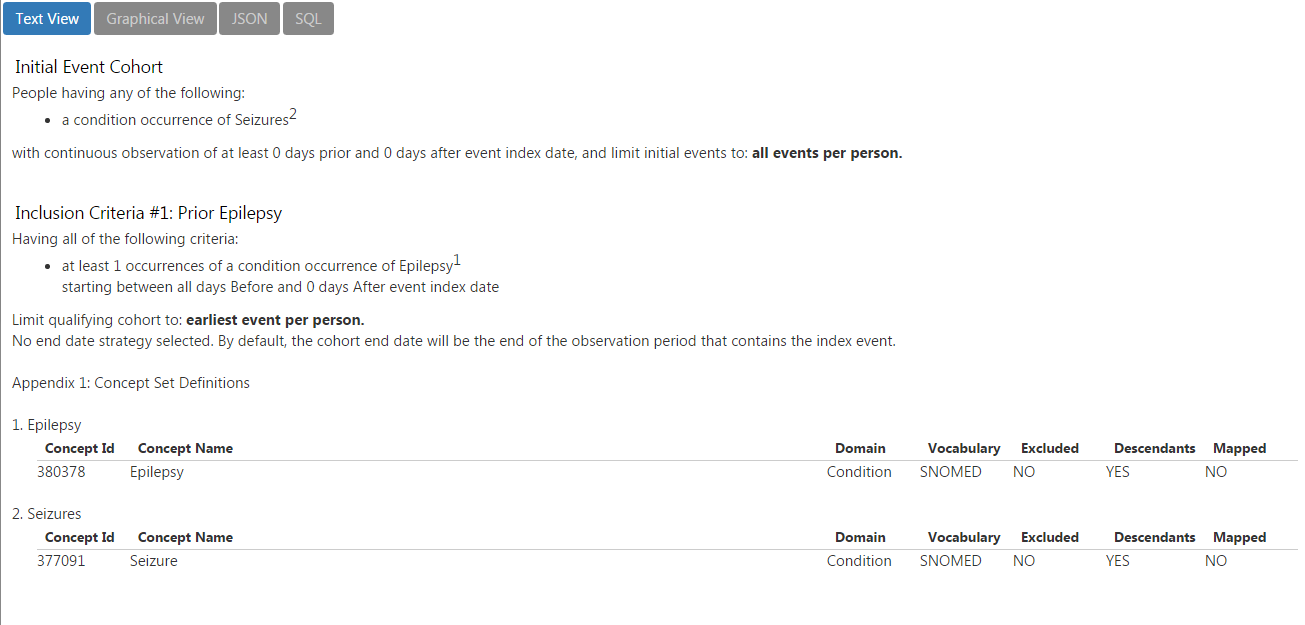
**Method:**

To define patients with Seizures and Epilepsy, the ICD10 codes defined by the Mini-Sentinel study were ran through the OMOP V5 Concept table and Concept Relationship table to find the appropriate SNOMED concept ids. Then, using Atlas we validated our code by searching and creating a cohort for Seizures. This gave a list of SNOMED codes which we ran across the QuintilesIMS data—France Disease Analyzer and German Disease Analyzer. Then, an inclusive cohort was created with Atlas to identify the patients that were diagnosed with Seizure how many had Epilepsy in the data. The inclusion of prior history of seizure was applied in the logic.

**Cohort Criteria:**

Using Atlas, we defined the cohort for Seizure and Epilepsy:



**Results:**

Using German DA data, we ran “Query 1” from this document (using ATLAS) and found 54,560 patients had seizures out of 30,162,584 total population of patients in the database. Out of the 54,560, using our inclusion criteria of a prior diagnosis of Seizure, we found that 8,144 patients were diagnosed with Epilepsy. This was using “Query 2” from this document (using ATLAS).

Using the France DA data, we ran “Query 1” and found 1,856 patients were diagnosed with the condition of Seizure out of a total population of 9,551,252 in the database. Out of these 1,856 patients were diagnosed with Epilepsy, who had at least 1 prior diagnosis of Seizures.

**Limitations:**

To clearly define the sample of patients diagnosed with Seizures and Epilepsy, the Mini-Sentinel study used the definition of inpatient and outpatient visits. The data they used, included this level of detail. They separated all patients with inpatient visits to validate Epilepsy ICD codes. However, the France Disease Analyzer and German Disease Analyzer databases which only consist of office visits. The data does not contain any hospital visits, therefore the methodology of inpatient vs outpatient could not be used. Also, both datasets only contain WHO-ICD codes, therefore the validation of ICD9 codes could not be completed.

Part 1: Take Sentinel codes for Seizure/Epilepsy and check in data if we have patients.

--France 16063

select count(distinct person\_id) from condition\_occurrence co

where co.CONDITION\_CONCEPT\_ID in (

select concept\_id from CONCEPT c

where c.concept\_id in (select Concept\_ID\_2 from concept\_relationship cr where cr.CONCEPT\_ID\_1 in

(select concept\_id from concept where CONCEPT\_CODE in

('G40.X','G40.0','G40.1','G40.2','G40.3','G40.4','G40.5','G40.6','G40.7','G40.8','G40.9','G41.X','G41.0','G41.1','G41.2','G41.8','G41.9','G43.1','G45.X','R55','R56.0','R56.8'

))

and RELATIONSHIP\_ID='Maps to')

and domain\_id = 'Condition'

)

and (TO\_CHAR(co.CONDITION\_START\_DATE, 'YYYY/MM/DD') >= '2005/01/01' and TO\_CHAR(co.CONDITION\_START\_DATE, 'YYYY/MM/DD') <= '2007/01/01')

--Germany 59374

--Find for SNOMED concept ids

select \* from (

select distinct condition\_concept\_id from condition\_occurrence co

where co.CONDITION\_CONCEPT\_ID in (

select concept\_id from CONCEPT c

where c.concept\_id in (select Concept\_ID\_2 from concept\_relationship cr where cr.CONCEPT\_ID\_1 in

(select concept\_id from concept where CONCEPT\_CODE in

('G40.X','G40.0','G40.1','G40.2','G40.3','G40.4','G40.5','G40.6','G40.7','G40.8','G40.9','G41.X','G41.0','G41.1','G41.2','G41.8','G41.9','G43.1','G45.X','R55','R56.0','R56.8'

))

and RELATIONSHIP\_ID='Maps to')

and domain\_id = 'Condition'

)

and (TO\_CHAR(co.CONDITION\_START\_DATE, 'YYYY/MM/DD') >= '2005/01/01' and TO\_CHAR(co.CONDITION\_START\_DATE, 'YYYY/MM/DD') <= '2007/01/01')

) join concept c on condition\_concept\_id = c.concept\_id

Query 1:

Part 2: How many patients have Seizure—

CREATE TABLE Codesets (

codeset\_id int NOT NULL,

concept\_id bigint NOT NULL

)

;

INSERT INTO Codesets (codeset\_id, concept\_id)

SELECT 0 as codeset\_id, c.concept\_id FROM (select distinct I.concept\_id FROM

(

select concept\_id from CONCEPT where concept\_id in (377091)and invalid\_reason is null

UNION select c.concept\_id

from CONCEPT c

join CONCEPT\_ANCESTOR ca on c.concept\_id = ca.descendant\_concept\_id

and ca.ancestor\_concept\_id in (377091)

and c.invalid\_reason is null

) I

) C;

CREATE TABLE qualified\_events

AS

WITH primary\_events (event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date) AS

(

-- Begin Primary Events

select row\_number() over (PARTITION BY P.person\_id order by P.start\_date) as event\_id, P.person\_id, P.start\_date, P.end\_date, OP.observation\_period\_start\_date as op\_start\_date, OP.observation\_period\_end\_date as op\_end\_date

FROM

(

select P.person\_id, P.start\_date, P.end\_date, ROW\_NUMBER() OVER (PARTITION BY person\_id ORDER BY start\_date ASC) ordinal

FROM

(

-- Begin Condition Occurrence Criteria

SELECT C.person\_id, C.condition\_occurrence\_id as event\_id, C.condition\_start\_date as start\_date, COALESCE(C.condition\_end\_date, (C.condition\_start\_date + 1)) as end\_date, C.CONDITION\_CONCEPT\_ID as TARGET\_CONCEPT\_ID

FROM

(

SELECT co.\*, ROW\_NUMBER() over (PARTITION BY co.person\_id ORDER BY co.condition\_start\_date, co.condition\_occurrence\_id) as ordinal

FROM CONDITION\_OCCURRENCE co

where co.condition\_concept\_id in (SELECT concept\_id from Codesets where codeset\_id = 0)

) C

-- End Condition Occurrence Criteria

) P

) P

JOIN observation\_period OP on P.person\_id = OP.person\_id and P.start\_date >= OP.observation\_period\_start\_date and P.start\_date <= op.observation\_period\_end\_date

WHERE (OP.OBSERVATION\_PERIOD\_START\_DATE + 0) <= P.START\_DATE AND (P.START\_DATE + 0) <= OP.OBSERVATION\_PERIOD\_END\_DATE

-- End Primary Events

)

SELECT

event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date

FROM

(

select pe.event\_id, pe.person\_id, pe.start\_date, pe.end\_date, pe.op\_start\_date, pe.op\_end\_date, row\_number() over (partition by pe.person\_id order by pe.start\_date ASC) as ordinal

FROM primary\_events pe

) QE

;

CREATE TABLE inclusionRuleCohorts

(

inclusion\_rule\_id bigint,

person\_id bigint,

event\_id bigint

)

;

CREATE TABLE included\_events

AS

WITH cteIncludedEvents(event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date, ordinal) AS

(

SELECT event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date, row\_number() over (partition by person\_id order by start\_date ASC) as ordinal

from

(

select Q.event\_id, Q.person\_id, Q.start\_date, Q.end\_date, Q.op\_start\_date, Q.op\_end\_date, SUM(coalesce(POWER(cast(2 as bigint), I.inclusion\_rule\_id), 0)) as inclusion\_rule\_mask

from qualified\_events Q

LEFT JOIN inclusionRuleCohorts I on I.person\_id = Q.person\_id and I.event\_id = Q.event\_id

GROUP BY Q.event\_id, Q.person\_id, Q.start\_date, Q.end\_date, Q.op\_start\_date, Q.op\_end\_date

) MG -- matching groups

)

SELECT

event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date

FROM

cteIncludedEvents Results

WHERE Results.ordinal = 1

;

-- Apply end date stratagies

-- by default, all events extend to the op\_end\_date.

CREATE TABLE cohort\_ends

AS

SELECT

event\_id, person\_id, op\_end\_date as end\_date

FROM

included\_events;

--DELETE FROM @target\_database\_schema.@target\_cohort\_table where cohort\_definition\_id = @target\_cohort\_id;

--INSERT INTO @target\_database\_schema.@target\_cohort\_table (cohort\_definition\_id, subject\_id, cohort\_start\_date, cohort\_end\_date)

select -1 as cohort\_definition\_id, F.person\_id, F.start\_date, F.end\_date

FROM (

select I.person\_id, I.start\_date, E.end\_date, row\_number() over (partition by I.person\_id, I.event\_id order by E.end\_date) as ordinal

from included\_events I

join cohort\_ends E on I.event\_id = E.event\_id and I.person\_id = E.person\_id and E.end\_date >= I.start\_date

) F

WHERE F.ordinal = 1

;

Query 2: Find Epilepsy patients out of the Seizure patients, inclusion criteria of prior history of Seizure

CREATE TABLE Codesets (

codeset\_id int NOT NULL,

concept\_id bigint NOT NULL

)

;

INSERT INTO Codesets (codeset\_id, concept\_id)

SELECT 0 as codeset\_id, c.concept\_id FROM (select distinct I.concept\_id FROM

(

select concept\_id from CONCEPT where concept\_id in (377091)and invalid\_reason is null

UNION select c.concept\_id

from CONCEPT c

join CONCEPT\_ANCESTOR ca on c.concept\_id = ca.descendant\_concept\_id

and ca.ancestor\_concept\_id in (377091)

and c.invalid\_reason is null

) I

) C;

INSERT INTO Codesets (codeset\_id, concept\_id)

SELECT 1 as codeset\_id, c.concept\_id FROM (select distinct I.concept\_id FROM

(

select concept\_id from CONCEPT where concept\_id in (380378)and invalid\_reason is null

UNION select c.concept\_id

from CONCEPT c

join CONCEPT\_ANCESTOR ca on c.concept\_id = ca.descendant\_concept\_id

and ca.ancestor\_concept\_id in (380378)

and c.invalid\_reason is null

) I

) C;

CREATE TABLE qualified\_events

AS

WITH primary\_events (event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date) AS

(

-- Begin Primary Events

select row\_number() over (PARTITION BY P.person\_id order by P.start\_date) as event\_id, P.person\_id, P.start\_date, P.end\_date, OP.observation\_period\_start\_date as op\_start\_date, OP.observation\_period\_end\_date as op\_end\_date

FROM

(

select P.person\_id, P.start\_date, P.end\_date, ROW\_NUMBER() OVER (PARTITION BY person\_id ORDER BY start\_date ASC) ordinal

FROM

(

-- Begin Condition Occurrence Criteria

SELECT C.person\_id, C.condition\_occurrence\_id as event\_id, C.condition\_start\_date as start\_date, COALESCE(C.condition\_end\_date, (C.condition\_start\_date + 1)) as end\_date, C.CONDITION\_CONCEPT\_ID as TARGET\_CONCEPT\_ID

FROM

(

SELECT co.\*, ROW\_NUMBER() over (PARTITION BY co.person\_id ORDER BY co.condition\_start\_date, co.condition\_occurrence\_id) as ordinal

FROM CONDITION\_OCCURRENCE co

where co.condition\_concept\_id in (SELECT concept\_id from Codesets where codeset\_id = 0)

) C

-- End Condition Occurrence Criteria

) P

) P

JOIN observation\_period OP on P.person\_id = OP.person\_id and P.start\_date >= OP.observation\_period\_start\_date and P.start\_date <= op.observation\_period\_end\_date

WHERE (OP.OBSERVATION\_PERIOD\_START\_DATE + 0) <= P.START\_DATE AND (P.START\_DATE + 0) <= OP.OBSERVATION\_PERIOD\_END\_DATE

-- End Primary Events

)

SELECT

event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date

FROM

(

select pe.event\_id, pe.person\_id, pe.start\_date, pe.end\_date, pe.op\_start\_date, pe.op\_end\_date, row\_number() over (partition by pe.person\_id order by pe.start\_date ASC) as ordinal

FROM primary\_events pe

) QE

;

CREATE TABLE inclusionRuleCohorts

(

inclusion\_rule\_id bigint,

person\_id bigint,

event\_id bigint

)

;

INSERT INTO inclusionRuleCohorts (inclusion\_rule\_id, person\_id, event\_id)

select 0 as inclusion\_rule\_id, person\_id, event\_id

FROM

(

select pe.person\_id, pe.event\_id

FROM qualified\_events pe

JOIN (

-- Begin Criteria Group

select 0 as index\_id, person\_id, event\_id

FROM

(

select E.person\_id, E.event\_id

FROM qualified\_events E

LEFT JOIN

(

-- Begin Correlated Criteria

SELECT 0 as index\_id, p.person\_id, p.event\_id

FROM qualified\_events P

LEFT JOIN

(

-- Begin Condition Occurrence Criteria

SELECT C.person\_id, C.condition\_occurrence\_id as event\_id, C.condition\_start\_date as start\_date, COALESCE(C.condition\_end\_date, (C.condition\_start\_date + 1)) as end\_date, C.CONDITION\_CONCEPT\_ID as TARGET\_CONCEPT\_ID

FROM

(

SELECT co.\*, ROW\_NUMBER() over (PARTITION BY co.person\_id ORDER BY co.condition\_start\_date, co.condition\_occurrence\_id) as ordinal

FROM CONDITION\_OCCURRENCE co

where co.condition\_concept\_id in (SELECT concept\_id from Codesets where codeset\_id = 1)

) C

-- End Condition Occurrence Criteria

) A on A.person\_id = P.person\_id and A.START\_DATE >= P.OP\_START\_DATE AND A.START\_DATE <= P.OP\_END\_DATE AND A.START\_DATE >= P.OP\_START\_DATE and A.START\_DATE <= (P.START\_DATE + 0)

GROUP BY p.person\_id, p.event\_id

HAVING COUNT(A.TARGET\_CONCEPT\_ID) >= 1

-- End Correlated Criteria

) CQ on E.person\_id = CQ.person\_id and E.event\_id = CQ.event\_id

GROUP BY E.person\_id, E.event\_id

HAVING COUNT(index\_id) = 1

) G

-- End Criteria Group

) AC on AC.person\_id = pe.person\_id AND AC.event\_id = pe.event\_id

) Results

;

CREATE TABLE included\_events

AS

WITH cteIncludedEvents(event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date, ordinal) AS

(

SELECT event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date, row\_number() over (partition by person\_id order by start\_date ASC) as ordinal

from

(

select Q.event\_id, Q.person\_id, Q.start\_date, Q.end\_date, Q.op\_start\_date, Q.op\_end\_date, SUM(coalesce(POWER(cast(2 as bigint), I.inclusion\_rule\_id), 0)) as inclusion\_rule\_mask

from qualified\_events Q

LEFT JOIN inclusionRuleCohorts I on I.person\_id = Q.person\_id and I.event\_id = Q.event\_id

GROUP BY Q.event\_id, Q.person\_id, Q.start\_date, Q.end\_date, Q.op\_start\_date, Q.op\_end\_date

) MG -- matching groups

-- the matching group with all bits set ( POWER(2,# of inclusion rules) - 1 = inclusion\_rule\_mask

WHERE (MG.inclusion\_rule\_mask = POWER(cast(2 as bigint),1)-1)

)

SELECT

event\_id, person\_id, start\_date, end\_date, op\_start\_date, op\_end\_date

FROM

cteIncludedEvents Results

WHERE Results.ordinal = 1

;

-- Apply end date stratagies

-- by default, all events extend to the op\_end\_date.

CREATE TABLE cohort\_ends

AS

SELECT

event\_id, person\_id, op\_end\_date as end\_date

FROM

included\_events;

--DELETE FROM @target\_database\_schema.@target\_cohort\_table where cohort\_definition\_id = @target\_cohort\_id;

--INSERT INTO @target\_database\_schema.@target\_cohort\_table (cohort\_definition\_id, subject\_id, cohort\_start\_date, cohort\_end\_date)

select -1 as cohort\_definition\_id, F.person\_id, F.start\_date, F.end\_date

FROM (

select I.person\_id, I.start\_date, E.end\_date, row\_number() over (partition by I.person\_id, I.event\_id order by E.end\_date) as ordinal

from included\_events I

join cohort\_ends E on I.event\_id = E.event\_id and I.person\_id = E.person\_id and E.end\_date >= I.start\_date

) F

WHERE F.ordinal = 1

;

TRUNCATE TABLE cohort\_ends;

DROP TABLE cohort\_ends;

TRUNCATE TABLE inclusionRuleCohorts;

DROP TABLE inclusionRuleCohorts;

TRUNCATE TABLE qualified\_events;

DROP TABLE qualified\_events;

TRUNCATE TABLE included\_events;

DROP TABLE included\_events;

TRUNCATE TABLE Codesets;

DROP TABLE Codesets;