Assumptions and constraints

- For every grading fact there is a possibility for more than one attempt. Due to this in our queries always the last attempt is taken into consideration

Warehouse usage scenarios

- Measuring of the performance of students separately and as groups
- Reverse analysis based on historical data about students' performance in tests
- Measuring the performance of teachers and their work balance
- Identifying students with worst performance across groups

Warehouse design description and justification

1. **Partitioning for Efficient Subset Queries**

• **Table:** Grading

• **Justification:** Partitioning the Grading table by group_id allows efficient queries at the group level, such as calculating average grades for a specific group. Dynamic partitioning enables seamless insertion of data into appropriate partitions without manual intervention.

• **Table:** Courses

• **Justification:** Static partitioning by course_year helps retrieve course data for a specific academic year efficiently, crucial for analyzing trends across years.

2. **Bucketing for Optimized Joins and Aggregations**

• **Table:** Grading

• **Justification:** Bucketing by student_id optimizes queries like identifying top-performing or low-performing students. Buckets distribute data into smaller, sorted partitions, enabling faster lookups and joins when filtering by student_id.

3. **Efficient Storage Formats for Query Performance**

• **Table:** Grading

• **Justification:** Stored in **ORC** format to handle large volumes of grading data with better compression and faster query execution. This is critical for analytical queries requiring scanning or aggregating large datasets.

• **Table:** Students

• **Justification:** Stored in **Parquet** format for optimized reading and writing performance. This is particularly helpful for queries requiring detailed student-level data.

4. **Complex Types for Rich Data Representation**

• **Table:** Grading

• **Justification:** The percentage_of_points column is an **ARRAY** type, allowing the storage of many attempts to the same test by the same student. This avoids creating multiple columns for each attempt.

• **Table:** Teachers

• **Justification:** The surnames column is a **MAP** type to store teacher surnames and exceptionally maiden surnames. This avoids creating additional column in which some rows would be empty.

5. **Internal Tables for Frequently Accessed Data**

• **Tables**: Grading

• **Justification:** Internal table ensures that frequently accessed and processed data (e.g., grades) are optimized for performance.

6. **External Tables for Flexibility and Integration**

• **Tables:** Students, Teachers, Dates, Groups, Courses

• **Justification:** External tables allow seamless integration with external systems, ensuring that data remains accessible for other applications or systems outside the data warehouse.

Competency questions description

1. What is the average grading for each group from September 2020?

This query helps track group-level academic performance to identify trends or issues and prioritize support where needed.

2. Who are the 5 students with the highest average percentage points?

Enables highlighting top-performing students for awards, scholarships, or other recognition.

3. Who are the 10 students with the lowest average percentage points?

Facilitates identifying students needing extra academic support or intervention.

4. Which teachers have the best average points obtained by their students in September 2020?

Evaluates teacher effectiveness and informs performance reviews or teaching method adjustments.

5. What is the performance for the selected student?

This query tracks the monthly academic performance of a specific student by calculating the average percentage of points they earned in tests.

6. Which group has the best average points?

Highlights high-performing groups and uses their practices as benchmarks for others.

7. What are the average percentage points for each academic year (course_year)?

Helps compare performance across years and measure the differences between each year.

8. What is the distribution of students across academic years?

Tells Us if there are inequalities in distribution of number of students on each academic year.

9. How has the average grading of groups changed over the 2021 (monthly trend)?

Tracks group progress or regress and shows if the changes in school affect the performance of students.

10. Which teachers have taught the most courses, and how does their students' performance compare?

Identifies good and overworked teachers for appropriate workload balancing.