Data Warehouse Queries

The queries provided in this document are used to exemplify how the data warehouse could be used, and to provide a starting point for creating queries against the CoursEval data warehouse.

Contents
Survey Fact .................................................................................................................................................. 2
Survey Assessment Data ............................................................................................................................ 2
Survey Submissions by Person ..................................................................................................................... 2
Student Submissions by Gender .................................................................................................................... 3
Missed Surveys by Person ............................................................................................................................ 3
Courses by Period ......................................................................................................................................... 3
Courses Across Periods .................................................................................................................................. 3
Survey Likert Responses .............................................................................................................................. 4
Survey Likert Responses – Score Leveler ...................................................................................................... 5
Survey Likert Responses for Assessed Instructors ....................................................................................... 5
**Survey Fact**

Show enough information to lookup the survey in CoursEval. Include the entire fact table.

```sql
SELECT s.CESurveyId, s.surveyName, s.SurveyType, 
p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester, 
sf.* 
FROM SurveyFact sf 
JOIN SurveyDim s 
    ON s.SurveyId = sf.SurveyId 
JOIN PeriodDim p 
    ON p.PeriodId = sf.PeriodId
```

**Survey Assessment Data**

Show each survey assessment defined by Survey, Course, Block, Site.

```sql
SELECT s.CESurveyId, s.surveyName, s.SurveyType, 
p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester, 
c.CourseId, c.CourseNumber, c.CourseName, c.CourseSection, 
bl.BlockCode, st.RotationSiteName, 
sf.* 
FROM SurveyAssessmentFact sf 
JOIN SurveyDim s 
    ON s.SurveyId = sf.SurveyId 
JOIN PeriodDim p 
    ON p.PeriodId = sf.PeriodId 
JOIN CourseDim c 
    ON c.CourseId = sf.CourseId 
JOIN BlockDim bl 
    ON bl.BlockId = sf.BlockId 
JOIN RotationSiteDim st 
    ON st.RotationSiteId = sf.RotationSiteId 
ORDER BY p.PeriodYear, p.PeriodName, 
s.surveyName, c.CourseName, bl.BlockCode, st.RotationSiteName
```

**Survey Submissions by Person**

This example shows submissions by person with lifetime statistics.

```sql
SELECT psn.PersonId, psn.FullName, psn.PrimaryRole, 
    COUNT(*) AS TotalSubmits, 
    AVG(TotalQuestionsAnswered) AS AvgQ, 
    AVG(TotalNumericQuestionsAnswered) AS AvgNumericQ, 
    AVG(TotalTextQuestionsAnswered) AS AvgTextQ, 
    AVG(DaysFromStart) AS AvgDaysFromStart, 
    AVG(DaysLeft) AS AvgDaysLeft 
FROM SubmittedSurveyFact sub 
JOIN PersonDim psn 
    ON psn.PersonId = sub.PersonEvaluatorId 
GROUP BY 1,2,3
```
Data Warehouse Queries

Student Submissions by Gender
Show submissions by gender from students only where the survey focus is students assessing faculty.

```
SELECT dem.gender,
       COUNT(*) AS TotalSubmits,
       AVG(TotalQuestionsAnswered) AS AvgQ,
       AVG(TotalNumericQuestionsAnswered) AS AvgNumericQ,
       AVG(TotalTextQuestionsAnswered) AS AvgTextQ,
       AVG(DaysFromStart) AS AvgDaysFromStart,
       AVG(DaysLeft) AS AvgDaysLeft
FROM SubmittedSurveyFact sub
JOIN DemographicDim dem
     ON dem.DemographicId = sub.DemographicId
JOIN SurveyDim s
     ON s.SurveyId = sub.SurveyId
WHERE s.SurveyFocusAbbrev = 'SF'
     AND s.SurveyType = 'Standard'
GROUP BY 1
```

Missed Surveys by Person
This example shows all missed surveys by person with lifetime statistics.

```
SELECT psn.PersonId, psn.FullName, psn.PrimaryRole,
       COUNT(*) AS TotalMissed,
       AVG(TotalDaysOpen) AS AvgDaysOpen
FROM MissedSurveyFact m
JOIN PersonDim psn
     ON psn.PersonId = m.PersonEvaluatorId
GROUP BY 1,2,3
```

Courses by Period
Count courses within a period.

```
SELECT p.PeriodName, COUNT(*) AS Courses
FROM CourseFact cf
JOIN PeriodDim p
     ON p.PeriodId = cf.PeriodId
GROUP BY 1
```

Courses Across Periods
Count courses by number and section. This shows courses that are re-used across periods.

```
SELECT c.CourseNumber, c.CourseSection, COUNT(*) AS Records
FROM CourseFact cf
JOIN CourseDim c
     ON c.CourseId = cf.CourseId
GROUP BY 1,2
ORDER BY 3 DESC
```
Survey Likert Responses
For a particular survey (note filter in WHERE clause), get count and average for each question. This will translate to N and Mean on CoursEval reports.

CoursEval always reports one-based results meaning response sets begin with 1. The DW also stores zero-based answers where the response set starts with 0. Show both 1 and 0-based mean.

```
SELECT COUNT(*) AS TotalAnswered,
       AVG(rl.ResponseValue) AS One_Based_Mean,
       AVG(rl.ResponseZeroValue) AS Zero_Based_Mean,
       q.QuestionId, q.Question, q.QuestionFocus, q.QuestionScope,
       Assessed.PersonId, Assessed.FullName, Assessed.PrimaryRole,
       s.SurveyId, s.CESurveyId, s.SurveyName, s.SurveyType,
       p.PeriodId, p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester,
       c.CourseId, c.CourseNumber, c.CourseName, c.CourseSection,
       bl.BlockCode, st.RotationSiteName,
       MAX(rs.ResponseSetName) AS ResponseSetName,
       MAX(rs.ResponseValueDefAbbrev) AS ResponsesDefinedAs
FROM ResponseLikertFact rl
JOIN SurveyDim s ON s.SurveyId = rl.SurveyId
JOIN PeriodDim p ON p.PeriodId = rl.PeriodId
JOIN CourseDim c ON c.CourseId = rl.CourseId
JOIN BlockDim bl ON bl.BlockId = rl.BlockId
JOIN RotationSiteDim st ON st.RotationSiteId = rl.RotationSiteId
JOIN PersonDim Assessed ON Assessed.PersonId = rl.PersonAssesseeId
JOIN QuestionDim q ON q.QuestionId = rl.QuestionId
JOIN ResponseSetDim rs ON rs.ResponseSetId = rl.ResponseSetId
WHERE rl.SurveyId = 14
GROUP BY q.QuestionId, q.Question, q.QuestionFocus, q.QuestionScope,
         Assessed.PersonId, Assessed.FullName, Assessed.PrimaryRole,
         s.SurveyId, s.CESurveyId, s.SurveyName, s.SurveyType,
         p.PeriodId, p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester,
         c.CourseId, c.CourseNumber, c.CourseName, c.CourseSection,
         bl.BlockCode, st.RotationSiteName
```
Survey Likert Responses – Score Leveler

For the same query above, level the scores to a 4-point scale. This will only work for 1-based results, not 0-based. The score leveler does not take into account the zero based scale for multipliers.

The score leveler dimension relies on the “scale to” to be a parameter (hard coded in this query). Join on the from scale (from ResponseSetDim) and from score (from ResponseLikertFact). The score leveler will always have scales 4, 5, & 10 to convert to regardless of the response sets in CoursEval. All defined response sets are setup to convert to every other defined response set in CoursEval.

```
SELECT COUNT(*) AS TotalAnswered,
       ROUND(AVG(rl.ResponseValue),3) AS Mean,
       rs.ResponseMaxScore AS MeanScale,
       ROUND(AVG(sl.ToScore),3) AS Leveled_Mean,
       4 AS LeveledToScale,
       q.QuestionId, q.QuestionAbbrev, q.QuestionFocus, q.QuestionScope,
       Assessed.PersonId, Assessed.FullName, Assessed.PrimaryRole,
       s.SurveyId, s.CESurveyId, s.SurveyName, s.SurveyType,
       p.PeriodId, p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester,
       c.CourseId, c.CourseNumber, c.CourseName, c.CourseSection,
       MAX(rs.ResponseSetName) AS ResponseSetName,
       MAX(rs.ResponseValueDefAbbrev) AS ResponsesDefinedAs
FROM ResponseLikertFact rl
JOIN SurveyDim s
  ON s.SurveyId = rl.SurveyId
JOIN PeriodDim p
  ON p.PeriodId = rl.PeriodId
JOIN CourseDim c
  ON c.CourseId = rl.CourseId
JOIN PersonDim Assessed
  ON Assessed.PersonId = rl.PersonAssesseeId
JOIN QuestionDim q
  ON q.QuestionId = rl.QuestionId
JOIN ResponseSetDim rs
  ON rs.ResponseSetId = rl.ResponseSetId
JOIN ScoreLevelerDim sl
  ON sl.FromScale = rs.ResponseMaxScore
       AND sl.FromScore = rl.ResponseValue
       AND sl.ToScale = 4
WHERE rl.SurveyId = 14
GROUP BY q.QuestionId, q.QuestionAbbrev, q.QuestionFocus, q.QuestionScope,
       Assessed.PersonId, Assessed.FullName, Assessed.PrimaryRole,
       s.SurveyId, s.CESurveyId, s.SurveyName, s.SurveyType,
       p.PeriodId, p.CEPeriodId, p.PeriodName, p.PeriodYear, p.Semester,
       c.CourseId, c.CourseNumber, c.CourseName, c.CourseSection
```

Survey Likert Responses for Assessed Instructors

Use the SurveyAssessmentBridge table to link instructors to course-based questions such that instructors in team taught courses are associated to:

1. Course question responses
2. People based question responses specifically for the instructor
Special filters:

1. Only include instructors for the course based questions if their CoursEval survey role is *Responsible Faculty* (RF).
2. Only include typical course evaluation surveys where students evaluate the course/instructor.
3. Use the person dimension placeholder for course based/overall questions (-4) to know when to apply the bridge to get the instructors.
4. The survey year is applied in the sample to narrow down the results.

```sql
SELECT
    p.PeriodAbbrev,
    s.SurveyName,
    c.CourseNumber,
    c.CourseSection,
    Assessed.LastNameFirst,
    q.Question,
    q.QuestionFocus,
    rs.ResponseDescription,
    rs.TotalResponseOptions AS Scale,
    rs.ResponseValue
FROM ResponseLikertFact rl
JOIN SurveyDim s
    ON s.SurveyId = rl.SurveyId
JOIN PeriodDim p
    ON p.PeriodId = rl.PeriodId
JOIN CourseDim c
    ON c.CourseId = rl.CourseId
JOIN SurveyAssessmentBridge bridge
    ON bridge.SurveyAssessmentFactId = rl.SurveyAssessmentFactId
JOIN PersonDim Assessed
    ON Assessed.PersonId = bridge.PersonAssesseeId
JOIN QuestionDim q
    ON q.QuestionId = rl.QuestionId
JOIN ResponseSetDim rs
    ON rs.ResponseSetId = rl.ResponseSetId
WHERE
    s.SurveyFocusAbbrev = 'SF'
    AND s.SurveyType = 'Standard'
    AND p.PeriodYear = 2012
    AND rl.PersonAssesseeId = -4
    AND bridge.SurveyRoleAbbrev = 'RF'
UNION ALL
SELECT
    p.PeriodAbbrev,
    s.SurveyName,
    c.CourseNumber,
    c.CourseSection,
    Assessed.LastNameFirst,
    q.Question,
    q.QuestionFocus,
    rs.ResponseDescription,
    rs.TotalResponseOptions AS Scale,
    rs.ResponseValue
FROM ResponseLikertFact rl
JOIN SurveyDim s
```
ON s.SurveyId = r1.SurveyId
JOIN PeriodDim p
  ON p.PeriodId = r1.PeriodId
JOIN CourseDim c
  ON c.CourseId = r1.CourseId
JOIN PersonDim Assessed
  ON Assessed.PersonId = r1.PersonAssesseeId
JOIN QuestionDim q
  ON q.QuestionId = r1.QuestionId
JOIN ResponseSetDim rs
  ON rs.ResponseSetId = r1.ResponseSetId
WHERE s.SurveyFocusAbbrev = 'SF'
AND s.SurveyType = 'Standard'
AND p.PeriodYear = 2012
AND r1.PersonAssesseeId <> -4