WisCALM 2022 – *E. coli* Assessment Parameter Documentation

EGAD # 3200-2020-28

A product of the State of Wisconsin Clean Water Act Water Quality Report to Congress

Nine Springs Creek, Wisconsin DNR

Ashley Beranek, Project Manager
Jacob Dickmann, SWIMS Database File Manager
Will Westbury, WATERS Database File Manager
Brian Tinberg, Systems Architect

WISCONSIN
DEPT. OF NATURAL RESOURCES

Contents

Parameter Name and Number

5 Year E-coli Assessment Impairment Flag	80427

Description

To protect humans from illness caused by bacterial contamination in surface waters during recreational contact new *Escherichia coli* (*E. coli*) criteria were approved in 2020 (**Table 1**). These criteria replace the previous Fecal coliform standards. There are two separate *E. coli* criteria, a Geometric Mean (GM) criterion and a Statistical Threshold Value (STV) criterion. The use of both GM and STV criteria protects against spikes in bacterial densities while allowing for natural variation in water quality. These criteria apply to lakes, reservoirs, impounded flowing waters, streams, rivers, inland beaches, and Great Lake beaches.

To assess the attainment of recreational uses in Wisconsin's waters, WDNR uses all data collected during the recreation season, defined as Memorial Day (last Monday of May) to Labor Day (first Monday of September), over the past five years for each station with data. For each year's recreation season 90-day rolling periods are used to compare collected data to the criteria. In each rolling 90-day period the GM of samples and percent of samples exceeding STV criteria are calculated. If one or more 90-day GM or % above STV exceed criteria then the water is considered impaired for *E. coli*. Delisting will occur when there are no 90-day GM or % above STV exceeding criteria. WDNR believes this is an appropriate way of recognizing chronic risk to human health associated with recreational activities in water with long-term elevated levels of E. coli.

Data Sources, Storage

This parameter is collected by DNR staff, volunteers, and by members of other organizations at high, medium, and low priority beaches throughout the summer season. These data are collected in the field as per the protocols listed below and then analyzed at certified laboratories, typically the State Lab of Hygiene (SLOH). The results are then loaded from the labs into the SWIMS database.

Methods and procedures to collect data.

- Wisconsin Great Lakes Beach Monitoring Program Quality Assurance Protection Plan
- Long Term Trend – Rivers
- Long Term Trend – Streams
- Lake Sampling Procedures – LTT Water Quality

Data Entry

E. coli data analyzed by the SLOH are sent to SWIMS via the Lab Data Entry System. If connections are established with other laboratories, data can be sent to SWIMS via those connections. A spreadsheet batch upload process may be utilized to enter *E. coli* data to SWIMS.

Presentation of Results

Data is in Water Condition Viewer

How does parameter fit into the existing multi-part assessment categorization process?

- The parameter may be assessed independently from other parameters.

Assessment Package Logic

Assessment Methodology Description (excerpted from WisCALM 2022)

Data Requirements

a) *Period of record.* Data from the most recent 5 years is given preference as it is more representative of current conditions.

b) *Sampling Frequency and Seasonal Range.* The recreational period in Wisconsin is considered Memorial Day (last Monday of May) to Labor Day (first Monday of September).

Within the recreational period a minimum number of values measured on separate calendar days within a 90-day period are required for comparison to the criteria. A minimum of 5 samples are required for comparison to the Geomean criterion; a minimum of 11 samples are required for comparison to the STV criterion (Table 22).

c) *Measurement Depth. E. coli* should be measured several inches below the surface, following *E. coli* collection protocols for each waterbody type. In lakes the maximum depth to use is 6 ft (2 m).

Calculations

a) *Calculations – Geometric Mean Criterion.* Calculate the geometric mean for each 90-day rolling period with distinct datasets (different set of data by even one value).

b) *Calculations – Statistical Threshold Value (STV) Criterion.* Calculate the percent of values that exceed the STV criterion for each 90-day rolling period with distinct datasets (different set of data by even one value).

Application

a) *Exceedance Frequency.* Exceedance of the Geometric Mean criterion in any 90-day rolling period indicates impairment. Exceedance of the STV criterion by more than 10% in any given 90-day period indicates impairment. Listing for *E. coli* occurs when either or both criteria are exceeded.

All samples that meet data requirements will be used unless determined to be unrepresentative by the regional biologists (best professional judgement, BPJ). Enforcement samples (e.g. manure spill or

sewerage overflow) will be taken into consideration when reviewing potential *E. coli* listings on a case by case basis. Use of enforcement samples does not preclude listing as it may be a chronic issue.

Table 1. The two criteria for *E. coli* in NR102 Wis. Adm. Code**.**

E. coli (counts[1] per 100 mL)	
Geometric Mean	Statistical Threshold Value
126	410

1. For determining attainment or compliance, counts are considered equivalent to either colony forming units (CFU) or most probable number (MPN).

Assessment Package Steps

1. Select representative stations and sample results.
    a. Query database for stations using the following criteria:
        i. Station Type = LAKE, RESERVOIR, RIVER/STREAM, RIVERINE IMPOUNDMENT (Use the station selection used for TP packages.)
        ii. Station Type = GREAT LAKE-BEACH; LAKE-BEACH; RIVER, STREAM-BEACH; RIVERINE IMPOUNDMENT-BEACH;
        OR Station Type = GREAT LAKE; GREAT LAKE-BAY/HARBOR; LAKE; RESERVOIR; RIVERINE IMPOUNDMENT; RIVER, STREAM AND Field Description = BEACH, OR Station Name includes the word "beach"
    b. From the selected stations, Select *E. coli* datasets using the DNR Parameter codes 98929, 98930, 98933, 99069, 99132, 99188, 99743, 99824, 99826, 99962, and 99964
    c. From the *E. coli* datasets, use only the samples collected during the recreation season and five-year assessment period
        i. Recreation season is from the last Monday in May to the first Monday in September.
        ii. Five-year assessment period for 2022 cycle is from 2016-2020.
    d. Sample depth: depth can be null, but any values greater than 6 ft (2 m) should exclude the sample from the analysis.

2. Edit datasets that include replicates and blanks and censored data (i.e. result values above/below limits of quantitation (LOQ))
    a. Remove replicates and blanks
        i. Multiple samples collected from the same station on the same data and time should be considered "replicate" samples. Replicate samples are collected for the purpose of measuring the amount of sampling error or sample variability, and all but one replicate should be removed from datasets to be assessed using this assessment tool. Randomly select the replicate value that is kept.
        ii. Samples that are identified as "blanks," which are collected for the purpose of quality assurance, should be removed from the datasets to be assessed using this assessment tool.
    b. Edit censored data (i.e. values reported as less or greater than the LOQ)
        i. For result values that are less than the LOQ (e.g. < 1.0, or < 10.0), which are results stored as text, convert to numerical value and divide by two. Replace the LOQ value with the calculated quotient.
        ii. For result values that are greater than the LOQ (e.g. > 2400, or > 240000), which are results stored as text, convert to numerical value and multiply by two. Replace the LOQ value with the calculated product.

3. Calculation A: Geometric mean calculation in any given 90-day rolling period during the recreation season.
    a. For each station there need to be at least 5 samples (5 days) within a 90-day period to calculate a geomean.
    b. For each station there may be anywhere from 0 to multiple geomeans.
    c. If a 90-day period uses the same dataset as a previous 90-day period, skip that calculation.
    d. Flag (Y/N) calculation if it contains Enforcement Samples.

4. Comparison A: Compare geometric mean values against the *E. coli* geometric mean criterion (126 counts/100ml)
    a. At least one 90-day geometric mean value is required to assess a station.
    b. If all available geometric mean values for a particular station are less than or equal to the criterion (≤126 counts/100ml), then the station is meeting the criterion (i.e. Impaired=N).
    c. If one or more geometric mean values for a particular station are greater than the criterion (≤126 counts /100ml), then the station is exceeding the criterion (i.e. Impaired=Y).

5. Calculation B: Percent of samples in each 90-day rolling period exceeding Statistical Threshold Value (STV) *E. coli* criterion (410 counts/100ml).
    a. For each station there need to be at least 11 samples within a 90-day period to compare against criteria.
    b. For each station there may be anywhere from 0 to multiple 90-day periods.
    c. For each 90-day period, calculate the percentage of samples above the criterion of 410 counts /100ml. Use all samples within the 90-day period.
    d. If a 90-day period uses the same dataset as a pervious 90-day period, skip that calculation.
    e. Flag (Y/N) calculation if it contains Enforcement Samples.

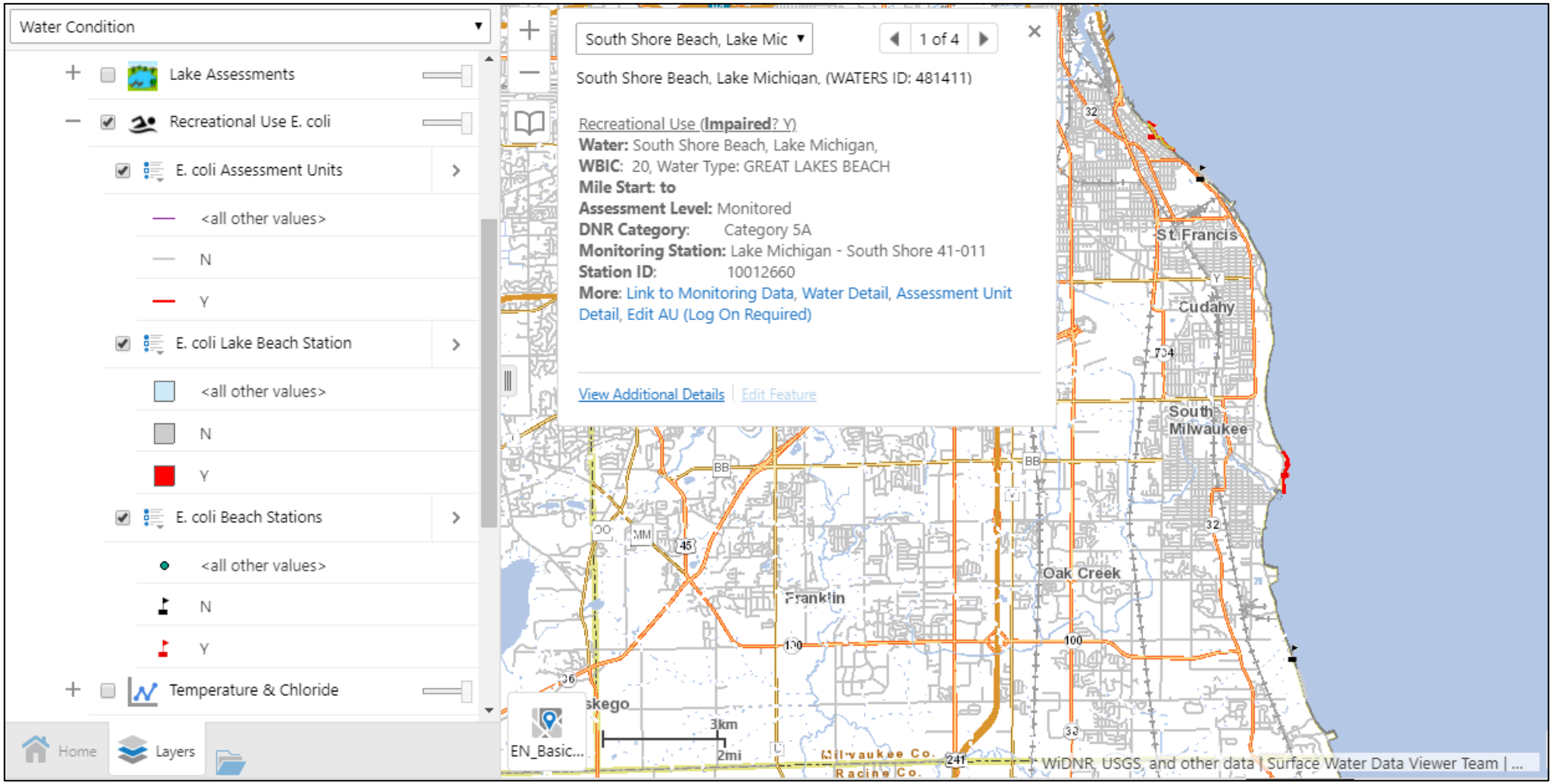
6. Comparison B: Compare percentage of samples above STV to the *E. coli* STV criteria (no more than 10% of samples above 410 counts/100ml)
    a. At least one 90-day period with minimum data required (11 samples) is needed to assess a station.
    b. If all 90-day periods for a particular station have ≤ 10% values exceeding, then the station is meeting the criterion (i.e. Impaired=N).
    c. If one or more 90-day period for a particular station have > 10% values exceeding, then the station is exceeding the criterion (i.e. Impaired=Y).

7. Table of calculations at station level:
    a. Station-level calculation export should be generated.

8. Station summary result output:
    a. A summary of all assessed monitoring stations should be generated.
    b. If a station is impaired for either the geomean comparison or the STV comparison, then the AU should be identified as Impaired.

9. Summary of assessment results by assessment unit.

View of Water Condition Viewer report for beach *E. coli.*



Assessment Package Code

The first assessment package piece gathers the raw data that meets the assessment requirements: generate view: WT_SWIMS_RIVER_TEMP_RESULT2_V.

```
/* Formatted on 9/10/2020 2:01:29 PM (QP5 v5.326) */
CREATE OR REPLACE FORCE VIEW W07510.WT_SWIMS_ECOLI_RESULT_V
(
   ECOLI_RESULT_SEQ_NO,
   RESULT_DATE_TIME,
   MONIT_STATION_SEQ_NO,
   DNR_PARAMETER_TYPE,
   DNR_PARAMETER_CODE,
   RESULT_VALUE_NO,
   RESULT_AMT,
   ECOLI_RESULT_AMT,
   RESULT_UNITS_TEXT,
   RESULT_QUALIFIER_CODE,
   LOQ_AMT,
   LOD_AMT,
   CREATE_DATE,
   CREATE_USER_ID,
   LAST_UPDATE_DATE,
   LAST_UPDATE_USER_ID,
   STATION_ID,
   PRIMARY_STATION_NAME,
   STATION_TYPE_CODE,
   SECONDARY_STATION_TYPE,
   ASSESSMENT_UNIT_SEQ_NOS
)
BEQUEATH DEFINER
AS
   SELECT ECR.ECOLI_RESULT_SEQ_NO
             ECOLI_RESULT_SEQ_NO,
          ECR.RESULT_DATE_TIME
             RESULT_DATE_TIME,
          ECR.MONIT_STATION_SEQ_NO
             MONIT_STATION_SEQ_NO,
          ECR.DNR_PARAMETER_TYPE
             DNR_PARAMETER_TYPE,
          ECR.DNR_PARAMETER_CODE
             DNR_PARAMETER_CODE,
          ECR.RESULT_VALUE_NO
             RESULT_VALUE_NO,
          ECR.RESULT_AMT
             RESULT_AMT,
          CASE
             WHEN SUBSTR (result_value_no, 1, 1) = '<' THEN result_amt / 2
             WHEN SUBSTR (result_value_no, 1, 1) = '>' THEN result_amt * 2
             ELSE result_amt
          END,
          ECR.RESULT_UNITS_TEXT
```

```
      RESULT_UNITS_TEXT,
   ECR.RESULT_QUALIFIER_CODE
      RESULT_QUALIFIER_CODE,
   ECR.LOQ_AMT
      LOQ_AMT,
   ECR.LOD_AMT
      LOD_AMT,
   ECR.CREATE_DATE
      CREATE_DATE,
   ECR.CREATE_USER_ID
      CREATE_USER_ID,
   ECR.LAST_UPDATE_DATE
      LAST_UPDATE_DATE,
   ECR.LAST_UPDATE_USER_ID
      LAST_UPDATE_USER_ID,
   WSN.STATION_ID
      STATION_ID,
   WSN.PRIMARY_STATION_NAME
      PRIMARY_STATION_NAME,
   WSN.STATION_TYPE_CODE
      STATION_TYPE_CODE,
   WSN.SECONDARY_STATION_TYPE
      SECONDARY_STATION_TYPE,
   pk_wt510_util.f_get_isect_key_by_station (
      ecr.monit_station_seq_no,
      'AU')
FROM WT_SWIMS_ECOLI_RESULT ECR, WT_SWIMS_MONIT_STATION WSN
WHERE ecr.monit_station_seq_no = wsn.monit_station_seq_no;

COMMENT ON TABLE W07510.WT_SWIMS_ECOLI_RESULT_V IS 'View created to run the ecoli
assessment package for integrated reporting. This package is updated every two years.';
```

The second assessment package piece calculates the geomean and STV exceedance rate within the available 90-day rolling day periods for a station: generate view: WT_SWIMS_ECOLI_STA_CALC2_V.

```
/* Formatted on 9/10/2020 2:19:12 PM (QP5 v5.326) */
CREATE OR REPLACE FORCE VIEW W07510.WT_SWIMS_ECOLI_STA_CALC2_V
(
    BEACH_CODE,
    MONIT_STATION_SEQ_NO,
    STATION_ID,
    PRIMARY_STATION_NAME,
    WINDOW_START_DATE,
    WINDOW_END_DATE,
    EARLIEST_SAMPLE_DATE,
    LATEST_SAMPLE_DATE,
    RESULT_CNT,
    MIN_DATA_REQ_GEOMEAN_FLAG,
    MIN_DATA_REQ_STV_FLAG,
    GEOMETRIC_MEAN_AMT,
    GEOMEAN_EXCEEDING_FLAG,
```

```
    PERCENT_EXCEEDING_STV,
    STV_EXCEEDING_FLAG,
    HAS_ENFORCEMENT_SAMPLES_FLAG
)
BEQUEATH DEFINER
AS
    WITH
        years
        AS
            (SELECT DISTINCT
                    TRUNC (result_date_time, 'YEAR')     AS year_date
               FROM wt_swims_ecoli_result2_v),
        seasons
        AS                                    /* memorial day thru labor day */
            (SELECT NEXT_DAY (
                        TO_DATE ('0524' || TO_CHAR(year_date, 'yyyy'),
                                 'mmddyyyy'),
                        'MONDAY')
                        AS season_start_date,
                    NEXT_DAY (
                        TO_DATE ('0831' || TO_CHAR(year_date, 'yyyy'),
                                 'mmddyyyy'),
                        'MONDAY')
                        AS season_end_date,
                      NEXT_DAY (
                          TO_DATE ('0831' || TO_CHAR(year_date, 'yyyy'),
                                   'mmddyyyy'),
                          'MONDAY')
                    - NEXT_DAY (
                          TO_DATE ('0524' || TO_CHAR(year_date, 'yyyy'),
                                   'mmddyyyy'),
                          'MONDAY')
                    - 88
                        AS window_cnt
               FROM years),
        windows
        AS                              /* all 90-day windows within the season */
            (SELECT season_start_date + add_days        AS window_start_date,
                    season_start_date + add_days + 89   AS window_end_date
               FROM seasons
                    JOIN (    SELECT LEVEL - 1     AS add_days
                                FROM DUAL
                          CONNECT BY LEVEL <= 25) /* 17 is the most windows we can
have in a Memorial Day to Labor Day season; any number larger than that will work here */
                        ON add_days < window_cnt),
        station_window_summary
        AS /* dates and result counts for windows in which a station had results */
            (  SELECT beach_code,
                      monit_station_seq_no,
                      w.window_start_date,
                      w.window_end_date,
                      MIN (TRUNC (r.result_date_time))     AS min_result_date,
                      COUNT (*)                            AS result_cnt
                 FROM wt_swims_ecoli_result2_v r
                      JOIN windows w
```

```
                        ON TRUNC (r.result_date_time) BETWEEN
w.window_start_date
                                                    AND w.window_end_date
             GROUP BY beach_code,
                      monit_station_seq_no,
                      w.window_start_date,
                      w.window_end_date),
        station_windows
        AS                                          /* windows of unique result sets */
            (  SELECT beach_code,
                      monit_station_seq_no,
                      MIN (window_start_date)     AS window_start_date,
                      MIN (window_end_date)       AS window_end_date,
                      min_result_date,
                      result_cnt
                 FROM station_window_summary
             GROUP BY beach_code,
                      monit_station_seq_no,
                      min_result_date,
                      result_cnt),
        station_geomeans
        AS                  /* calculate geometric means for each qualifying station window */
            (  SELECT w.beach_code,
                      r.monit_station_seq_no,
                      r.station_id,
                      r.primary_station_name,
                      w.window_start_date,
                      w.window_end_date,
                      MIN (TRUNC (r.result_date_time))
                          AS earliest_sample_date,
                      MAX (TRUNC (r.result_date_time))
                          AS latest_sample_date,
                      COUNT (*)
                          AS result_cnt,
                      CASE WHEN COUNT (*) >= 5 THEN 'Y' ELSE 'N' END
                          AS min_data_req_geomean_flag,
                      CASE WHEN COUNT (*) >= 11 THEN 'Y' ELSE 'N' END
                          AS min_data_req_stv_flag,
                      POWER (
                          10,
                            SUM (LOG (10, r.ecoli_result_amt))
                          / COUNT (r.ecoli_result_amt))
                          AS geometric_mean_amt,
                        100
                      * SUM (
                            CASE
                                WHEN r.exceeds_stv_flag = 'Y' THEN 1
                                ELSE 0
                            END)
                      / COUNT (*)
                          AS percent_exceeding_stv,
                      MAX (r.enforcement_flag)
                          AS has_enforcement_samples_flag
                 FROM wt_swims_ecoli_result2_v r
                      JOIN station_windows w
                          ON     r.monit_station_seq_no =
```

```
                                    w.monit_station_seq_no
                                AND TRUNC (r.result_date_time) BETWEEN
w.window_start_date
                                                                AND w.window_end_date
             GROUP BY w.beach_code,
                      r.monit_station_seq_no,
                      r.station_id,
                      r.primary_station_name,
                      w.window_start_date,
                      w.window_end_date)
    SELECT beach_code,
           monit_station_seq_no,
           station_id,
           primary_station_name,
           window_start_date,
           window_end_date,
           earliest_sample_date,
           latest_sample_date,
           result_cnt,
           CASE WHEN result_cnt >= 5 THEN 'Y' ELSE 'N' END
               AS min_data_req_geomean_flag,
           CASE WHEN result_cnt >= 11 THEN 'Y' ELSE 'N' END
               AS min_data_req_stv_flag,
           geometric_mean_amt,
           CASE WHEN geometric_mean_amt > 126 THEN 'Y' ELSE 'N' END
               AS geomean_exceeding_flag,
           percent_exceeding_stv,
           CASE WHEN percent_exceeding_stv > 10 THEN 'Y' ELSE 'N' END
               AS stv_exceeding_flag,
           has_enforcement_samples_flag
      FROM station_geomeans;
```

The third assessment package piece is a station summary of the number of exceedances: generate view: WT_ ECOLI_STATION_SUMMARY2_V.

```
* Formatted on 9/10/2020 2:47:33 PM (QP5 v5.326) */
CREATE OR REPLACE FORCE VIEW W23321.WT_ECOLI_STATION_SUMMARY2_V
(
   BEACH_CODE,
   MONIT_STATION_SEQ_NO,
   STATION_ID,
   PRIMARY_STATION_NAME,
   EARLIEST_SAMPLE_DATE,
   LATEST_SAMPLE_DATE,
   GEOMEAN_QUALIFYING_CNT,
   GEOMEAN_EXCEEDING_CNT,
   STV_QUALIFYING_CNT,
   STV_EXCEEDING_CNT,
   RECENT_DATA_FLAG,
   HAS_ENFORCEMENT_SAMPLES_FLAG,
   ASSESSMENT_UNIT_SEQ_NO,
   IMPAIRED_FLAG,
   USE_FOR_ASSESSMENT_FLAG
```

```
)
BEQUEATH DEFINER
AS
   WITH
      max_sample_year
      AS
         (SELECT   ADD_MONTHS (TRUNC (MAX (result_date_time), 'YEAR'), 12)
               - 1
                  AS last_day_of_year
           FROM w07510.wt_swims_ecoli_result2),
      station_au
      AS
         (SELECT e.beach_code,
                 e.monit_station_seq_no,
                 x.assessment_unit_seq_no
            FROM w07510.wt_swims_ecoli_result2  e
                 JOIN wt_waters_ecoli_station_au x
                    ON e.monit_station_seq_no = x.monit_station_seq_no
           WHERE e.beach_code = 'BEACH'
          UNION
          SELECT e.beach_code,
                 e.monit_station_seq_no,
                 s.assessment_unit_seq_no
            FROM w07510.wt_swims_ecoli_result2  e
                 JOIN
                 (SELECT DISTINCT
                         tpms.monit_station_seq_no,
                         tpms.assessment_unit_seq_no
                    FROM wt_tp_monit_station  tpms
                         JOIN wt_assessment_unit au
                            ON tpms.assessment_unit_seq_no =
                               au.assessment_unit_seq_no
                   WHERE tpms.include_flag = 'Y' AND au.status_code = 'A')
                 s
                    ON e.monit_station_seq_no = s.monit_station_seq_no
           WHERE e.beach_code = 'NON_BEACH'),
      station_summary
      AS
         (  SELECT sc.beach_code,
                   sc.monit_station_seq_no,
                   sc.station_id,
                   sc.primary_station_name,
                   MIN (sc.earliest_sample_date)
                      AS earliest_sample_date,
                   MAX (sc.latest_sample_date)
                      AS latest_sample_date,
                   SUM (
                      CASE
                         WHEN sc.min_data_req_geomean_flag = 'Y' THEN 1
                         ELSE 0
                      END)
```

```
            AS geomean_qualifying_cnt,
          SUM (
            CASE
              WHEN     sc.min_data_req_geomean_flag = 'Y'
                  AND sc.geomean_exceeding_flag = 'Y'
              THEN
                1
              ELSE
                0
            END)
            AS geomean_exceeding_cnt,
          SUM (
            CASE
              WHEN sc.min_data_req_stv_flag = 'Y' THEN 1
              ELSE 0
            END)
            AS stv_qualifying_cnt,
          SUM (
            CASE
              WHEN     sc.min_data_req_stv_flag = 'Y'
                  AND sc.stv_exceeding_flag = 'Y'
              THEN
                1
              ELSE
                0
            END)
            AS stv_exceeding_cnt,
          CASE
            WHEN MAX (sc.latest_sample_date) >=
                   ADD_MONTHS (max_sample_year.last_day_of_year,
                               -24)
                 + 1
            THEN
              'Y'
            ELSE
              'N'
          END
            AS recent_data_flag,
          sc.has_enforcement_samples_flag
        FROM w07510.wt_swims_ecoli_sta_calc2_v sc
          CROSS JOIN max_sample_year
       WHERE (   sc.min_data_req_geomean_flag = 'Y'
              OR sc.min_data_req_stv_flag = 'Y')
    GROUP BY sc.beach_code,
          sc.monit_station_seq_no,
          sc.station_id,
          sc.primary_station_name,
          max_sample_year.last_day_of_year,
          sc.has_enforcement_samples_flag)
SELECT station_summary.beach_code,
    station_summary.monit_station_seq_no,
```

```
        station_summary.station_id,
        station_summary.primary_station_name,
        station_summary.earliest_sample_date,
        station_summary.latest_sample_date,
        station_summary.geomean_qualifying_cnt,
        station_summary.geomean_exceeding_cnt,
        station_summary.stv_qualifying_cnt,
        station_summary.stv_exceeding_cnt,
        station_summary.recent_data_flag,
        station_summary.has_enforcement_samples_flag,
        station_au.assessment_unit_seq_no,
        CASE
           WHEN geomean_exceeding_cnt > 0 OR stv_exceeding_cnt > 0
           THEN
              'Y'
           ELSE
              'N'
        END
           AS impaired_flag,
     /* this CASE is the use_for_assessment_flag. At most one station per AU will have this flag equal to Y.
           So, we're going to use the row_number() function to find the first row where we're partitioning
           by the AU and ordering by stations with the most desirable criteria, which is:

1) impaired stations over non-impaired
2) stations with recent data
3) most geomeans exceeding (for impaired stations) or most geomeans not exceeding (for non-impaired
stations)
4) most STV exceeding (for impaired stations) or most STV not exceeding (for non-impaired stations)
        */
        CASE ROW_NUMBER ()
              OVER (
                 PARTITION BY station_au.assessment_unit_seq_no
                 ORDER BY
                    CASE
                       WHEN    geomean_exceeding_cnt > 0
                          OR stv_exceeding_cnt > 0
                       THEN
                          1
                       ELSE
                          2
                    END,
                    CASE station_summary.recent_data_flag
                       WHEN 'Y' THEN 1
                       ELSE 2
                    END,
                    CASE
                       WHEN    geomean_exceeding_cnt > 0
                          OR stv_exceeding_cnt > 0
                       THEN
                          geomean_exceeding_cnt
                       ELSE
```

```
                    geomean_qualifying_cnt
                  - geomean_exceeding_cnt
            END DESC,
            CASE
               WHEN    geomean_exceeding_cnt > 0
                  OR stv_exceeding_cnt > 0
               THEN
                  stv_exceeding_cnt
               ELSE
                  stv_qualifying_cnt - stv_exceeding_cnt
            END DESC)
         WHEN 1
         THEN
            'Y'
         ELSE
            'N'
      END
         AS use_for_assessment_flag
  FROM station_summary
      JOIN station_au
         ON     station_summary.beach_code = station_au.beach_code
           AND station_summary.monit_station_seq_no =
               station_au.monit_station_seq_no;
```

The fourth assessment package piece is an assessment unit summary: generate view: WT_ ECOLI_AU_STATION_SUMMARY2_V.

```
/* Formatted on 9/10/2020 3:23:11 PM (QP5 v5.326) */
CREATE OR REPLACE FORCE VIEW W23321.WT_ECOLI_AU_STATION_SUMMARY2_V
(
   WBIC,
   OFFICIAL_NAME,
   LOCAL_WATERBODY_NAME,
   ASSESSMENT_UNIT_SEQ_NO,
   START_MILE_NO,
   END_MILE_NO,
   WATER_TYPE_SIZE_AMT,
   BEACH_CODE,
   STATION_ID,
   PRIMARY_STATION_NAME,
   MONIT_STATION_SEQ_NO,
   EARLIEST_SAMPLE_DATE,
   LATEST_SAMPLE_DATE,
   GEOMEAN_QUALIFYING_CNT,
   GEOMEAN_EXCEEDING_CNT,
   STV_QUALIFYING_CNT,
   STV_EXCEEDING_CNT,
   IMPAIRED_FLAG
)
BEQUEATH DEFINER
AS
```

```
SELECT au.wbic,
       rw.official_name,
       au.local_waterbody_name,
       au.assessment_unit_seq_no,
       au.start_mile_no,
       au.end_mile_no,
       au.water_type_size_amt,
       ss.beach_code,
       station_id,
       ss.primary_station_name,
       ss.monit_station_seq_no,
       ss.earliest_sample_date,
       ss.latest_sample_date,
       ss.geomean_qualifying_cnt,
       ss.geomean_exceeding_cnt,
       ss.stv_qualifying_cnt,
       ss.stv_exceeding_cnt,
       ss.impaired_flag
  FROM wt_ecoli_station_summary2_v  ss
       JOIN wt_assessment_unit au
          ON ss.assessment_unit_seq_no = au.assessment_unit_seq_no
       LEFT OUTER JOIN wl_row_waterbody_mv rw ON au.wbic = rw.wbic
 WHERE ss.use_for_assessment_flag = 'Y';
```