Switching Transformers/Importers From 2019 to 2021 QIP Value Sets

Programmers Meeting

November 1, 2021

New 2021 QIP Value Set

- This applies only to Transformers or Data Elements (AKA, Importers) that have a JOIN to the 2019 version of the QIP Value Sets
- Old table name in Relevant: relevant_qip_2019_value_set
- New table name in Relevant: hedis_value_set_codes

- The 2019 Value Set table will eventually be eliminated from the data scheme

Upgrade to New HEDIS Table

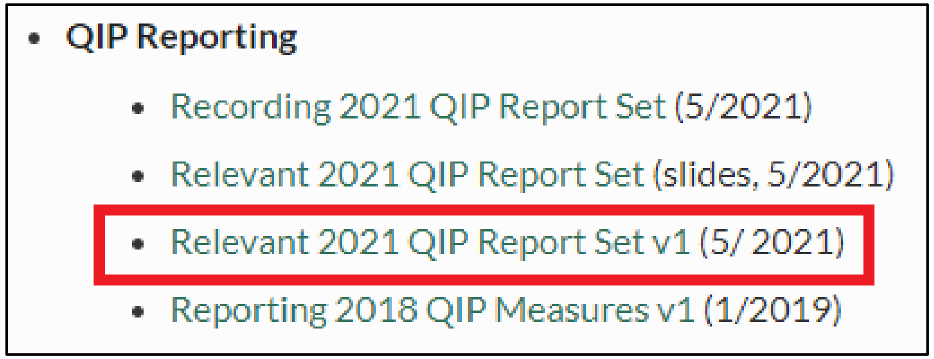
- New table name in Relevant: hedis_value_set_codes
- Similar structure and columns as the eCQM Value Set table
- Some columns have slightly different names (e.g., code vs code_value)
- New Value Sets in the future will be appended to this table
- In your SQL code, use the column latest = ‘TRUE’ to identify the most recent Value Set added. This is the same as the approach with the eCQM Value Set table
- The “switch” to the new table is a one-time procedure

eCQM Value Sets

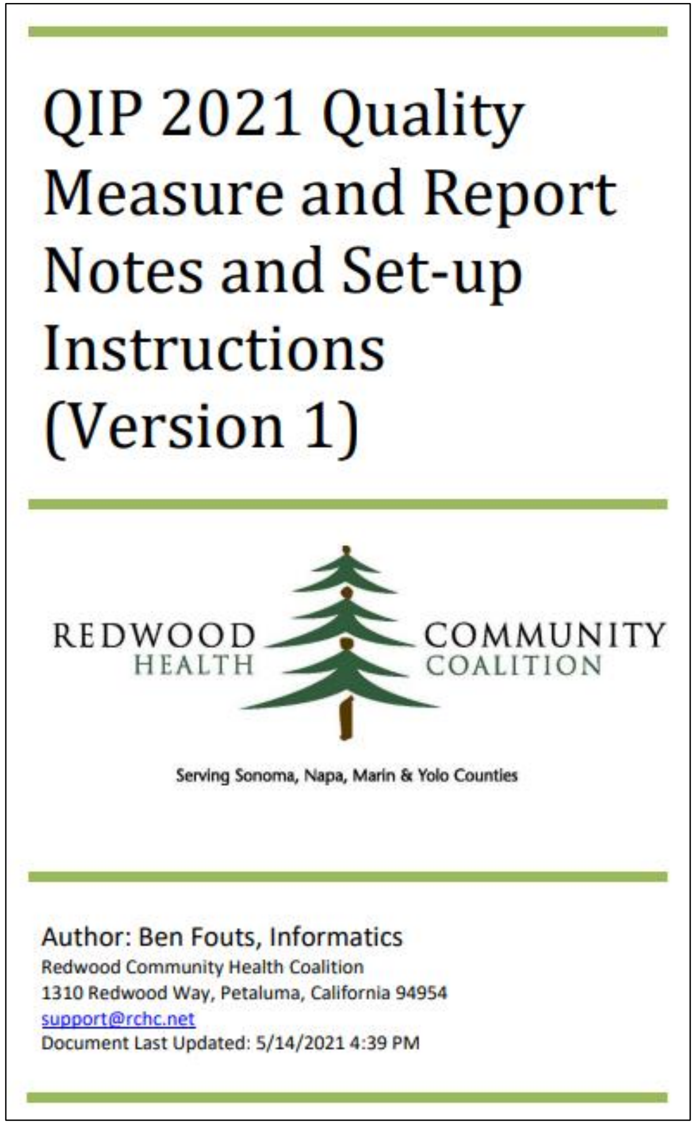
- eCQM Value Set table: cqm_value_set_codes
- Generally, we are using the eCQM Value Sets for those Transformers or Data Elements that are used commonly between UDS or QIP Quality Measures
- For example, there is only one Data Element to identify for patients with diabetes: relevant_diabetics. There are not separate Data Elements with JOINS to the UDS and QIP Value Sets
- From my research/experience, the Value Sets tend to be the same. If you come across a significant difference, let me know!

Which Transformers or Data Elements Need to be Switched?

https://www.rchc.net/population-health/data-analytics-and-governance/#toggle-id-2



- **QIP Reporting**
  - Recording 2021 QIP Report Set (5/2021)
  - Relevant 2021 QIP Report Set (slides, 5/2021)
  - Relevant 2021 QIP Report Set v1 (5/ 2021)
  - Reporting 2018 QIP Measures v1 (1/2019)



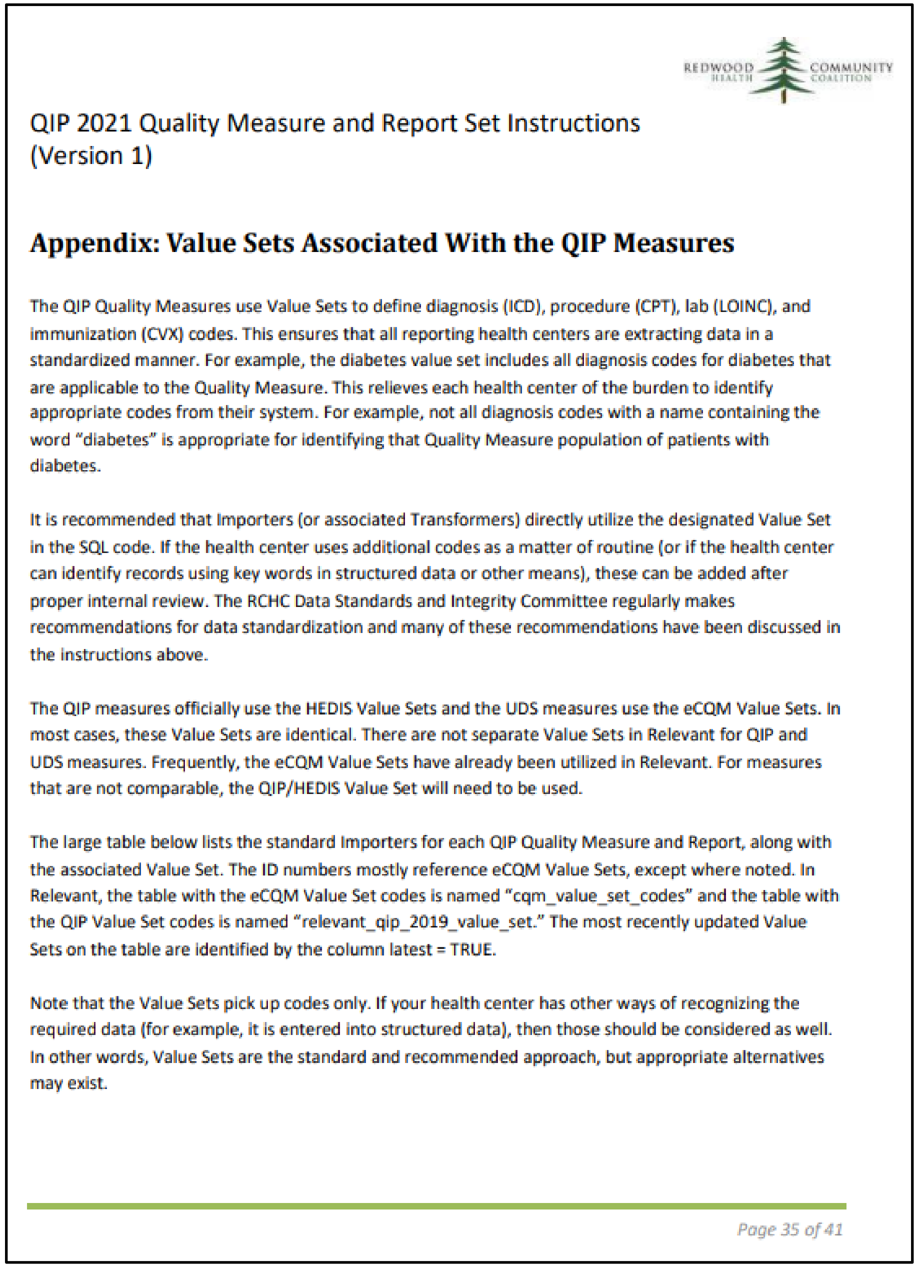
QIP 2021 Quality Measure and Report Notes and Set-up Instructions (Version 1)

REDWOOD HEALTH COMMUNITY COALITION

Serving Sonoma, Napa, Marin & Yolo Counties

Author: Ben Fouts, Informatics
Redwood Community Health Coalition
1310 Redwood Way, Petaluma, California 94954
support@rchc.net
Document Last Updated: 5/14/2021 4:39 PM

Appendix Table



REDWOOD HEALTH COMMUNITY COALITION

QIP 2021 Quality Measure and Report Set Instructions
(Version 1)

Appendix: Value Sets Associated With the QIP Measures

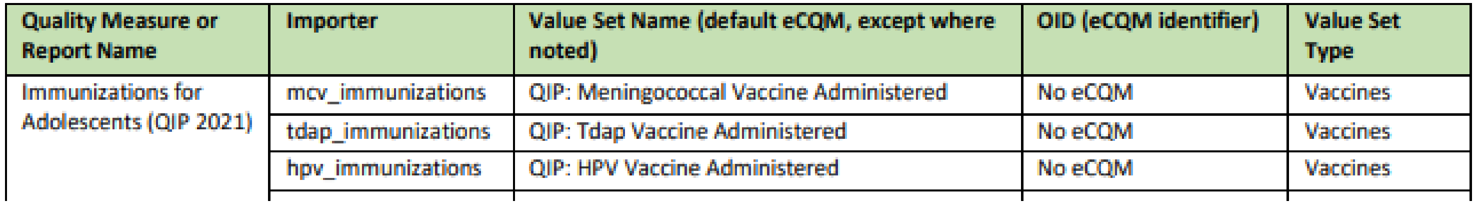
The QIP Quality Measures use Value Sets to define diagnosis (ICD), procedure (CPT), lab (LOINC), and immunization (CVX) codes. This ensures that all reporting health centers are extracting data in a standardized manner. For example, the diabetes value set includes all diagnosis codes for diabetes that are applicable to the Quality Measure. This relieves each health center of the burden to identify appropriate codes from their system. For example, not all diagnosis codes with a name containing the word "diabetes" is appropriate for identifying that Quality Measure population of patients with diabetes.

It is recommended that Importers (or associated Transformers) directly utilize the designated Value Set in the SQL code. If the health center uses additional codes as a matter of routine (or if the health center can identify records using key words in structured data or other means), these can be added after proper internal review. The RCHC Data Standards and Integrity Committee regularly makes recommendations for data standardization and many of these recommendations have been discussed in the instructions above.

The QIP measures officially use the HEDIS Value Sets and the UDS measures use the eCQM Value Sets. In most cases, these Value Sets are identical. There are not separate Value Sets in Relevant for QIP and UDS measures. Frequently, the eCQM Value Sets have already been utilized in Relevant. For measures that are not comparable, the QIP/HEDIS Value Set will need to be used.

The large table below lists the standard Importers for each QIP Quality Measure and Report, along with the associated Value Set. The ID numbers mostly reference eCQM Value Sets, except where noted. In Relevant, the table with the eCQM Value Set codes is named "cqm_value_set_codes" and the table with the QIP Value Set codes is named "relevant_qip_2019_value_set." The most recently updated Value Sets on the table are identified by the column latest = TRUE.

Note that the Value Sets pick up codes only. If your health center has other ways of recognizing the required data (for example, it is entered into structured data), then those should be considered as well. In other words, Value Sets are the standard and recommended approach, but appropriate alternatives may exist.

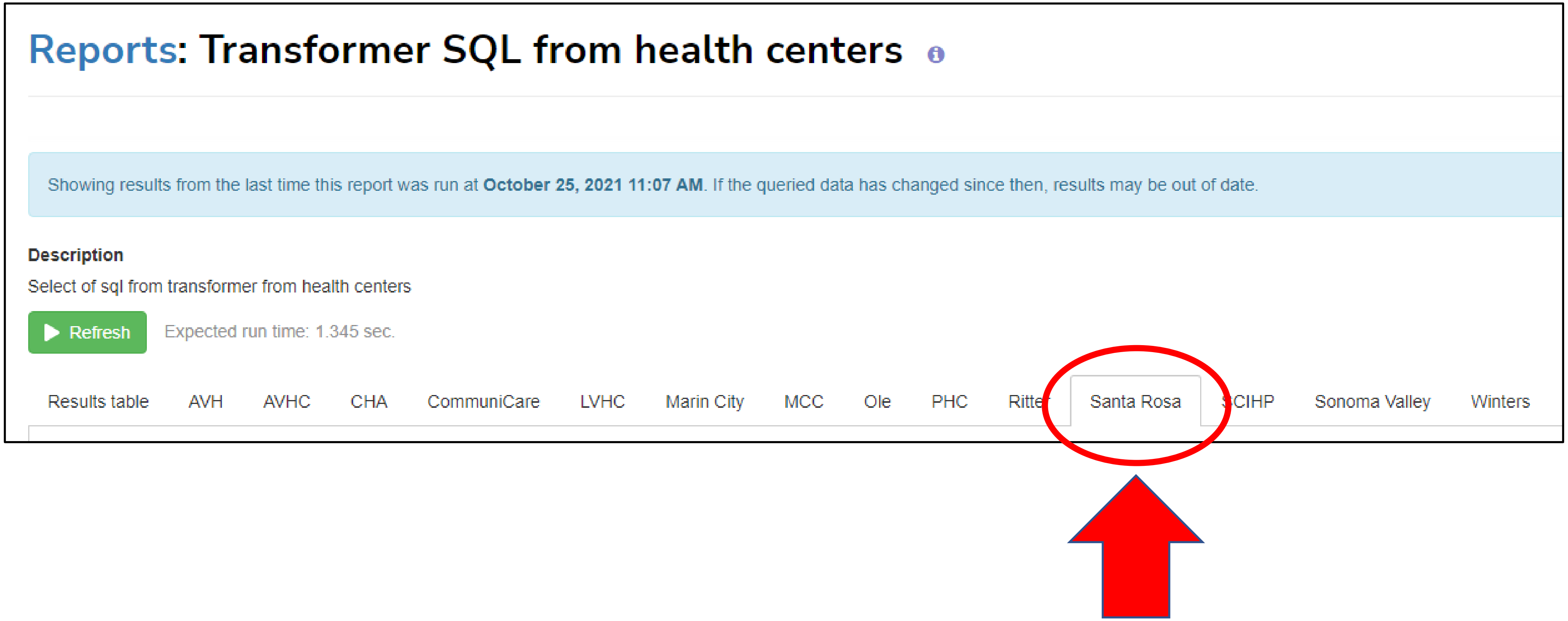


Quality Measure or Report Name	Importer	Value Set Name (default eCQM, except where noted)	OID (eCQM identifier)	Value Set Type
Immunizations for Adolescents (QIP 2021)	mcv_immunizations	QIP: Meningococcal Vaccine Administered	No eCQM	Vaccines
	tdap_immunizations	QIP: Tdap Vaccine Administered	No eCQM	Vaccines
	hpv_immunizations	QIP: HPV Vaccine Administered	No eCQM	Vaccines

- Generally, reference names to Value Sets should be the same. This document was created earlier in the year with references to the 2019 Value Sets

Other Research

- Use the SQL reports on the RCHC Aggregate
- Example:



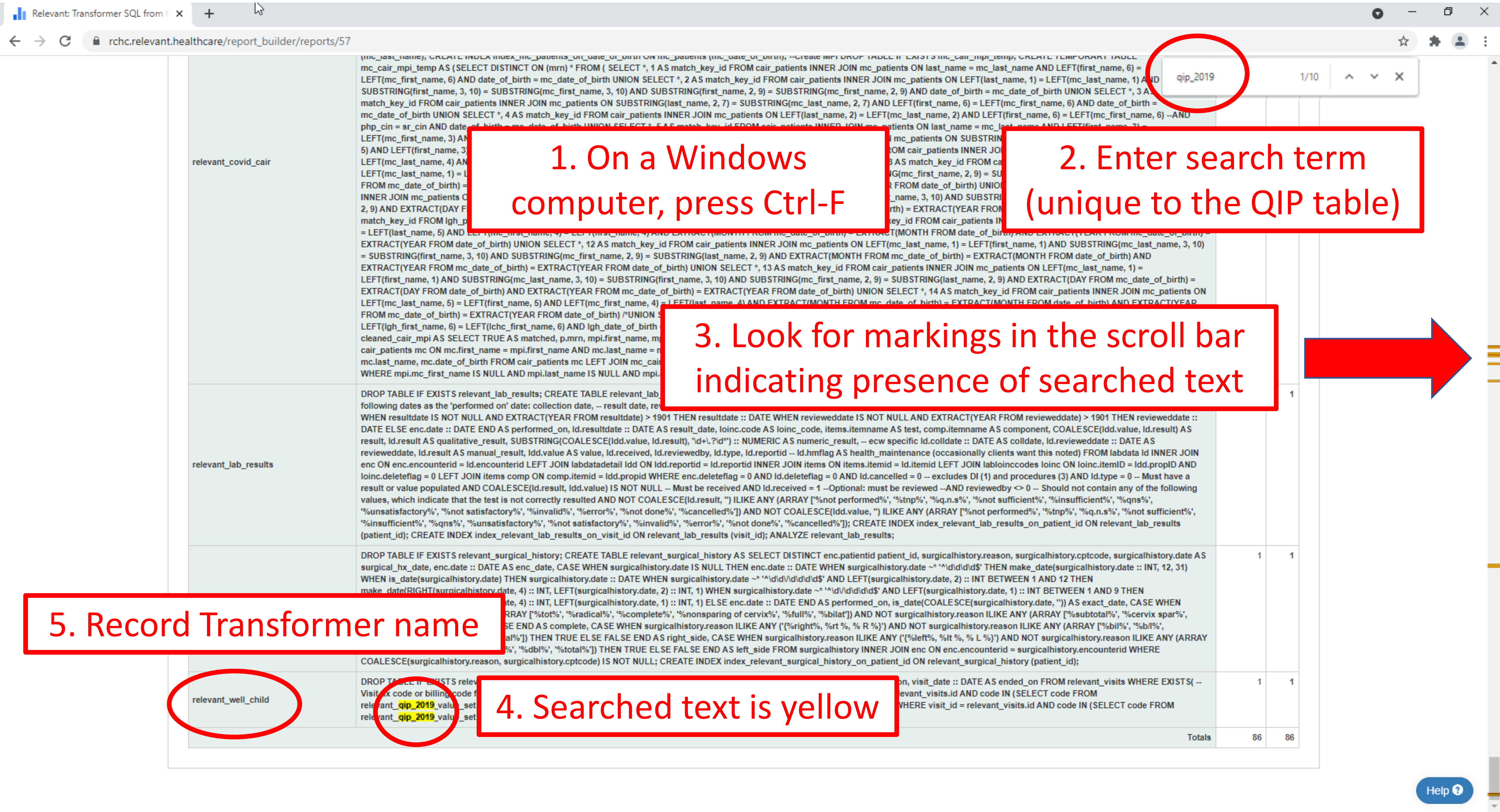
Reports: Transformer SQL from health centers

Showing results from the last time this report was run at **October 25, 2021 11:07 AM**. If the queried data has changed since then, results may be out of date.

Description

Select of sql from transformer from health centers

Expected run time: 1.345 sec.



Relevant: Transformer SQL from I
rchc.relevant.healthcare/report_builder/reports/57
qip_2019
1/10
Help

relevant_covid_cair	(mc_last_name); CREATE INDEX index_mc_patients_on_date_of_birth ON mc_patients (mc_date_of_birth); --Create MPI DROP TABLE IF EXISTS mc_cair_mpi_temp; CREATE TEMPORARY TABLE mc_cair_mpi_temp AS (SELECT DISTINCT ON (mrn) * FROM (SELECT *, 1 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON last_name = mc_last_name AND LEFT(first_name, 6) = LEFT(mc_first_name, 6) AND date_of_birth = mc_date_of_birth UNION SELECT *, 2 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON LEFT(last_name, 1) = LEFT(mc_last_name, 1) AND SUBSTRING(first_name, 3, 10) = SUBSTRING(mc_first_name, 3, 10) AND SUBSTRING(first_name, 2, 9) = SUBSTRING(mc_first_name, 2, 9) AND date_of_birth = mc_date_of_birth UNION SELECT *, 3 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON SUBSTRING(last_name, 2, 7) = SUBSTRING(mc_last_name, 2, 7) AND LEFT(first_name, 6) = LEFT(mc_first_name, 6) AND date_of_birth = mc_date_of_birth UNION SELECT *, 4 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON LEFT(last_name, 2) = LEFT(mc_last_name, 2) AND LEFT(first_name, 6) = LEFT(mc_first_name, 6) --AND php_cin = sr_cin AND date_of_birth = mc_date_of_birth UNION SELECT *, 5 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON last_name = mc_last_name AND LEFT(first_name, 3) = LEFT(mc_first_name, 3) AN mc_patients ON SUBSTRIN 5) AND LEFT(first_name, 3 ROM cair_patients INNER JO LEFT(mc_last_name, 4) AN 8 AS match_key_id FROM ca LEFT(mc_last_name, 1) = L NG(mc_first_name, 2, 9) = SU FROM mc_date_of_birth) = R FROM date_of_birth) UNIO INNER JOIN mc_patients O _name, 3, 10) AND SUBSTRI 2, 9) AND EXTRACT(DAY F rth) = EXTRACT(YEAR FROM match_key_id FROM lgh_p key_id FROM cair_patients IN = LEFT(last_name, 5) AND LEFT(mc_first_name, 4) = LEFT(first_name, 4) AND EXTRACT(MONTH FROM mc_date_of_birth) = EXTRACT(MONTH FROM date_of_birth) AND EXTRACT(YEAR FROM mc_date_of_birth) = EXTRACT(YEAR FROM date_of_birth) UNION SELECT *, 12 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON LEFT(mc_last_name, 1) = LEFT(first_name, 1) AND SUBSTRING(mc_last_name, 3, 10) = SUBSTRING(first_name, 3, 10) AND SUBSTRING(mc_first_name, 2, 9) = SUBSTRING(last_name, 2, 9) AND EXTRACT(MONTH FROM mc_date_of_birth) = EXTRACT(MONTH FROM date_of_birth) AND EXTRACT(YEAR FROM mc_date_of_birth) = EXTRACT(YEAR FROM date_of_birth) UNION SELECT *, 13 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON LEFT(mc_last_name, 1) = LEFT(first_name, 1) AND SUBSTRING(mc_last_name, 3, 10) = SUBSTRING(first_name, 3, 10) AND SUBSTRING(mc_first_name, 2, 9) = SUBSTRING(last_name, 2, 9) AND EXTRACT(DAY FROM mc_date_of_birth) = EXTRACT(DAY FROM date_of_birth) AND EXTRACT(YEAR FROM mc_date_of_birth) = EXTRACT(YEAR FROM date_of_birth) UNION SELECT *, 14 AS match_key_id FROM cair_patients INNER JOIN mc_patients ON LEFT(mc_last_name, 5) = LEFT(first_name, 5) AND LEFT(mc_first_name, 4) = LEFT(last_name, 4) AND EXTRACT(MONTH FROM mc_date_of_birth) = EXTRACT(MONTH FROM date_of_birth) AND EXTRACT(YEAR FROM mc_date_of_birth) = EXTRACT(YEAR FROM date_of_birth) /*UNION S LEFT(lgh_first_name, 6) = LEFT(lchc_first_name, 6) AND lgh_date_of_birth cleaned_cair_mpi AS SELECT TRUE AS matched, p.mrn, mpi.first_name, mp cair_patients mc ON mc.first_name = mpi.first_name AND mc.last_name = m mc.last_name, mc.date_of_birth FROM cair_patients mc LEFT JOIN mc_cai WHERE mpi.mc_first_name IS NULL AND mpi.last_name IS NULL AND mpi.		
relevant_lab_results	DROP TABLE IF EXISTS relevant_lab_results; CREATE TABLE relevant_lab following dates as the 'performed on' date: collection date, -- result date, re WHEN resultdate IS NOT NULL AND EXTRACT(YEAR FROM resultdate) > 1901 THEN resultdate :: DATE WHEN revieweddate IS NOT NULL AND EXTRACT(YEAR FROM revieweddate) > 1901 THEN revieweddate :: DATE ELSE enc.date :: DATE END AS performed_on, ld.resultdate :: DATE AS result_date, loinc.code AS loinc_code, items.itemname AS test, comp.itemname AS component, COALESCE(ldd.value, ld.result) AS result, ld.result AS qualitative_result, SUBSTRING(COALESCE(ldd.value, ld.result), '\d+\.?\d*') :: NUMERIC AS numeric_result, -- ecw specific ld.colldate :: DATE AS colldate, ld.revieweddate :: DATE AS revieweddate, ld.result AS manual_result, ldd.value AS value, ld.received, ld.reviewedby, ld.type, ld.reportid -- ld.hmflag AS health_maintenance (occasionally clients want this noted) FROM labdata ld INNER JOIN enc ON enc.encounterid = ld.encounterid LEFT JOIN labdatadetail ldd ON ldd.reportid = ld.reportid INNER JOIN items ON items.itemid = ld.itemid LEFT JOIN labloinccodes loinc ON loinc.itemID = ldd.propID AND loinc.deleteflag = 0 LEFT JOIN items comp ON comp.itemid = ldd.propid WHERE enc.deleteflag = 0 AND ld.deleteflag = 0 AND ld.cancelled = 0 -- excludes DI (1) and procedures (3) AND ld.type = 0 -- Must have a result or value populated AND COALESCE(ld.result, ldd.value) IS NOT NULL -- Must be received AND ld.received = 1 --Optional: must be reviewed --AND reviewedby <> 0 -- Should not contain any of the following values, which indicate that the test is not correctly resulted AND NOT COALESCE(ld.result, '') ILIKE ANY (ARRAY ['%not performed%', '%tnp%', '%q.n.s%', '%not sufficient%', '%insufficient%', '%qns%', '%unsatisfactory%', '%not satisfactory%', '%invalid%', '%error%', '%not done%', '%cancelled%']) AND NOT COALESCE(ldd.value, '') ILIKE ANY (ARRAY ['%not performed%', '%tnp%', '%q.n.s%', '%not sufficient%', '%insufficient%', '%qns%', '%unsatisfactory%', '%not satisfactory%', '%invalid%', '%error%', '%not done%', '%cancelled%']); CREATE INDEX index_relevant_lab_results_on_patient_id ON relevant_lab_results (patient_id); CREATE INDEX index_relevant_lab_results_on_visit_id ON relevant_lab_results (visit_id); ANALYZE relevant_lab_results;		1
	DROP TABLE IF EXISTS relevant_surgical_history; CREATE TABLE relevant_surgical_history AS SELECT DISTINCT enc.patientid patient_id, surgicalhistory.reason, surgicalhistory.cptcode, surgicalhistory.date AS surgical_hx_date, enc.date :: DATE AS enc_date, CASE WHEN surgicalhistory.date IS NULL THEN enc.date :: DATE WHEN surgicalhistory.date ~* '^\d\d\d\d$' THEN make_date(surgicalhistory.date :: INT, 12, 31) WHEN is_date(surgicalhistory.date) THEN surgicalhistory.date :: DATE WHEN surgicalhistory.date ~* '^\d\d\/\d\d\d\d$' AND LEFT(surgicalhistory.date, 2) :: INT BETWEEN 1 AND 12 THEN make_date(RIGHT(surgicalhistory.date, 4) :: INT, LEFT(surgicalhistory.date, 2) :: INT, 1) WHEN surgicalhistory.date ~* '^\d\/\d\d\d\d$' AND LEFT(surgicalhistory.date, 1) :: INT BETWEEN 1 AND 9 THEN te, 4) :: INT, LEFT(surgicalhistory.date, 1) :: INT, 1) ELSE enc.date :: DATE END AS performed_on, is_date(COALESCE(surgicalhistory.date, '')) AS exact_date, CASE WHEN RRAY ['%tot%', '%radical%', '%complete%', '%nonsparing of cervix%', '%full%', '%bilat']) AND NOT surgicalhistory.reason ILIKE ANY (ARRAY ['%subtotal%', '%cervix spar%', SE END AS complete, CASE WHEN surgicalhistory.reason ILIKE ANY ('{%right%, %rt %, % R %}') AND NOT surgicalhistory.reason ILIKE ANY (ARRAY ['%bil%', '%b/l%', al%']) THEN TRUE ELSE FALSE END AS right_side, CASE WHEN surgicalhistory.reason ILIKE ANY ('{%left%, %lt %, % L %}') AND NOT surgicalhistory.reason ILIKE ANY (ARRAY %', '%dbl%', '%total%']) THEN TRUE ELSE FALSE END AS left_side FROM surgicalhistory INNER JOIN enc ON enc.encounterid = surgicalhistory.encounterid WHERE COALESCE(surgicalhistory.reason, surgicalhistory.cptcode) IS NOT NULL; CREATE INDEX index_relevant_surgical_history_on_patient_id ON relevant_surgical_history (patient_id);	1	1
relevant_well_child	DROP TABLE IF EXISTS relev on, visit_date :: DATE AS ended_on FROM relevant_visits WHERE EXISTS(-- Visit dx code or billing code f levant_visits.id AND code IN (SELECT code FROM relevant_qip_2019_value_set WHERE visit_id = relevant_visits.id AND code IN (SELECT code FROM relevant_qip_2019_value_set	1	1
	Totals	86	86

1. On a Windows computer, press Ctrl-F

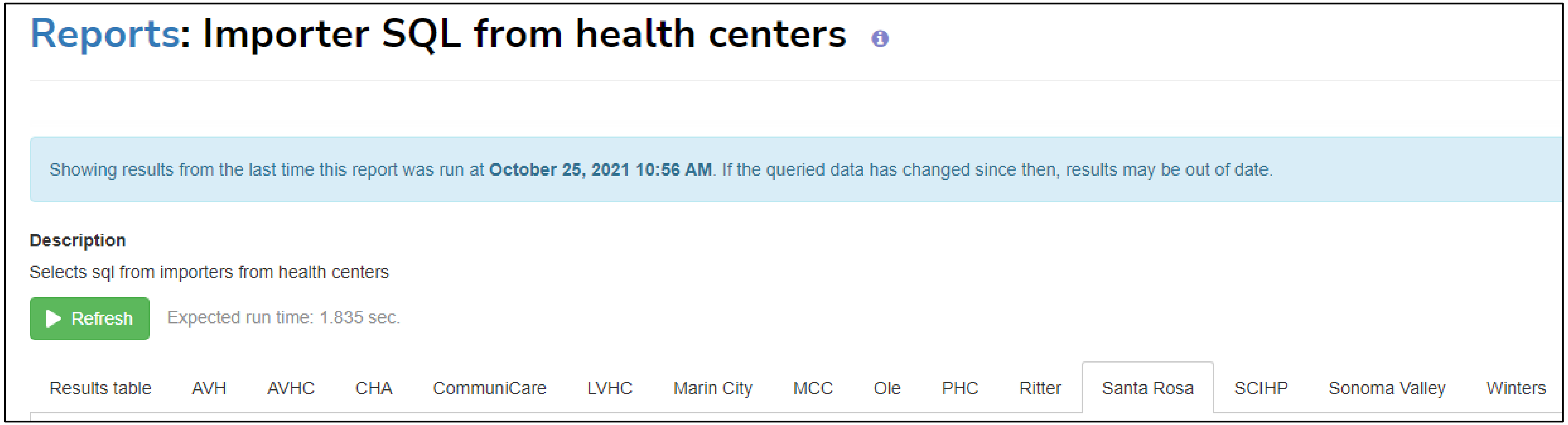
2. Enter search term (unique to the QIP table)

3. Look for markings in the scroll bar indicating presence of searched text

4. Searched text is yellow

5. Record Transformer name

Other RCHC Aggregate Reports



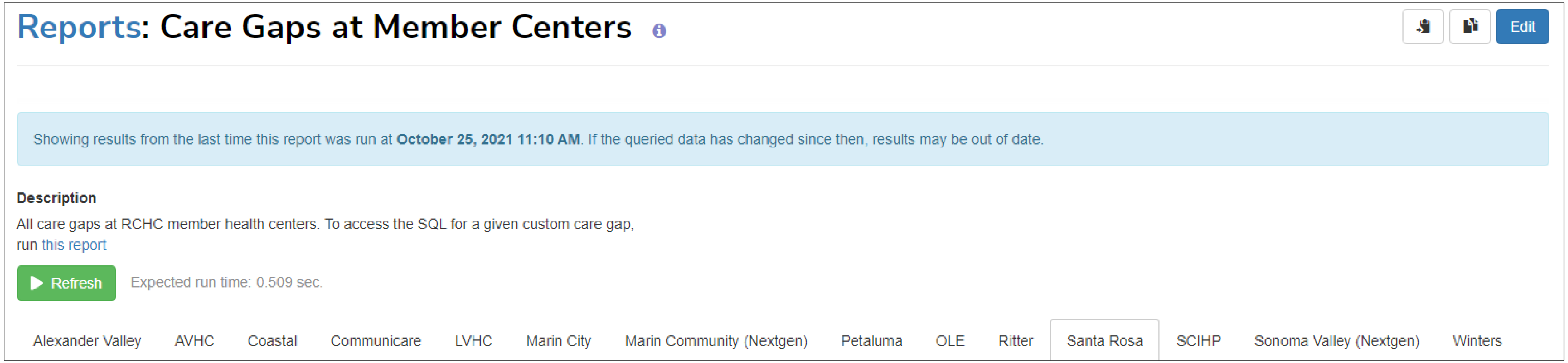
Reports: Importer SQL from health centers

Showing results from the last time this report was run at **October 25, 2021 10:56 AM**. If the queried data has changed since then, results may be out of date.

Description

Selects sql from importers from health centers

Expected run time: 1.835 sec.



Reports: Care Gaps at Member Centers

Showing results from the last time this report was run at **October 25, 2021 11:10 AM**. If the queried data has changed since then, results may be out of date.

Description

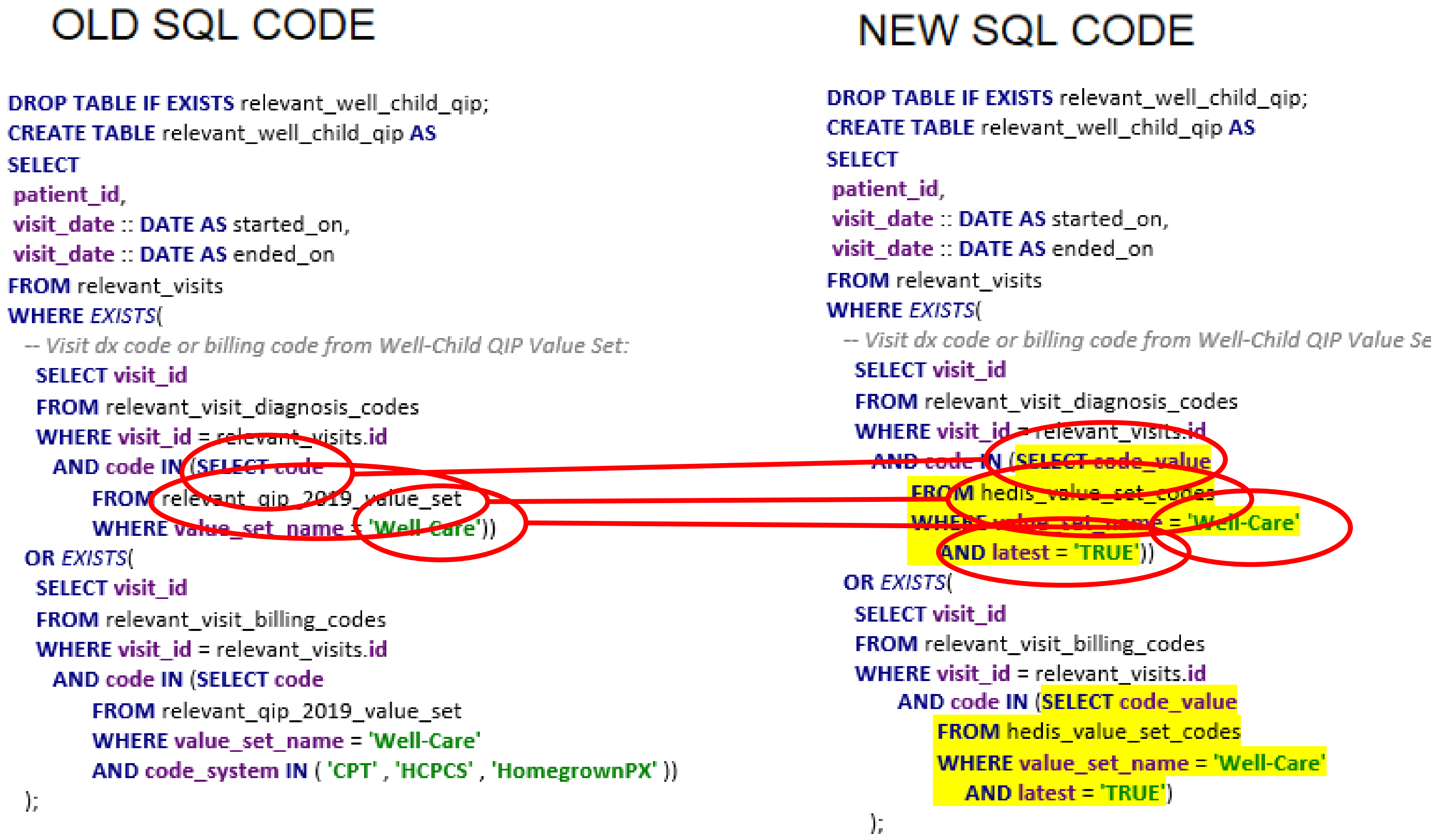
All care gaps at RCHC member health centers. To access the SQL for a given custom care gap, run this report

Expected run time: 0.509 sec.

Key QIP Quality Measures and Value Sets

- Well-Child Visits in the First 15 Months of Life (QIP 2021)
  - ➢ Well Care
- Child and Adolescent Well-Care Visits (QIP 2021)
  - ➢ Well Care
- Immunizations for Adolescents (QIP 2021)
  - ➢ Meningococcal Vaccine Administered
  - ➢ Tdap Vaccine Administered
  - ➢ HPV Vaccine Administered

SQL Change Example



OLD SQL CODE

```
DROP TABLE IF EXISTS relevant_well_child_qip;
CREATE TABLE relevant_well_child_qip AS
SELECT
 patient_id,
 visit_date :: DATE AS started_on,
 visit_date :: DATE AS ended_on
FROM relevant_visits
WHERE EXISTS(
  -- Visit dx code or billing code from Well-Child QIP Value Set:
   SELECT visit_id
   FROM relevant_visit_diagnosis_codes
   WHERE visit_id = relevant_visits.id
     AND code IN (SELECT code
          FROM relevant_qip_2019_value_set
          WHERE value_set_name = 'Well-Care'))
  OR EXISTS(
   SELECT visit_id
   FROM relevant_visit_billing_codes
   WHERE visit_id = relevant_visits.id
     AND code IN (SELECT code
          FROM relevant_qip_2019_value_set
          WHERE value_set_name = 'Well-Care'
          AND code_system IN ( 'CPT' , 'HCPCS' , 'HomegrownPX' ))
  );
```

NEW SQL CODE

```
DROP TABLE IF EXISTS relevant_well_child_qip;
CREATE TABLE relevant_well_child_qip AS
SELECT
 patient_id,
 visit_date :: DATE AS started_on,
 visit_date :: DATE AS ended_on
FROM relevant_visits
WHERE EXISTS(
  -- Visit dx code or billing code from Well-Child QIP Value Se
   SELECT visit_id
   FROM relevant_visit_diagnosis_codes
   WHERE visit_id = relevant_visits.id
     AND code IN (SELECT code_value
          FROM hedis_value_set_codes
          WHERE value_set_name = 'Well-Care'
            AND latest = 'TRUE'))
  OR EXISTS(
   SELECT visit_id
   FROM relevant_visit_billing_codes
   WHERE visit_id = relevant_visits.id
       AND code IN (SELECT code_value
            FROM hedis_value_set_codes
            WHERE value_set_name = 'Well-Care'
              AND latest = 'TRUE')
   );
```

Other Notes

- You might find Transformers/Importers from "old" measures that are connected to the 2019 QIP Value Set. Examples: diabetes nephropathy screens and retinal eye exams. You might as well "fix" those as well. They are still part of the model and might be used for Care Gaps (or they might return as measures)
- There might be references in the new Data Elements, like relevant_cases or relevant_medications
- The HEDIS Value Set table contains all of the Value Sets, not just the QIP. Therefore, you can also use it to find codes for *ad hoc* reports

Other Notes

- Ben will update the RCHC Validation Reports on the RCHC Aggregate that use the old 2019 QIP Value Set table. You should update these in your Relevant instance as well
- Ben is researching the “HomegrownPx” codes in the 2019 QIP Value Sets
- The HEDIS Value Set table exists currently in the Staging Database. Ben is asking Relevant about also putting it in the Production Database