
claim_extension_1.product_selection_np_frmlly_attr_amt,
claim_extension_1.product_selection_np_brand_frmlly_attr_amt,
claim_extension_1.ingredient_cost_reimbursable_amt,
claim_extension_1.dispensing_fee_reimbursable_amt,
claim_extension_1.version_release_id,
claim_extension_1.prescription_service_reference_type_cd,
claim_extension_1.prior_authorization_type_cd,
claim_extension_1.percentage_sales_tax_submitted_amt,
claim_extension_1.percentage_sales_tax_submitted_rt,
claim_extension_1.percentage_sales_tax_submitted_basis_cd,
claim_extension_1.medicaid_paid_amt,
claim_extension_1.response_message_txt,
claim_extension_1.authorization_nbr_id,
claim_extension_1.tax_exempt_cd,
claim_extension_1.sales_tax_attributed_amt,
claim_extension_1.accumulated_deductible_amt,
claim_extension_1.remaining_deductible_amt,
claim_extension_1.remaining_benefit_amt,
claim_extension_1.periodic_deductible_applied_amt,
claim_extension_1.exceeding_period_benefit_maximum_amt,
claim_extension_1.flat_sales_tax_submitted_amt,
claim_extension_1.flat_sales_tax_paid_amt,
claim_extension_1.flat_sales_tax_calculation_basis_cd,
claim_extension_1.alternate_pricing_flat_sales_tax_paid_amt,
claim_extension_1.alternate_pricing_percentage_tax_paid_amt,
claim_extension_1.alternate_pricing_percentage_tax_paid_rt,
claim_extension_1.alternate_pricing_percentage_tax_basis_cd,
claim_extension_1.customer_calc_flat_sales_tax_paid_amt,
claim_extension_1.customer_calc_percentage_tax_paid_amt,
claim_extension_1.customer_calc_percentage_tax_paid_rt,
claim_extension_1.customer_calc_percentage_tax_basis_cd,
claim_extension_1.percent_sales_tax_paid_amt AS percentage_sales_tax_paid_amt,
claim_extension_1.percent_sales_tax_paid_basis_cd AS percentage_sales_tax_paid_basis_cd,
claim_extension_1.percent_sales_tax_calculation_basis_cd,
claim_extension_1.percent_sales_tax_rt,
claim_extension_1.intended_ingredient_cost_amt,
claim_extension_1.intended_dispensing_fee_amt,
claim_extension_1.intended_flat_sales_tax_amt,
claim_extension_1.intended_percentage_sales_tax_rt,
claim_extension_1.intended_percentage_sales_tax_amt,
claim_extension_1.intended_percentage_tax_basis_cd,
claim_extension_1.prescribed_qty,
claim_extension_1.other_coverage_cd,
claim_extension_1.original_prescribed_qty,
claim_extension_1.product_unit_of_measure_cd,
claim_extension_1.original_prescribed_prod_id_type_cd,
```

```

claim_extension_1.generic_equivalent_prod_id_type_cd,
claim_extension_1.delay_reason_cd,
claim_extension_1.pharmacy_service_type_cd,
claim_extension_1.apply_spread_partial_fill_copay_cd,
claim_extension_1.spread_partial_fill_dispense_fee_amt,
claim_extension_1.preventative_drug_edit_performed_cd,
claim_extension_1.submitted_member_first_nm,
claim_extension_1.submitted_member_last_nm,
claim_extension_1.drug_pricing_source_cd,
CASE WHEN claim_extension_1.drug_pricing_source_cd = '00'
      THEN ''
      ELSE COALESCE(claim_extension_1.drug_pricing_source_cd, '')
END AS client_side_price_source_cd,
claim_extension_1.ndc_source_cd,
claim_extension_1.base_price_source_substitute_cd,
claim_extension_1.penalty_amt,
claim_extension_1.incentive_amt,
claim_extension_1.allowance_applies_cd AS allowance_applied_cd,
claim_extension_1.copay_value_id,
claim_extension_1.dispensing_limit_maintenance_cd,
claim_extension_1.cob_msp_patient_pay_limit_amt,
claim_extension_1.negotiated_patient_pay_amt,
claim_extension_1.opap_like_parameter_maximum_total_amt,
claim_extension_1.submitted_other_payer_total_amt,
claim_extension_1.oral_chemotherapy_parity_cd,
claim_extension_1.drug_discount_340b_program_ind,
claim_extension_1.ninety_day_retail_cd,
claim_extension_1.pricing_factor_rt_1,
claim_extension_1.pricing_factor_rt_2,
claim_extension_1.pricing_list_nm_2,
claim_extension_1.mac_unit_price_1_amt AS mac_unit_price_amt_1,
claim_extension_1.mac_unit_price_2_amt AS mac_unit_price_amt_2,
claim_extension_1.submitted_pregnancy_cd,
claim_extension_1.gcn_sequence_nbr,
claim_extension_1.enhanced_therapeutic_class_cd AS enhanced_therapy_class_cd,
claim_extension_1.dosage_form_cd,
claim_extension_1.route_administration_cd,
y04.route_desc_tx AS route_administration_description_txt,
claim_extension_1.rxcui_id,
claim_extension_1.billing_customer_id,
claim_extension_1.reporting_customer_id,
claim_extension_1.pharmacy_audit_discrepancy_cd,
claim_extension_1.pharmacy_audit_penalty_amt,
claim_extension_1.dur_age_safety_edit_id,
claim_extension_1.dur_minimum_maximum_safety_edit_id,
claim_extension_1.dur_drug_interaction_edit_id,
claim_extension_1.dur_duplicate_therapy_safety_edit_id,
claim_extension_1.dur_gender_safety_edit_id,
claim_extension_1.pharmacy_network_type_cd,
claim_extension_1.client_pricing_mac_reimbursement_comptn_id,
claim_extension_1.rate_sheet_id AS ratesheet_id,
claim_extension_1.network_suite_id,
claim_extension_1.client_pricing_rate_sheet_id AS client_pricing_ratesheet_id,
claim_extension_1.contract_unit_price_amt,
claim_extension_1.contract_ingredient_amt,
claim_extension_1.vaccine_paid_amt,
claim_extension_1.rate_sheet_mic_applies_type_cd AS ratesheet_mic_applied_type_cd,
claim_extension_1.rate_sheet_generic_drug_incentive_fee_amt AS ratesheet_generic_drug_incentive_fee_amt,
claim_extension_1.rate_sheet_dispense_fee_amt AS ratesheet_dispense_fee_amt,
claim_extension_1.rate_sheet_vaccine_fee_amt AS ratesheet_vaccine_fee_amt,
claim_extension_1.rate_sheet_delivery_fee_amt AS ratesheet_delivery_fee_amt,
claim_extension_1.rate_sheet_shipping_fee_amt AS ratesheet_shipping_fee_amt,
claim_extension_1.rate_sheet_incentive_fee_amt AS ratesheet_incentive_fee_amt,
claim_extension_1.rate_sheet_postage_fee_amt AS ratesheet_postage_fee_amt,
claim_extension_1.rate_sheet_mic_fee_amt AS ratesheet_mic_fee_amt,
claim_extension_1.client_pricing_ingredient_amt,
claim_extension_1.client_pricing_dispense_fee_amt,
claim_extension_1.client_pricing_copay_amt,
claim_extension_1.drug_pricing_source_id,
claim_extension_1.client_pricing_sales_tax_amt,
claim_extension_1.client_pricing_flat_sales_tax_pd_amt AS client_pricing_flat_sales_tax_paid_amt,

```

```
claim_extension_1.client_pricing_percentage_tax_paid_amt,
claim_extension_1.client_pricing_percentage_tax_paid_rt,
claim_extension_1.client_pricing_percentage_tax_basis_cd,
claim_extension_1.rate_sheet_type_id AS ratesheet_type_id,
claim_extension_1.rate_sheet_mic_loe_cd AS ratesheet_mic_loe_cd,
claim_extension_1.rate_sheet_administration_fee_amt AS ratesheet_administration_fee_amt,
claim_extension_1.rate_sheet_compound_prep_fee_amt AS ratesheet_compound_prep_fee_amt,
claim_extension_1.client_pricing_vaccine_fee_amt,
claim_extension_1.client_pricing_incentive_fee_amt,
claim_extension_1.client_pricing_generic_incentive_fee_amt,
claim_extension_1.client_pricing_delivery_fee_amt,
claim_extension_1.client_pricing_shipping_fee_amt,
claim_extension_1.client_pricing_postage_fee_amt,
claim_extension_1.client_pricing_administration_fee_amt,
claim_extension_1.client_pricing_compound_prep_fee_amt,
claim_extension_1.mpd_calculation_generic_price_amt,
claim_extension_1.customer_defined_plan_id,
claim_extension_1.prescriber_information_level_cd,
claim_extension_1.payment_center_id,
claim_extension_1.payment_center_record_type_cd,
claim_extension_1.payment_center_customer_set_id,
claim_extension_1.relationship_id,
claim_extension_1.relationship_record_type_cd,
claim_extension_1.relationship_customer_set_id,
claim_extension_1.dur_medication_safety_edit_id,
claim_extension_1.mic_total_paid_difference_amt,
claim_extension_1.pharmacy_mac_price_id,
claim_extension_1.client_pricing_mac_price_id,
claim_extension_1.financial_suite_id,
claim_extension_1.financial_cycle_cd,
claim_extension_1.financial_payment_group_id,
claim_extension_1.client_pricing_package_nm,
claim_extension_1.dur_protocol_id,
claim_extension_1.dur_claim_age_conflict_cd,
claim_extension_1.dur_claim_gender_conflict_cd,
claim_extension_1.dur_claim_medication_conflict_cd,
claim_extension_1.allow_pharmacy_over_payment_recovery_cd,
claim_extension_1.anda_status_cd,
claim_extension_1.orange_book_available_cd,
claim_extension_1.generic_available_cd,
claim_extension_1.fda_marketing_effective_end_dt,
claim_extension_1.fda_listed_cd,
claim_extension_1.medispan_awp_price_amt,
claim_extension_1.mbi_id,
claim_extension_1.patient_street_address_line_1 AS patient_street_address_line_txt_1,
claim_extension_1.patient_street_address_line_2 AS patient_street_address_line_txt_2,
claim_extension_1.patient_country_cd,
claim_extension_1.species,
claim_extension_1.submitted_customer_location_cd,
claim_extension_1.submitted_other_coverage_cd,
claim_extension_1.submitted_eligibility_clarification_cd,
claim_extension_1.submitted_service_level_cd,
claim_extension_1.submitted_prescription_clarification_cd,
claim_extension_1.submitted_unit dosage_cd,
claim_extension_1.submitted_postage_amt,
claim_extension_1.submitted_other_payer_amt AS submitted_other_payer_amt,
claim_extension_1.submitted_other_payer_dt,
claim_extension_1.submitted_cardholder_first_nm,
claim_extension_1.submitted_cardholder_last_nm,
claim_extension_1.submitted_patient_first_nm,
claim_extension_1.submitted_patient_last_nm,
claim_extension_1.submitted_patient_street_address,
claim_extension_1.submitted_patient_city,
claim_extension_1.submitted_patient_state,
claim_extension_1.submitted_patient_zip_cd,
claim_extension_1.software_vendor_certification_id,
claim_extension_1.clinical_product_id,
claim_extension_1.submitted_patient_phone_nbr,
claim_extension_1.benefit_stage_ind_1,
claim_extension_1.benefit_stage_ind_2,
claim_extension_1.benefit_stage_ind_3,
```

```

claim_extension_1.benefit_stage_ind_4,
claim_extension_1.dispense_event_id,
claim_extension_1.dispense_seq_id,
claim_extension_2.external_transaction_tracking_id,
claim_extension_2.renewal_reference_dt,
claim_extension_2.accumulator_medication_strength_qty,
claim_extension_2.member_contract_pbp_id,
claim_extension_2.contract_pbp_effective_begin_dt,
claim_extension_2.contract_pbp_process_dt,
claim_extension_2.contract_pbp_effective_end_dt,
claim_extension_2.cms_contract_id_nbr,
claim_extension_2.prescriber_id AS internal_prescriber_id,
claim_extension_2.medispan_awp_unit_cost_amt,
claim_extension_2.internal_specialty_ind,
claim_extension_2.rebate_drug_list_id,
claim_extension_2.rebate_drug_list_entry_id,
claim_extension_2.submitted_scheduled_prescription_id,
claim_extension_2.unique_person_id,
claim_extension_2.rate_sheet_association_typ_cd AS ratesheet_association_typ_cd,
claim_extension_2.rate_sheet_association_custom_grp_shrt_nm AS ratesheet_association_custom_grp_shrt_nm,
claim_extension_2.rate_sheet_association_relationship_id AS ratesheet_association_relationship_id,
claim_extension_2.rate_list_id,
claim_extension_2.covid_claim_ind,
claim_extension_2.covid_incentive_submitted_amt,
claim_extension_2.covid_incentive_paid_amt,
claim_extension_2.transition_letter_max_qty AS transition_letter_maximum_qty,
claim_extension_2.nadac_preferred_amt,
claim_extension_2.nadac_brand_unit_price,
claim_extension_2.nadac_generic_unit_price,
claim_extension_2.acaful_unit_price,
claim_extension_2.basis_of_reimbursement_cd,
claim_extension_2.dmr_prescriber_tax_id,
claim_extension_2.adjudicated_program_typ_cd AS adjudicated_program_type_cd,
claim_extension_2.msp_recovered_amt,
claim_extension_2.patient_email_address,
claim_extension_2.response_question_amt,
claim_extension_2.authorizedRepresentativeFirstNm,
claim_extension_2.authorizedRepresentativeLastNm,
claim_extension_2.assignedPreauthorizationId,
claim_extension_2.professional_paid_service_fee_amt AS paid_professional_service_fee_amt,
claim_extension_2.multiplePrescriptionGroupId,
claim_extension_2.multiplePrescriptionGroupReasonCd,
claim_extension_2.totalPrescribedRemainingQty,
claim_extension_2.preparationEnvironmentTyp,
claim_extension_2.preparationEnvironmentEventCd,
claim_extension_2.associatedPrescriptionRefNbrQlfr,
claim_extension_2.associatedPrescriptionRefFillNbr,
claim_extension_2.originalManufacturerProductIdQlfr,
claim_extension_2.originalManufacturerProductId,
claim_extension_2.ltpacDispenseFrequency,
claim_extension_2.ltpacBillingMethodology,
claim_extension_2.ltpacDispensingEventsNbr,
claim_extension_2.doNotDispenseBeforeDt,
claim_extension_2.providerFirstNm,
claim_extension_2.providerLastNm,
claim_extension_2.prescriberPhoneNbrExt,
claim_extension_2.prescriberStreetAddressLine1 AS prescriberStreetAddressLineTxt1,
claim_extension_2.prescriberStreetAddressLine2 AS prescriberStreetAddressLineTxt2,
TRIM(BOTH FROM al0.prescriberAddressCtyNm) AS prescriberAddressCityNm,
(SELECT b22.governmentUnitPostalCd
   FROM global.b22
 WHERE b22.governmentUnitId = al0.governmentUnitId) AS prescriberStateCd,
(al0.zipDeliveryOfficeCd || '-'::TEXT) || TRIM(BOTH FROM TO_CHAR(al0.presbAdrZipCodeSuffix,
'0000'::TEXT)) AS prescriberZipCd,
claim_extension_2.prescriberCountryCd,
claim_extension_2.prescriberDeaNbr,
claim_extension_2.prescriberPlaceOfService,
claim_extension_2.compoundLevelOfComplexity,
claim_extension_2.reconciliationId,
claim_extension_2.adjudicatedProgramType,
claim_extension_2.nextAvailableFillDt,

```

```

claim_extension_2.plan_benefit_override_ind,
claim_extension_2.plan_benefit_override_value_cnt,
claim_extension_2.plan_benefit_override_value,
claim_extension_2.maximum_age_qlfr,
claim_extension_2.maximum_age,
claim_extension_2.minimum_age_qlfr,
claim_extension_2.minimum_age,
claim_extension_2.minimum_amt,
claim_extension_2.maximum_amt_qlfr,
claim_extension_2.maximum_amt,
claim_extension_2.maximum_amt,
claim_extension_2.maximum_amt_time_period,
claim_extension_2.maximum_amt_time_period_effective_begin_dt,
claim_extension_2.maximum_amt_time_period_effective_end_dt,
claim_extension_2.maximum_amt_time_period_units,
claim_extension_2.remaining_amt_qlfr,
claim_extension_2.remaining_amt,
claim_extension_2.subrogation_requestors_reconciliation_id,
claim_extension_2.regulatory_fee_cnt,
claim_extension_2.regulatory_fee_typ_cd AS regulatory_fee_type_cd,
claim_extension_2.regulatory_fee_exempt_ind,
claim_extension_2.patient_regulatory_fee_amt,
claim_extension_2.plan_type,
claim_extension_2.disability_effective_end_dt,
claim_extension_2.esrd_ind,
claim_extension_2.esrd_effective_begin_dt,
claim_extension_2.esrd_effective_end_dt,
claim_extension_2.hospice_effective_begin_dt,
claim_extension_2.hospice_effective_end_dt,
claim_extension_2.hospice_facility_nbr,
claim_extension_2.hospice_facility_nm,
claim_extension_2.hospice_phone_nbr,
claim_extension_2.institutional_ind,
claim_extension_2.institutional_effective_end_dt,
claim_extension_2.other_benefit_cnt,
claim_extension_2.other_benefit_typ_cd AS other_benefit_type_cd,
claim_extension_2.other_benefit_effective_begin_dt,
claim_extension_2.other_benefit_effective_end_dt,
claim_extension_2.other_benefit_state_cd,
claim_extension_2.other_benefit_typ_id AS other_benefit_type_id,
claim_extension_2.other_benefit_facility_nm,
claim_extension_2.other_benefit_facility_phone_nbr,
claim_extension_2.other_benefit_detail_information_cnt,
claim_extension_2.other_benefit_detail_information_ind,
claim_extension_2.other_benefit_detail_info_eff_begin_dt AS other_benefit_detail_info_effv_begin_dt,
claim_extension_2.other_benefit_detail_info_eff_end_dt AS other_benefit_detail_info_effv_end_dt,
claim_extension_2.other_benefit_detail_info_prescriber_nbr,
claim_extension_2.other_benefit_detail_info_facility_nm,
claim_extension_2.other_benefit_dtl_info_facility_phone_nbr,
claim_extension_2.invalid_prescriber_data_source,
claim_extension_2.invalid_prescriber_data_source_state_cd,
claim_extension_2.covered_plan_paid_amt,
claim_extension_2.noncovered_plan_paid_amt,
claim_extension_2.special_packaging_ind,
claim_extension_2.other_payer_health_plan_id_qlfr,
claim_extension_2.other_payer_health_plan_id,
claim_extension_2.transaction_reference_nbr,
claim_core.plan_ingredient_cost_amt + claim_core.plan_dispense_fee_amt +
claim_core.incentive_paid_amt +
claim_extension_1.generic_incentive_fee_amt +
((SELECT COALESCE(SUM(claim_other_amount_paid,
claim_other_amount_paid_amt), 0.00) AS "coalesce"
FROM claimsprocess.claim_other_amount_paid
WHERE claim_other_amount_paid.customer_id = claim_core.
customer_id
AND claim_other_amount_paid.claim_id::TEXT = claim_core.
claim_id::TEXT)) +
claim_extension_1.flat_sales_tax_paid_amt +
claim_extension_1.percent_sales_tax_paid_amt AS
plan_total_cost_amt,
claim_core.customer_calculated_sales_tax_amt + claim_core.customer_calculated_ingredient_cost_amt +

```

```

customer_calculated_total_amt,
claim_core.customer_calculated_dispense_fee_amt AS
customer_calculated_plan_paid_amt,
CASE WHEN (claim_core.customer_calculated_ingredient_cost_amt + claim_core.customer_calculated_dispense_fee_amt +
claim_core.customer_calculated_sales_tax_amt - claim_core.customer_calculated_copay_amt) < 0
THEN 0.00
ELSE (claim_core.customer_calculated_ingredient_cost_amt + claim_core.customer_calculated_dispense_fee_amt +
claim_core.customer_calculated_sales_tax_amt - claim_core.customer_calculated_copay_amt)
END AS customer_member_reimbursement_amt,
claim_submission_clarification_codes_1.submission_clarification_cd AS submission_clarification_cd_1,
(SELECT r60.submission_clrfn_lng_nm
FROM global.r60
WHERE r60.submission_clarification_cd = claim_submission_clarification_codes_1.
submission_clarification_cd
LIMIT 1) AS submission_clarification_long_nm_1,
claim_submission_clarification_codes_2.submission_clarification_cd AS submission_clarification_cd_2,
(SELECT r60.submission_clrfn_lng_nm
FROM global.r60
WHERE r60.submission_clarification_cd = claim_submission_clarification_codes_2.
submission_clarification_cd
LIMIT 1) AS submission_clarification_long_nm_2,
claim_submission_clarification_codes_3.submission_clarification_cd AS submission_clarification_cd_3,
(SELECT r60.submission_clrfn_lng_nm
FROM global.r60
WHERE r60.submission_clarification_cd = claim_submission_clarification_codes_3.
submission_clarification_cd
LIMIT 1) AS submission_clarification_long_nm_3,
claim_error.claim_error_code_deny_ignore_print_set,
claim_additional_message.claim_additional_message_txt_set,
claim_client_pricing_unit.client_pricing_unit_id_set,
claim_client_pricing_unit.client_pricing_mac_component_id_set,
claim_client_pricing_unit.client_pricing_mac_price_id_set,
claim_client_pricing_unit.client_pricing_unit_amt_set,
claim_client_pricing_unit.client_pricing_unit_sheet_nm_set,
claim_client_pricing_unit.discount_rt_set,
claim_client_pricing_unit.dispense_fee_amt_set,
claim_client_pricing_unit.ingredient_cost_amt_set,
claim_client_pricing_unit.patient_pay_amt_set,
claim_client_pricing_unit.pricing_source_cd_set,
claim_client_pricing_unit.rate_sheet_sheet_nm_set,
claim_client_pricing_unit.sales_tax_amt_set,
claim_client_pricing_unit.total_cost_amt_set,
claim_client_pricing_unit.total_paid_amt_set,
claim_cob_other_payer.claim_other_payer_cardholder_id_set,
claim_cob_other_payer.claim_other_payer_group_id_set,
claim_cob_other_payer.claim_other_payer_id_set,
claim_cob_other_payer.claim_other_payer_person_cd_set,
claim_compound_product.compound_product_id_type_cd_set,
claim_compound_product.ingredient_cost_basis_cd_set,
claim_compound_product.ingredient_id_set,
claim_compound_product.ingredient_quantity_cnt_set,
claim_compound_product.plan_ingredient_cost_amt_set,
ncpdp_error_codes.ncpdp_error_cd_set,
claim_customer_reporting_indicators.customer_generic_reporting_ind_position_set,
claim_customer_reporting_indicators.customer_generic_reporting_ind_set,
claim_customer_reporting_indicators.customer_generic_defined_field_short_nm_set,
claim_customer_reporting_indicators.customer_generic_defined_field_long_nm_set,
claim_global_reporting_indicators.global_generic_reporting_ind_position_set,
claim_global_reporting_indicators.global_generic_reporting_ind_set,
claim_global_reporting_indicators.global_generic_defined_field_short_nm_set,
claim_global_reporting_indicators.global_generic_defined_field_long_nm_set,
claim_core.pharmacy_plan_id,
claim_core.dur_co_agent_qualifier_1,
claim_core.dur_co_agent_qualifier_2,
claim_core.dur_co_agent_qualifier_3,
claim_core.dur_co_agent_qualifier_4,
claim_core.dur_co_agent_qualifier_5,
claim_core.dur_co_agent_qualifier_6

```

```

claim_core.dur_co_agent_qualifier_7,
claim_core.dur_co_agent_qualifier_8,
claim_core.dur_co_agent_qualifier_9,
claim_core.dur_co_agent_id_1,
claim_core.dur_co_agent_id_2,
claim_core.dur_co_agent_id_3,
claim_core.dur_co_agent_id_4,
claim_core.dur_co_agent_id_5,
claim_core.dur_co_agent_id_6,
claim_core.dur_co_agent_id_7,
claim_core.dur_co_agent_id_8,
claim_core.dur_co_agent_id_9,
(SELECT u78.network_suite_sht_nm
     FROM customer.u78 u78
    JOIN customer.u91 u91 ON u91.network_suite_id = u78.network_suite_id
                           AND claim_core.fill_dt >= u91.effective_dt
                           AND claim_core.fill_dt <= u91.termination_dt
   WHERE u78.customer_set_id = e21.customer_set_id
         AND claim_core.customer_id::NUMERIC = e21.customer_id
         AND u78.pharmacy_network_id = claim_core.network_id::NUMERIC
  ORDER BY u91.last_mnt_ts DESC
 LIMIT 1) AS network_suite_short_nm,
(SELECT u78.network_suite_status_cd
     FROM customer.u78 u78
    JOIN customer.u91 u91 ON u91.network_suite_id = u78.network_suite_id
                           AND claim_core.fill_dt >= u91.effective_dt
                           AND claim_core.fill_dt <= u91.termination_dt
   WHERE u78.customer_set_id = e21.customer_set_id
         AND claim_core.customer_id::NUMERIC = e21.customer_id
         AND u78.pharmacy_network_id = claim_core.network_id::NUMERIC
  ORDER BY u91.last_mnt_ts DESC
 LIMIT 1) AS network_suite_status_cd,
claim_core.ect_batch_nbr,
COALESCE(s50c.pharmacy_nm, s50m.pharmacy_nm, s50a.pharmacy_nm) AS pharmacy_nm,
COALESCE(s50c.address_line_1_tx, s50m.address_line_1_tx, s50a.address_line_1_tx) AS
pharmacy_address_line_txt_1,
COALESCE(s50c.address_line_2_tx, s50m.address_line_2_tx, s50a.address_line_2_tx) AS
pharmacy_address_line_txt_2,
COALESCE(s50c.city_nm, s50m.city_nm, s50a.city_nm) AS pharmacy_city_nm,
COALESCE(s50c.postal_st_cd, s50m.postal_st_cd, s50a.postal_st_cd) AS pharmacy_state_cd,
COALESCE(s50c.zip_cd, s50m.zip_cd, s50a.zip_cd) AS pharmacy_zip_cd,
COALESCE(s50c.phone_nbr, s50m.phone_nbr, s50a.phone_nbr) AS pharmacy_phone_nbr,
COALESCE(s50c.county_cd, s50m.county_cd, s50a.county_cd) AS pharmacy_county_cd,
claim_fee_extension.transaction_type_cd_set,
claim_fee_extension.claim_fee_extension_owner_cd_set,
claim_fee_extension.adjust_fee_cd_set,
claim_fee_extension.fee_amt_set,
1 AS rx_count,
u92.pharmacy_component_id,
u92.pharmacy_component_sht_nm AS pharmacy_component_short_nm,
u92.pharmacy_component_lng_nm AS pharmacy_component_long_nm,
bq8.mac_price_component_id,
bq8.mac_price_cmpont_sht_nm AS mac_price_component_short_nm,
bq8.mac_price_cmpont_lng_nm AS mac_price_component_long_nm,
u80.ratesheet_short_nm AS rate_sheet_short_nm,
u80.ratesheet_long_nm AS rate_sheet_long_nm,
u90.rate_sheet_type_lng_nm AS rate_sheet_type_long_nm,
t08.pricing_component_sht_nm AS pricing_component_short_nm,
CASE WHEN claim_extension_2.rate_sheet_quality_network_ind = 'Y' OR claim_extension_2.
network_quality_network_ind = 'Y'
      THEN 'Y'
      ELSE 'N'
END AS quality_network_eligible_ind,
CASE WHEN claim_core.record_source_ind = 'A' AND claim_core.medicaid_agency_id > ''
      THEN 'Y'
      ELSE 'N'
END AS medicaid_subrogation_ind,
CASE WHEN claim_extension_2.rate_sheet_check_suppress_ind = 'Y' OR claim_extension_2.
network_check_suppress_ind = 'Y'
      THEN 'Y'
      ELSE 'N'

```

```

        END AS check_suppression_ind,
CASE WHEN (claim_extension_1.version_release_id = 'D0')
      THEN claim_core.total_paid_amt + claim_core.plan_calculated_copay_plus_coins_amt +
claim_extension_1.other_payer_recognized_amt
      ELSE claim_core.total_paid_amt + claim_core.plan_calculated_copay_plus_coins_amt + claim_core.
member_deductible_cost_amt
      END AS claim_total_cost_amt,
      claim_core.pca_rollover_amt,
      groupsl_bnft_date.deductible_plan_id,
      COALESCE(s50c.contact_first_nm, s50m.contact_first_nm, s50a.contact_first_nm) AS
pharmacy_contact_first_nm,
      COALESCE(s50c.contact_last_nm, s50m.contact_last_nm, s50a.contact_last_nm) AS pharmacy_contact_last_nm,
(SELECT h92.ytd_drg_spnd_amt
     FROM customer.ded_pde07_clmdtl h92
    WHERE h92.cust_id = claim_core.customer_id
      AND h92.clm_id = claim_core.claim_id
    ORDER BY CASE WHEN h92.pln_typ_cde = 'DD1'
                  THEN 1
                  WHEN h92.pln_typ_cde = 'IC1'
                  THEN 2
                  WHEN h92.pln_typ_cde = 'CG1'
                  THEN 3
                  WHEN h92.pln_typ_cde = 'CAT'
                  THEN 4
                  ELSE 5
                END DESC
LIMIT 1) AS total_covered_drug_cost_amt,
(SELECT h92.ytd_troop_amt
     FROM customer.ded_pde07_clmdtl h92
    WHERE h92.cust_id = claim_core.customer_id
      AND h92.clm_id = claim_core.claim_id
    ORDER BY CASE WHEN h92.pln_typ_cde = 'DD1'
                  THEN 1
                  WHEN h92.pln_typ_cde = 'IC1'
                  THEN 2
                  WHEN h92.pln_typ_cde = 'CG1'
                  THEN 3
                  WHEN h92.pln_typ_cde = 'CAT'
                  THEN 4
                  ELSE 5
                END DESC
LIMIT 1) AS ytd_accumulated_troop_amt,
ded_pde07_clm.gdca_amt AS gross_drug_cost_above_amt,
ded_pde07_clm.gdcb_amt AS gross_drug_cost_below_amt
--      claim_extension_2.custom_group_id,                                -- claim_extension_2.custom_group_id NOT YET
AVAILABLE IN DOMANI ODS
--      u85.custom_group_long_nm,                                     -- claim_extension_2.custom_group_id NOT YET
AVAILABLE IN DOMANI ODS
--      claim_extension_2.preferred_ind,                                -- claim_extension_2.preferred_ind NOT YET
AVAILABLE IN DOMANI ODS
--      claim_extension_2.partd_m3p_associated_claim_id,          -- To be added once available in UAT/HUAT/Prod
--      claim_extension_2.partd_m3p_associated_claim_id,          -- To be added once available in UAT/HUAT/Prod
FROM claimsprocess.claim_core
JOIN customer.e21 ON e21.customer_id = claim_core.customer_id
      AND e21.customer_set_type_id = 2
JOIN customer.e21 e21b ON e21b.customer_id = claim_core.customer_id
      AND e21b.customer_set_type_id = 3
LEFT JOIN claimsprocess.a_claim_core ON a_claim_core.customer_id = claim_core.customer_id
      AND a_claim_core.claim_id = claim_core.claim_id
      AND a_claim_core.process_adjust_created_ts = (SELECT MAX(acc.
process_adjust_created_ts)
FROM claimsprocess.
a_claim_core acc
WHERE acc.customer_id =
claim_core.customer_id
      AND acc.claim_id =
claim_core.claim_id)
LEFT JOIN claimsprocess.claim_extension_1 ON claim_extension_1.customer_id = claim_core.customer_id
      AND claim_extension_1.claim_id = claim_core.claim_id
LEFT JOIN claimsprocess.claim_extension_2 ON claim_extension_2.customer_id = claim_core.customer_id
      AND claim_extension_2.claim_id = claim_core.claim_id

```

```

LEFT JOIN customer.customr0 ON customr0.customer_no = claim_core.customer_id
LEFT JOIN customer.clients0 ON clients0.customer_no = claim_core.customer_id
    AND clients0.client_no = claim_core.client_id
LEFT JOIN customer.groups1 ON groups1.customer_no = claim_core.customer_id
    AND groups1.client_no = claim_core.client_id
    AND groups1.group_no = claim_core.group_id
LEFT JOIN customer.o58 ON o58.cust_id = claim_core.customer_id
    AND o58.clnt_id = claim_core.client_id
    AND o58.grp_intl_id = claim_core.group_id
    AND o58.member_id = claim_core.member_id
LEFT JOIN claimsprocess.claim_vendor_drug ON claim_vendor_drug.customer_id = claim_core.customer_id
    AND claim_vendor_drug.claim_id = claim_core.claim_id
LEFT JOIN claimsprocess.claim_medicare_partd ON claim_medicare_partd.customer_id = claim_core.customer_id
    AND claim_medicare_partd.claim_id = claim_core.claim_id
LEFT JOIN misc.claim_component_column_information_vw ON claim_component_column_information_vw.customer_id =
claim_core.customer_id
    AND claim_component_column_information_vw.claim_id =
claim_core.claim_id
LEFT JOIN claimsprocess.claim_other_amount_paid m47_seq_1 ON m47_seq_1.customer_id = claim_core.customer_id
    AND m47_seq_1.claim_id = claim_core.claim_id
    AND m47_seq_1.claim_other_amount_paid_sequence_id = 1
LEFT JOIN claimsprocess.claim_other_amount_paid m47_seq_2 ON m47_seq_2.customer_id = claim_core.customer_id
    AND m47_seq_2.claim_id = claim_core.claim_id
    AND m47_seq_2.claim_other_amount_paid_sequence_id = 2
LEFT JOIN claimsprocess.claim_other_amount_paid m47_seq_3 ON m47_seq_3.customer_id = claim_core.customer_id
    AND m47_seq_3.claim_id = claim_core.claim_id
    AND m47_seq_3.claim_other_amount_paid_sequence_id = 3
LEFT JOIN (SELECT t1.cust_id,
    t1.ndc,
    t1.bgn_dte,
    t1.awp_disc_pctg,
    t1.end_dte,
    t1.prcs strt_tmst
    FROM customer.prcg_ndc_1 t1) e70 ON e70.cust_id = claim_core.customer_id
    AND e70.ndc = claim_core.ndc_id
    AND claim_core.fill_dt BETWEEN e70.bgn_dte AND e70.end_dte
    AND e70.prcs strt_tmst = (SELECT MAX(e70a.prcs strt_tmst)
        FROM customer.prcg_ndc_1 e70a
        WHERE e70a.cust_id = claim_core.
customer_id
    AND e70a.ndc = claim_core.ndc_id
    AND claim_core.fill_dt BETWEEN e70a.
bgn_dte AND e70a.end_dte)
LEFT JOIN (SELECT x.coverage_id,
    x.benefit_plan_type_id,
    x.effective_dt,
    x.termination_dt,
    p23.customer_id,
    p23.client_id,
    f69.customer_set_id,
    f69.coverage_1_cd,
    f69.coverage_2_cd,
    f69.coverage_description_tx
    FROM customer.f69 f69
    INNER JOIN customer.p32 x ON x.coverage_id = f69.coverage_id
    INNER JOIN customer.p23 p23 ON p23.coverage_id = x.coverage_id) AS p32 ON p32.customer_id =
claim_core.customer_id
    AND p32.client_id =
claim_core.client_id
    AND p32.customer_set_id =
e21b.customer_set_id
    AND p32.coverage_1_cd =
claim_core.coverage_cd_1
    AND p32.coverage_2_cd =
claim_core.coverage_cd_2
    AND claim_core.fill_dt
BETWEEN p32.effective_dt AND p32.termination_dt
LEFT JOIN global.f96 f96 ON f96.drug_pricing_source_cd = claim_core.price_source_id
LEFT JOIN customer.c76 c76 ON c76.customer_set_id = e21.customer_set_id
LEFT JOIN claimsprocess.claim_dmr_external ON claim_dmr_external.customer_id = claim_core.customer_id
    AND claim_dmr_external.claim_id = claim_core.claim_id

```

```

LEFT JOIN global.x80 x80 ON x80.ndc_id = claim_core.ndc_id
LEFT JOIN global.x92 x92 ON x92.ndc_id = claim_core.ndc_id
LEFT JOIN global.y00 y00 ON y00.ndc_id = claim_core.ndc_id
LEFT JOIN global.y11 y11 ON y11.ahfs_tc_cd = claim_core.ahfs_therapy_class
LEFT JOIN global.y32 y32 ON y32.evd_fdb_vocab_id = claim_core.ndc_id
    AND y32.evd_fdb_vocab_type_id = 100
    AND y32.evd_link_type_id = 4
LEFT JOIN global.y04 y04 ON y04.route_admin_cd = claim_extension_1.route_administration_cd
LEFT JOIN global.x85 x85 ON x85.gcn_seqno_id = x80.gcn_seqno_id
LEFT JOIN global.x90 x90 ON x90.labeler_id = x80.labeler_id
LEFT JOIN global.y02 y02 ON y02.drug_strength_desc_tx = x85.drug_strength_desc_tx
LEFT JOIN global.y03 y03 ON y03.dosage_form_cd = x85.dosage_form_cd
LEFT JOIN global.y06 y06 ON y06.ingredient_list_id = x85.ingredient_list_id
LEFT JOIN global.y10 y10 ON y10.specific_tc_cd = x85.specific_tc_cd
LEFT JOIN global.y12 y12 ON y12.generic_tc_id = x85.generic_tc_id
LEFT JOIN global.y13 y13 ON y13.gcn_seqno_id = x85.gcn_seqno_id
LEFT JOIN global.y16 y16 ON y16.gcn_seqno_id = x85.gcn_seqno_id
LEFT JOIN global.y37 y37 ON y37.standard_tc_cd = x85.standard_tc_cd
LEFT JOIN customer.al4 al4 ON al4.accum_component_id = claim_component_column_information_vw.
accumulator_component_id::NUMERIC
    LEFT JOIN claimsprocess.claim_submission_clarification_codes claim_submission_clarification_codes_1 ON
claim_submission_clarification_codes_1.customer_id = claim_core.customer_id
        AND
claim_submission_clarification_codes_1.claim_id = claim_core.claim_id
        AND
claim_submission_clarification_codes_1.claim_submission_clarification_sequence_id = 1
    LEFT JOIN claimsprocess.claim_submission_clarification_codes claim_submission_clarification_codes_2 ON
claim_submission_clarification_codes_2.customer_id = claim_core.customer_id
        AND
claim_submission_clarification_codes_2.claim_id = claim_core.claim_id
        AND
claim_submission_clarification_codes_2.claim_submission_clarification_sequence_id = 2
    LEFT JOIN claimsprocess.claim_submission_clarification_codes claim_submission_clarification_codes_3 ON
claim_submission_clarification_codes_3.customer_id = claim_core.customer_id
        AND
claim_submission_clarification_codes_3.claim_id = claim_core.claim_id
        AND
claim_submission_clarification_codes_3.claim_submission_clarification_sequence_id = 3
    LEFT JOIN LATERAL (SELECT string_agg((((claim_error_1.claim_error_cd::TEXT || claim_error_1.auto_deny_error::TEXT) || claim_error_1.auto_ignore_error::TEXT) || claim_error_1.auto_no_print_error::TEXT) || ' '::TEXT) || error_configuration.error_txt::TEXT, E'\n'
        ORDER BY claim_error_1.customer_id, claim_error_1.claim_id, claim_error_1.
claim_error_sequence_id) AS claim_error_code_deny_ignore_print_set,
        string_agg(error_configuration.error_reason_subcategory_medium_txt, E'\n'
        ORDER BY claim_error_1.customer_id, claim_error_1.claim_id, claim_error_1.
claim_error_sequence_id) AS claim_error_reason_subcategory_set
        FROM claimsprocess.claim_error claim_error_1
        LEFT JOIN claimsprocess.error_configuration ON error_configuration.
internal_error_cd = claim_error_1.claim_error_cd
            AND error_configuration.
error_abbreviation_txt = 'POS'
            WHERE claim_core.customer_id = claim_error_1.customer_id
                AND claim_core.claim_id = claim_error_1.claim_id) AS claim_error ON 1 = 1
    LEFT JOIN LATERAL (SELECT string_agg(claim_additional_message_1.claim_additional_message_txt::TEXT, E'\n'.
claim_id, claim_additional_message_1.claim_additional_message_information_seq_id) AS
claim_additional_message_txt_set
        FROM claimsprocess.claim_additional_message AS claim_additional_message_1
        WHERE claim_additional_message_1.customer_id = claim_core.customer_id
            AND claim_additional_message_1.claim_id::TEXT = claim_core.claim_id::TEXT) AS
claim_additional_message ON 1 = 1
    LEFT JOIN LATERAL (SELECT string_agg(claim_client_pricing_unit_1.client_pricing_unit_id::CHARACTER VARYING::TEXT, E'\n'
        ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS client_pricing_unit_id_set,
        string_agg(claim_client_pricing_unit_1.client_pricing_mac_component_id::CHARACTER VARYING::TEXT, E'\n'
        ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS client_pricing_mac_component_id_set,

```

```

        string_agg(claim_client_pricing_unit_1.client_pricing_mac_price_id::CHARACTER
VARYING::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS client_pricing_mac_price_id_set,
        string_agg(claim_client_pricing_unit_1.client_pricing_unit_amnt::CHARACTER VARYING::
TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS client_pricing_unit_amnt_set,
        string_agg(claim_client_pricing_unit_1.client_pricing_unit_sheet_nm::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS client_pricing_unit_sheet_nm_set,
        string_agg(claim_client_pricing_unit_1.discount_rt::CHARACTER VARYING::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS discount_rt_set,
        string_agg(claim_client_pricing_unit_1.dispense_fee_amnt::CHARACTER VARYING::TEXT,
E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS dispense_fee_amnt_set,
        string_agg(claim_client_pricing_unit_1.ingredient_cost_amnt::CHARACTER VARYING::
TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS ingredient_cost_amnt_set,
        string_agg(claim_client_pricing_unit_1.patient_pay_amnt::CHARACTER VARYING::TEXT,
E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS patient_pay_amnt_set,
        string_agg(claim_client_pricing_unit_1.pricing_source_cd::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS pricing_source_cd_set,
        string_agg(claim_client_pricing_unit_1.rate_sheet_sheet_nm::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS rate_sheet_sheet_nm_set,
        string_agg(claim_client_pricing_unit_1.sales_tax_amnt::CHARACTER VARYING::TEXT, E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS sales_tax_amnt_set,
        string_agg(claim_client_pricing_unit_1.total_cost_amnt::CHARACTER VARYING::TEXT,
E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS total_cost_amnt_set,
        string_agg(claim_client_pricing_unit_1.total_paid_amnt::CHARACTER VARYING::TEXT,
E'\n'
            ORDER BY claim_client_pricing_unit_1.customer_id, claim_client_pricing_unit_1.
claim_id, claim_client_pricing_unit_1.client_pricing_unit_id, claim_client_pricing_unit_1.
client_pricing_unit_sequence_id) AS total_paid_amnt_set
    FROM claimsprocess.claim_client_pricing_unit AS claim_client_pricing_unit_1
    WHERE claim_client_pricing_unit_1.customer_id = claim_core.customer_id
        AND claim_client_pricing_unit_1.claim_id::TEXT = claim_core.claim_id::TEXT) AS
claim_client_pricing_unit ON 1 = 1
    LEFT JOIN LATERAL (SELECT string_agg(claim_cob_other_payer_1.claim_other_payer_cardholder_id::TEXT, E'\n'
        ORDER BY claim_cob_other_payer_1.customer_id, claim_cob_other_payer_1.claim_id,
claim_cob_other_payer_1.claim_other_payer_sequence_id) AS claim_other_payer_cardholder_id_set,
        string_agg(claim_cob_other_payer_1.claim_other_payer_group_id::TEXT, E'\n'
            ORDER BY claim_cob_other_payer_1.customer_id, claim_cob_other_payer_1.claim_id,
claim_cob_other_payer_1.claim_other_payer_sequence_id) AS claim_other_payer_group_id_set,
        string_agg(claim_cob_other_payer_1.claim_other_payer_id::TEXT, E'\n'
            ORDER BY claim_cob_other_payer_1.customer_id, claim_cob_other_payer_1.claim_id,
claim_cob_other_payer_1.claim_other_payer_sequence_id) AS claim_other_payer_id_set,
        string_agg(claim_cob_other_payer_1.claim_other_payer_person_cd::TEXT, E'\n'
            ORDER BY claim_cob_other_payer_1.customer_id, claim_cob_other_payer_1.claim_id,
claim_cob_other_payer_1.claim_other_payer_sequence_id) AS claim_other_payer_person_cd_set
    ) AS
claim_cob_other_payer_1

```



```

WHEN claim_fee_extension_1.claim_fee_extension_owner_cd = 'Z'
THEN claim_fee_extension_1.claim_fee_extension_owner_cd || ' - '
Domani_ddc_Fee'
ELSE claim_fee_extension_1.claim_fee_extension_owner_cd || ' - '
Invalid'
END, E'\n'
ORDER BY claim_fee_extension_1.customer_id, claim_fee_extension_1.claim_id,
claim_fee_extension_1.claim_fee_extension_sequence_id) AS claim_fee_extension_owner_cd_set,
string_agg(claim_fee_extension_1.adjust_fee_cd::TEXT, E'\n'
ORDER BY claim_fee_extension_1.customer_id, claim_fee_extension_1.claim_id,
claim_fee_extension_1.claim_fee_extension_sequence_id) AS adjust_fee_cd_set,
string_agg('$)::TEXT || claim_fee_extension_1.fee_amt::TEXT, E'\n'
ORDER BY claim_fee_extension_1.customer_id, claim_fee_extension_1.claim_id,
claim_fee_extension_1.claim_fee_extension_sequence_id) AS fee_amt_set
FROM claimsprocess.claim_fee_extension claim_fee_extension_1
WHERE claim_core.customer_id = claim_fee_extension_1.customer_id
AND claim_core.claim_id = claim_fee_extension_1.claim_id) claim_fee_extension ON l = 1
LEFT JOIN misc.prescriber_max_record_vw ak6 ON ak6.external_prescriber_id = claim_core.submitted_prescriber_id
AND ak6.prescriber_id_effective_dt <= claim_core.fill_dt
AND ak6.prescriber_id_term_dt >= claim_core.fill_dt
LEFT JOIN global.a10 a10 ON a10.prescriber_id = ak6.prescriber_id
AND ak6.vendor_source_cd = 'L'
AND a10.prescriber_id_type_id IS NULL
AND a10.prescriber_id_type_seq_id IS NULL
AND a10.presb_address_rank_cd = '1'
AND a10.prescriber_address_type_cd = 'B'
LEFT JOIN global.at8 at8 ON at8.prescriber_id = claim_extension_2.prescriber_id
LEFT JOIN global.au0 au0 ON au0.prescriber_specialty_id = at8.prescriber_specialty_id
LEFT JOIN finance.finance_claim fc ON fc.customer_id = claim_core.customer_id
AND fc.claim_id = claim_core.claim_id
AND fc.process_create_ts = (SELECT MAX(fc1.process_create_ts)
FROM finance.finance_claim fc1
WHERE fc1.customer_id = claim_core.customer_id
AND fc1.claim_id = claim_core.claim_id)
LEFT JOIN finance.outbound_payment op ON op.finance_claim_id = fc.finance_claim_id
AND op.cycle_id = fc.cycle_id
AND fc.process_create_ts = (SELECT MAX(op1.process_create_ts)
FROM finance.outbound_payment op1
WHERE op1.finance_claim_id = op.
finance_claim_id
AND op1.cycle_id = op.cycle_id)
LEFT JOIN finance.payment p ON p.payment_id = op.payment_id
LEFT JOIN finance.cycle c ON c.cycle_id = fc.cycle_id
LEFT JOIN customer.t08 t08 ON t08.pricing_component_id = claim_component_column_information_vw.
pricing_component_id::NUMERIC
LEFT JOIN customer.u92 u92 ON u92.pharmacy_component_id = claim_component_column_information_vw.
pharmacy_component_id::NUMERIC
LEFT JOIN customer.u80 u80.ratesheet_id = claim_extension_1.rate_sheet_id
LEFT JOIN global.u90 u90.rate_sheet_type_id = u80.rate_sheet_type_id
LEFT JOIN customer.bq8 bq8.mac_price_component_id = claim_extension_1.pharmacy_mac_price_id
LEFT JOIN LATERAL (SELECT string_agg(customer_indicators_1.customer_position_cnt::TEXT, E'\n'
ORDER BY customer_indicators_1.customer_id, customer_indicators_1.claim_id,
customer_indicators_1.customer_position_cnt) AS customer_generic_reporting_ind_position_set,
string_agg(customer_indicators_1.customer_generic_reporting_ind, E'\n'
ORDER BY customer_indicators_1.customer_id, customer_indicators_1.claim_id,
customer_indicators_1.customer_position_cnt) AS customer_generic_reporting_ind_set,
string_agg(COALESCE(gdf.generic_defined_field_short_nm, ' '), E'\n'
ORDER BY customer_indicators_1.customer_id, customer_indicators_1.claim_id,
customer_indicators_1.customer_position_cnt) AS customer_generic_defined_field_short_nm_set,
string_agg(COALESCE(gdf.generic_defined_field_long_nm, ' '), E'\n'
ORDER BY customer_indicators_1.customer_id, customer_indicators_1.claim_id,
customer_indicators_1.customer_position_cnt) AS customer_generic_defined_field_long_nm_set
FROM customer_indicators AS customer_indicators_1
LEFT JOIN claimsprocess.customer_generic_defined_field cgdf ON cgdf.customer_id =
customer_indicators_1.customer_id
AND cgdf.position_cnt =
customer_indicators_1.customer_position_cnt
LEFT JOIN claimsprocess.generic_defined_field gdf ON gdf.generic_defined_field_id = cgdf.
generic_defined_field_id
AND gdf.generic_defined_field_type_cd =
'C'
```

```

        WHERE customer_indicators_1.customer_id = claim_core.customer_id
              AND customer_indicators_1.claim_id = claim_core.claim_id) AS
claim_customer_reporting_indicators ON 1 = 1
    LEFT JOIN LATERAL (SELECT string_agg(global_indicators_1.global_position_cnt::TEXT, E'\n'
                                ORDER BY global_indicators_1.customer_id, global_indicators_1.claim_id,
global_indicators_1.global_position_cnt) AS global_generic_reporting_ind_position_set,
                                string_agg(global_indicators_1.global_generic_reporting_ind, E'\n'
                                ORDER BY global_indicators_1.customer_id, global_indicators_1.claim_id,
global_indicators_1.global_position_cnt) AS global_generic_reporting_ind_set,
                                string_agg(COALESCE(gdf.generic_defined_field_short_nm, ' '), E'\n'
                                ORDER BY global_indicators_1.customer_id, global_indicators_1.claim_id,
global_indicators_1.global_position_cnt) AS global_generic_defined_field_short_nm_set,
                                string_agg(COALESCE(gdf.generic_defined_field_long_nm, ' '), E'\n'
                                ORDER BY global_indicators_1.customer_id, global_indicators_1.claim_id,
global_indicators_1.global_position_cnt) AS global_generic_defined_field_long_nm_set
                                FROM global_indicators AS global_indicators_1
                                LEFT JOIN claimsprocess.generic_defined_field gdf ON gdf.position_cnt =
global_indicators_1.global_position_cnt
                                AND gdf.generic_defined_field_type_cd =
'G'
                                WHERE global_indicators_1.customer_id = claim_core.customer_id
                                      AND global_indicators_1.claim_id = claim_core.claim_id) AS
claim_global_reporting_indicators ON 1 = 1
    LEFT JOIN customer.groups1_bnft_date ON groups1_bnft_date.customer_no = groups1.customer_no
                                      AND groups1_bnft_date.client_no = groups1.client_no
                                      AND groups1_bnft_date.group_no = groups1.group_no
                                      AND groups1.bnft_date.array_index = 1
    LEFT JOIN misc.field_configuration_lookup mpddtc1 ON mpddtc1.field_configuration_type_cd =
claim_medicare_partd.partd_drug_type_cd
                                      AND mpddtc1.field_configuration_type_txt = 'Medicare Part-D
Drug Type Code'
    LEFT JOIN misc.field_configuration_lookup ndasic ON ndasic.field_configuration_type_cd::INTEGER = x92.
nda_status_cd
                                      AND ndasic.field_configuration_type_txt = 'New Drug
Application Status Indicator Code'
    LEFT JOIN misc.field_configuration_lookup ndasi ON ndasi.field_configuration_type_cd::INTEGER = x92.
nda_status_cd
                                      AND ndasi.field_configuration_type_txt = 'New Drug Application
Status Indicator'
    LEFT JOIN misc.field_configuration_lookup andasic ON andasic.field_configuration_type_cd::INTEGER = x92.
anda_status_cd
                                      AND andasic.field_configuration_type_txt = 'New Drug
Application Status Indicator Code'
    LEFT JOIN misc.field_configuration_lookup andasi ON andasi.field_configuration_type_cd::INTEGER = x92.
anda_status_cd
                                      AND andasi.field_configuration_type_txt = 'New Drug
Application Status Indicator'
    LEFT JOIN misc.field_configuration_lookup diic ON diic.field_configuration_type_cd = x80.innovator_ind_cd
                                      AND diic.field_configuration_type_txt = 'Drug Innovator
Indicator Code'
    LEFT JOIN misc.field_configuration_lookup dii ON dii.field_configuration_type_cd = x80.innovator_ind_cd
                                      AND dii.field_configuration_type_txt = 'Drug Innovator Indicator'
    LEFT JOIN misc.field_configuration_lookup dipic ON dipic.field_configuration_type_cd = x80.
instnl_product_ind_cd
                                      AND dipic.field_configuration_type_txt = 'Drug Institutional
Product Indicator Code'
    LEFT JOIN misc.field_configuration_lookup dbac ON dbac.field_configuration_type_cd = x80.bla_approved_cd
                                      AND dbac.field_configuration_type_txt = 'Drug BLA Approved Code'
    LEFT JOIN misc.field_configuration_lookup dcc ON dcc.field_configuration_type_cd = x80.drug_class_cd
                                      AND dcc.field_configuration_type_txt = 'Drug Class Code'
    LEFT JOIN misc.field_configuration_lookup dric ON dric.field_configuration_type_cd = x80.repackaged_ind_cd
                                      AND dric.field_configuration_type_txt = 'Drug Repackage
Indicator Code'
    LEFT JOIN misc.field_configuration_lookup csdc ON csdc.field_configuration_type_cd = claim_core.
cics_claim_status_cd
                                      AND csdc.field_configuration_type_txt = 'Claim Status Detail
Code'
    LEFT JOIN misc.field_configuration_lookup csc ON csc.field_configuration_type_cd::INTEGER = claim_core.
claim_status_cd
                                      AND csc.field_configuration_type_txt = 'Claim Status Code'
    LEFT JOIN misc.field_configuration_lookup cc ON cc.field_configuration_type_cd::INTEGER = claim_core.

```

```

compound_cd
    AND cc.field_configuration_type_txt = 'Compound Code'
LEFT JOIN misc.field_configuration_lookup gi ON gi.field_configuration_type_cd = claim_core.generic_ind
    AND gi.field_configuration_type_txt = 'Generic Indicator'
LEFT JOIN misc.field_configuration_lookup gpi ON gpi.field_configuration_type_cd = claim_core.
generic_product_ind
    AND gpi.field_configuration_type_txt = 'Generic Product
Indicator'
LEFT JOIN misc.field_configuration_lookup psi ON psi.field_configuration_type_cd = claim_core.price_source_ind
    AND psi.field_configuration_type_txt = 'Price Source Indicator'
LEFT JOIN misc.field_configuration_lookup sd ON sd.field_configuration_type_cd = claim_core.specialty_drug_ind
    AND sd.field_configuration_type_txt = 'Specialty Drug Indicator'
LEFT JOIN misc.field_configuration_lookup sdi ON sdi.field_configuration_type_cd = claim_core.
submitted_dispense_as_written_ind
    AND sdi.field_configuration_type_txt = 'Submitted DAW Indicator'
LEFT JOIN misc.field_configuration_lookup csrc ON csrc.field_configuration_type_cd = claim_extension_1.
cost_share_reduction_cd
    AND csrc.field_configuration_type_txt = 'Cost Share Reduction
Code'
LEFT JOIN misc.field_configuration_lookup dc ON dc.field_configuration_type_cd = claim_core.dea_cd
    AND dc.field_configuration_type_txt = 'DEA Code'
LEFT JOIN misc.field_configuration_lookup mpddtc2 ON mpddtc2.field_configuration_type_cd = x80.
part_d_drug_type_cd
    AND mpddtc2.field_configuration_type_txt = 'Medicare Part-D
Drug Type Code'
LEFT JOIN misc.field_configuration_lookup dfc ON dfc.field_configuration_type_cd = x80.drug_form_cd
    AND dfc.field_configuration_type_txt = 'Drug Form Code'
LEFT JOIN misc.field_configuration_lookup rsi ON rsi.field_configuration_type_cd = claim_core.
record_source_ind
    AND rsi.field_configuration_type_txt = 'Record Source Indicator'
LEFT JOIN misc.field_configuration_lookup cfi ON cfi.field_configuration_type_cd = claim_core.
customized_formulary_ind
    AND cfi.field_configuration_type_txt = 'Customized Formulary
Indicator'
LEFT JOIN misc.field_configuration_lookup ic ON ic.field_configuration_type_cd = claim_core.institutional_cd
    AND ic.field_configuration_type_txt = 'Institutional Code'
LEFT JOIN customer.ded_pde07_clm ON ded_pde07_clm.cust_id = claim_core.customer_id
    AND ded_pde07_clm.clm_id = claim_core.claim_id
-- LEFT JOIN customer.u85 ON u85.custom_group_id = claim_extension_2.custom_group_id -- claim_extension_2.
custom_group_id NOT YET AVAILABLE IN DOMANI ODS
)
;
COMMENT ON VIEW misc.claim_price_detail_vw IS 'Claim-level price data for all claim-related pricing/cost
information; including additional claim-related information.';

```