LIBNAME COHORT '/userdata06/room241/data\_source/COHORT';

LIBNAME HOME '/userdata06/room241/data\_source/youb';

/\* ============================================

STEP 1: Create indexed patient lookup with demographics

400K patients - perfect size for indexed SQL joins

============================================ \*/

PROC SQL;

CREATE TABLE HOME.valid\_patients\_lookup

(INDEX=(INDI\_DSCM\_NO)) AS

SELECT

INDI\_DSCM\_NO,

/\* Calculate treatment end yearmonth \*/

CASE

WHEN INT(MOD(TRTFN\_DATE, 10000) / 100) = 12

THEN PUT(INT(TRTFN\_DATE / 10000) + 1, 4.) || '01'

ELSE PUT(INT(TRTFN\_DATE / 10000), 4.) ||

PUT(INT(MOD(TRTFN\_DATE, 10000) / 100) + 1, Z2.)

END AS TRT\_END\_YEARMONTH LENGTH=6,

/\* Demographics for final dataset \*/

EPISODE, SEX\_TYPE, AGE\_NHIS, RSLT\_FN, SIGUNGU, SIDO,

GAIBJA\_TYPE, INC5, TBCD\_KCDA, xray1, smear1, cul1, xpert1

FROM COHORT.tblndb

WHERE TRTFN\_DATE IS NOT NULL

AND INDI\_DSCM\_NO IS NOT NULL;

QUIT;

/\* ============================================

STEP 2: Union all monthly tables with SQL

SQL UNION is efficient for 2-3M total records

Filters patients and calculates quarters in one pass

============================================ \*/

PROC SQL;

CREATE TABLE HOME.visits\_with\_quarters AS

SELECT DISTINCT

t.INDI\_DSCM\_NO,

t.YEAR\_MONTH,

t.CMN\_KEY,

/\* Calculate quarter immediately \*/

INT(INTCK('MONTH',

INPUT(CATS(p.TRT\_END\_YEARMONTH, '01'), YYMMDD8.),

INPUT(CATS(t.YEAR\_MONTH, '01'), YYMMDD8.)

) / 3) AS quarter

FROM (

/\* Union all monthly tables \*/

%MACRO gen\_union;

%LOCAL year month ym tablename first;

%LET first = 1;

%DO year = 2010 %TO 2017;

%DO month = 1 %TO 12;

%LET ym = &year.%SYSFUNC(PUTN(&month, Z2.));

%LET tablename = HOME.t20\_&ym;

%IF %SYSFUNC(EXIST(&tablename)) %THEN %DO;

%IF &first = 0 %THEN %DO;

UNION ALL

%END;

%LET first = 0;

SELECT INDI\_DSCM\_NO, CMN\_KEY, "&ym" AS YEAR\_MONTH

FROM &tablename

WHERE FORM\_CD IN ('03', '08')

%END;

%END;

%END;

%MEND;

%gen\_union

) AS t

INNER JOIN HOME.valid\_patients\_lookup AS p

ON t.INDI\_DSCM\_NO = p.INDI\_DSCM\_NO

AND t.YEAR\_MONTH >= p.TRT\_END\_YEARMONTH

WHERE CALCULATED quarter BETWEEN 0 AND 30;

QUIT;

/\* ============================================

STEP 3: Aggregate to patient-quarter level

============================================ \*/

PROC SQL;

CREATE TABLE HOME.quarterly\_visits\_agg AS

SELECT

INDI\_DSCM\_NO,

quarter,

COUNT(\*) AS hospital\_visits

FROM HOME.visits\_with\_quarters

GROUP BY INDI\_DSCM\_NO, quarter;

QUIT;

/\* ============================================

STEP 4: Fill missing quarters with zeros

Only for patients with 20+ quarters max

============================================ \*/

PROC SQL;

/\* Identify patients with sufficient follow-up \*/

CREATE TABLE HOME.eligible\_patients AS

SELECT

INDI\_DSCM\_NO,

MAX(quarter) AS max\_quarter

FROM HOME.quarterly\_visits\_agg

GROUP BY INDI\_DSCM\_NO

HAVING max\_quarter >= 20;

/\* Generate complete quarter grid (0-20) for eligible patients \*/

CREATE TABLE HOME.quarter\_grid AS

SELECT

p.INDI\_DSCM\_NO,

q.quarter

FROM HOME.eligible\_patients AS p

CROSS JOIN (

SELECT 0 AS quarter UNION ALL SELECT 1 UNION ALL SELECT 2 UNION ALL

SELECT 3 UNION ALL SELECT 4 UNION ALL SELECT 5 UNION ALL SELECT 6 UNION ALL

SELECT 7 UNION ALL SELECT 8 UNION ALL SELECT 9 UNION ALL SELECT 10 UNION ALL

SELECT 11 UNION ALL SELECT 12 UNION ALL SELECT 13 UNION ALL SELECT 14 UNION ALL

SELECT 15 UNION ALL SELECT 16 UNION ALL SELECT 17 UNION ALL SELECT 18 UNION ALL

SELECT 19 UNION ALL SELECT 20

) AS q;

/\* Merge with actual visits, fill zeros \*/

CREATE TABLE HOME.quarterly\_visits\_complete AS

SELECT

g.INDI\_DSCM\_NO,

g.quarter,

COALESCE(v.hospital\_visits, 0) AS hospital\_visits

FROM HOME.quarter\_grid AS g

LEFT JOIN HOME.quarterly\_visits\_agg AS v

ON g.INDI\_DSCM\_NO = v.INDI\_DSCM\_NO

AND g.quarter = v.quarter

ORDER BY g.INDI\_DSCM\_NO, g.quarter;

QUIT;

/\* ============================================

STEP 5: Final dataset with demographics

============================================ \*/

PROC SQL;

CREATE TABLE HOME.lcmm\_data AS

SELECT

q.INDI\_DSCM\_NO,

q.quarter AS time,

q.hospital\_visits,

p.EPISODE,

p.SEX\_TYPE,

p.AGE\_NHIS,

p.RSLT\_FN,

p.SIGUNGU,

p.SIDO,

p.GAIBJA\_TYPE,

p.INC5,

p.TBCD\_KCDA,

p.xray1,

p.smear1,

p.cul1,

p.xpert1

FROM HOME.quarterly\_visits\_complete AS q

INNER JOIN HOME.valid\_patients\_lookup AS p

ON q.INDI\_DSCM\_NO = p.INDI\_DSCM\_NO

ORDER BY q.INDI\_DSCM\_NO, q.quarter;

QUIT;

/\* ============================================

STEP 6: Summary statistics

============================================ \*/

PROC SQL;

CREATE TABLE HOME.quarterly\_summary AS

SELECT

COUNT(DISTINCT INDI\_DSCM\_NO) AS unique\_patients FORMAT=COMMA12.,

COUNT(DISTINCT time) AS quarters\_with\_data,

COUNT(\*) AS patient\_quarter\_records FORMAT=COMMA12.,

SUM(hospital\_visits) AS total\_visits FORMAT=COMMA12.,

MIN(hospital\_visits) AS min\_quarterly\_visits,

MAX(hospital\_visits) AS max\_quarterly\_visits,

MEAN(hospital\_visits) AS avg\_quarterly\_visits FORMAT=8.2,

SUM(CASE WHEN hospital\_visits = 0 THEN 1 ELSE 0 END) AS zero\_visit\_quarters FORMAT=COMMA12.

FROM HOME.lcmm\_data;

SELECT \* FROM HOME.quarterly\_summary;

QUIT;

/\* ============================================

STEP 7: Clean up intermediate datasets

============================================ \*/

PROC DATASETS LIBRARY=HOME NOLIST;

DELETE visits\_with\_quarters

quarterly\_visits\_agg

eligible\_patients

quarter\_grid

quarterly\_visits\_complete;

QUIT;