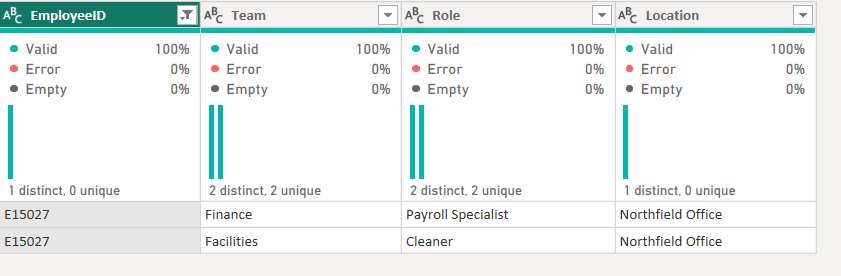
Based on the datasets, the goal is to support analytics for employee learning and development, including tracking behaviour, skill improvement, course engagement, and demographic insights. Since the structure is relatively de-normalised and need flexible slicing across time, teams, course types, and behaviour, **a snowflake schema** is most appropriate due to the need for normalized dimensions like Employee, Course, Role, Location and Team.

Employee-related info (Role, Team, and Location) can change over time and have multiple attributes. Normalizing these into separate dimension tables avoids data duplication, making maintenance easier.

**Note: Employee’s promotion or transfer date details are not available. So any trends over the time by role / team / location may not be accurate. We can assume those dates based on course enrolment date.**



**Fact table: fact\_courseengagement**

ID (PK)

EmployeeHistID (FK)

CourseID (FK)

FeedbackID (FK)

DeviceID (FK)

IsMandatory

EntrollmentDate

CompletionDate

CompletionStatus

Score

DurationMinutes

LastAccessDate

SkillScore\_Communication

SkillScore\_TechEfficiency

SkillScore\_Teamwork

SkillGapFlag

UsedScreenReader

**Measures:**

AvgFeedbackRating

AvgSkillScore\_Communication

AvgSkillScore\_Teamwork

AvgSkillScore\_TechEfficiency

TotalTimeSpent- in minutes

TotalMandatoryCoursesEnrolled

NumMandatoryCoursesCompleted

PercentofCompletedMandatoryCourse

OverdueTrainingsByEnrollmentDate

PercentageSkillGaps- Skill Gap Percentage

CompletionPercentage: Course completion Percentage

DistinctEmployeesFromFact – It counts distinct employees who have at least one fact record in the fact table

**Newly added Columns**

CommunicationScoreCategory- this will return category of score. Eg: 1-2: Low, 3-4: Medium, 5: High

TeamworkScoreCategory- this will return category of score. Eg: 1-2: Low, 3-4: Medium, 5: High

TechEfficiencyScoreCategory- this will return category of score. Eg: 1-2: Low, 3-4: Medium, 5: High

FormattedDuration- return course duration in hh:mm format

**Dimension Tables**

**dim\_employee :** EmployeeID (PK), FirstName, LastName, Email, JoinedDate

**dim\_role**: RoleID (PK), Role

**dim\_location**: LocationID (PK), Location

**dim\_team**: TeamID (PK), Team

**dim\_course**: CourseID (PK), CourseTitle,CourseProvider,CourseCategory

**dim\_employeeHistory**: EmployeeHistID (PK), EmployeeID (FK), RoleID(FK), LocationID (FK), TeamID (FK)

**dim\_date**: Date,Year,Month,Day,MonthName,Month\_Year,Sortcolumn

**dim\_device**: DeviceID (PK), AccessViaDevice

**dim\_feedback**: FeedbackID (PK), FeedbackComment,FeedbackRating

**Relationships**

|  |  |  |
| --- | --- | --- |
| Link | Relationship | Description |
| fact\_courseengagement.EmployeeHistID > dim\_employeehistory.EmployeeHistID | Many to One | Links training activity to the employee’s role/team/location at the time. |
| dim\_employeehistory.EmployeeID > dim\_employee.EmployeeID | Many to one | Connects employee history records to the employee master |
| fact\_courseengagement.CourseID > dim\_course.CourseID | Many to one | Links each training record to course info. |
| fact\_courseengagement.EntrollmentDate > dim\_date.Date | Many-to-One (Active) |  |
| fact\_courseengagement.CompletionDate > dim\_date.Date | Many-to-One (In Active) |  |
| fact\_courseengagement.LastAccessDate > dim\_date.Date | Many-to-One (In Active) |  |
| dim\_employeehistory.RoleID > dim\_role.RoleID | Many to one |  |
| dim\_employeehistory.TeamID > dim\_team.TeamID | Many to one |  |
| dim\_employeehistory.LocationID > dim\_location.LocationID | Many to one |  |
| fact\_courseengagement.DeviceID > dim\_device.DeviceID | Many to one |  |
| fact\_courseengagement.FeedbackID > dim\_feedback.FeedbackID | Many to one |  |

I’ve connected Dim tables to EmployeeHistory instead of directly to Fact, because role, team, location can change over time for an employee, and EmployeeHistory captures that timeline.

Coursecategorylookup table was created to get coursecategory from fact table. Merge query cannot be use directly to dim\_course and fact because of circular dependency.

Common date table is created for reusability across multiple date fields. It keeps model cleaner.

Second reason is unified filtering across all metrics.