

SQL Pipeline for Modeling Dataset

-- SQL Pipeline for Modeling Dataset

-- Average Exam Score with target pass/fail rate

```
CREATE OR REPLACE TABLE STUDENT_EXAM_SUMMARY AS
SELECT
    STUDENT_ID,
    ROUND(AVG(SCORE), 2) AS AVERAGE_SCORE,
    CASE WHEN AVG(SCORE) >= 40 THEN 1 ELSE 0 END AS PASS_FAIL_RATE
FROM
    EXAM_SCORES
GROUP BY
    STUDENT_ID;
```

-- Attendance Percentage

```
CREATE OR REPLACE TABLE STUDENT_ATTENDANCE_SUMMARY AS
SELECT
    STUDENT_ID,
    ROUND(100*SUM(CASE WHEN STATUS = 'Present' THEN 1 ELSE 0 END)/COUNT(*), 2) AS
ATTENDANCE_PERCENT
FROM
    ATTENDANCE
GROUP BY
    STUDENT_ID;
```

--Activities Features

```
CREATE OR REPLACE TABLE STUDENT_ACTIVITIES_SUMMARY AS
SELECT
    STUDENT_ID,
    CASE WHEN COUNT(*) > 0 THEN 1 ELSE 0 END AS ACTIVITY_FLAG,
    MAX(ACHIEVEMENT_LEVEL) AS HIGHEST_ACHIEVEMENT
FROM
    ACTIVITIES
GROUP BY
    STUDENT_ID;
```

-- Disciplinary Records

-- Get incident counts per type per student

```
CREATE OR REPLACE TABLE STUDENT_DISCIPLINE_SUMMARY AS
WITH INCIDENT_COUNTS AS(
```

```

SELECT
    STUDENT_ID,
    INCIDENT_TYPE,
    COUNT(*) AS INCIDENT_COUNT
FROM DISCIPLINARY_RECORDS
GROUP BY STUDENT_ID, INCIDENT_TYPE
),
PRIMARY_INCIDENT AS(
    SELECT
        STUDENT_ID,
        INCIDENT_TYPE AS PRIMARY_INCIDENT_TYPE
    FROM (
        SELECT
            STUDENT_ID,
            INCIDENT_TYPE,
            INCIDENT_COUNT,
            ROW_NUMBER() OVER (PARTITION BY STUDENT_ID ORDER BY INCIDENT_COUNT DESC,
INCIDENT_TYPE ASC) AS RN
        FROM INCIDENT_COUNTS
    )
    WHERE RN = 1
),
DISCIPLINE_SUMMARY AS(
    SELECT
        STUDENT_ID,
        COUNT(*) AS DISCIPLINE_COUNT,
        MAX(
            CASE SEVERITY
                WHEN 'Low' THEN 1
                WHEN 'Medium' THEN 2
                WHEN 'High' THEN 3
            END
        ) AS DISCIPLINE_SEVERITY
    FROM DISCIPLINARY_RECORDS
    GROUP BY STUDENT_ID
)
SELECT
    D.STUDENT_ID,
    D.DISCIPLINE_COUNT,
    D.DISCIPLINE_SEVERITY,
    P.PRIMARY_INCIDENT_TYPE
FROM DISCIPLINE_SUMMARY D
LEFT JOIN PRIMARY_INCIDENT P ON D.STUDENT_ID = P.STUDENT_ID;

```

--Teacher Features (Classes & Teachers)

```
CREATE OR REPLACE TABLE STUDENT_TEACHER_SUMMARY AS
SELECT
    C.STUDENT_ID,
    ROUND(AVG(T.TEACHER_RATING), 2) AS AVG_TEACHER_RATING,
    ROUND(AVG(T.YEARS_EXPERIENCE), 2) AS AVG_TEACHER_EXPERIENCE
FROM CLASSES C
JOIN TEACHERS T ON C.TEACHER_ID = T.TEACHER_ID
GROUP BY C.STUDENT_ID;
```

--Combining everything into final modeling dataset

```
CREATE OR REPLACE TABLE STUDENT_MODELING_DATASET AS
SELECT
    S.STUDENT_ID,
    S.GENDER,
    S.GRADE_LEVEL,
    S.PARENT_EDUCATION_LEVEL,
    S.FAMILY_INCOME,

    EX.AVERAGE_SCORE,
    EX.PASS_FAIL_RATE,

    AT.ATTENDANCE_PERCENT,

    AC.ACTIVITY_FLAG,
    AC.ACTIVITY_HOURS,
    AC.HIGHEST_ACHIEVEMENT,

    SD.DISCIPLINE_COUNT,
    SD.DISCIPLINE_SEVERITY,
    SD.PRIMARY_INCIDENT_TYPE,

    ST.AVG_TEACHER_RATING,
    ST.AVG_TEACHER_EXPERIENCE
FROM STUDENTS S
LEFT JOIN STUDENT_EXAM_SUMMARY EX ON S.STUDENT_ID = EX.STUDENT_ID
LEFT JOIN STUDENT_ATTENDANCE_SUMMARY AT ON S.STUDENT_ID = AT.STUDENT_ID
LEFT JOIN STUDENT_ACTIVITIES_SUMMARY AC ON S.STUDENT_ID = AC.STUDENT_ID
LEFT JOIN STUDENT_DISCIPLINE_SUMMARY SD ON S.STUDENT_ID = SD.STUDENT_ID
LEFT JOIN STUDENT_TEACHER_SUMMARY ST ON S.STUDENT_ID = ST.STUDENT_ID;

SELECT * FROM STUDENT_MODELING_DATASET
```