Hogwarts School of Witchcraft and Wizardry

Advanced Databases

Group B1
Oliwer Figura, Łukasz Machnik, Zofia Wiora

April 2025

1 Data model

1.1 Database Description

The Hogwarts School of Witchcraft and Wizardry Database System is designed to efficiently manage all aspects of the magical educational institution. This comprehensive system tracks students, teachers, academic records, house points, and extracurricular activities.

1.2 Purpose and Scope

This database serves multiple critical functions:

- Student information management including personal details, house affiliations, and dormitory assignments
- Keeping track of grades
- House points tracking for the annual House Cup competition
- Quidditch team roster management
- Teacher records and subject assignments

1.3 Database Schema

Figure 1 shows the Entity-Relationship diagram of the Hogwarts database. The schema illustrates the relationships between the nine primary tables that form the core of the system.

1.4 Key Entities and Relationships

The database consists of the following key tables:

1. **Students**: Contains personal information about each student, including their house and dormitory assignments.
2. **Houses**: The four houses of Hogwarts (Gryffindor, Hufflepuff, Ravenclaw, and Slytherin) with their attributes.
3. **Teachers**: Faculty information including employment dates and specializations.
4. **Grades**: Academic performance records with subject, teacher, student, and evaluation date.
5. **Subjects**: Course information including year level requirements and assigned classrooms.
6. **Students_Subjects**: Many-to-many relationship table connecting students to their enrolled subjects.
7. **Dormitories**: Housing information with gender and house affiliations.
8. **Points**: Records of house points awarded or deducted, with detailed attribution.
9. **Quidditch_Team_Members**: Manages Quidditch team rosters with positions and captain status.

2 Workload

2.1 Transactions querying the data

2.1.1 Grades analysis

This transaction takes all the students and for each of them displays the most interesting statistics, such as position in quidditch team, number of received points, and average of all grades.

Parameters: `house_name, year_range_start, year_range_end, date_start, date_end, grade_range_start, grade_range_end, teacher_names`

1. IF
   - student's house_id value belongs to a house named `house_name`
   - student's year value belongs to a range [`year_range_start, year_range_end`]
   - in the time period [`date_start, date_end`] student obtained only grades from range [`grade_range_start, grade_range_end`]
2. SHOW Student's
   - name
   - surname
   - gender
   - number of received points in the specified time period
   - position in quidditch team (if applicable, NULL otherwise)
   - average of all grades awarded by teachers included in the list `teacher_names` in the specified time period
   - (a) for this student for each subject calculate average grade in the specified time period
     (b) SHOW name of subject with the highest average and this average
   - (a) for this student for each subject calculate average grade in the specified time period
     (b) SHOW name of subject with the lowest average and this average
3. ORDERED BY
   - average grade
   - descending

Tables used: `Quidditch_Team_Members, Students, Houses, Points, Teachers, Grades, Subjects, Students_Subjects`

2.1.2 House points summary

This transaction shows all houses and for each of them displays useful data connected to the House Cup competition.

Parameters: `houses_list, teachers_list, only_players, date_start, date_end`

1. IF
   - house's name value belongs to a house listed in `houses_list`
   - student is assigned to any quidditch team (only applicable if `only_players` is set to true)
   - points were awarded in the specified time period
2. SHOW
   - house name
   - sum of points awarded by teachers listed in `teachers_list` collected by students assigned to this house
   - a teacher that awarded the most points for this house
   - a teacher that awarded the least points for this house
   - a month in which the most points were awarded for this house
   - a student which collected the most points for this house
3. ORDERED BY
   - house name
   - ascending

Tables used: `Quidditch_Team_Members, Students, Houses, Points, Teachers`

2.1.3 Best students display

This transaction is used to pick the best students divided by house, year and gender.

Parameters: `top_considered, date_start, date_end`

1. FOR EACH House
   (a) FOR EACH Year
      i. FOR EACH Gender
      SHOW
      - students ordered by points earned in specified time period
      - limit number of displayed students to `top_considered`
      - From this list SHOW
        – Student with the best average grade in the specified time period

Tables used: `Houses, Students, Points`

2.2 Transactions changing the data

2.2.1 Raising grades

This transaction is used to motivate students to earn points for their houses by rewarding academic improvements. Students who have received a lot of points are rewarded by raising their worst grades.

Parameters: `houses_list, teachers_list, min_points, raise_amount, number_of_affected`

For all grades received by all the students

IF

- student's house is in `houses_list`
- number of points received by student is higher than `min_points` (query)
- grade was assigned by one of the teachers from the `teachers_list`
- grade is lesser than maximum possible grade
- grade is one of the top `number_of_affected` worst grades received from the teacher from the `teachers_list` by the student (if the grades are the same, the older one is increased) (query)

UPDATE the grade by increasing its value by `raise_amount`.

Tables used: `Students, Houses, Teachers, Points, Grades`

2.2.2 Advancing students

This transaction would usually be performed at the beginning of a school year to move students to the next class, but only if they passed all the subjects. As a result of advancing to the next class, all students have to be assigned to new subjects and dormitories. Additionally, students finishing school need to be deleted, and new students need to be added.

Parameters: `new_students_list, minimal_average, max_dorm_capacity`

1. For all students

   IF

   - worst average of the subject of the student is higher than `minimal_average` (query based on 2.1.1)

   THEN UPDATE

   - increase student's year by one
   - change student's dormitory value to NULL

2. For all students

   IF

- student's year is higher than 7

THEN DELETE this student

3. INSERT new students from `new_students_list`
4. For all students and first dormitory fulfilling all conditions
   IF
   - value of student's dormitory is equal to NULL
   - dormitory house is equal to student house
   - dormitory year is equal to student year
   - dormitory gender is equal to student gender
   - list of students with dormitory value equal to dormitory id is not higher than `max_dorm_capacity` (query).

   THEN UPDATE
   - assign the dormitory to the student
5. DELETE all records from Points table
6. DELETE all records from Grades table
7. DELETE all records from Students_Subjects table
8. For all students and all subjects
   IF
   - student's year is equal to subject's year

   INSERT student and subject to Students_Subjects

Tables used: `Students, Dormitories, Quidditch_Team_Members, Points, Grades, Students_Subjects, Subjects`

2.2.3 Points normalization

This transaction addresses potential imbalances in the house points system caused by varying teacher grading standards by applying the linear normalization of the points values.

Parameters: `min_points_value, max_points_value, start_date, end_date, teachers_list`

For all teachers

IF

- teacher is on the `teachers_list`

UPDATE value of the points by

- finding the lowest value of points awarded by the teacher between `start_date` and `end_date` (query)
- finding the highest value of points awarded by the teacher between `start_date` and `end_date` (query)
- applying linear normalization to all values of points awarded by the teacher between `start_date` and `end_date` (query) so they all fall in the range of [`min_points_value`, `max_points_value`]

Tables used: `Points, Teachers`

3 DBMS

We chose and installed the following DBMS: Oracle 21c Express Edition. We also installed Oracle SQL Developer - software that helps to manage the Oracle database and allows generating an image with all relations. At last, we created the whole database structure with SQL code visible below and generated the aforementioned image of all relations (Fig. 2)

```
CREATE TABLE Students (
    id NUMBER NOT NULL PRIMARY KEY,
    name VARCHAR2(64) NOT NULL,
    surname VARCHAR2(64) NOT NULL,
    gender CHAR NOT NULL,
    date_of_birth DATE NOT NULL,
    year NUMBER NOT NULL,
    hogsmeade_consent NUMBER(1) DEFAULT 0 NOT NULL,
    house_id NUMBER NOT NULL,
    dormitory_id NUMBER
);

CREATE TABLE Dormitories (
    id NUMBER NOT NULL PRIMARY KEY,
    gender CHAR NOT NULL,
    room_number NUMBER NOT NULL,
    house_id NUMBER NOT NULL
);

CREATE TABLE Houses (
    id NUMBER NOT NULL PRIMARY KEY,
    name VARCHAR2(16) NOT NULL,
    symbol VARCHAR2(16) NOT NULL,
    location VARCHAR2(128),
    teacher_id NUMBER
);

CREATE TABLE Quidditch_Team_Members (
    id NUMBER NOT NULL PRIMARY KEY,
    position VARCHAR2(64),
    is_captain NUMBER(1) DEFAULT 0 NOT NULL,
    student_id NUMBER NOT NULL
);

CREATE TABLE Teachers (
    id NUMBER NOT NULL PRIMARY KEY,
    name VARCHAR2(64) NOT NULL,
```

```
    surname VARCHAR2(64) NOT NULL,
    date_of_birth DATE NOT NULL,
    date_of_employment DATE NOT NULL
);

CREATE TABLE Points (
    id NUMBER NOT NULL PRIMARY KEY,
    value NUMBER NOT NULL,
    description VARCHAR2(1024),
    award_date DATE NOT NULL,
    student_id NUMBER NOT NULL,
    teacher_id NUMBER
);

CREATE TABLE Grades (
    id NUMBER NOT NULL PRIMARY KEY,
    value VARCHAR2(1) NOT NULL,
    award_date DATE NOT NULL,
    student_id NUMBER NOT NULL,
    subject_id NUMBER NOT NULL,
    teacher_id NUMBER NOT NULL
);

CREATE TABLE Subjects (
    id NUMBER NOT NULL PRIMARY KEY,
    name VARCHAR2(64) NOT NULL,
    classroom NUMBER NOT NULL,
    year NUMBER NOT NULL,
    teacher_id NUMBER NOT NULL
);

CREATE TABLE Students_Subjects (
    student_id NUMBER NOT NULL,
    subject_id NUMBER NOT NULL
);


ALTER TABLE Students
    ADD FOREIGN KEY (house_id) REFERENCES Houses(id);

ALTER TABLE Students
    ADD FOREIGN KEY (dormitory_id) REFERENCES Dormitories(id);
```

```
ALTER TABLE Dormitories
    ADD FOREIGN KEY (house_id) REFERENCES Houses(id);

ALTER TABLE Houses
    ADD FOREIGN KEY (teacher_id) REFERENCES Teachers(id);

ALTER TABLE Quidditch_Team_Members
    ADD FOREIGN KEY (student_id) REFERENCES Students(id);

ALTER TABLE Points
    ADD FOREIGN KEY (student_id) REFERENCES Students(id);

ALTER TABLE Points
    ADD FOREIGN KEY (teacher_id) REFERENCES Teachers(id);

ALTER TABLE Grades
    ADD FOREIGN KEY (student_id) REFERENCES Students(id);

ALTER TABLE Grades
    ADD FOREIGN KEY (subject_id) REFERENCES Subjects(id);

ALTER TABLE Grades
    ADD FOREIGN KEY (teacher_id) REFERENCES Teachers(id);

ALTER TABLE Subjects
    ADD FOREIGN KEY (teacher_id) REFERENCES Teachers(id);

ALTER TABLE Students_Subjects
    ADD FOREIGN KEY (student_id) REFERENCES Students(id);

ALTER TABLE Students_Subjects
    ADD FOREIGN KEY (subject_id) REFERENCES Subjects(id);
```

4 Data

In order to populate the database, we've created the **Python script** generating required data.

4.1 Methods of generating the data

- **Students**: For each student, gender is randomly selected, and then a random name and surname are selected from the list of British names and surnames from Wikipedia. Date of birth is drawn from the range [2007-01-01, 2013-12-31] and the year is calculated accordingly. Students are assigned to houses randomly with a uniform distribution. Dormitories are assigned randomly in accordance with the constraints of the student's house, gender, and year, so that in one dormitory there are no more students than 6.
- **Houses**: There are four houses statically generated.
- **Teachers**: For each teacher, a name and surname are randomly selected from the list of British names and surnames from Wikipedia. Then, the date of birth is drawn from the range [1900-01-01, 2000-01-01]. The date of employment is also selected randomly, but the employee can be no younger than 24 to become a teacher.
- **Grades**: Grades are generated with a normal distribution (mean = 3, standard deviation = 1.25, min value = 0, max value = 5) and then translated to corresponding letters. Students and teachers are assigned randomly, taking into account that teacher and student must match the subject. The award date is drawn from the range [2023-09-01, 2024-06-30].
- **Subjects**: Subjects are generated from the provided list of subjects, assigned to corresponding years. Classroom and teacher are selected randomly.
- **Students_Subjects**: This table is generated based on the Students and Subjects table, so that each student is assigned to all subjects that have the same year.
- **Dormitories**: For each house, gender, and year, there are between 58 and 62 dormitories. Room number consists of year number and identification number which can occur only once in the house.
- **Points**: The value of points is randomly generated with a normal distribution (mean = 0, standard deviation = 5) and rounded to integer values. If the value is equal to 0, it is omitted. Students and teachers are assigned randomly, and the description is generated based on the name of the student and information if the number of received points is negative or positive. The content of the description is prepared using randomly selected words with various probabilities, so they feel natural to the reader. The award date is drawn from the range [2023-09-01, 2024-06-30].
- **Quidditch_Team_Members**: Every student is with probability 1% assigned to be a member of Quidditch team, then the roles are distributed with one of the members being the captain.

4.2 Number of records in each table

For each script run, the number of records generated may vary slightly. In the table, there are exact values shown for the data generated in one of the runs.

- **Students**: $10\,000 \pm 10\%$
- **Houses**: 4
- **Teachers**: $1000 \pm 10\%$
- **Grades**: from 10 to 30 grades for each subject assigned to a student

Table	Rows
Students	9 346
Houses	4
Teachers	1097
Grades	1 977 314
Subjects	76
Students_Subjects	101 583
Dormitories	3 369
Points	2 986 448
Quidditch_Team_Members	97

Table 1: Tables dimensions

- **Subjects**: 76
- **Students_Subjects**: from 10 to 12 subjects assigned to a student
- **Dormitories**: from 58 to 62 dormitories for each house, gender, and year
- **Points**: $3\,000\,000 \pm 10\%$
- **Quidditch_Team_Members**: about 100

4.3 Distribution of data

While generating data for **Points** and **Grades** tables, we assumed that the occurrence of each grade/point value is subject to a normal distribution.

5 Database workload

5.1 Transactions querying the data

All execution times are for default parameters.

5.1.1 Grades analysis

Query displaying students' statistics.

```
SELECT q.name,
       q.surname,
       q.gender,
       q.total_points,
       q.position,
       g1.symbol AS best_grade,
```

```
       s1.name AS best_subject,
       g2.symbol AS worst_grade,
       s2.name AS worst_subject
FROM (
    SELECT students.name,
           students.surname,
           students.gender,
           SUM(points.value) AS total_points,
           quidditch_team_members.position,
           MAX(averages.rounded_avg) KEEP (DENSE_RANK FIRST ORDER BY averages.accurate_avg DESC) AS max_grade,
           MAX(averages.subject_id) KEEP (DENSE_RANK FIRST ORDER BY averages.accurate_avg DESC) AS best_subject,
           MAX(averages.rounded_avg) KEEP (DENSE_RANK FIRST ORDER BY averages.accurate_avg ASC) AS min_grade,
           MAX(averages.subject_id) KEEP (DENSE_RANK FIRST ORDER BY averages.accurate_avg ASC) AS worst_subject
    FROM students
        INNER JOIN houses
            ON students.house_id = houses.id
        INNER JOIN points
            ON students.id = points.student_id
        INNER JOIN grades
            ON students.id = grades.student_id
        LEFT JOIN quidditch_team_members
            ON students.id = quidditch_team_members.student_id
    INNER JOIN (
        SELECT grades.student_id,
               grades.subject_id,
               AVG(grades_enum.value) AS ACCURATE_AVG,
               ROUND(AVG(grades_enum.value), 0) AS ROUNDED_AVG
        FROM grades
            INNER JOIN grades_enum
                ON grades.value = grades_enum.symbol
        GROUP BY grades.student_id,
                 grades.subject_id
    ) averages ON averages.student_id = students.id
    WHERE houses.name = 'Gryffindor'
      AND students.year >= 1
      AND students.year <= 7
      AND points.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
      AND points.award_date <= TO_DATE('2023-09-07', 'YYYY-MM-DD')
      AND students.id NOT IN (
        SELECT students.id
        FROM students
            INNER JOIN grades
```

```
                ON students.id = grades.student_id
            INNER JOIN grades_enum
                ON grades.value = grades_enum.symbol
        WHERE grades.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
          AND grades.award_date <= TO_DATE('2023-09-07', 'YYYY-MM-DD')
          AND (
            grades_enum.value <
            (
                SELECT grades_enum.value
                FROM grades_enum
                WHERE grades_enum.symbol = 'T'
            )
            OR grades_enum.value >
            (
                SELECT grades_enum.value
                FROM grades_enum
                WHERE grades_enum.symbol = 'O'
            )
          )
      )
    GROUP BY students.id,
    students.name,
    students.surname,
    students.gender,
    quidditch_team_members.position
) q
    INNER JOIN grades_enum g1
        ON q.max_grade = g1.value
    INNER JOIN grades_enum g2
        ON q.min_grade = g2.value
    INNER JOIN subjects s1
        ON q.best_subject = s1.id
    INNER JOIN subjects s2
        ON q.worst_subject = s2.id
ORDER BY 4 DESC;
```

5.1.2 House points summary

Query displaying points summary for all houses.

```sql
SELECT houses.name,
       SUM(points.value),
       t1.name AS BEST_TEACHER_NAME,
       t1.surname AS BEST_TEACHER_SURNAME,
       t2.name AS WORST_TEACHER_NAME,
       t2.surname AS WORST_TEACHER_SURNAME,
       m.best_month, s.best_student_name,
       s.best_student_surname
FROM houses
    INNER JOIN students
        ON houses.id = students.house_id
    INNER JOIN points
        ON students.id = points.student_id
    INNER JOIN (
        SELECT q.house_id,
               MAX(q.teacher_id) KEEP (DENSE_RANK FIRST ORDER BY q.total_points DESC) AS FAVOURITE_TEACHER,
               MAX(q.teacher_id) KEEP (DENSE_RANK FIRST ORDER BY q.total_points ASC) AS LEAST_FAVOURITE_TEACHER
        FROM (
            SELECT students.house_id,
                   points.teacher_id,
                   SUM(points.value) as TOTAL_POINTS
            FROM students
                INNER JOIN points
                    ON students.id = points.student_id
            WHERE points.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
              AND points.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
            GROUP BY students.house_id,
                     points.teacher_id
        ) q
        GROUP BY q.house_id
    ) t
        ON t.house_id = houses.id
    INNER JOIN (
        SELECT q.house_id,
               MAX(q.award_month) KEEP (DENSE_RANK FIRST ORDER BY q.total_points DESC) AS BEST_MONTH
        FROM (
            SELECT students.house_id,
                   EXTRACT(MONTH FROM points.award_date) AS AWARD_MONTH,
                   SUM(points.value) AS TOTAL_POINTS
            FROM points
                INNER JOIN students
                    ON points.student_id = students.id
```

```
            WHERE points.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
              AND points.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
            GROUP BY students.house_id,
                     EXTRACT(MONTH FROM points.award_date)
        ) q
        GROUP BY q.house_id
    ) m
        ON m.house_id = houses.id
    INNER JOIN teachers t1
        ON t1.id = t.favourite_teacher
    INNER JOIN teachers t2
        ON t2.id = t.least_favourite_teacher
    INNER JOIN (
        SELECT students.house_id,
               MAX(students.name) KEEP (DENSE_RANK FIRST ORDER BY q.total_points DESC) AS BEST_STUDENT_NAME,
               MAX(students.surname) KEEP (DENSE_RANK FIRST ORDER BY q.total_points DESC) AS BEST_STUDENT_SURNAME
        FROM students
            INNER JOIN (
                SELECT points.student_id,
                       SUM(points.value) as TOTAL_POINTS
                FROM points
                WHERE points.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
                  AND points.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
                GROUP BY points.student_id
            ) q
                ON students.id = q.student_id
        GROUP BY students.house_id
    ) s
        ON s.house_id = houses.id
WHERE points.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
  AND points.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
GROUP BY houses.name,
         t1.name,
         t1.surname,
         t2.name,
         t2.surname,
         m.best_month,
         s.best_student_name,
         s.best_student_surname;
```

5.1.3 Best students display

Query selecting best students from the database.

```
CREATE OR REPLACE PROCEDURE display_best_students(
    p_top_count IN NUMBER DEFAULT 10,
    p_date_start IN DATE DEFAULT TO_DATE('2023-01-01', 'YYYY-MM-DD'),
    p_date_end IN DATE DEFAULT TO_DATE('2023-12-31', 'YYYY-MM-DD')
) AS
    v_total_students NUMBER;
    v_start_time TIMESTAMP;
    v_end_time TIMESTAMP;
    v_execution_time NUMBER;
BEGIN
    v_start_time := SYSTIMESTAMP;
    DECLARE
        v_rank NUMBER := 1;
    BEGIN
        FOR student_rec IN (
            SELECT
                s.name || ' ' || s.surname AS student_name,
                h.name AS house_name,
                s.year,
                ROUND(AVG(ge.value), 2) AS avg_grade,
                NVL(SUM(p.value), 0) AS total_points
            FROM
                Students s
                JOIN Houses h ON s.house_id = h.id
                LEFT JOIN Grades g ON s.id = g.student_id
                    AND g.award_date BETWEEN p_date_start AND p_date_end
                LEFT JOIN Grades_Enum ge ON g.value = ge.symbol
                LEFT JOIN Points p ON s.id = p.student_id
                    AND p.award_date BETWEEN p_date_start AND p_date_end
            GROUP BY
                s.name, s.surname, h.name, s.year
            ORDER BY
                avg_grade DESC, total_points DESC
            FETCH FIRST p_top_count ROWS ONLY
        ) LOOP
            DBMS_OUTPUT.PUT_LINE(
                LPAD(v_rank, 4) || '  ' ||
                RPAD(student_rec.student_name, 26) || '  ' ||
                RPAD(student_rec.house_name, 10) || '  ' ||
```

```
                LPAD(student_rec.year, 4) || '  ' ||
                LPAD(TO_CHAR(student_rec.avg_grade, '90.99'), 9) || '  ' ||
                LPAD(TO_CHAR(student_rec.total_points), 6)
            );

            v_rank := v_rank + 1;
        END LOOP;
    END;

    SELECT COUNT(*) INTO v_total_students FROM Students;

    v_end_time := SYSTIMESTAMP;
    v_execution_time := EXTRACT(SECOND FROM (v_end_time - v_start_time)) +
                        EXTRACT(MINUTE FROM (v_end_time - v_start_time)) * 60;
END;
```

5.1.4 Best students by average grade

Query displaying top students by average grade.

```
WITH top_students AS (
    SELECT
        ROW_NUMBER() OVER (ORDER BY ROUND(AVG(ge.value), 2) DESC, NVL(SUM(p.value), 0) DESC) AS rank,
        s.name || ' ' || s.surname AS student_name,
        h.name AS house_name,
        s.year,
        ROUND(AVG(ge.value), 2) AS avg_grade,
        NVL(SUM(p.value), 0) AS total_points
    FROM
        Students s
        JOIN Houses h ON s.house_id = h.id
        LEFT JOIN Grades g ON s.id = g.student_id
            AND g.award_date BETWEEN TO_DATE('&date_start', 'YYYY-MM-DD') AND TO_DATE('&date_end', 'YYYY-MM-DD')
        LEFT JOIN Grades_Enum ge ON g.value = ge.symbol
        LEFT JOIN Points p ON s.id = p.student_id
            AND p.award_date BETWEEN TO_DATE('&date_start', 'YYYY-MM-DD') AND TO_DATE('&date_end', 'YYYY-MM-DD')
    GROUP BY
        s.name, s.surname, h.name, s.year
)
SELECT
    rank,
    student_name,
```

```
    house_name,
    year,
    avg_grade,
    total_points
FROM
    top_students
WHERE
    rank <= &top_count
ORDER BY
    rank;
```

5.2 Transactions changing the data

5.2.1 Raising grades

Transaction raising students' grades based on earned points.

```
CREATE OR REPLACE PROCEDURE raise_student_grades(
    p_houses VARCHAR2,
    p_teachers VARCHAR2,
    p_min_points NUMBER,
    p_raise_amount NUMBER,
    p_num_affected NUMBER
) AS
    v_start TIMESTAMP;
    v_end TIMESTAMP;

    v_count NUMBER := 0;
    v_students_found NUMBER := 0;
    v_teachers_found NUMBER := 0;
    v_display_limit CONSTANT NUMBER := 5;

    v_houses VARCHAR2(1000);
    v_teachers VARCHAR2(1000);

    PROCEDURE short_delay IS
        v_dummy NUMBER;
    BEGIN
        FOR i IN 1..10 LOOP
            SELECT 1 INTO v_dummy FROM dual;
        END LOOP;
    END short_delay;
```

```
26
27  BEGIN
28      v_start := SYSTIMESTAMP;
29      v_houses := ',' || p_houses || ',';
30      v_teachers := ',' || p_teachers || ',';
31
32      DBMS_OUTPUT.PUT_LINE('Processing houses: ' || p_houses);
33
34      FOR r IN (
35          SELECT
36              s.id AS student_id,
37              s.name || ' ' || s.surname AS student_name,
38              h.name AS house_name,
39              NVL(SUM(p.value), 0) AS total_points
40          FROM
41              Students s
42              JOIN Houses h ON s.house_id = h.id
43              LEFT JOIN Points p ON s.id = p.student_id
44          WHERE
45              INSTR(v_houses, ',' || h.name || ',') > 0
46          GROUP BY
47              s.id, s.name, s.surname, h.name
48          HAVING
49              NVL(SUM(p.value), 0) >= p_min_points
50          ORDER BY
51              total_points DESC
52      ) LOOP
53          v_students_found := v_students_found + 1;
54
55          IF v_students_found <= v_display_limit THEN
56              DBMS_OUTPUT.PUT_LINE('Processing student #' || v_students_found || ': '
57              || r.student_name ||
58              ' (' || r.house_name || ', ' || r.total_points || ' points)');
59          ELSIF v_students_found = v_display_limit + 1 THEN
60              DBMS_OUTPUT.PUT_LINE('... Processing remaining students ...');
61          END IF;
62
63          short_delay;
64
65          DECLARE
66              v_teacher_count NUMBER := 0;
67              v_student_grades_updated NUMBER := 0;
```

```
        BEGIN
            FOR t IN (
                SELECT id, name || ' ' || surname AS teacher_name
                FROM Teachers
                WHERE p_teachers = 'ALL' OR INSTR(v_teachers, ',' || id || ',') > 0
                FETCH FIRST 5 ROWS ONLY
            ) LOOP
                v_teachers_found := v_teachers_found + 1;
                v_teacher_count := v_teacher_count + 1;

                DECLARE
                    v_grades_updated NUMBER := 0;
                BEGIN
                    FOR g IN (
                        SELECT
                            g.id AS grade_id,
                            g.value AS current_grade,
                            ge.value AS numeric_value,
                            CASE
                                WHEN g.subject_id IS NOT NULL THEN (
                                    SELECT name
                                    FROM Subjects
                                    WHERE id = g.subject_id)
                                ELSE 'Unknown Subject'
                            END AS subject_name
                        FROM
                            Grades g
                            JOIN Grades_Enum ge ON g.value = ge.symbol
                        WHERE
                            g.student_id = r.student_id
                            AND g.teacher_id = t.id
                            AND g.value != 'O'
                        ORDER BY
                            ge.value ASC, g.award_date ASC
                        FETCH FIRST p_num_affected ROWS ONLY
                    ) LOOP
                        short_delay;

                        DECLARE
                            v_new_grade VARCHAR2(1);
                        BEGIN
                            v_new_grade :=
```

```
                            CASE g.current_grade
                                WHEN 'T' THEN 'D'
                                WHEN 'D' THEN 'P'
                                WHEN 'P' THEN 'A'
                                WHEN 'A' THEN 'E'
                                WHEN 'E' THEN 'O'
                                ELSE g.current_grade
                            END;

                        UPDATE Grades
                            SET value = v_new_grade
                            WHERE id = g.grade_id;

                        v_count := v_count + SQL%ROWCOUNT;
                        v_grades_updated := v_grades_updated + SQL%ROWCOUNT;
                        v_student_grades_updated := v_student_grades_updated
                            + SQL%ROWCOUNT;
                    END;
                END LOOP;
                IF v_grades_updated > 0 AND v_students_found <= v_display_limit
                THEN
                    DBMS_OUTPUT.PUT_LINE('    - Updated ' || v_grades_updated ||
                                ' grades for teacher: ' || t.teacher_name);
                END IF;
            END;
        END LOOP;
        IF v_teacher_count > 0 AND v_students_found <= v_display_limit THEN
            DBMS_OUTPUT.PUT_LINE('  Processed ' || v_teacher_count
            || ' teachers for this student');
            IF v_student_grades_updated > 0 THEN
                DBMS_OUTPUT.PUT_LINE('  Total grades updated for this student: '
                || v_student_grades_updated);
            END IF;
        END IF;
    END;
    IF MOD(v_students_found, 10) = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Progress: ' || v_students_found
        || ' students processed, ' ||
                        v_count || ' grades updated so far (' ||
                        TO_CHAR(ROUND(EXTRACT(SECOND FROM
                        (SYSTIMESTAMP - v_start)), 1)) ||
                        ' seconds elapsed)');
```

```
        END IF;
    END LOOP;

    IF v_students_found = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No eligible students ');
    END IF;

    v_end := SYSTIMESTAMP;

    DBMS_OUTPUT.PUT_LINE('=== EXECUTION SUMMARY ===');
    DBMS_OUTPUT.PUT_LINE('Students found and processed: ' || v_students_found);
    DBMS_OUTPUT.PUT_LINE('Teachers processed: ' || v_teachers_found);
    DBMS_OUTPUT.PUT_LINE('Grades updated: ' || v_count);
    DBMS_OUTPUT.PUT_LINE('Execution time: ' ||
        TO_CHAR(ROUND(EXTRACT(SECOND FROM (v_end - v_start)) +
                EXTRACT(MINUTE FROM (v_end - v_start)) * 60, 2)) ||
        ' seconds');

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Changes committed.');

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Changes rolled back.');
END;
```

5.2.2 Raising grades transaction

Transaction raising grades of students with high points.

```
MERGE INTO Grades g
USING (
    WITH eligible_students AS (
        SELECT
            s.id AS student_id,
            s.name || ' ' || s.surname AS student_name,
            h.name AS house_name,
            NVL(SUM(p.value), 0) AS total_points
        FROM
            Students s
```

```
        JOIN Houses h ON s.house_id = h.id
        LEFT JOIN Points p ON s.id = p.student_id
    WHERE
        INSTR(',' || '&houses_list' || ',', ',' || h.name || ',') > 0
    GROUP BY
        s.id, s.name, s.surname, h.name
    HAVING
        NVL(SUM(p.value), 0) >= &min_points
),
eligible_teachers AS (
    SELECT id, name || ' ' || surname AS teacher_name
    FROM Teachers
    WHERE
        UPPER('&teachers_list') = 'ALL' OR
        INSTR(',' || '&teachers_list' || ',', ',' || id || ',') > 0
),
grades_to_update AS (
    SELECT
        g.id AS grade_id,
        g.student_id,
        g.teacher_id,
        g.value AS current_grade,
        CASE g.value
            WHEN 'T' THEN 'D'
            WHEN 'D' THEN 'P'
            WHEN 'P' THEN 'A'
            WHEN 'A' THEN 'E'
            WHEN 'E' THEN 'O'
            ELSE g.value
        END AS new_grade,
        ROW_NUMBER() OVER (
            PARTITION BY g.student_id
            ORDER BY ge.value ASC, DBMS_RANDOM.VALUE) AS grade_rank
    FROM
        Grades g
        JOIN Grades_Enum ge ON g.value = ge.symbol
        JOIN eligible_students es ON g.student_id = es.student_id
        JOIN eligible_teachers et ON g.teacher_id = et.id
    WHERE
        g.value != 'O'
        AND g.value != 'E'
)
```

```
    SELECT
        grade_id,
        new_grade
    FROM
        grades_to_update
    WHERE
        grade_rank <= &number_of_affected
) src
ON (g.id = src.grade_id)
WHEN MATCHED THEN
UPDATE SET g.value = src.new_grade;
```

5.2.3 Advancing students

Transaction moving students to the next class.

```
CREATE OR REPLACE PROCEDURE ADVANCING_STUDENTS
(
  MIN_AVERAGE IN NUMBER DEFAULT 3.5
, MAX_DORM_CAPACITY IN NUMBER DEFAULT 6
) AS
    time_start NUMBER;
    time_end NUMBER;
    time_elapsed NUMBER;
BEGIN
    time_start := DBMS_UTILITY.GET_TIME;
-- Advancing eligible students and removing them from dormitory
    UPDATE students
    SET students.year = students.year + 1, students.dormitory_id = NULL
    WHERE students.id NOT IN (
        SELECT DISTINCT failed.student_id
        FROM (
            SELECT grades.student_id, grades.subject_id, AVG(grades_enum.value)
            FROM grades
            INNER JOIN grades_enum ON grades.value = grades_enum.symbol
            GROUP BY grades.student_id, grades.subject_id
            HAVING AVG(grades_enum.value) < min_average
        ) failed
    );
-- Assigning students to a new dormitory
    UPDATE students s
    SET s.dormitory_id = (
```

```
        SELECT available.id
        FROM (
            SELECT dormitories.id, COUNT(students.id) AS OCCUPANCY
            FROM dormitories
            LEFT JOIN students ON dormitories.id = students.dormitory_id
            WHERE dormitories.year = s.year
            AND dormitories.house_id = s.house_id
            AND dormitories.gender = s.gender
            HAVING COUNT(students.id) < max_dorm_capacity
            GROUP BY dormitories.id
            ORDER BY 2 ASC
            FETCH FIRST ROW ONLY
        ) available
    )
    WHERE s.dormitory_id IS NULL;
-- Deleting all grades, points and subject assignments
    DELETE FROM points;
    DELETE FROM grades;
    DELETE FROM students_subjects;
-- Assigning all students to new subjects
    INSERT INTO students_subjects (student_id, subject_id)
    SELECT students.id, subjects.id
    FROM students
    INNER JOIN subjects ON students.year = subjects.year;
-- Calculate elapsed time
    time_end := DBMS_UTILITY.GET_TIME;
    time_elapsed := (time_end - time_start) / 100;
    DBMS_OUTPUT.PUT_LINE('Execution time: ' || time_elapsed || ' seconds');
END ADVANCING_STUDENTS;
```

5.2.4 Points normalization

Transaction normalizing house points values.

```
UPDATE points pOuter
    SET pOuter.value = ROUND(pOuter.value * 10 / (
        SELECT MAX(pInner.value)
            FROM points pInner
        WHERE pInner.teacher_id = pouter.teacher_id
    ))
    WHERE pOuter.value > 0
      AND pOuter.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
```

```
      AND pOuter.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
      AND teacher.id >= 0
      AND teacher.id <= 100
```

5.2.5 Removing bad students' accomplishments

Transaction that removes all positive points acquired by students that didn't get average positive grades from 4 or more subjects.

```
DELETE FROM points
WHERE points.student_id IN(
    SELECT q.student_id
    FROM (
        SELECT grades.student_id,
               grades.subject_id,
               AVG(grades_enum.value)
        FROM grades
            INNER JOIN grades_enum
                ON grades.value = grades_enum.symbol
        WHERE grades.award_date >= TO_DATE('2023-09-01', 'YYYY-MM-DD')
          AND grades.award_date <= TO_DATE('2024-06-30', 'YYYY-MM-DD')
        HAVING AVG(grades_enum.value) < 4
        GROUP BY grades.student_id,
                 grades.subject_id
    ) q
    HAVING COUNT(q.subject_id) >= 4
    GROUP BY q.student_id
)
  AND points.value > 0
```

6 Query plans

6.1 Transactions querying the data

6.1.1 Grades analysis

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                 | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                      |     3 |   882 |       |   144K (82)| 00:00:06 |
|   1 |  SORT ORDER BY                     |                      |     3 |   882 |       |   144K (82)| 00:00:06 |
```

```
|   2 |   NESTED LOOPS                     |                          |     3 |   882 |       |   144K (82)| 00:00:06 |
|   3 |    NESTED LOOPS                    |                          |     3 |   882 |       |   144K (82)| 00:00:06 |
|   4 |     NESTED LOOPS                   |                          |     3 |   741 |       |   144K (82)| 00:00:06 |
|*  5 |      HASH JOIN                     |                          |     3 |   600 |       |   144K (82)| 00:00:06 |
|*  6 |       HASH JOIN                    |                          |     3 |   555 |       |   144K (82)| 00:00:06 |
|   7 |        VIEW                        |                          |     3 |   510 |       |   144K (82)| 00:00:06 |
|   8 |         SORT GROUP BY              |                          |     3 |   879 |       |   144K (82)| 00:00:06 |
|*  9 |          HASH JOIN                 |                          |   829M|   226G|       | 70819  (63)| 00:00:03 |
|* 10 |           TABLE ACCESS FULL        | HOUSES                   |     1 |    23 |       |     3   (0)| 00:00:01 |
|  11 |           MERGE JOIN ANTI          |                          |  3316M|   834G|       | 48783  (47)| 00:00:02 |
|  12 |            MERGE JOIN              |                          |  3316M|   793G|       | 46283  (49)| 00:00:02 |
|  13 |             MERGE JOIN             |                          |    15M|  3647M|       | 12587   (3)| 00:00:01 |
|  14 |              MERGE JOIN OUTER      |                          |  1937K|   386M|       |  2886   (6)| 00:00:01 |
|  15 |               MERGE JOIN           |                          |  1937K|   299M|       |  2882   (6)| 00:00:01 |
|  16 |                SORT JOIN           |                          |  1937K|    96M|       |  2632   (7)| 00:00:01 |
|  17 |                 VIEW               |                          |  1937K|    96M|       |  2632   (7)| 00:00:01 |
|  18 |                  HASH GROUP BY     |                          |  1937K|    79M|       |  2632   (7)| 00:00:01 |
|* 19 |                   HASH JOIN        |                          |  1937K|    79M|       |  2510   (2)| 00:00:01 |
|  20 |                    TABLE ACCESS FULL| GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  21 |                    TABLE ACCESS FULL| GRADES                   |  1937K|    51M|       |  2494   (2)| 00:00:01 |
|* 22 |               SORT JOIN            |                          |  9156 |   983K|  2344K|   250   (1)| 00:00:01 |
|* 23 |                TABLE ACCESS FULL   | STUDENTS                 |  9156 |   983K|       |    19   (0)| 00:00:01 |
|* 24 |              SORT JOIN             |                          |    97 |  4559 |       |     4  (25)| 00:00:01 |
|  25 |               TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS   |    97 |  4559 |       |     3   (0)| 00:00:01 |
|* 26 |             SORT JOIN              |                          | 77469 |  2647K|  7304K|  9610   (1)| 00:00:01 |
|* 27 |              TABLE ACCESS FULL     | POINTS                   | 77469 |  2647K|       |  8886   (1)| 00:00:01 |
|* 28 |            SORT JOIN               |                          |  1937K|    24M|    74M| 11766   (2)| 00:00:01 |
|  29 |             TABLE ACCESS FULL      | GRADES                   |  1937K|    24M|       |  2491   (2)| 00:00:01 |
|* 30 |           SORT UNIQUE              |                          |  4804 | 62452 |       |  2500   (2)| 00:00:01 |
|  31 |            VIEW                    | VW_NSO_1                 |  4804 | 62452 |       |  2499   (2)| 00:00:01 |
|* 32 |             HASH JOIN              |                          |  4804 |   182K|       |  2499   (2)| 00:00:01 |
|* 33 |              TABLE ACCESS FULL     | GRADES_ENUM              |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 34 |               TABLE ACCESS FULL    | GRADES_ENUM              |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 35 |               TABLE ACCESS FULL    | GRADES_ENUM              |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 36 |              TABLE ACCESS FULL     | GRADES                   | 49275 |  1154K|       |  2496   (2)| 00:00:01 |
|  37 |        TABLE ACCESS FULL           | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  38 |       TABLE ACCESS FULL            | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  39 |      TABLE ACCESS BY INDEX ROWID   | SUBJECTS                 |     1 |    47 |       |     1   (0)| 00:00:01 |
|* 40 |       INDEX UNIQUE SCAN            | SYS_C008771              |     1 |       |       |     0   (0)| 00:00:01 |
|* 41 |     INDEX UNIQUE SCAN              | SYS_C008771              |     1 |       |       |     0   (0)| 00:00:01 |
|  42 |    TABLE ACCESS BY INDEX ROWID     | SUBJECTS                 |     1 |    47 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   5 - access("Q"."MIN_GRADE"="G2"."VALUE")
   6 - access("Q"."MAX_GRADE"="G1"."VALUE")
   9 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  10 - filter("HOUSES"."NAME"='Gryffindor')
  19 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  22 - access("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
       filter("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
```

```
  23 - filter("STUDENTS"."YEAR">=1 AND "STUDENTS"."YEAR"<=7)
  24 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
       filter("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
  26 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
       filter("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  27 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2023-09-07 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  28 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
       filter("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  30 - access("STUDENTS"."ID"="ID")
       filter("STUDENTS"."ID"="ID")
  32 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  33 - filter("GRADES_ENUM"."VALUE"< (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM" WHERE
              "GRADES_ENUM"."SYMBOL"='T') OR "GRADES_ENUM"."VALUE"> (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM"
              "GRADES_ENUM" WHERE "GRADES_ENUM"."SYMBOL"='O'))
  34 - filter("GRADES_ENUM"."SYMBOL"='T')
  35 - filter("GRADES_ENUM"."SYMBOL"='O')
  36 - filter("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE"<=TO_DATE(' 2023-09-07 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  40 - access("Q"."BEST_SUBJECT"="S1"."ID")
  41 - access("Q"."WORST_SUBJECT"="S2"."ID")
Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - this is an adaptive plan
```

Candidates for improvements:

Operations with highest cost: 0-8 (SELECT STATEMENT, SORT ORDER BY, NESTED LOOPS, HASH JOIN, VIEW, SORT GROUP BY)

Other operations with high cost: 9, 11-13, 28 (HASH JOIN, MERGE JOIN ANTI, MERGE JOIN, SORT JOIN)

6.1.2 House points summary

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |             |  2986K|   535M|       |   159K  (1)| 00:00:07 |
|   1 |  HASH GROUP BY                     |             |  2986K|   535M|   583M|   159K  (1)| 00:00:07 |
|*  2 |   HASH JOIN                        |             |  2986K|   535M|       | 36394   (3)| 00:00:02 |
|   3 |    TABLE ACCESS FULL               | TEACHERS    |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  4 |    HASH JOIN                       |             |  2986K|   481M|       | 36369   (3)| 00:00:02 |
|   5 |     TABLE ACCESS FULL              | TEACHERS    |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  6 |     HASH JOIN                      |             |  2986K|   427M|       | 36344   (3)| 00:00:02 |
|   7 |      VIEW                          |             |     4 |    64 |       |  9127   (4)| 00:00:01 |
|   8 |       SORT GROUP BY                |             |     4 |   116 |       |  9127   (4)| 00:00:01 |
|   9 |        VIEW                        |             |   858 | 24882 |       |  9127   (4)| 00:00:01 |
|  10 |         HASH GROUP BY              |             |   858 | 19734 |       |  9127   (4)| 00:00:01 |
|* 11 |          HASH JOIN                 |             |  2986K|    65M|       |  8934   (2)| 00:00:01 |
|  12 |           TABLE ACCESS FULL        | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|* 13 |           TABLE ACCESS FULL        | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |
|* 14 |      HASH JOIN                     |             |  2986K|   381M|       | 27197   (3)| 00:00:02 |
```

```
|  15 |       VIEW                          |             |     4 |   116 |       |  9131   (4)| 00:00:01 |
|  16 |        SORT GROUP BY                |             |     4 |   116 |       |  9131   (4)| 00:00:01 |
|  17 |         VIEW                        |             |  3103 | 89987 |       |  9131   (4)| 00:00:01 |
|  18 |          HASH GROUP BY              |             |  3103 | 83781 |       |  9131   (4)| 00:00:01 |
|* 19 |           HASH JOIN                 |             |  2986K|    76M|       |  8938   (2)| 00:00:01 |
|  20 |            TABLE ACCESS FULL        | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|* 21 |            TABLE ACCESS FULL        | POINTS      |  2986K|    56M|       |  8899   (1)| 00:00:01 |
|* 22 |       HASH JOIN                     |             |  2986K|   299M|       | 18046   (3)| 00:00:01 |
|  23 |        VIEW                         | VW_GBF_15   |  9346 |   812K|       |  9131   (4)| 00:00:01 |
|* 24 |         HASH JOIN                   |             |  9346 |   930K|       |  9131   (4)| 00:00:01 |
|  25 |          MERGE JOIN                 |             |     4 |   380 |       |  9112   (4)| 00:00:01 |
|  26 |           TABLE ACCESS BY INDEX ROWID| HOUSES      |     4 |    56 |       |     2   (0)| 00:00:01 |
|  27 |            INDEX FULL SCAN          | SYS_C008743 |     4 |       |       |     1   (0)| 00:00:01 |
|* 28 |           SORT JOIN                 |             |     4 |   324 |       |  9110   (4)| 00:00:01 |
|  29 |            VIEW                     |             |     4 |   324 |       |  9109   (4)| 00:00:01 |
|  30 |             SORT GROUP BY           |             |     4 |   156 |       |  9109   (4)| 00:00:01 |
|* 31 |              HASH JOIN              |             |  9346 |   355K|       |  9108   (4)| 00:00:01 |
|  32 |               VIEW                  |             |  9346 |   155K|       |  9088   (4)| 00:00:01 |
|  33 |                HASH GROUP BY        |             |  9346 |   146K|       |  9088   (4)| 00:00:01 |
|* 34 |                 TABLE ACCESS FULL   | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |
|  35 |               TABLE ACCESS FULL     | STUDENTS    |  9346 |   200K|       |    19   (0)| 00:00:01 |
|  36 |          TABLE ACCESS FULL          | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|* 37 |        TABLE ACCESS FULL            | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("T2"."ID"="T"."LEAST_FAVOURITE_TEACHER")
   4 - access("T1"."ID"="T"."FAVOURITE_TEACHER")
   6 - access("M"."HOUSE_ID"="ITEM_2")
  11 - access("POINTS"."STUDENT_ID"="STUDENTS"."ID")
  13 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  14 - access("T"."HOUSE_ID"="ITEM_2")
  19 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  21 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  22 - access("ITEM_1"="POINTS"."STUDENT_ID")
  24 - access("HOUSES"."ID"="STUDENTS"."HOUSE_ID")
  28 - access("S"."HOUSE_ID"="HOUSES"."ID")
       filter("S"."HOUSE_ID"="HOUSES"."ID")
  31 - access("STUDENTS"."ID"="Q"."STUDENT_ID")
  34 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  37 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

Candidates for improvements:

Operations with highest cost: 0, 1 (SELECT STATEMENT, HASH GROUP BY)

Other operations with high cost: 2, 4, 6, 14 (HASH JOIN)

6.1.3 Best Students Display

This query ranks students based on their average grades and accumulated points within a specified time period. The query involves complex joins across multiple tables (Students, Houses, Grades, Grades_Enum, Points) and includes window functions for ranking.

```
| Id  | Operation                     | Name         | Rows  | Cost  | Time     |
|-----|-------------------------------|--------------|-------|-------|----------|
|   0 | SELECT STATEMENT              |              |    10 |  1254 | 00:00:16 |
|   1 |  WINDOW SORT                  |              |    10 |  1254 | 00:00:16 |
|   2 |   VIEW                        |              |  9346 |  1250 | 00:00:16 |
|*  3 |    FILTER                     |              |       |       |          |
|   4 |     WINDOW SORT               |              |  9346 |  1250 | 00:00:16 |
|   5 |      HASH GROUP BY            |              |  9346 |  1178 | 00:00:15 |
|   6 |       HASH JOIN               |              |  9346 |   978 | 00:00:12 |
|   7 |        TABLE ACCESS FULL      | HOUSES       |     4 |     3 | 00:00:01 |
|   8 |        HASH JOIN              |              |  9346 |   975 | 00:00:12 |
|   9 |         TABLE ACCESS FULL     | STUDENTS     |  9346 |    87 | 00:00:02 |
|  10 |         HASH JOIN OUTER       |              | 65422 |   882 | 00:00:11 |
|  11 |          HASH JOIN OUTER      |              | 65422 |   456 | 00:00:06 |
|  12 |           HASH JOIN           |              | 65422 |   240 | 00:00:03 |
|* 13 |            TABLE ACCESS FULL  | GRADES       | 65422 |   129 | 00:00:02 |
|  14 |            TABLE ACCESS FULL  | GRADES_ENUM  |     6 |     2 | 00:00:01 |
|* 15 |           TABLE ACCESS FULL   | POINTS       | 65422 |   129 | 00:00:02 |

Predicate Information (identified by operation id):
---------------------------------------------------
   3 - filter(RANK<=:TOP_COUNT)
  13 - filter(("G"."AWARD_DATE">=TO_DATE(:DATE_START) AND "G"."AWARD_DATE"<=TO_DATE(:DATE_END)))
  15 - filter(("P"."AWARD_DATE">=TO_DATE(:DATE_START) AND "P"."AWARD_DATE"<=TO_DATE(:DATE_END)))
```

Candidates for improvements

- **Window Sort Operations (1, 4)**: These operations have the highest cost and consume the most execution time. The first window sort is for the ROW_NUMBER function, the second to apply the final ranking.
- **Hash Group By (5)**: With a cost of 1178, this operation aggregates data for calculating average grades and total points per student.
- **Multiple Hash Joins (6, 8, 10, 11, 12)**: The operations join all tables involved in the query.
- **Full Table Scans (7, 9, 13, 14, 15)**: The query performs full table scans on all five tables.
- **Date Range Filtering (13, 15)**: Post-scan filtering on award_date columns.

6.2 Transactions changing the data

6.2.1 Raising Grades (UPDATE)

This transaction updates student grades based on specific criteria. It identifies students from selected houses with points above a threshold, then updates their worst grades to higher values. The transaction involves complex logic with multiple joins, sorting, and data modification.

```
| Id  | Operation                          | Name          | Rows  | Cost  | Time     |
|-----|------------------------------------|---------------|-------|-------|----------|
|   0 | MERGE STATEMENT                    |               |  4099 |  2240 | 00:00:28 |
|   1 |  MERGE                             | GRADES        |       |       |          |
|   2 |   VIEW                             |               |  4099 |  1785 | 00:00:22 |
|   3 |    WINDOW SORT PUSHED RANK         |               |  4099 |  1785 | 00:00:22 |
|*  4 |     FILTER                         |               |       |       |          |
|*  5 |      HASH JOIN                     |               |  4099 |  1743 | 00:00:21 |
|*  6 |       HASH JOIN                    |               |  4099 |  1625 | 00:00:20 |
|   7 |        HASH JOIN                   |               |   438 |   242 | 00:00:03 |
|   8 |         VIEW                       |               |   438 |   139 | 00:00:02 |
|   9 |          HASH GROUP BY             |               |   438 |   139 | 00:00:02 |
|* 10 |           HASH JOIN                |               |   585 |   119 | 00:00:02 |
|* 11 |            TABLE ACCESS FULL       | HOUSES        |     1 |     3 | 00:00:01 |
|  12 |            HASH JOIN OUTER         |               |  2336 |   115 | 00:00:02 |
|  13 |             TABLE ACCESS FULL      | STUDENTS      |  9346 |    87 | 00:00:02 |
|* 14 |             TABLE ACCESS FULL      | POINTS        | 65422 |    28 | 00:00:01 |
|  15 |         VIEW                       |               |    25 |     3 | 00:00:01 |
|  16 |          TABLE ACCESS FULL         | TEACHERS      |    25 |     3 | 00:00:01 |
|* 17 |        TABLE ACCESS FULL           | GRADES        | 65422 |  1365 | 00:00:17 |
|  18 |       TABLE ACCESS FULL            | GRADES_ENUM   |     6 |     3 | 00:00:01 |
|  19 |   BUFFER SORT                      |               |  4099 |       |          |
|  20 |    TABLE ACCESS BY USER ROWID      | GRADES        |  4099 |   455 | 00:00:06 |

Predicate Information (identified by operation id):
---------------------------------------------------
   4 - filter("GRADE_RANK"<=TO_NUMBER(:NUMBER_OF_AFFECTED))
   5 - access("G"."VALUE"="GE"."SYMBOL")
   6 - access("G"."STUDENT_ID"="ES"."STUDENT_ID" AND "G"."TEACHER_ID"="ET"."ID")
  10 - access("S"."HOUSE_ID"="H"."ID")
  11 - filter(INSTR(','||:HOUSES_LIST||',',','||"H"."NAME"||',')>0)
  14 - filter("P"."STUDENT_ID"="S"."ID")
  17 - filter("G"."VALUE"<>'O' AND "G"."VALUE"<>'E')
```

Candidates for improvements

- **Merge Statement (0) and Window Sort Pushed Rank (3)**: This operations have the highest cost.

- **Multiple Hash Joins (5, 6, 7, 10, 12)**: These operations join multiple tables, with operations 5 and 6 being particularly expensive.
- **Full Table Access on GRADES (17)**: With a cost of 1365, this operation scans 65,422 grade records to find grades eligible for updating.
- **Buffer Sort (19) and Table Access by User ROWID (20)**: These operations are part of the final MERGE update, accessing the actual grade records to be updated.
- **Hash Group By (9)**: This operation aggregates points data to identify eligible students.

6.2.2 Assign subjects (INSERT)

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                     | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT             |                          |  1016 | 52868 |       | 12318   (3)| 00:00:01 |
|   1 |  LOAD TABLE CONVENTIONAL     | STUDENTS_SUBJECTS        |       |       |       |            |          |
|   2 |   HASH UNIQUE                |                          |  1016 | 52868 |       | 12318   (3)| 00:00:01 |
|   3 |    UNION-ALL                 |                          |       |       |       |            |          |
|*  4 |     HASH JOIN                |                          |  1015 | 26390 |       |  6155   (3)| 00:00:01 |
|*  5 |      HASH JOIN ANTI          |                          |    93 |  1860 |       |  6152   (3)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL      | STUDENTS                 |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|   7 |       VIEW                   | VW_NSO_1                 | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|*  8 |        FILTER                |                          |       |       |       |            |          |
|   9 |         SORT GROUP BY        |                          | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 10 |          HASH JOIN           |                          |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  11 |           TABLE ACCESS FULL  | GRADES_ENUM              |     6 |    30 |       |     3   (0)| 00:00:01 |
|  12 |           TABLE ACCESS FULL  | GRADES                   |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 13 |      TABLE ACCESS FULL       | SUBJECTS                 |    76 |   456 |       |     3   (0)| 00:00:01 |
|* 14 |     HASH JOIN ANTI           |                          |     1 |    44 |       |  6158   (3)| 00:00:01 |
|* 15 |      HASH JOIN               |                          |    97 |  3007 |       |    25   (0)| 00:00:01 |
|  16 |       MERGE JOIN CARTESIAN   |                          |    97 |  2328 |       |     6   (0)| 00:00:01 |
|* 17 |        TABLE ACCESS FULL     | SUBJECTS                 |     1 |    20 |       |     3   (0)| 00:00:01 |
|  18 |        BUFFER SORT           |                          |    97 |   388 |       |     3   (0)| 00:00:01 |
|  19 |         TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS   |    97 |   388 |       |     3   (0)| 00:00:01 |
|* 20 |       TABLE ACCESS FULL      | STUDENTS                 |  8011 | 56077 |       |    19   (0)| 00:00:01 |
|* 21 |        TABLE ACCESS FULL     | SUBJECTS                 |     1 |    20 |       |     3   (0)| 00:00:01 |
|  22 |      VIEW                    | VW_NSO_2                 | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|* 23 |       FILTER                 |                          |       |       |       |            |          |
|  24 |        SORT GROUP BY         |                          | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 25 |         HASH JOIN            |                          |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  26 |          TABLE ACCESS FULL   | GRADES_ENUM              |     6 |    30 |       |     3   (0)| 00:00:01 |
|  27 |          TABLE ACCESS FULL   | GRADES                   |  1977K|    16M|       |  2495   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   4 - access("STUDENTS"."YEAR"="SUBJECTS"."YEAR")
   5 - access("STUDENTS"."ID"="STUDENT_ID")
   6 - filter("STUDENTS"."YEAR">=1 AND "STUDENTS"."YEAR"<=7)
   8 - filter(SUM("GRADES_ENUM"."VALUE")/COUNT("GRADES_ENUM"."VALUE")<3.5)
  10 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  13 - filter("SUBJECTS"."YEAR">=1 AND "SUBJECTS"."YEAR"<=7)
```

```
  14 - access("STUDENTS"."ID"="STUDENT_ID")
  15 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID")
  17 - filter("SUBJECTS"."NAME"='Flying')
  20 - filter("STUDENTS"."YEAR"<> (SELECT "SUBJECTS"."YEAR" FROM "SUBJECTS" "SUBJECTS" WHERE
             "SUBJECTS"."NAME"='Flying'))
  21 - filter("SUBJECTS"."NAME"='Flying')
  23 - filter(SUM("GRADES_ENUM"."VALUE")/COUNT("GRADES_ENUM"."VALUE")<3.5)
  25 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
```

Candidates for improvements

- **Hash Unique (2)**: This operation has the highest cost.
- **Multiple Hash Joins (4, 5, 10, 14, 15, 25)**: These operations join multiple tables, with operations 4, 5 and 14 being particularly expensive.
- **Full Table Access (12, 27)**: Both with a cost of 2495 and accessing 1 977 000 rows are very costly.
- **Sort Group By (9, 24)**: These operations are also one of the more expensive

6.2.3 Removing bad students' accomplishments (DELETE)

```
-----------------------------------------------------------------------------------------------
| Id  | Operation                | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT         |              | 74812 |  2264K|       | 16247   (2)| 00:00:01 |
|   1 |  DELETE                  | POINTS       |       |       |       |            |          |
|*  2 |   HASH JOIN RIGHT SEMI   |              | 74812 |  2264K|       | 16247   (2)| 00:00:01 |
|   3 |    VIEW                  | VW_NSO_1     |   468 |  6084 |       |  7356   (3)| 00:00:01 |
|*  4 |     SORT GROUP BY        |              |   468 |  7956 |       |  7356   (3)| 00:00:01 |
|   5 |      VIEW                |              | 25113 |   416K|       |  7356   (3)| 00:00:01 |
|*  6 |       FILTER             |              |       |       |       |            |          |
|   7 |        HASH GROUP BY     |              | 25113 |   539K|    68M|  7356   (3)| 00:00:01 |
|*  8 |         HASH JOIN        |              |  1977K|    41M|       |  2524   (3)| 00:00:01 |
|   9 |          TABLE ACCESS FULL| GRADES_ENUM |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 10 |          TABLE ACCESS FULL| GRADES       |  1977K|    32M|       |  2508   (2)| 00:00:01 |
|* 11 |    TABLE ACCESS FULL     | POINTS       |  1494K|    25M|       |  8881   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("POINTS"."STUDENT_ID"="STUDENT_ID")
   4 - filter(COUNT("Q"."SUBJECT_ID")>=4)
   6 - filter(SUM("GRADES_ENUM"."VALUE")/COUNT("GRADES_ENUM"."VALUE")<4)
   8 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  10 - filter("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd
             hh24:mi:ss') AND "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd
             hh24:mi:ss'))
  11 - filter("POINTS"."VALUE">0)
```

Candidates for improvements

- **Hash Joins (2, 8)**: These operations join multiple tables, which is quite expensive

- **Hash Group By (7)**: This operation groups all students for every subject they attend to calculate average for each of them.
- **Table Access Full (11)**: This operation accesses table points which is the biggest table in our database.

7 Testing application

7.1 Tester script

The script for this task was written in SQL. It runs the whole workload (given as a parameter) multiple times and saves observed times in a `workload_metrics` table. Another procedure saves query plans in a separate `workload_plans` table.

```
1  create or replace PROCEDURE WORKLOAD_TESTER
2  (
3      P_WORKLOAD IN SYS_REFCURSOR
4  ) AS
5      V_NAME VARCHAR2(32);
6      V_SQL VARCHAR2(4000);
7      V_PLAN VARCHAR2(32767);
8      V_PLAN_ID NUMBER;
9      V_START TIMESTAMP;
10     V_END TIMESTAMP;
11     V_ELAPSED NUMBER;
12 BEGIN
13     LOOP
14         FETCH P_WORKLOAD INTO V_NAME, V_SQL;
15         EXIT WHEN P_WORKLOAD%NOTFOUND;
16
17         SELECT MAX(workload_plans.id) INTO V_PLAN_ID
18         FROM workload_plans
19             INNER JOIN workload_sql ON workload_plans.sql_id = workload_sql.id
20         WHERE workload_sql.name = V_NAME;
21         EXECUTE IMMEDIATE 'ALTER SYSTEM FLUSH BUFFER_CACHE';
22         EXECUTE IMMEDIATE 'ALTER SYSTEM FLUSH SHARED_POOL';
23         V_START := SYSTIMESTAMP;
24         EXECUTE IMMEDIATE V_SQL;
25         V_END := SYSTIMESTAMP;
26         ROLLBACK;
27         V_ELAPSED := EXTRACT(SECOND FROM (V_END - V_START)) * 1000;
28         INSERT INTO workload_metrics (plan_id, execution_time) VALUES (v_plan_id, v_elapsed);
29         COMMIT;
30     END LOOP;
31 END WORKLOAD_TESTER;
```

7.2 Running tests

the workload is stored in `workload_sql` table and passed to tester script as a cursor. The tests are being run 10 times in the following way:

```
DECLARE
   workload SYS_REFCURSOR;
BEGIN
    DELETE FROM workload_metrics;
    DELETE FROM workload_plans;

    OPEN workload FOR
        SELECT name, code FROM WORKLOAD_SQL;
    generate_plans(workload);
    CLOSE workload;

    FOR i IN 1..10 LOOP
        OPEN workload FOR
            SELECT name, code FROM WORKLOAD_SQL;
        workload_tester(workload);
        CLOSE workload;
    END LOOP;
END;
```

7.3 Tests results

Transaction Name	Tests Amount	Min Time	Max Time	Avg Time	Plan cost
Grades analysis	10	703	754	724	144 K
House points summary	10	754	790	771	159 K
Best Students Display	10	271	307	288	1254
Raising Grades (UPDATE)	10	2384	2707	2500	2240
Assign subjects (INSERT)	10	3830	4123	3939	12 318
Removing points (DELETE)	10	5159	6616	5422	16 247

7.4 Conclusions

Transactions querying the data in general are much faster than the transactions changing the data. That's because the second type of transactions use selects within them and also because UPDATE, INSERT and DELETE commands take more time.

8 Indexes

8.1 Index on Points(award_date)

Purpose: Optimization of transaction **House Points Summary**.
Query plan points to be changed: 13, 21, 34, 37

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |             |  2986K|   535M|       |   159K  (1)| 00:00:07 |
|   1 |  HASH GROUP BY                      |             |  2986K|   535M|   583M|   159K  (1)| 00:00:07 |
|*  2 |   HASH JOIN                         |             |  2986K|   535M|       | 36394   (3)| 00:00:02 |
|   3 |    TABLE ACCESS FULL                | TEACHERS    |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  4 |    HASH JOIN                        |             |  2986K|   481M|       | 36369   (3)| 00:00:02 |
|   5 |     TABLE ACCESS FULL               | TEACHERS    |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  6 |     HASH JOIN                       |             |  2986K|   427M|       | 36344   (3)| 00:00:02 |
|   7 |      VIEW                           |             |     4 |    64 |       |  9127   (4)| 00:00:01 |
|   8 |       SORT GROUP BY                 |             |     4 |   116 |       |  9127   (4)| 00:00:01 |
|   9 |        VIEW                         |             |   858 | 24882 |       |  9127   (4)| 00:00:01 |
|  10 |         HASH GROUP BY               |             |   858 | 19734 |       |  9127   (4)| 00:00:01 |
|* 11 |          HASH JOIN                  |             |  2986K|    65M|       |  8934   (2)| 00:00:01 |
|  12 |           TABLE ACCESS FULL         | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |

|* 13 |           TABLE ACCESS FULL         | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |

|* 14 |      HASH JOIN                      |             |  2986K|   381M|       | 27197   (3)| 00:00:02 |
|  15 |       VIEW                          |             |     4 |   116 |       |  9131   (4)| 00:00:01 |
|  16 |        SORT GROUP BY                |             |     4 |   116 |       |  9131   (4)| 00:00:01 |
|  17 |         VIEW                        |             |  3103 | 89987 |       |  9131   (4)| 00:00:01 |
|  18 |          HASH GROUP BY              |             |  3103 | 83781 |       |  9131   (4)| 00:00:01 |
|* 19 |           HASH JOIN                 |             |  2986K|    76M|       |  8938   (2)| 00:00:01 |
|  20 |            TABLE ACCESS FULL        | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |

|* 21 |            TABLE ACCESS FULL        | POINTS      |  2986K|    56M|       |  8899   (1)| 00:00:01 |

|* 22 |       HASH JOIN                     |             |  2986K|   299M|       | 18046   (3)| 00:00:01 |
|  23 |        VIEW                         | VW_GBF_15   |  9346 |   812K|       |  9131   (4)| 00:00:01 |
|* 24 |         HASH JOIN                   |             |  9346 |   930K|       |  9131   (4)| 00:00:01 |
|  25 |          MERGE JOIN                 |             |     4 |   380 |       |  9112   (4)| 00:00:01 |
|  26 |           TABLE ACCESS BY INDEX ROWID| HOUSES      |     4 |    56 |       |     2   (0)| 00:00:01 |
|  27 |            INDEX FULL SCAN          | SYS_C008743 |     4 |       |       |     1   (0)| 00:00:01 |
|* 28 |           SORT JOIN                 |             |     4 |   324 |       |  9110   (4)| 00:00:01 |
|  29 |            VIEW                     |             |     4 |   324 |       |  9109   (4)| 00:00:01 |
|  30 |             SORT GROUP BY           |             |     4 |   156 |       |  9109   (4)| 00:00:01 |
|* 31 |              HASH JOIN              |             |  9346 |   355K|       |  9108   (4)| 00:00:01 |
|  32 |               VIEW                  |             |  9346 |   155K|       |  9088   (4)| 00:00:01 |
|  33 |                HASH GROUP BY        |             |  9346 |   146K|       |  9088   (4)| 00:00:01 |

|* 34 |                 TABLE ACCESS FULL    | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |

|  35 |               TABLE ACCESS FULL      | STUDENTS    |  9346 |   200K|       |    19   (0)| 00:00:01 |
|  36 |          TABLE ACCESS FULL           | STUDENTS    |  9346 | 65422 |       |    19   (0)| 00:00:01 |

|* 37 |        TABLE ACCESS FULL             | POINTS      |  2986K|    45M|       |  8895   (1)| 00:00:01 |

-------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("T2"."ID"="T"."LEAST_FAVOURITE_TEACHER")
   4 - access("T1"."ID"="T"."FAVOURITE_TEACHER")
   6 - access("M"."HOUSE_ID"="ITEM_2")
  11 - access("POINTS"."STUDENT_ID"="STUDENTS"."ID")
```

```
  13 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  14 - access("T"."HOUSE_ID"="ITEM_2")
  19 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  21 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  22 - access("ITEM_1"="POINTS"."STUDENT_ID")
  24 - access("HOUSES"."ID"="STUDENTS"."HOUSE_ID")
  28 - access("S"."HOUSE_ID"="HOUSES"."ID")
       filter("S"."HOUSE_ID"="HOUSES"."ID")
  31 - access("STUDENTS"."ID"="Q"."STUDENT_ID")
  34 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  37 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-30 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

8.1.1 Conclusion

Dates are often selected as range and can have many values, that's why it's best to use **B-tree** index type in this case.

8.2 Index on Grades(student_id, award_date)

Purpose: Optimization of transaction **Grades Analysis**.
Query plan points to be changed: 17, 29, 36

```
-------------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name                 | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                      |    84M|    17G|       |  6392K  (1)| 00:04:10 |
|   1 |  SORT ORDER BY                 |                      |    84M|    17G|    18G|  6392K  (1)| 00:04:10 |
|*  2 |   HASH JOIN                    |                      |    84M|    17G|       |  2384K  (1)| 00:01:34 |
|   3 |    TABLE ACCESS FULL           | SUBJECTS             |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                   |                      |    84M|    15G|       |  2383K  (1)| 00:01:34 |
|   5 |     MERGE JOIN CARTESIAN       |                      |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN      |                      |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL        | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT              |                      |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL       | GRADES_ENUM          |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT               |                      |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL        | SUBJECTS             |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                       |                      |    84M|    13G|       |  2382K  (1)| 00:01:34 |
|  13 |      SORT GROUP BY             |                      |    84M|     9G|    10G|  2382K  (1)| 00:01:34 |
|* 14 |       HASH JOIN                |                      |    84M|     9G|    30M| 26186   (4)| 00:00:02 |
|  15 |        TABLE ACCESS FULL       | GRADES               |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
|* 16 |        HASH JOIN               |                      |   397K|    45M|       | 19118   (2)| 00:00:01 |

|* 17 |         HASH JOIN              |                      |  1256 |   128K|       | 10204   (3)| 00:00:01 |

|  18 |          JOIN FILTER CREATE    | :BF0000              |    23 |  1426 |       |  2548   (3)| 00:00:01 |
|* 19 |          HASH JOIN OUTER       |                      |    23 |  1426 |       |  2548   (3)| 00:00:01 |
|* 20 |           HASH JOIN ANTI       |                      |    23 |  1035 |       |  2545   (3)| 00:00:01 |
|* 21 |            HASH JOIN           |                      |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL  | HOUSES               |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 23 |             TABLE ACCESS FULL  | STUDENTS             |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  24 |            VIEW                | VW_NSO_1             |   190K|   745K|       |  2521   (3)| 00:00:01 |
|* 25 |             HASH JOIN          |                      |   190K|  3541K|       |  2521   (3)| 00:00:01 |
|* 26 |              TABLE ACCESS FULL | GRADES_ENUM          |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL| GRADES_ENUM          |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |               TABLE ACCESS FULL| GRADES_ENUM          |     1 |     5 |       |     3   (0)| 00:00:01 |
```

```
|* 29 |               TABLE ACCESS FULL | GRADES                 |  1957K|    26M|       |  2505   (2)| 00:00:01 |

|  30 |            TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  31 |          VIEW                   |                        |   502K|    20M|       |  7653   (3)| 00:00:01 |
|  32 |           HASH GROUP BY         |                        |   502K|  6866K|    45M|  7653   (3)| 00:00:01 |
|* 33 |            HASH JOIN            |                        |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  34 |             TABLE ACCESS FULL   | GRADES_ENUM            |     6 |    30 |       |     3   (0)| 00:00:01 |
|  35 |             JOIN FILTER USE     | :BF0000                |  1977K|    16M|       |  2495   (2)| 00:00:01 |

|* 36 |              TABLE ACCESS FULL  | GRADES                 |  1977K|    16M|       |  2495   (2)| 00:00:01 |

|* 37 |         TABLE ACCESS FULL       | POINTS                 |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("Q"."WORST_SUBJECT"="S2"."ID")
   4 - access("Q"."BEST_SUBJECT"="S1"."ID" AND "Q"."MIN_GRADE"="G2"."VALUE" AND
              "Q"."MAX_GRADE"="G1"."VALUE")
  14 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  16 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  17 - access("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
  19 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
  20 - access("STUDENTS"."ID"="ID")
  21 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  22 - filter("HOUSES"."NAME"='Gryffindor')
  23 - filter("STUDENTS"."YEAR">=1 AND "STUDENTS"."YEAR"<=7)
  25 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  26 - filter("GRADES_ENUM"."VALUE"< (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM" WHERE
              "GRADES_ENUM"."SYMBOL"='T') OR "GRADES_ENUM"."VALUE"> (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM"
              "GRADES_ENUM" WHERE "GRADES_ENUM"."SYMBOL"='O'))
  27 - filter("GRADES_ENUM"."SYMBOL"='T')
  28 - filter("GRADES_ENUM"."SYMBOL"='O')
  29 - filter("GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  33 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  36 - filter(SYS_OP_BLOOM_FILTER(:BF0000,"GRADES"."STUDENT_ID"))
  37 - filter("POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

Students' IDs and award dates appear multiple times in our rows. However, there are still many of them, that's why it's not obvious if bitmap indexes would be better than b-tree indexes. For that reason, we've compared query plans for both of these types.

8.2.1 Experiment 1 - B-tree index type

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                   | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                        |    12M|  2592M|       |  1662K  (1)| 00:01:05 |
|   1 |  SORT ORDER BY                           |                        |    12M|  2592M|  2757M|  1662K  (1)| 00:01:05 |
|*  2 |   HASH JOIN                              |                        |    12M|  2592M|       |  1073K  (1)| 00:00:42 |
|   3 |    TABLE ACCESS FULL                     | SUBJECTS               |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                        |    12M|  2356M|       |  1073K  (1)| 00:00:42 |
|   5 |     MERGE JOIN CARTESIAN                 |                        |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN                |                        |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL                  | GRADES_ENUM            |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT                        |                        |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL                 | GRADES_ENUM            |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                         |                        |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL                  | SUBJECTS               |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                                 |                        |    12M|  2003M|       |  1073K  (1)| 00:00:42 |
|  13 |      SORT GROUP BY                       |                        |    12M|  1425M|    10G|  1073K  (1)| 00:00:42 |
```

```
|* 14 |       HASH JOIN                              |                          |    84M|  9698M|    30M| 44918   (2)| 00:00:02 |
|  15 |        INDEX FAST FULL SCAN                   | TESTINDEX1               |  1977K|  7723K|       |  1736   (2)| 00:00:01 |
|* 16 |        HASH JOIN                              |                          |   397K|    44M|       | 38682   (1)| 00:00:02 |
|  17 |         NESTED LOOPS                          |                          |  1256 |   123K|       | 29768   (1)| 00:00:02 |
|* 18 |          HASH JOIN OUTER                      |                          |    23 |  1380 |       | 28087   (1)| 00:00:02 |
|  19 |           NESTED LOOPS ANTI                   |                          |    23 |   989 |       | 28084   (1)| 00:00:02 |
|* 20 |            HASH JOIN                          |                          |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL                 | HOUSES                   |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL                 | STUDENTS                 |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  23 |            VIEW PUSHED PREDICATE              | VW_NSO_1                 |     2 |     4 |       |    12   (0)| 00:00:01 |
|  24 |             NESTED LOOPS                      |                          |     2 |    38 |       |    12   (0)| 00:00:01 |
|  25 |              NESTED LOOPS                     |                          |    21 |    38 |       |    12   (0)| 00:00:01 |
|* 26 |               TABLE ACCESS FULL               | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |                TABLE ACCESS FULL              | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |                TABLE ACCESS FULL              | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 29 |               INDEX RANGE SCAN                | TESTINDEX1               |    21 |       |       |     2   (0)| 00:00:01 |
|* 30 |              TABLE ACCESS BY INDEX ROWID      | GRADES                   |     3 |    42 |       |     9   (0)| 00:00:01 |
|  31 |           TABLE ACCESS FULL                   | QUIDDITCH_TEAM_MEMBERS   |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  32 |          VIEW PUSHED PREDICATE                |                          |     1 |    41 |       |    73   (2)| 00:00:01 |
|  33 |           SORT GROUP BY                       |                          |    51 |   714 |       |    73   (2)| 00:00:01 |
|* 34 |            HASH JOIN                          |                          |   212 |  2968 |       |    72   (0)| 00:00:01 |
|  35 |             TABLE ACCESS FULL                 | GRADES_ENUM              |     6 |    30 |       |     3   (0)| 00:00:01 |
|  36 |             TABLE ACCESS BY INDEX ROWID BATCHED| GRADES                   |   212 |  1908 |       |    69   (0)| 00:00:01 |
|* 37 |              INDEX RANGE SCAN                 | TESTINDEX1               |   212 |       |       |     3   (0)| 00:00:01 |
|* 38 |         TABLE ACCESS FULL                     | POINTS                   |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("Q"."WORST_SUBJECT"="S2"."ID")
   4 - access("Q"."BEST_SUBJECT"="S1"."ID" AND "Q"."MIN_GRADE"="G2"."VALUE" AND "Q"."MAX_GRADE"="G1"."VALUE")
  14 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  16 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  18 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
  20 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  21 - filter("HOUSES"."NAME"='Gryffindor')
  22 - filter("STUDENTS"."YEAR">=1 AND "STUDENTS"."YEAR"<=7)
  26 - filter("GRADES_ENUM"."VALUE"< (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM" WHERE
              "GRADES_ENUM"."SYMBOL"='T') OR "GRADES_ENUM"."VALUE"> (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM"
              WHERE "GRADES_ENUM"."SYMBOL"='O'))
  27 - filter("GRADES_ENUM"."SYMBOL"='T')
  28 - filter("GRADES_ENUM"."SYMBOL"='O')
  29 - access("GRADES"."STUDENT_ID"="STUDENTS"."ID" AND "GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00',
              'syyyy-mm-dd hh24:mi:ss') AND "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  30 - filter("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  34 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  37 - access("GRADES"."STUDENT_ID"="STUDENTS"."ID")
  38 - filter("POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

8.2.2 Experiment 2 - Bitmap index type

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                     |    12M|  2592M|       |  1658K  (1)| 00:01:05 |
|   1 |  SORT ORDER BY                           |                     |    12M|  2592M|  2757M|  1658K  (1)| 00:01:05 |
|*  2 |   HASH JOIN                              |                     |    12M|  2592M|       |  1069K  (1)| 00:00:42 |
|   3 |    TABLE ACCESS FULL                     | SUBJECTS            |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                     |    12M|  2356M|       |  1069K  (1)| 00:00:42 |
|   5 |     MERGE JOIN CARTESIAN                 |                     |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN                |                     |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL                  | GRADES_ENUM         |     6 |    30 |       |     3   (0)| 00:00:01 |
```

```
|   8 |       BUFFER SORT                              |                          |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL                       | GRADES_ENUM              |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                               |                          |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL                        | SUBJECTS                 |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                                       |                          |    12M|  2003M|       |  1069K  (1)| 00:00:42 |
|  13 |      SORT GROUP BY                             |                          |    12M|  1425M|    10G|  1069K  (1)| 00:00:42 |
|* 14 |       HASH JOIN                                |                          |    84M|  9698M|    30M| 41009   (3)| 00:00:02 |
|  15 |        TABLE ACCESS FULL                       | GRADES                   |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
|* 16 |        HASH JOIN                               |                          |   397K|    44M|       | 34016   (1)| 00:00:02 |
|  17 |         NESTED LOOPS                           |                          |  1256 |   123K|       | 25102   (1)| 00:00:01 |
|* 18 |          HASH JOIN OUTER                       |                          |    23 |  1380 |       | 23946   (1)| 00:00:01 |
|  19 |           NESTED LOOPS ANTI                    |                          |    23 |   989 |       | 23943   (1)| 00:00:01 |
|* 20 |            HASH JOIN                           |                          |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL                  | HOUSES                   |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL                  | STUDENTS                 |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  23 |            VIEW PUSHED PREDICATE               | VW_NSO_1                 |     2 |     4 |       |    10   (0)| 00:00:01 |
|* 24 |             HASH JOIN                          |                          |     2 |    38 |       |    10   (0)| 00:00:01 |
|* 25 |              TABLE ACCESS FULL                 | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 26 |               TABLE ACCESS FULL                | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL                | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|  28 |              TABLE ACCESS BY INDEX ROWID BATCHED| GRADES                   |    21 |   294 |       |     7   (0)| 00:00:01 |
|  29 |               BITMAP CONVERSION TO ROWIDS      |                          |       |       |       |            |          |
|* 30 |                BITMAP INDEX RANGE SCAN         | TESTINDEX1               |       |       |       |            |          |
|  31 |           TABLE ACCESS FULL                    | QUIDDITCH_TEAM_MEMBERS   |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  32 |          VIEW PUSHED PREDICATE                 |                          |     1 |    41 |       |    50   (2)| 00:00:01 |
|  33 |           SORT GROUP BY                        |                          |    51 |   714 |       |    50   (2)| 00:00:01 |
|* 34 |            HASH JOIN                           |                          |   212 |  2968 |       |    49   (0)| 00:00:01 |
|  35 |             TABLE ACCESS FULL                  | GRADES_ENUM              |     6 |    30 |       |     3   (0)| 00:00:01 |
|  36 |             TABLE ACCESS BY INDEX ROWID BATCHED| GRADES                   |   212 |  1908 |       |    46   (0)| 00:00:01 |
|  37 |              BITMAP CONVERSION TO ROWIDS       |                          |       |       |       |            |          |
|* 38 |               BITMAP INDEX RANGE SCAN          | TESTINDEX1               |       |       |       |            |          |
|* 39 |         TABLE ACCESS FULL                      | POINTS                   |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("Q"."WORST_SUBJECT"="S2"."ID")
   4 - access("Q"."BEST_SUBJECT"="S1"."ID" AND "Q"."MIN_GRADE"="G2"."VALUE" AND "Q"."MAX_GRADE"="G1"."VALUE")
  14 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  16 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  18 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
  20 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  21 - filter("HOUSES"."NAME"='Gryffindor')
  22 - filter("STUDENTS"."YEAR">=1 AND "STUDENTS"."YEAR"<=7)
  24 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  25 - filter("GRADES_ENUM"."VALUE"< (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM" WHERE
              "GRADES_ENUM"."SYMBOL"='T') OR "GRADES_ENUM"."VALUE"> (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM"
              WHERE "GRADES_ENUM"."SYMBOL"='O'))
  26 - filter("GRADES_ENUM"."SYMBOL"='T')
  27 - filter("GRADES_ENUM"."SYMBOL"='O')
  30 - access("GRADES"."STUDENT_ID"="STUDENTS"."ID" AND "GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00',
              'syyyy-mm-dd hh24:mi:ss') AND "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter("GRADES"."STUDENT_ID"="STUDENTS"."ID" AND "GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00',
              'syyyy-mm-dd hh24:mi:ss') AND "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  34 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  38 - access("GRADES"."STUDENT_ID"="STUDENTS"."ID")
```

8.2.3 Conclusion

Both indexes greatly improve the efficiency of the sample query. In this particular case the **bitmap** index is slightly better than the b-tree.

8.3 Index on Points(EXTRACT(MONTH FROM award_date))

Purpose: Optimization of transaction **House Points Summary**.
Query plan points to be changed: 37

```
Plan hash value: 1621774207
```

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		2986K	535M		159K (1)	00:00:07
1	HASH GROUP BY		2986K	535M	583M	159K (1)	00:00:07
* 2	HASH JOIN		2986K	535M		36394 (3)	00:00:02
3	TABLE ACCESS FULL	TEACHERS	1097	20843		5 (0)	00:00:01
* 4	HASH JOIN		2986K	481M		36369 (3)	00:00:02
5	TABLE ACCESS FULL	TEACHERS	1097	20843		5 (0)	00:00:01
* 6	HASH JOIN		2986K	427M		36344 (3)	00:00:02
7	VIEW		4	64		9127 (4)	00:00:01
8	SORT GROUP BY		4	116		9127 (4)	00:00:01
9	VIEW		4	116		9127 (4)	00:00:01
10	HASH GROUP BY		4	144		9127 (4)	00:00:01
* 11	HASH JOIN		2986K	102M		8934 (2)	00:00:01
12	TABLE ACCESS FULL	STUDENTS	9346	65422		19 (0)	00:00:01
* 13	TABLE ACCESS FULL	POINTS	2986K	82M		8895 (1)	00:00:01
* 14	HASH JOIN		2986K	381M		27197 (3)	00:00:02
15	VIEW		4	116		9131 (4)	00:00:01
16	SORT GROUP BY		4	116		9131 (4)	00:00:01
17	VIEW		3103	89987		9131 (4)	00:00:01
18	HASH GROUP BY		3103	83781		9131 (4)	00:00:01
* 19	HASH JOIN		2986K	76M		8938 (2)	00:00:01
20	TABLE ACCESS FULL	STUDENTS	9346	65422		19 (0)	00:00:01
* 21	TABLE ACCESS FULL	POINTS	2986K	56M		8899 (1)	00:00:01
* 22	HASH JOIN		2986K	299M		18046 (3)	00:00:01
23	VIEW	VW_GBF_15	9346	812K		9131 (4)	00:00:01
* 24	HASH JOIN		9346	930K		9131 (4)	00:00:01
25	MERGE JOIN		4	380		9112 (4)	00:00:01
26	TABLE ACCESS BY INDEX ROWID	HOUSES	4	56		2 (0)	00:00:01
27	INDEX FULL SCAN	SYS_C008743	4			1 (0)	00:00:01
* 28	SORT JOIN		4	324		9110 (4)	00:00:01
29	VIEW		4	324		9109 (4)	00:00:01
30	SORT GROUP BY		4	156		9109 (4)	00:00:01
* 31	HASH JOIN		9346	355K		9108 (4)	00:00:01
32	VIEW		9346	155K		9088 (4)	00:00:01
33	HASH GROUP BY		9346	146K		9088 (4)	00:00:01
* 34	TABLE ACCESS FULL	POINTS	2986K	45M		8895 (1)	00:00:01
35	TABLE ACCESS FULL	STUDENTS	9346	200K		19 (0)	00:00:01
36	TABLE ACCESS FULL	STUDENTS	9346	65422		19 (0)	00:00:01
* 37	TABLE ACCESS FULL	POINTS	2986K	45M		8895 (1)	00:00:01

```
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("T2"."ID"="T"."LEAST_FAVOURITE_TEACHER")
   4 - access("T1"."ID"="T"."FAVOURITE_TEACHER")
   6 - access("M"."HOUSE_ID"="ITEM_2")
  11 - access("POINTS"."STUDENT_ID"="STUDENTS"."ID")
  13 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  14 - access("T"."HOUSE_ID"="ITEM_2")
  19 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  21 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

```
22 - access("ITEM_1"="POINTS"."STUDENT_ID")
24 - access("HOUSES"."ID"="STUDENTS"."HOUSE_ID")
28 - access("S"."HOUSE_ID"="HOUSES"."ID")
     filter("S"."HOUSE_ID"="HOUSES"."ID")
31 - access("STUDENTS"."ID"="Q"."STUDENT_ID")
34 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
            "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
37 - filter("POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
            "POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

8.3.1 Conclusion

One of the often used functions in filtering is month extraction. As there is small amount of months, the type of this **functional** index should be **bitmap**.

9 Indexes - implementation

9.1 Implemented indexes

Indexes that have been selected after experiments, and have been tested during cost comparison had been created as follows:

```
CREATE BITMAP INDEX index1 ON grades(student_id, award_date);
CREATE INDEX index2 ON points(student_id, award_date);
CREATE INDEX index3 ON grades(EXTRACT(MONTH FROM award_date));
```

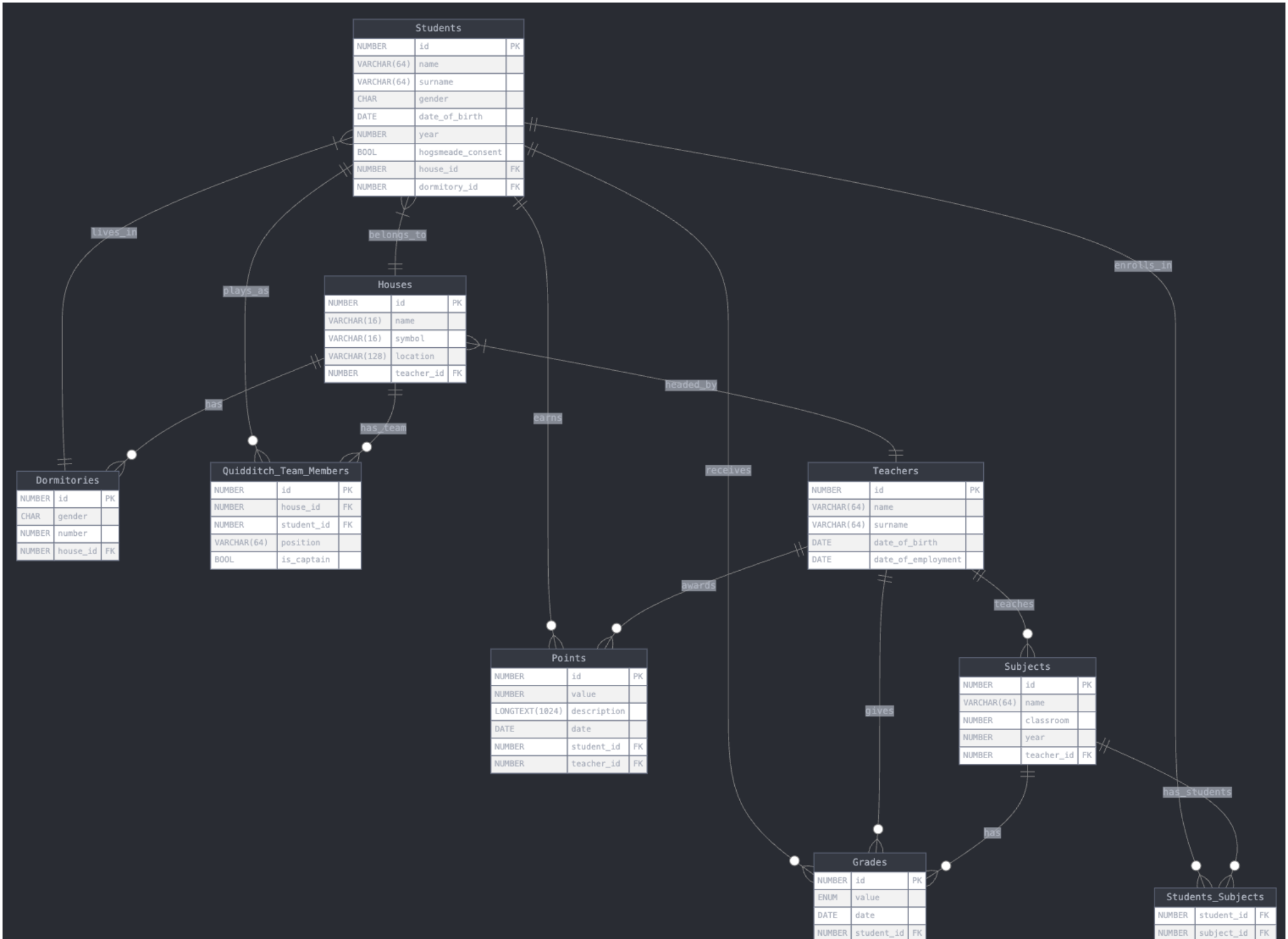
9.2 Cost comparison

We ran the whole workload before and after creating indexes, **10 times** in each case. Creating them positively influenced 4 out of 6 query plans in our workload. What is very interesting is that for 3 of them the actual running times didn't actually improve - actually quite the contrary; now they are a lot bigger. The changes of times of procedures, of which query plans didn't change, are a result of natural differences in each running time.

Procedure name	**Min. measured time**				**Max. measured time**				**Average measured time**				**Query plan cost**			
	Default	Index	Diff.	Profit	Default	Index	Diff.	Profit	Default	Index	Diff.	Profit	Default	Index	Diff.	Profit
Grades analysis	552	1078	-526	-95%	831	1560	-729	-88%	628.4	1335.9	-707.5	-113%	6392 K	1658 K	4734 K	74%
Points summary	392	757	-365	-93%	567	1098	-531	-94%	452.6	935.4	-482.8	-107%	39 M	38 M	1 M	3%
Best students	232	244	-12	-5%	303	331	-28	-9%	262.7	278.6	-15.9	-6%	22 M	19 M	3 M	14%
Raise grades	1925	1883	42	2%	2809	2712	97	3%	2168.4	2303.2	-134.8	-6%	21196	21196	0	0%
Assign subjects	2757	2835	-78	-3%	3934	4081	-147	-4%	3061.4	3459.5	-398.1	-13%	12318	12318	0	0%
Remove points	26407	3279	23128	88%	34829	31564	3265	9%	29593.6	16934.5	12659.1	43%	11394	6771	4623	41%
Whole workload	32265	10076	22189	69%	43273	41346	1927	4%	36167,1	24247,1	11920,6	33%	67437 K	58698 K	8739 K	13%

9.3 Query plans comparison

Created indexes have affected four of all six procedures' query plans, including Grade Analysis, Points Summary, Best Students Display, and Removing Points. Their changed query plans are presented below.


Students
NUMBER id PK
VARCHAR(64) name
VARCHAR(64) surname
CHAR gender
DATE date_of_birth
NUMBER year
BOOL hogsmeade_consent
NUMBER house_id FK
NUMBER dormitory_id FK
lives_in
belongs_to
plays_as
enrolls_in
Houses
NUMBER id PK
VARCHAR(16) name
VARCHAR(16) symbol
VARCHAR(128) location
NUMBER teacher_id FK
has
has_team
headed_by
earns
receives
Dormitories
NUMBER id PK
CHAR gender
NUMBER number
NUMBER house_id FK
Quidditch_Team_Members
NUMBER id PK
NUMBER house_id FK
NUMBER student_id FK
VARCHAR(64) position
BOOL is_captain
Teachers
NUMBER id PK
VARCHAR(64) name
VARCHAR(64) surname
DATE date_of_birth
DATE date_of_employment
awards
teaches
Points
NUMBER id PK
NUMBER value
LONGTEXT(1024) description
DATE date
NUMBER student_id FK
NUMBER teacher_id FK
Subjects
NUMBER id PK
VARCHAR(64) name
NUMBER classroom
NUMBER year
NUMBER teacher_id FK
gives
has
has_students
Grades
NUMBER id PK
ENUM value
DATE date
NUMBER student_id FK
Students_Subjects
NUMBER student_id FK
NUMBER subject_id FK

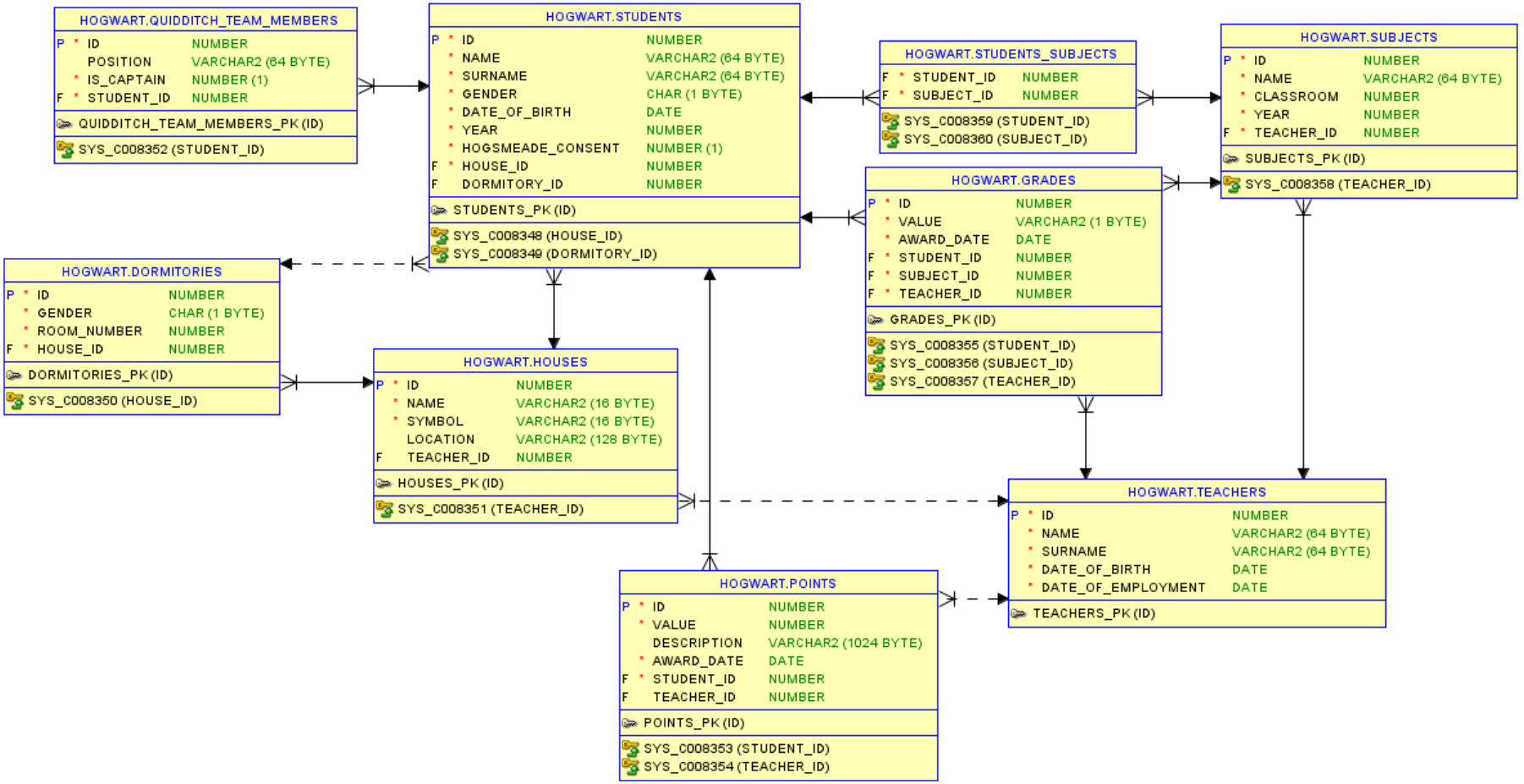


Figure 2: Database schema generated with Oracle SQL Developer Software

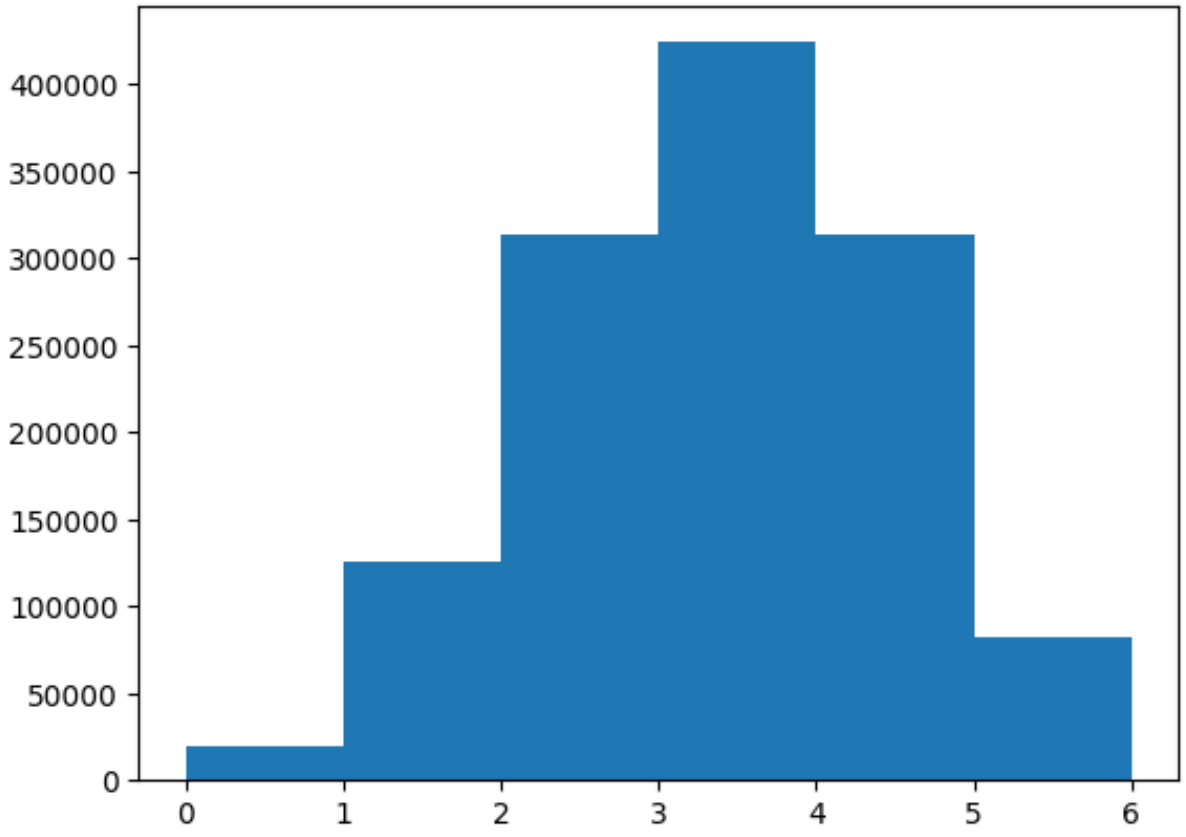


Figure 3: Grades distribution

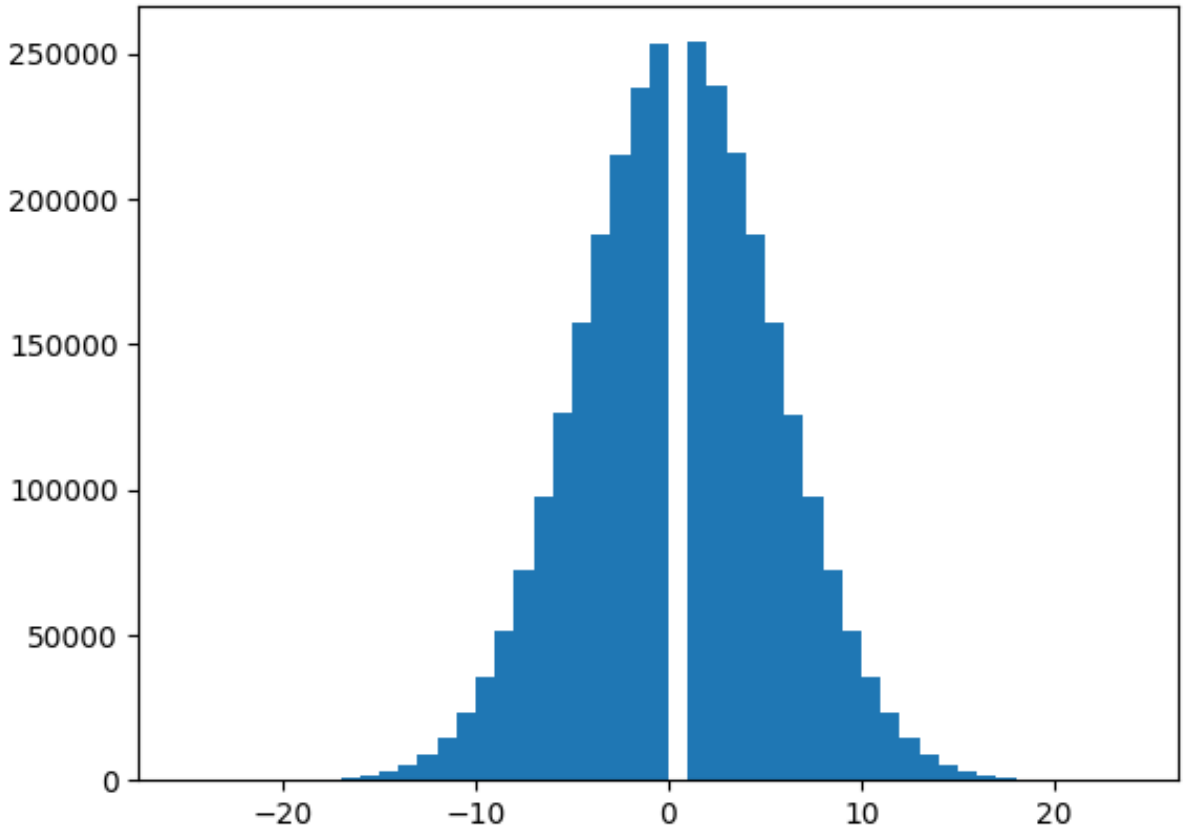


Figure 4: Points distribution

9.3.1 Grades analysis without indexes

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                         |    84M|    17G|       |  6392K  (1)| 00:04:10 |
|   1 |  SORT ORDER BY                 |                         |    84M|    17G|    18G|  6392K  (1)| 00:04:10 |
|*  2 |   HASH JOIN                    |                         |    84M|    17G|       |  2384K  (1)| 00:01:34 |
|   3 |    TABLE ACCESS FULL           | SUBJECTS                |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                   |                         |    84M|    15G|       |  2383K  (1)| 00:01:34 |
|   5 |     MERGE JOIN CARTESIAN       |                         |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN      |                         |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL        | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT              |                         |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL       | GRADES_ENUM             |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT               |                         |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL        | SUBJECTS                |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                       |                         |    84M|    13G|       |  2382K  (1)| 00:01:34 |
|  13 |      SORT GROUP BY             |                         |    84M|     9G|   10G|  2382K  (1)| 00:01:34 |
|* 14 |       HASH JOIN                |                         |    84M|     9G|   30M| 26186   (4)| 00:00:02 |
|  15 |        TABLE ACCESS FULL       | GRADES                  |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
|* 16 |        HASH JOIN               |                         |   397K|    45M|       | 19118   (2)| 00:00:01 |
|* 17 |         HASH JOIN              |                         |  1256 |   128K|       | 10204   (3)| 00:00:01 |
|  18 |          JOIN FILTER CREATE    | :BF0000                 |    23 |  1426 |       |  2548   (3)| 00:00:01 |
|                                                                                                                     |
|* 19 |           HASH JOIN OUTER      |                         |    23 |  1426 |       |  2548   (3)| 00:00:01 |
|* 20 |            HASH JOIN ANTI      |                         |    23 |  1035 |       |  2545   (3)| 00:00:01 |
|                                                                                                                     |
|* 21 |             HASH JOIN          |                         |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 22 |              TABLE ACCESS FULL | HOUSES                  |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 23 |              TABLE ACCESS FULL | STUDENTS                |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  24 |             VIEW               | VW_NSO_1                |   190K|   745K|       |  2521   (3)| 00:00:01 |
|                                                                                                                     |
|* 25 |              HASH JOIN         |                         |   190K|  3541K|       |  2521   (3)| 00:00:01 |
|* 26 |               TABLE ACCESS FULL| GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL| GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |               TABLE ACCESS FULL| GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 29 |              TABLE ACCESS FULL | GRADES                  |  1957K|    26M|       |  2505   (2)| 00:00:01 |
|                                                                                                                     |
|                                                                                                                     |
|                                                                                                                     |
|  30 |           TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS  |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  31 |          VIEW                  |                         |   502K|    20M|       |  7653   (3)| 00:00:01 |
|                                                                                                                     |
|  32 |           HASH GROUP BY        |                         |   502K|  6866K|    45M|  7653   (3)| 00:00:01 |
|                                                                                                                     |
|* 33 |            HASH JOIN           |                         |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  34 |             TABLE ACCESS FULL  | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|  35 |             JOIN FILTER USE    | :BF0000                 |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 36 |              TABLE ACCESS FULL | GRADES                  |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|                                                                                                                     |
|                                                                                                                     |
|                                                                                                                     |
|* 37 |         TABLE ACCESS FULL      | POINTS                  |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

9.3.2 Grades analysis with indexes

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                         |    12M|  2592M|       |  1658K  (1)| 00:01:05 |
|   1 |  SORT ORDER BY                           |                         |    12M|  2592M|  2757M|  1658K  (1)| 00:01:05 |
|*  2 |   HASH JOIN                              |                         |    12M|  2592M|       |  1069K  (1)| 00:00:42 |
|   3 |    TABLE ACCESS FULL                     | SUBJECTS                |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                         |    12M|  2356M|       |  1069K  (1)| 00:00:42 |
|   5 |     MERGE JOIN CARTESIAN                 |                         |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN                |                         |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL                  | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT                        |                         |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL                 | GRADES_ENUM             |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                         |                         |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL                  | SUBJECTS                |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                                 |                         |    12M|  2003M|       |  1069K  (1)| 00:00:42 |
|  13 |      SORT GROUP BY                       |                         |    12M|  1425M|   10G|  1069K  (1)| 00:00:42 |
|* 14 |       HASH JOIN                          |                         |    84M|  9698M|   30M| 41009   (3)| 00:00:02 |
|  15 |        TABLE ACCESS FULL                 | GRADES                  |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
|* 16 |        HASH JOIN                         |                         |   397K|    44M|       | 34016   (1)| 00:00:02 |
|                                                                                                                     |
|                                                                                                                     |
|  17 |         NESTED LOOPS                     |                         |  1256 |   123K|       | 25102   (1)| 00:00:01 |
|* 18 |          HASH JOIN OUTER                 |                         |    23 |  1380 |       | 23946   (1)| 00:00:01 |
|                                                                                                                     |
|  19 |           NESTED LOOPS ANTI              |                         |    23 |   989 |       | 23943   (1)| 00:00:01 |
|* 20 |            HASH JOIN                     |                         |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL            | HOUSES                  |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL            | STUDENTS                |  9346 |   246K|       |    19   (0)| 00:00:01 |
|                                                                                                                     |
|  23 |            VIEW PUSHED PREDICATE         | VW_NSO_1                |     2 |     4 |       |    10   (0)| 00:00:01 |
|* 24 |            HASH JOIN                     |                         |     2 |    38 |       |    10   (0)| 00:00:01 |
|* 25 |             TABLE ACCESS FULL            | GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 26 |              TABLE ACCESS FULL           | GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |             TABLE ACCESS FULL            | GRADES_ENUM             |     1 |     5 |       |     3   (0)| 00:00:01 |
|                                                                                                                     |
|  28 |            TABLE ACCESS BY INDEX ROWID BATCHED| GRADES             |    21 |   294 |       |     7   (0)| 00:00:01 |
|  29 |             BITMAP CONVERSION TO ROWIDS  |                         |       |       |       |            |          |
|* 30 |              BITMAP INDEX RANGE SCAN     | INDEX1                  |       |       |       |            |          |
|  31 |           TABLE ACCESS FULL              | QUIDDITCH_TEAM_MEMBERS  |    97 |  1649 |       |     3   (0)| 00:00:01 |
|                                                                                                                     |
|  32 |          VIEW PUSHED PREDICATE           |                         |     1 |    41 |       |    50   (2)| 00:00:01 |
|                                                                                                                     |
|  33 |           SORT GROUP BY                  |                         |    51 |   714 |       |    50   (2)| 00:00:01 |
|* 34 |            HASH JOIN                     |                         |   212 |  2968 |       |    49   (0)| 00:00:01 |
|  35 |             TABLE ACCESS FULL            | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|                                                                                                                     |
|                                                                                                                     |
|  36 |             TABLE ACCESS BY INDEX ROWID BATCHED | GRADES           |   212 |  1908 |       |    46   (0)| 00:00:01 |
|  37 |              BITMAP CONVERSION TO ROWIDS |                         |       |       |       |            |          |
|* 38 |               BITMAP INDEX RANGE SCAN    | INDEX1                  |       |       |       |            |          |
|* 39 |         TABLE ACCESS FULL                | POINTS                  |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

9.3.3 Points summary without indexes

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                         |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|*  1 |  HASH JOIN                   |                         |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|   2 |   TABLE ACCESS FULL          | TEACHERS                |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                         |  1105M|   186G|       |    39M  (1)| 00:25:29 |
|   4 |    MERGE JOIN CARTESIAN      |                         |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                         |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS                |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                         |  1105M|   161G|       |    39M  (1)| 00:25:29 |
|   9 |     SORT GROUP BY            |                         |  1105M|   118G|   135G|    39M  (1)| 00:25:29 |
|* 10 |      HASH JOIN               |                         |  1120M|   120G|   119M| 65365  (13)| 00:00:03 |
|                                                                                                                     |
|  11 |       VIEW                   |                         |  2986K|    85M|       | 22617   (2)| 00:00:01 |
|                                                                                                                     |
|  12 |        HASH GROUP BY         |                         |  2986K|    34M|    68M| 22617   (2)| 00:00:01 |
|                                                                                                                     |
|  13 |         TABLE ACCESS FULL    | POINTS                  |  2986K|    34M|       |  8879   (1)| 00:00:01 |
|* 14 |        HASH JOIN             |                         |  3506K|   284M|       | 13234   (2)| 00:00:01 |
|                                                                                                                     |
|  15 |         TABLE ACCESS FULL    | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 16 |         HASH JOIN            |                         |  3506K|   267M|  1400K| 13207   (2)| 00:00:01 |
|* 17 |          HASH JOIN           |                         | 16576 |  1197K|       |  8928   (1)| 00:00:01 |
|* 18 |           HASH JOIN          |                         |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 19 |            HASH JOIN         |                         |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  20 |             TABLE ACCESS FULL| QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL| STUDENTS                |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 22 |            TABLE ACCESS FULL | HOUSES                  |     3 |    42 |       |     3   (0)| 00:00:01 |
|                                                                                                                     |
|                                                                                                                     |
|                                                                                                                     |
|                                                                                                                     |
|* 23 |          TABLE ACCESS FULL   | POINTS                  |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|  24 |         TABLE ACCESS FULL    | GRADES                  |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

9.3.4 Points summary with indexes

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                            | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                     |                         |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|*  1 |  HASH JOIN                           |                         |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|   2 |   TABLE ACCESS FULL                  | TEACHERS                |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                          |                         |  1105M|   186G|       |    38M  (1)| 00:25:07 |
|   4 |    MERGE JOIN CARTESIAN              |                         |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL                | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT                      |                         |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL               | TEACHERS                |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                              |                         |  1105M|   161G|       |    38M  (1)| 00:25:07 |
|   9 |     SORT GROUP BY                    |                         |  1105M|   116G|   131G|    38M  (1)| 00:25:07 |
|* 10 |      HASH JOIN                       |                         |  1120M|   117G|    56M| 72682  (11)| 00:00:03 |
|* 11 |       TABLE ACCESS FULL              | POINTS                  |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|                                                                                                                     |
|* 12 |       HASH JOIN                      |                         |  4917K|   454M|       | 28129   (1)| 00:00:02 |
|                                                                                                                     |
|  13 |        TABLE ACCESS FULL             | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|                                                                                                                     |
|* 14 |        HASH JOIN                     |                         |  4917K|   431M|  2232K| 28093   (1)| 00:00:02 |
|  15 |         NESTED LOOPS                 |                         | 23243 |  1952K|       | 23763   (1)| 00:00:01 |
|                                                                                                                     |
|* 16 |          HASH JOIN                   |                         |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 17 |           HASH JOIN                  |                         |    97 |  4268 |       |    22   (0)| 00:00:01 |
|                                                                                                                     |
|                                                                                                                     |
|  18 |            TABLE ACCESS FULL         | QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 19 |            TABLE ACCESS FULL         | STUDENTS                |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 20 |           TABLE ACCESS FULL          | HOUSES                  |     3 |    42 |       |     3   (0)| 00:00:01 |
|  21 |          VIEW PUSHED PREDICATE       |                         |     1 |    28 |       |   325   (1)| 00:00:01 |
|  22 |           SORT GROUP BY              |                         |   196 |  2352 |       |   325   (1)| 00:00:01 |
|  23 |            TABLE ACCESS BY INDEX ROWID BATCHED| POINTS         |   320 |  3840 |       |   324   (0)| 00:00:01 |
|* 24 |             INDEX RANGE SCAN         | INDEX2                  |   320 |       |       |     4   (0)| 00:00:01 |
|                                                                                                                     |
|  25 |         TABLE ACCESS FULL            | GRADES                  |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

9.3.5 Best students without indexes

```
-------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|   1 |  SORT ORDER BY               |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|*  2 |   VIEW                       |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|*  3 |    WINDOW SORT PUSHED RANK   |              |   631M|    45G|    52G|    22M  (1)| 00:14:52 |
|   4 |     HASH GROUP BY            |              |   631M|    45G|    52G|    22M  (1)| 00:14:52 |
|*  5 |      HASH JOIN               |              |   631M|    45G|    79M| 38180  (12)| 00:00:02 |
|*  6 |       TABLE ACCESS FULL      | POINTS       |  2986K|    45M|       |  8895   (1)| 00:00:01 |
|*  7 |       HASH JOIN              |              |  1977K|   115M|       | 14265   (2)| 00:00:01 |
|                                                                                                     |
|*  8 |        HASH JOIN             |              |  9346 |   355K|       |    22   (0)| 00:00:01 |
|   9 |         TABLE ACCESS FULL    | HOUSES       |     4 |    56 |       |     3   (0)| 00:00:01 |
|  10 |         TABLE ACCESS FULL    | STUDENTS     |  9346 |   228K|       |    19   (0)| 00:00:01 |
|  11 |        VIEW                  |              |  1977K|    41M|       | 14229   (2)| 00:00:01 |
|                                                                                                     |
|  12 |         HASH GROUP BY        |              |  1977K|    35M|    60M| 14229   (2)| 00:00:01 |
|                                                                                                     |
|* 13 |          HASH JOIN           |              |  1977K|    35M|       |  2521   (3)| 00:00:01 |
|  14 |           TABLE ACCESS FULL  | GRADES_ENUM  |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 15 |           TABLE ACCESS FULL  | GRADES       |  1977K|    26M|       |  2505   (2)| 00:00:01 |
|                                                                                                     |
|                                                                                                     |
|                                                                                                     |
-------------------------------------------------------------------------------------------------------
```

9.3.6 Best students with indexes

```
-------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |    10 |  1330 |       |    19M  (1)| 00:12:42 |
|   1 |  SORT ORDER BY               |              |    10 |  1330 |       |    19M  (1)| 00:12:42 |
|*  2 |   VIEW                       |              |    10 |  1330 |       |    19M  (1)| 00:12:42 |
|*  3 |    WINDOW SORT PUSHED RANK   |              |    48M|  4012M|    56G|    19M  (1)| 00:12:42 |
|   4 |     HASH GROUP BY            |              |    48M|  4012M|    56G|    19M  (1)| 00:12:42 |
|*  5 |      HASH JOIN               |              |   631M|    50G|    79M|   494K  (3)| 00:00:20 |
|*  6 |       TABLE ACCESS FULL      | POINTS       |  2986K|    45M|       |  8895   (1)| 00:00:01 |
|                                                                                                     |
|   7 |       NESTED LOOPS           |              |  1977K|   131M|       |   469K  (3)| 00:00:19 |
|*  8 |        HASH JOIN             |              |  9346 |   355K|       |    22   (0)| 00:00:01 |
|   9 |         TABLE ACCESS FULL    | HOUSES       |     4 |    56 |       |     3   (0)| 00:00:01 |
|  10 |         TABLE ACCESS FULL    | STUDENTS     |  9346 |   228K|       |    19   (0)| 00:00:01 |
|                                                                                                     |
|  11 |        VIEW PUSHED PREDICATE |              |     1 |    31 |       |    50   (2)| 00:00:01 |
|                                                                                                     |
|  12 |         SORT GROUP BY        |              |   108 |  2052 |       |    50   (2)| 00:00:01 |
|* 13 |          HASH JOIN           |              |   212 |  4028 |       |    49   (0)| 00:00:01 |
|  14 |           TABLE ACCESS FULL  | GRADES_ENUM  |     6 |    30 |       |     3   (0)| 00:00:01 |
|                                                                                                     |
|  15 |           TABLE ACCESS BY INDEX ROWID BATCHED| GRADES       |   212 |  2968 |       |    46   (0)| 00:00:01 |
|  16 |            BITMAP CONVERSION TO ROWIDS       |              |       |       |       |            |          |
|* 17 |             BITMAP INDEX RANGE SCAN          | INDEX1       |       |       |       |            |          |
-------------------------------------------------------------------------------------------------------
```

9.3.7 Remove points without indexes

```
-------------------------------------------------------------------------------------------
| Id  | Operation                | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT         |             | 18383 |   556K| 11394   (2)| 00:00:01 |
|   1 |  DELETE                  | POINTS      |       |       |            |          |
|*  2 |   HASH JOIN RIGHT SEMI   |             | 18383 |   556K| 11394   (2)| 00:00:01 |
|     |                                                                                   |
|     |                                                                                   |
|   3 |    VIEW                  | VW_NSO_1    |   115 |  1495 |  2503   (2)| 00:00:01 |
|*  4 |     SORT GROUP BY        |             |   115 |  1955 |  2503   (2)| 00:00:01 |
|   5 |      VIEW                |             |  2291 | 38947 |  2503   (2)| 00:00:01 |
|*  6 |       FILTER             |             |       |       |            |          |
|   7 |        HASH GROUP BY     |             |  2291 | 50402 |  2503   (2)| 00:00:01 |
|*  8 |         HASH JOIN        |             | 45810 |   984K|  2499   (2)| 00:00:01 |
|   9 |          TABLE ACCESS FULL| GRADES_ENUM |     6 |    30 |     3   (0)| 00:00:01 |
|* 10 |          TABLE ACCESS FULL| GRADES      | 45810 |   760K|  2496   (2)| 00:00:01 |
|     |                                                                                   |
|* 11 |    TABLE ACCESS FULL     | POINTS      |  1494K|    25M|  8881   (1)| 00:00:01 |
|     |                                                                                   |
-------------------------------------------------------------------------------------------
```

9.3.8 Remove points with indexes

```
-------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT              |             |  2078 | 81042 |  6771   (2)| 00:00:01 |
|   1 |  DELETE                       | POINTS      |       |       |            |          |
|     |                                                                                         |
|   2 |   NESTED LOOPS                |             |  2078 | 81042 |  6771   (2)| 00:00:01 |
|   3 |    NESTED LOOPS               |             |  4160 | 81042 |  6771   (2)| 00:00:01 |
|   4 |     VIEW                      | VW_NSO_1    |    13 |   169 |  2569   (5)| 00:00:01 |
|*  5 |      SORT GROUP BY            |             |    13 |   221 |  2569   (5)| 00:00:01 |
|   6 |       VIEW                    |             |   248 |  4216 |  2569   (5)| 00:00:01 |
|*  7 |        FILTER                 |             |       |       |            |          |
|   8 |         HASH GROUP BY         |             |   248 |  6696 |  2569   (5)| 00:00:01 |
|*  9 |          HASH JOIN            |             |  4943 |   130K|  2568   (5)| 00:00:01 |
|  10 |           TABLE ACCESS FULL   | GRADES_ENUM |     6 |    30 |     3   (0)| 00:00:01 |
|* 11 |           TABLE ACCESS FULL   | GRADES      |  4943 |   106K|  2565   (5)| 00:00:01 |
|* 12 |     INDEX RANGE SCAN          | INDEX2      |   320 |       |     3   (0)| 00:00:01 |
|     |                                                                                         |
|* 13 |    TABLE ACCESS BY INDEX ROWID| POINTS      |   160 |  4160 |   323   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

9.4 Experiments

Two experiments were conducted, checking differences in performance after using two different types of indexes: B-tree and bitmap. Tested indexes included `index1 ON grades(student_id, award_date)` which was tested on **Grades Analysis** procedure, and `index2 ON points(student_id, award_date)` tested on **Points Summary** procedure.

Procedure name	**Min. measured time**				**Max. measured time**				**Average measured time**				**Query plan cost**			
	B-tree	Bitmap	Diff.	Profit	B-tree	Bitmap	Diff.	Profit	B-tree	Bitmap	Diff.	Profit	B-tree	Bitmap	Diff.	Profit
Grades analysis (Index 1)	607	477	130	21%	814	497	317	39%	686.2	485.5	200.7	29%	1662 K	1658 K	4 K	0.2%
Points summary (Index 2)	315	330	-15	-5%	341	374	-33	-10%	322.8	340.8	-18.0	-6%	38 M	38 M	253	0%

Query plan for `index1` showed a small advantage of using the bitmap type of index over the B-tree one. However, the measured time has proven that the difference between implementations using these two types of indexes is much higher in practice. Therefore, for `index1`, the **bitmap** type was selected.

Second experiment has shown the negligible difference between the query plan costs, however the tests shown, that the b-tree index performs better in practice. Therefore, for `index2`, the **B-tree** type was selected.

9.4.1 Grades analysis with B-tree Index

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                       |    12M|  2592M|       |  1662K  (1)| 00:01:05 |
|   1 |  SORT ORDER BY                           |                       |    12M|  2592M|  2757M|  1662K  (1)| 00:01:05 |
|*  2 |   HASH JOIN                              |                       |    12M|  2592M|       |  1073K  (1)| 00:00:42 |
|   3 |    TABLE ACCESS FULL                     | SUBJECTS              |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                       |    12M|  2356M|       |  1073K  (1)| 00:00:42 |
|   5 |     MERGE JOIN CARTESIAN                 |                       |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN                |                       |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL                  | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT                        |                       |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL                 | GRADES_ENUM           |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                         |                       |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL                  | SUBJECTS              |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                                 |                       |    12M|  2003M|       |  1073K  (1)| 00:00:42 |
|  13 |      SORT GROUP BY                       |                       |    12M|  1425M|    10G|  1073K  (1)| 00:00:42 |
|* 14 |       HASH JOIN                          |                       |    84M|  9698M|    30M| 44918   (2)| 00:00:02 |
|  15 |        INDEX FAST FULL SCAN              | INDEX1                |  1977K|  7723K|       |  1736   (2)| 00:00:01 |
|                                                                                                                             |
|* 16 |        HASH JOIN                         |                       |   397K|    44M|       | 38682   (1)| 00:00:02 |
|  17 |         NESTED LOOPS                     |                       |  1256 |   123K|       | 29768   (1)| 00:00:02 |
|* 18 |          HASH JOIN OUTER                 |                       |    23 |  1380 |       | 28087   (1)| 00:00:02 |
|  19 |           NESTED LOOPS ANTI              |                       |    23 |   989 |       | 28084   (1)| 00:00:02 |
|* 20 |            HASH JOIN                     |                       |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL            | HOUSES                |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL            | STUDENTS              |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  23 |            VIEW PUSHED PREDICATE         | VW_NSO_1              |     2 |     4 |       |    12   (0)| 00:00:01 |
|  24 |             NESTED LOOPS                 |                       |     2 |    38 |       |    12   (0)| 00:00:01 |
|  25 |              NESTED LOOPS                |                       |    21 |    38 |       |    12   (0)| 00:00:01 |
|                                                                                                                             |
|* 26 |               TABLE ACCESS FULL          | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL          | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |               TABLE ACCESS FULL          | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 29 |              INDEX RANGE SCAN            | INDEX1                |    21 |       |       |     2   (0)| 00:00:01 |
|* 30 |             TABLE ACCESS BY INDEX ROWID  | GRADES                |     3 |    42 |       |     9   (0)| 00:00:01 |
|                                                                                                                             |
|                                                                                                                             |
|                                                                                                                             |
|  31 |          TABLE ACCESS FULL               | QUIDDITCH_TEAM_MEMBERS|    97 |  1649 |       |     3   (0)| 00:00:01 |
|  32 |         VIEW PUSHED PREDICATE            |                       |     1 |    41 |       |    73   (2)| 00:00:01 |
|  33 |          SORT GROUP BY                   |                       |    51 |   714 |       |    73   (2)| 00:00:01 |
|* 34 |           HASH JOIN                      |                       |   212 |  2968 |       |    72   (0)| 00:00:01 |
|  35 |            TABLE ACCESS FULL             | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|  36 |            TABLE ACCESS BY INDEX ROWID BATCHED| GRADES           |   212 |  1908 |       |    69   (0)| 00:00:01 |
|                                                                                                                             |
|* 37 |             INDEX RANGE SCAN             | INDEX1                |   212 |       |       |     3   (0)| 00:00:01 |
|                                                                                                                             |
|* 38 |        TABLE ACCESS FULL                 | POINTS                |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------
```

9.4.2 Grades analysis with Bitmap Index

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                       |    12M|  2592M|       |  1658K  (1)| 00:01:05 |
|   1 |  SORT ORDER BY                           |                       |    12M|  2592M|  2757M|  1658K  (1)| 00:01:05 |
|*  2 |   HASH JOIN                              |                       |    12M|  2592M|       |  1069K  (1)| 00:00:42 |
|   3 |    TABLE ACCESS FULL                     | SUBJECTS              |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                       |    12M|  2356M|       |  1069K  (1)| 00:00:42 |
|   5 |     MERGE JOIN CARTESIAN                 |                       |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN                |                       |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL                  | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT                        |                       |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL                 | GRADES_ENUM           |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                         |                       |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL                  | SUBJECTS              |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                                 |                       |    12M|  2003M|       |  1069K  (1)| 00:00:42 |
|  13 |      SORT GROUP BY                       |                       |    12M|  1425M|    10G|  1069K  (1)| 00:00:42 |
|* 14 |       HASH JOIN                          |                       |    84M|  9698M|    30M| 41009   (3)| 00:00:02 |
|                                                                                                                             |
|  15 |        TABLE ACCESS FULL                 | GRADES                |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
|* 16 |        HASH JOIN                         |                       |   397K|    44M|       | 34016   (1)| 00:00:02 |
|  17 |         NESTED LOOPS                     |                       |  1256 |   123K|       | 25102   (1)| 00:00:01 |
|* 18 |          HASH JOIN OUTER                 |                       |    23 |  1380 |       | 23946   (1)| 00:00:01 |
|  19 |           NESTED LOOPS ANTI              |                       |    23 |   989 |       | 23943   (1)| 00:00:01 |
|* 20 |            HASH JOIN                     |                       |  2337 | 95817 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL            | HOUSES                |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL            | STUDENTS              |  9346 |   246K|       |    19   (0)| 00:00:01 |
|  23 |            VIEW PUSHED PREDICATE         | VW_NSO_1              |     2 |     4 |       |    10   (0)| 00:00:01 |
|                                                                                                                             |
|                                                                                                                             |
|* 24 |             HASH JOIN                    |                       |     2 |    38 |       |    10   (0)| 00:00:01 |
|* 25 |              TABLE ACCESS FULL           | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 26 |              TABLE ACCESS FULL           | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |              TABLE ACCESS FULL           | GRADES_ENUM           |     1 |     5 |       |     3   (0)| 00:00:01 |
|                                                                                                                             |
|                                                                                                                             |
|  28 |             TABLE ACCESS BY INDEX ROWID BATCHED| GRADES          |    21 |   294 |       |     7   (0)| 00:00:01 |
|  29 |              BITMAP CONVERSION TO ROWIDS |                       |       |       |       |            |          |
|* 30 |               BITMAP INDEX RANGE SCAN    | INDEX1                |       |       |       |            |          |
|  31 |          TABLE ACCESS FULL               | QUIDDITCH_TEAM_MEMBERS|    97 |  1649 |       |     3   (0)| 00:00:01 |
|  32 |         VIEW PUSHED PREDICATE            |                       |     1 |    41 |       |    50   (2)| 00:00:01 |
|  33 |          SORT GROUP BY                   |                       |    51 |   714 |       |    50   (2)| 00:00:01 |
|* 34 |           HASH JOIN                      |                       |   212 |  2968 |       |    49   (0)| 00:00:01 |
|  35 |            TABLE ACCESS FULL             | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|  36 |            TABLE ACCESS BY INDEX ROWID BATCHED | GRADES          |   212 |  1908 |       |    46   (0)| 00:00:01 |
|  37 |             BITMAP CONVERSION TO ROWIDS  |                       |       |       |       |            |          |
|                                                                                                                             |
|* 38 |              BITMAP INDEX RANGE SCAN     | INDEX1                |       |       |       |            |          |
|* 39 |        TABLE ACCESS FULL                 | POINTS                |  2956K|    45M|       |  8895   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------
```

9.4.3 Points summary with B-tree Index

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                 | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                      |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|*  1 |  HASH JOIN                               |                      |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|   2 |   TABLE ACCESS FULL                      | TEACHERS             |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                              |                      |  1105M|   186G|       |    38M  (1)| 00:25:07 |
|   4 |    MERGE JOIN CARTESIAN                  |                      |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL                    | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT                          |                      |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL                   | TEACHERS             |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                                  |                      |  1105M|   161G|       |    38M  (1)| 00:25:07 |
|   9 |     SORT GROUP BY                        |                      |  1105M|   116G|   131G|    38M  (1)| 00:25:07 |
|* 10 |      HASH JOIN                           |                      |  1120M|   117G|    56M| 72682  (11)| 00:00:03 |
|* 11 |       TABLE ACCESS FULL                  | POINTS               |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|* 12 |       HASH JOIN                          |                      |  4917K|   454M|       | 28129   (1)| 00:00:02 |
|  13 |        TABLE ACCESS FULL                 | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 14 |        HASH JOIN                         |                      |  4917K|   431M|  2232K| 28093   (1)| 00:00:02 |
|  15 |         NESTED LOOPS                     |                      | 23243 |  1952K|       | 23763   (1)| 00:00:01 |
|* 16 |          HASH JOIN                       |                      |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 17 |           HASH JOIN                      |                      |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  18 |            TABLE ACCESS FULL             | QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 19 |            TABLE ACCESS FULL             | STUDENTS             |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 20 |           TABLE ACCESS FULL              | HOUSES               |     3 |    42 |       |     3   (0)| 00:00:01 |
|  21 |          VIEW PUSHED PREDICATE           |                      |     1 |    28 |       |   325   (1)| 00:00:01 |
|  22 |           SORT GROUP BY                  |                      |   196 |  2352 |       |   325   (1)| 00:00:01 |
|  23 |            TABLE ACCESS BY INDEX ROWID BATCHED| POINTS               |   320 |  3840 |       |   324   (0)| 00:00:01 |
|                                                                                                           |
|* 24 |             INDEX RANGE SCAN             | INDEX2               |   320 |       |       |     4   (0)| 00:00:01 |
|                                                                                                           |
|  25 |        TABLE ACCESS FULL                 | GRADES               |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------
```

9.4.4 Points summary with Bitmap Index

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                 | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                      |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|*  1 |  HASH JOIN                               |                      |  1105M|   205G|       |    38M  (1)| 00:25:07 |
|   2 |   TABLE ACCESS FULL                      | TEACHERS             |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                              |                      |  1105M|   186G|       |    38M  (1)| 00:25:06 |
|   4 |    MERGE JOIN CARTESIAN                  |                      |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL                    | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT                          |                      |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL                   | TEACHERS             |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                                  |                      |  1105M|   161G|       |    38M  (1)| 00:25:06 |
|   9 |     SORT GROUP BY                        |                      |  1105M|   116G|   131G|    38M  (1)| 00:25:06 |
|* 10 |      HASH JOIN                           |                      |  1120M|   117G|    56M| 54235  (15)| 00:00:03 |
|* 11 |       TABLE ACCESS FULL                  | POINTS               |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|* 12 |       HASH JOIN                          |                      |  4917K|   454M|       |  9682   (2)| 00:00:01 |
|  13 |        TABLE ACCESS FULL                 | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 14 |        HASH JOIN                         |                      |  4917K|   431M|  2232K|  9647   (2)| 00:00:01 |
|  15 |         NESTED LOOPS                     |                      | 23243 |  1952K|       |  5316   (2)| 00:00:01 |
|* 16 |          HASH JOIN                       |                      |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 17 |           HASH JOIN                      |                      |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  18 |            TABLE ACCESS FULL             | QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 19 |            TABLE ACCESS FULL             | STUDENTS             |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 20 |           TABLE ACCESS FULL              | HOUSES               |     3 |    42 |       |     3   (0)| 00:00:01 |
|  21 |          VIEW PUSHED PREDICATE           |                      |     1 |    28 |       |    72   (2)| 00:00:01 |
|  22 |           SORT GROUP BY                  |                      |   196 |  2352 |       |    72   (2)| 00:00:01 |
|  23 |            TABLE ACCESS BY INDEX ROWID BATCHED| POINTS               |   320 |  3840 |       |    71   (0)| 00:00:01 |
|  24 |             BITMAP CONVERSION TO ROWIDS  |                      |       |       |       |            |          |
|                                                                                                           |
|* 25 |              BITMAP INDEX RANGE SCAN     | INDEX2               |       |       |       |            |          |
|  26 |        TABLE ACCESS FULL                 | GRADES               |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------
```

10 Partitions

10.1 Proposed partitions

10.1.1 List Partitioning on GRADES.value

Details: There are only six possible values in this column, so we can divide the data into partitions using one partition for each possible value.
Purpose: Optimization of transaction **Grades Analysis**
Possible improvements: Query Plan rows 24 and 32

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                 | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                      |    47M|  9981M|       |  3628K  (1)| 00:02:22 |
|   1 |  SORT ORDER BY                     |                      |    47M|  9981M|    10G|  3628K  (1)| 00:02:22 |
|*  2 |   HASH JOIN                        |                      |    47M|  9981M|       |  1359K  (1)| 00:00:54 |
|   3 |    TABLE ACCESS FULL               | SUBJECTS             |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                       |                      |    47M|  9074M|       |  1359K  (1)| 00:00:54 |
|   5 |     MERGE JOIN CARTESIAN           |                      |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN          |                      |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL            | GRADES_ENUM          |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT                  |                      |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL           | GRADES_ENUM          |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT                   |                      |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL            | SUBJECTS             |    76 |  1520 |       |     1   (0)| 00:00:01 |
```

```
|  12 |     VIEW                       |                          |    47M|  7712M|       |  1358K  (1)| 00:00:54 |
|  13 |      SORT GROUP BY             |                          |    47M|  5671M|  6092M|  1358K  (1)| 00:00:54 |
|* 14 |       HASH JOIN                |                          |    47M|  5671M|    28M| 24853   (3)| 00:00:01 |
|* 15 |        HASH JOIN               |                          |   224K|    25M|       | 19118   (2)| 00:00:01 |
|* 16 |         HASH JOIN              |                          |   711 | 74655 |       | 10204   (3)| 00:00:01 |
|  17 |          JOIN FILTER CREATE    | :BF0000                  |    13 |   806 |       |  2548   (3)| 00:00:01 |
|* 18 |           HASH JOIN OUTER      |                          |    13 |   806 |       |  2548   (3)| 00:00:01 |
|* 19 |            HASH JOIN ANTI      |                          |    13 |   585 |       |  2545   (3)| 00:00:01 |
|* 20 |             HASH JOIN          |                          |  1323 | 54243 |       |    22   (0)| 00:00:01 |
|* 21 |              TABLE ACCESS FULL | HOUSES                   |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |              TABLE ACCESS FULL | STUDENTS                 |  5291 |   139K|       |    19   (0)| 00:00:01 |
|  23 |             VIEW               | VW_NSO_1                 |   190K|   745K|       |  2521   (3)| 00:00:01 |
|                                                                                                                  |
|* 24 |              HASH JOIN         |                          |   190K|  3541K|       |  2521   (3)| 00:00:01 |
|                                                                                                                  |
|* 25 |               TABLE ACCESS FULL | GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 26 |               TABLE ACCESS FULL| GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL| GRADES_ENUM              |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |              TABLE ACCESS FULL | GRADES                   |  1957K|    26M|       |  2505   (2)| 00:00:01 |
|  29 |            TABLE ACCESS FULL   | QUIDDITCH_TEAM_MEMBERS   |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  30 |           VIEW                 |                          |   502K|    20M|       |  7653   (3)| 00:00:01 |
|  31 |            HASH GROUP BY       |                          |   502K|  6866K|    45M|  7653   (3)| 00:00:01 |
|                                                                                                                  |
|* 32 |             HASH JOIN          |                          |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|                                                                                                                  |
|  33 |              TABLE ACCESS FULL | GRADES_ENUM              |     6 |    30 |       |     3   (0)| 00:00:01 |
|  34 |              JOIN FILTER USE   | :BF0000                  |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 35 |               TABLE ACCESS FULL| GRADES                   |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 36 |         TABLE ACCESS FULL      | POINTS                   |  2956K|    45M|       |  8895   (1)| 00:00:01 |
|  37 |        TABLE ACCESS FULL       | GRADES                   |  1977K|  7723K|       |  2492   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("Q"."WORST_SUBJECT"="S2"."ID")
   4 - access("Q"."BEST_SUBJECT"="S1"."ID" AND "Q"."MIN_GRADE"="G2"."VALUE" AND
              "Q"."MAX_GRADE"="G1"."VALUE")
  14 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  15 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  16 - access("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
  18 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID"(+))
  19 - access("STUDENTS"."ID"="ID")
  20 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  21 - filter("HOUSES"."NAME"='Gryffindor')
  22 - filter("STUDENTS"."YEAR"=1 OR "STUDENTS"."YEAR"=3 OR "STUDENTS"."YEAR"=5 OR "STUDENTS"."YEAR"=7)
->24 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  25 - filter("GRADES_ENUM"."VALUE"< (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM" "GRADES_ENUM" WHERE
              "GRADES_ENUM"."SYMBOL"='T') OR "GRADES_ENUM"."VALUE"> (SELECT "GRADES_ENUM"."VALUE" FROM "GRADES_ENUM"
              "GRADES_ENUM" WHERE "GRADES_ENUM"."SYMBOL"='O'))
  26 - filter("GRADES_ENUM"."SYMBOL"='T')
  27 - filter("GRADES_ENUM"."SYMBOL"='O')
  28 - filter("GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
->32 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  35 - filter(SYS_OP_BLOOM_FILTER(:BF0000,"GRADES"."STUDENT_ID"))
  36 - filter("POINTS"."AWARD_DATE"<=TO_DATE(' 2024-06-25 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "POINTS"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

10.1.2 Range Partitioning on POINTS.award_date

Details: We can divide the data into partitions such that points awarded in different months are on different partitions.
Purpose: Optimization of transaction **Points Summary**
Possible improvements: Query Plan row 23

```
-----------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                         |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|*  1 |  HASH JOIN                   |                         |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|   2 |   TABLE ACCESS FULL          | TEACHERS                |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                         |  1105M|   186G|       |    39M  (1)| 00:25:29 |
|   4 |    MERGE JOIN CARTESIAN      |                         |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                         |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS                |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                         |  1105M|   161G|       |    39M  (1)| 00:25:29 |
|   9 |     SORT GROUP BY            |                         |  1105M|   118G|   135G|    39M  (1)| 00:25:29 |
|* 10 |      HASH JOIN               |                         |  1120M|   120G|   119M| 65365  (13)| 00:00:03 |
|  11 |       VIEW                   |                         |  2986K|    85M|       | 22617   (2)| 00:00:01 |
|  12 |        HASH GROUP BY         |                         |  2986K|    34M|    68M| 22617   (2)| 00:00:01 |
|  13 |         TABLE ACCESS FULL    | POINTS                  |  2986K|    34M|       |  8879   (1)| 00:00:01 |
|* 14 |       HASH JOIN              |                         |  3506K|   284M|       | 13234   (2)| 00:00:01 |
|  15 |        TABLE ACCESS FULL     | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 16 |        HASH JOIN             |                         |  3506K|   267M|  1400K| 13207   (2)| 00:00:01 |
|* 17 |         HASH JOIN            |                         | 16576 |  1197K|       |  8928   (1)| 00:00:01 |
|* 18 |          HASH JOIN           |                         |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 19 |           HASH JOIN          |                         |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  20 |            TABLE ACCESS FULL| QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 21 |            TABLE ACCESS FULL| STUDENTS                |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 22 |           TABLE ACCESS FULL | HOUSES                  |     3 |    42 |       |     3   (0)| 00:00:01 |
|                                                                                                               |
|* 23 |          TABLE ACCESS FULL   | POINTS                  |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|                                                                                                               |
|  24 |         TABLE ACCESS FULL    | GRADES                  |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - access("Q"."WORST_TEACHER"="T2"."ID")
   3 - access("Q"."BEST_TEACHER"="T1"."ID" AND "Q"."AVG_GRADE"="GRADES_ENUM"."VALUE")
  10 - access("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
  14 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  16 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  17 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  18 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  19 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID")
  21 - filter("STUDENTS"."YEAR">4 OR "STUDENTS"."YEAR"<3)
  22 - filter("HOUSES"."NAME"<>'Hufflepuff')
->23 - filter("POINTS"."AWARD_DATE">TO_DATE(' 2024-01-06 00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
              "POINTS"."AWARD_DATE"<TO_DATE(' 2023-12-06 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

10.1.3 Hash Partitioning on STUDENTS.id

Details: We want to evenly distribute students of different IDs between partitions.
Purpose: Optimization of transaction **Points Summary**
Possible improvements: Query Plan rows 10 and 16

```
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                       |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|*  1 |  HASH JOIN                   |                       |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|   2 |   TABLE ACCESS FULL          | TEACHERS              |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                       |  1105M|   186G|       |    39M  (1)| 00:25:29 |
|   4 |    MERGE JOIN CARTESIAN      |                       |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                       |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS              |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                       |  1105M|   161G|       |    39M  (1)| 00:25:29 |
|   9 |     SORT GROUP BY            |                       |  1105M|   118G|   135G|    39M  (1)| 00:25:29 |
|                                                                                                         |
|* 10 |      HASH JOIN               |                       |  1120M|   120G|   119M| 65365  (13)| 00:00:03 |
|                                                                                                         |
|  11 |       VIEW                   |                       |  2986K|    85M|       | 22617   (2)| 00:00:01 |
|  12 |        HASH GROUP BY         |                       |  2986K|    34M|    68M| 22617   (2)| 00:00:01 |
|  13 |         TABLE ACCESS FULL    | POINTS                |  2986K|    34M|       |  8879   (1)| 00:00:01 |
|* 14 |       HASH JOIN              |                       |  3506K|   284M|       | 13234   (2)| 00:00:01 |
|  15 |        TABLE ACCESS FULL     | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|                                                                                                         |
|* 16 |        HASH JOIN             |                       |  3506K|   267M|  1400K| 13207   (2)| 00:00:01 |
|                                                                                                         |
|* 17 |         HASH JOIN            |                       | 16576 |  1197K|       |  8928   (1)| 00:00:01 |
|* 18 |          HASH JOIN           |                       |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 19 |           HASH JOIN          |                       |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  20 |            TABLE ACCESS FULL| QUIDDITCH_TEAM_MEMBERS |    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 21 |            TABLE ACCESS FULL| STUDENTS               |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 22 |           TABLE ACCESS FULL | HOUSES                 |     3 |    42 |       |     3   (0)| 00:00:01 |
|* 23 |          TABLE ACCESS FULL  | POINTS                 |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|  24 |         TABLE ACCESS FULL   | GRADES                 |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - access("Q"."WORST_TEACHER"="T2"."ID")
   3 - access("Q"."BEST_TEACHER"="T1"."ID" AND "Q"."AVG_GRADE"="GRADES_ENUM"."VALUE")
->10 - access("AVERAGES"."STUDENT_ID"="STUDENTS"."ID")
  14 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
->16 - access("STUDENTS"."ID"="GRADES"."STUDENT_ID")
  17 - access("STUDENTS"."ID"="POINTS"."STUDENT_ID")
  18 - access("STUDENTS"."HOUSE_ID"="HOUSES"."ID")
  19 - access("STUDENTS"."ID"="QUIDDITCH_TEAM_MEMBERS"."STUDENT_ID")
  21 - filter("STUDENTS"."YEAR">4 OR "STUDENTS"."YEAR"<3)
  22 - filter("HOUSES"."NAME"<>'Hufflepuff')
  23 - filter("POINTS"."AWARD_DATE">TO_DATE(' 2024-01-06 00:00:00', 'syyyy-mm-dd hh24:mi:ss') OR
              "POINTS"."AWARD_DATE"<TO_DATE(' 2023-12-06 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

10.2 Proposed experiments

10.2.1 Experiment 1: Partitioning the GRADES table – List vs. Hash

Partitioning schemes being compared:

- *Scheme A (Proposed):* List partitioning of the `GRADES` table based on the `value` column (6 partitions – one for each grade).
- *Scheme B (Alternative):* Hash partitioning of the `GRADES` table using the `student_id` column (e.g., into 4 or 8 partitions).

Goal/Hypothesis: The idea is to check which approach performs better. The experiment will test whether a more even data spread from hashing by `student_id` helps when running queries on `GRADES`, especially when grades aren't evenly distributed or when queries don't hit just one list partition. This will be compared to using list partitioning on the low-cardinality `value` column.
Tested transactions: *Grades analysis*

10.2.2 Experiment 2: Partitioning the POINTS table – Range vs. Composite (Range + Hash)

Partitioning schemes being compared:

- *Scheme A (Proposed):* Range partitioning of the `POINTS` table based on `award_date` (monthly).
- *Scheme B (Alternative):* Composite partitioning – first by range on `award_date` (monthly), and then subpartitioned by hash on `teacher_id` (e.g., 4 subpartitions per month).

Goal/Hypothesis: The goal is to see whether combining range and hash partitioning improves performance when queries filter by both date and `teacher_id`. This setup might allow for more accurate pruning and better workload distribution.
Tested transactions: *Points summary*

10.2.3 Experiment 3: Logical placement of partitions (Tablespaces) in the POINTS table

Base partitioning: Range partitioning of the `POINTS` table by `award_date` (monthly).
Configurations being compared:

- *Configuration A:* All partitions stored in the default single tablespace.
- *Configuration B:* Some partitions (e.g., older vs. newer months) stored in different, dedicated tablespaces.

Goal/Hypothesis: The goal is to find out whether putting partitions in different tablespaces (even if they're on the same disk) changes how the database handles queries or tracks I/O stats. This could give insight into whether using separate tablespaces helps with managing data and optimizing systems with more than one storage device.
Tested transactions: *Points Summary*

11 Partitions - implementation

11.1 Implemented partitions

As declared before, we specified three partitions: GRADES.value, POINTS.award_date, STUDENTS.id. In order to test them efficiently, we created a second database with the same data, but slightly altered creation process:

```
CREATE TABLE Grades (
    id NUMBER NOT NULL PRIMARY KEY,
    value VARCHAR2(1) NOT NULL,
    award_date DATE NOT NULL,
    student_id NUMBER NOT NULL,
    subject_id NUMBER NOT NULL,
    teacher_id NUMBER NOT NULL
)
PARTITION BY LIST(value) (
    PARTITION troll VALUES ('T'),
    PARTITION dread VALUES ('D'),
    PARTITION poor VALUES ('P'),
    PARTITION accept VALUES ('A'),
    PARTITION expec VALUES ('E'),
    PARTITION outst VALUES ('O'),
    PARTITION other VALUES (DEFAULT)
);

CREATE TABLE Points (
    id NUMBER NOT NULL PRIMARY KEY,
    value NUMBER NOT NULL,
    description VARCHAR2(1024),
    award_date DATE NOT NULL,
    student_id NUMBER NOT NULL,
    teacher_id NUMBER
)
PARTITION BY RANGE(award_date) (
    PARTITION september VALUES LESS THAN (TO_DATE('2023-10-01', 'YYYY-MM-DD')),
    PARTITION october VALUES LESS THAN (TO_DATE('2023-11-01', 'YYYY-MM-DD')),
    PARTITION november VALUES LESS THAN (TO_DATE('2023-12-01', 'YYYY-MM-DD')),
    PARTITION december VALUES LESS THAN (TO_DATE('2024-01-01', 'YYYY-MM-DD')),
    PARTITION january VALUES LESS THAN (TO_DATE('2024-02-01', 'YYYY-MM-DD')),
    PARTITION february VALUES LESS THAN (TO_DATE('2024-03-01', 'YYYY-MM-DD')),
    PARTITION march VALUES LESS THAN (TO_DATE('2024-04-01', 'YYYY-MM-DD')),
    PARTITION april VALUES LESS THAN (TO_DATE('2024-05-01', 'YYYY-MM-DD')),
    PARTITION may VALUES LESS THAN (TO_DATE('2024-06-01', 'YYYY-MM-DD')),
    PARTITION june VALUES LESS THAN (TO_DATE('2024-07-01', 'YYYY-MM-DD')),
    PARTITION future VALUES LESS THAN (MAXVALUE)
);

CREATE TABLE Students (
    id NUMBER NOT NULL PRIMARY KEY,
```

```
    name VARCHAR2(64) NOT NULL,
    surname VARCHAR2(64) NOT NULL,
    gender CHAR NOT NULL,
    date_of_birth DATE NOT NULL,
    year NUMBER NOT NULL,
    hogsmeade_consent NUMBER(1) DEFAULT 0 NOT NULL,
    house_id NUMBER NOT NULL,
    dormitory_id NUMBER
)
PARTITION BY HASH(id)
    PARTITIONS 4
    STORE IN (users, second_disk);
```

11.2 Cost comparison

We ran the whole workload on the database with and without partitions, **5 times** in each case. Dividing the database into partitions positively influenced most of our queries (according to query plans). Similar to what was the case with indexes - actual running times on our machine were often actually bigger with than without partitions.

Procedure name	Min. measured time				Max. measured time				Average measured time				Query plan cost			
	Default	Parti.	Diff.	Profit	Default	Parti.	Diff.	Profit	Default	Parti.	Diff.	Profit	Default	Parti.	Diff.	Profit
Grades analysis	564	498	66	11.7 %	813	537	276	33.9 %	677.0	514.4	162,6	24.0 %	3628 K	18300	3609 K	99.5 %
Points summary	377	468	-91	-24.1 %	556	540	16	2.9 %	468.6	507.8	-39,2	-8.4 %	39 M	24273	38.9 M	99.9 %
Best students	221	214	7	3.2 %	297	248	49	16.5 %	250.0	229.4	20,6	8.2 %	22 M	13368	21.9 M	99.9 %
Assign subjects	2735	6249	-3514	-128.5 %	3932	8869	-4937	-125.6 %	3403.8	7079.2	-3675,4	-108.0 %	12318	4942	7376	60 %
Remove points	26694	37578	-10884	-40.8 %	33966	40328	-6362	-18.7 %	30823.8	38952.6	-8128,8	-26.4 %	11460	13365	-1905	-16.6 %
Whole workload	30591	45007	-14416	-47.1 %	39564	50522	-10958	-27.7 %	35623.2	47283.4	-11660.2	-32.7 %	64.7 M	74248	64.6 M	99.9 %

11.3 Query plans comparison

Creating partitions changed all the query plans in our workload, and the new query plans are very different from the old ones. Below we present their comparison.

11.3.1 Grades analysis without partitions

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                       |   47M|  9981M|       |  3628K  (1)| 00:02:22 |
|   1 |  SORT ORDER BY                 |                       |   47M|  9981M|    10G|  3628K  (1)| 00:02:22 |
|*  2 |   HASH JOIN                    |                       |   47M|  9981M|       |  1359K  (1)| 00:00:54 |
|   3 |    TABLE ACCESS FULL           | SUBJECTS              |   76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                   |                       |   47M|  9074M|       |  1359K  (1)| 00:00:54 |
|   5 |     MERGE JOIN CARTESIAN       |                       | 2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN      |                       |   36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL        | GRADES_ENUM           |    6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT              |                       |    6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL       | GRADES_ENUM           |    6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT               |                       |   76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL        | SUBJECTS              |   76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                       |                       |   47M|  7712M|       |  1358K  (1)| 00:00:54 |
|  13 |      SORT GROUP BY             |                       |   47M|  5671M|  6092M|  1358K  (1)| 00:00:54 |
|* 14 |       HASH JOIN                |                       |   47M|  5671M|    28M| 24853   (3)| 00:00:01 |
|* 15 |        HASH JOIN               |                       |  224K|    25M|       | 19118   (2)| 00:00:01 |
|* 16 |         HASH JOIN              |                       |  711 | 74655 |       | 10204   (3)| 00:00:01 |
|                                                                                                           |
|  17 |          JOIN FILTER CREATE    | :BF0000               |   13 |   806 |       |  2548   (3)| 00:00:01 |
|* 18 |          HASH JOIN OUTER       |                       |   13 |   806 |       |  2548   (3)| 00:00:01 |
|* 19 |           HASH JOIN ANTI       |                       |   13 |   585 |       |  2545   (3)| 00:00:01 |
|                                                                                                           |
|* 20 |            HASH JOIN           |                       | 1323 | 54243 |       |    22   (0)| 00:00:01 |
|* 21 |             TABLE ACCESS FULL  | HOUSES                |    1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |             TABLE ACCESS FULL  | STUDENTS              | 5291 |  139K |       |    19   (0)| 00:00:01 |
|  23 |            VIEW                | VW_NSO_1              |  190K|  745K |       |  2521   (3)| 00:00:01 |
|* 24 |             HASH JOIN          |                       |  190K| 3541K |       |  2521   (3)| 00:00:01 |
|* 25 |              TABLE ACCESS FULL | GRADES_ENUM           |    1 |     5 |       |     3   (0)| 00:00:01 |
|                                                                                                           |
|* 26 |               TABLE ACCESS FULL| GRADES_ENUM           |    1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL| GRADES_ENUM           |    1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |              TABLE ACCESS FULL | GRADES                | 1957K|    26M|       |  2505   (2)| 00:00:01 |
|  29 |           TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS|   97 |  1649 |       |     3   (0)| 00:00:01 |
|  30 |          VIEW                  |                       |  502K|    20M|       |  7653   (3)| 00:00:01 |
|  31 |           HASH GROUP BY        |                       |  502K|  6866K|   45M|  7653   (3)| 00:00:01 |
|* 32 |            HASH JOIN           |                       | 1977K|    26M|       |  2511   (2)| 00:00:01 |
|  33 |             TABLE ACCESS FULL  | GRADES_ENUM           |    6 |    30 |       |     3   (0)| 00:00:01 |
|  34 |             JOIN FILTER USE    | :BF0000               | 1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 35 |              TABLE ACCESS FULL | GRADES                | 1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 36 |           TABLE ACCESS FULL    | POINTS                | 2956K|    45M|       |  8895   (1)| 00:00:01 |
|  37 |          TABLE ACCESS FULL     | GRADES                | 1977K|  7723K|       |  2492   (2)| 00:00:01 |
|                                                                                                           |
|                                                                                                           |
-------------------------------------------------------------------------------------------------------------
```

11.3.2 Grades analysis with partitions

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                  | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                       |    1 |   294 | 18300   (1)| 00:00:01 |       |       |
|   1 |  SORT ORDER BY                           |                       |    1 |   294 | 18300   (1)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                           |                       |    1 |   294 | 18299   (1)| 00:00:01 |       |       |
|   3 |    NESTED LOOPS                          |                       |    1 |   294 | 18299   (1)| 00:00:01 |       |       |
|   4 |     NESTED LOOPS                         |                       |    1 |   247 | 18299   (1)| 00:00:01 |       |       |
|*  5 |      HASH JOIN                           |                       |    1 |   200 | 18299   (1)| 00:00:01 |       |       |
|   6 |       MERGE JOIN CARTESIAN               |                       |    1 |    30 |     6   (0)| 00:00:01 |       |       |
|   7 |        TABLE ACCESS FULL                 | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|   8 |        BUFFER SORT                       |                       |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|   9 |         TABLE ACCESS FULL                | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|  10 |       VIEW                               |                       |    1 |   170 | 18293   (1)| 00:00:01 |       |       |
|  11 |        SORT GROUP BY                     |                       |    1 |   280 | 18293   (1)| 00:00:01 |       |       |
|* 12 |         HASH JOIN                        |                       |    1 |   280 | 15829   (1)| 00:00:01 |       |       |
|* 13 |          HASH JOIN OUTER                 |                       |    1 |   228 | 13364   (1)| 00:00:01 |       |       |
|* 14 |           HASH JOIN                      |                       |    1 |   181 | 13361   (1)| 00:00:01 |       |       |
|  15 |            NESTED LOOPS                  |                       |    1 |   146 |  2460   (1)| 00:00:01 |       |       |
|  16 |             NESTED LOOPS                 |                       |    1 |   146 |  2460   (1)| 00:00:01 |       |       |
|  17 |              NESTED LOOPS                |                       |    1 |   123 |  2460   (1)| 00:00:01 |       |       |
|  18 |               PARTITION LIST ALL         |                       |    1 |    13 |  2460   (1)| 00:00:01 |     1 |     7 |
|  19 |                TABLE ACCESS FULL         | GRADES                |    1 |    13 |  2460   (1)| 00:00:01 |     1 |     7 |
|* 20 |               TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS      |    1 |   110 |     0   (0)| 00:00:01 | ROWID | ROWID |
|* 21 |                INDEX UNIQUE SCAN         | SYS_C009092           |    1 |       |     0   (0)| 00:00:01 |       |       |
|  22 |              NESTED LOOPS SEMI           |                       |    1 |    39 |  2463   (1)| 00:00:01 |       |       |
|* 23 |               TABLE ACCESS FULL          | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|* 24 |               TABLE ACCESS FULL          | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|* 25 |               TABLE ACCESS FULL          | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|  26 |              PARTITION LIST ITERATOR     |                       |    1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 27 |               TABLE ACCESS FULL          | GRADES                |    1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 28 |             INDEX UNIQUE SCAN            | SYS_C009102           |    1 |       |     0   (0)| 00:00:01 |       |       |
|* 29 |            TABLE ACCESS BY INDEX ROWID   | HOUSES                |    1 |    23 |     0   (0)| 00:00:01 |       |       |
|  30 |           PARTITION RANGE ITERATOR       |                       |    1 |    35 | 10901   (1)| 00:00:01 |     1 |    10 |
|* 31 |            TABLE ACCESS FULL             | POINTS                |    1 |    35 | 10901   (1)| 00:00:01 |     1 |    10 |
|  32 |          TABLE ACCESS FULL               | QUIDDITCH_TEAM_MEMBERS|    1 |    47 |     3   (0)| 00:00:01 |       |       |
|  33 |         VIEW                             |                       |    1 |    52 |  2464   (1)| 00:00:01 |       |       |
|  34 |          HASH GROUP BY                   |                       |    1 |    43 |  2464   (1)| 00:00:01 |       |       |
|  35 |           NESTED LOOPS                   |                       |    1 |    43 |  2463   (1)| 00:00:01 |       |       |
|  36 |            TABLE ACCESS FULL             | GRADES_ENUM           |    1 |    15 |     3   (0)| 00:00:01 |       |       |
|  37 |            PARTITION LIST ITERATOR       |                       |    1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 38 |             TABLE ACCESS FULL            | GRADES                |    1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|  39 |     TABLE ACCESS BY INDEX ROWID          | SUBJECTS              |    1 |    47 |     0   (0)| 00:00:01 |       |       |
|* 40 |      INDEX UNIQUE SCAN                   | SYS_C009130           |    1 |       |     0   (0)| 00:00:01 |       |       |
|* 41 |     INDEX UNIQUE SCAN                    | SYS_C009130           |    1 |       |     0   (0)| 00:00:01 |       |       |
|  42 |    TABLE ACCESS BY INDEX ROWID           | SUBJECTS              |    1 |    47 |     0   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------------------------
```

11.3.3 Points summary without partitions

```
-------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                      | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                           |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|*  1 |  HASH JOIN                   |                           |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|   2 |   TABLE ACCESS FULL          | TEACHERS                  |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                           |  1105M|   186G|       |    39M  (1)| 00:25:29 |
|   4 |    MERGE JOIN CARTESIAN      |                           |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM               |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                           |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS                  |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                           |  1105M|   161G|       |    39M  (1)| 00:25:29 |
|   9 |     SORT GROUP BY            |                           |  1105M|   118G|   135G|    39M  (1)| 00:25:29 |
|* 10 |      HASH JOIN               |                           |  1120M|   120G|   119M| 65365  (13)| 00:00:03 |
|  11 |       VIEW                   |                           |  2986K|    85M|       | 22617   (2)| 00:00:01 |
|  12 |        HASH GROUP BY         |                           |  2986K|    34M|    68M| 22617   (2)| 00:00:01 |
|  13 |         TABLE ACCESS FULL    | POINTS                    |  2986K|    34M|       |  8879   (1)| 00:00:01 |
|* 14 |       HASH JOIN              |                           |  3506K|   284M|       | 13234   (2)| 00:00:01 |
|                                                                                                     |
|  15 |        TABLE ACCESS FULL     | GRADES_ENUM               |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 16 |        HASH JOIN             |                           |  3506K|   267M|  1400K| 13207   (2)| 00:00:01 |
|* 17 |         HASH JOIN            |                           | 16576 |  1197K|       |  8928   (1)| 00:00:01 |
|* 18 |          HASH JOIN           |                           |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 19 |           HASH JOIN          |                           |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  20 |            TABLE ACCESS FULL | QUIDDITCH_TEAM_MEMBERS    |    97 |  1649 |       |     3   (0)| 00:00:01 |
|                                                                                                     |
|* 21 |            TABLE ACCESS FULL | STUDENTS                  |  5504 |   145K|       |    19   (0)| 00:00:01 |
|                                                                                                     |
|* 22 |           TABLE ACCESS FULL  | HOUSES                    |     3 |    42 |       |     3   (0)| 00:00:01 |
|                                                                                                     |
|                                                                                                     |
|                                                                                                     |
|* 23 |          TABLE ACCESS FULL   | POINTS                    |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|  24 |         TABLE ACCESS FULL    | GRADES                    |  1977K|    11M|       |  2492   (2)| 00:00:01 |
|                                                                                                     |
|                                                                                                     |
|                                                                                                     |
-------------------------------------------------------------------------------------------------------
```

11.3.4 Points summary with partitions

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                  | Name                      | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                           |                           |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   1 |  NESTED LOOPS                              |                           |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                             |                           |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   3 |    NESTED LOOPS                            |                           |     1 |   253 | 24273   (1)| 00:00:01 |       |       |
|*  4 |     HASH JOIN                              |                           |     1 |   172 | 24273   (1)| 00:00:01 |       |       |
|   5 |      TABLE ACCESS FULL                     | GRADES_ENUM               |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|   6 |      VIEW                                  |                           |     1 |   157 | 24270   (1)| 00:00:01 |       |       |
|   7 |       SORT GROUP BY                        |                           |     1 |   284 | 24270   (1)| 00:00:01 |       |       |
|*  8 |        HASH JOIN                           |                           |     1 |   284 | 24269   (1)| 00:00:01 |       |       |
|*  9 |         HASH JOIN                          |                           |     1 |   245 | 13367   (1)| 00:00:01 |       |       |
|  10 |          NESTED LOOPS                      |                           |     1 |   210 |  2466   (1)| 00:00:01 |       |       |
|  11 |           NESTED LOOPS                     |                           |     1 |   210 |  2466   (1)| 00:00:01 |       |       |
|* 12 |            HASH JOIN                       |                           |     1 |   187 |  2466   (1)| 00:00:01 |       |       |
|  13 |             NESTED LOOPS                   |                           |     1 |   140 |  2463   (1)| 00:00:01 |       |       |
|  14 |              NESTED LOOPS                  |                           |     1 |   140 |  2463   (1)| 00:00:01 |       |       |
|  15 |               NESTED LOOPS                 |                           |     1 |    30 |  2463   (1)| 00:00:01 |       |       |
|  16 |                TABLE ACCESS FULL           | GRADES_ENUM               |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  17 |                PARTITION LIST ITERATOR     |                           |     1 |    15 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 18 |                 TABLE ACCESS FULL          | GRADES                    |     1 |    15 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 19 |               INDEX UNIQUE SCAN            | SYS_C009092               |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 20 |              TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS              |     1 |   110 |     0   (0)| 00:00:01 | ROWID | ROWID |
|  21 |             TABLE ACCESS FULL              | QUIDDITCH_TEAM_MEMBERS    |     1 |    47 |     3   (0)| 00:00:01 |       |       |
|* 22 |            INDEX UNIQUE SCAN               | SYS_C009102               |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 23 |           TABLE ACCESS BY INDEX ROWID      | HOUSES                    |     1 |    23 |     0   (0)| 00:00:01 |       |       |
|  24 |          PARTITION RANGE OR                |                           |     1 |    35 | 10901   (1)| 00:00:01 |KEY(OR)|KEY(OR)|
|* 25 |           TABLE ACCESS FULL                | POINTS                    |     1 |    35 | 10901   (1)| 00:00:01 |KEY(OR)|KEY(OR)|
|  26 |         VIEW                               |                           |     1 |    39 | 10902   (1)| 00:00:01 |       |       |
|  27 |          HASH GROUP BY                     |                           |     1 |    39 | 10902   (1)| 00:00:01 |       |       |
|  28 |           PARTITION RANGE ALL              |                           |     1 |    39 | 10901   (1)| 00:00:01 |     1 |    11 |
|  29 |            TABLE ACCESS FULL               | POINTS                    |     1 |    39 | 10901   (1)| 00:00:01 |     1 |    11 |
|  30 |     TABLE ACCESS BY INDEX ROWID            | TEACHERS                  |     1 |    81 |     0   (0)| 00:00:01 |       |       |
|* 31 |      INDEX UNIQUE SCAN                     | SYS_C009112               |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 32 |    INDEX UNIQUE SCAN                       | SYS_C009112               |     1 |       |     0   (0)| 00:00:01 |       |       |
|  33 |   TABLE ACCESS BY INDEX ROWID              | TEACHERS                  |     1 |    81 |     0   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------------------
```

11.3.5 Best students without partitions

```
-----------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|   1 |  SORT ORDER BY               |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|*  2 |   VIEW                       |              |    10 |  1330 |       |    22M  (1)| 00:14:52 |
|*  3 |    WINDOW SORT PUSHED RANK   |              |   631M|    45G|    52G|    22M  (1)| 00:14:52 |
|   4 |     HASH GROUP BY            |              |   631M|    45G|    52G|    22M  (1)| 00:14:52 |
|                                                                                                   |
|*  5 |      HASH JOIN               |              |   631M|    45G|    79M| 38180  (12)| 00:00:02 |
|*  6 |       TABLE ACCESS FULL      | POINTS       |  2986K|    45M|       |  8895   (1)| 00:00:01 |
|*  7 |       HASH JOIN              |              |  1977K|   115M|       | 14265   (2)| 00:00:01 |
|*  8 |        HASH JOIN             |              |  9346 |   355K|       |    22   (0)| 00:00:01 |
|   9 |         TABLE ACCESS FULL    | HOUSES       |     4 |    56 |       |     3   (0)| 00:00:01 |
|  10 |         TABLE ACCESS FULL    | STUDENTS     |  9346 |   228K|       |    19   (0)| 00:00:01 |
|  11 |        VIEW                  |              |  1977K|    41M|       | 14229   (2)| 00:00:01 |
|  12 |         HASH GROUP BY        |              |  1977K|    35M|    60M| 14229   (2)| 00:00:01 |
|* 13 |          HASH JOIN           |              |  1977K|    35M|       |  2521   (3)| 00:00:01 |
|                                                                                                   |
|                                                                                                   |
|                                                                                                   |
|                                                                                                   |
|  14 |           TABLE ACCESS FULL| GRADES_ENUM |     6 |    30 |       |     3   (0)| 00:00:01 |
|                                                                                                   |
|* 15 |           TABLE ACCESS FULL| GRADES       |  1977K|    26M|       |  2505   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------
```

11.3.6 Best students with partitions

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name         | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |              |    10 |  1330 | 13368   (1)| 00:00:01 |       |       |
|   1 |  SORT ORDER BY                           |              |    10 |  1330 | 13368   (1)| 00:00:01 |       |       |
|*  2 |   VIEW                                   |              |    10 |  1330 | 13367   (1)| 00:00:01 |       |       |
|*  3 |    WINDOW SORT PUSHED RANK               |              |     1 |   135 | 13367   (1)| 00:00:01 |       |       |
|   4 |     HASH GROUP BY                        |              |     1 |   135 | 13367   (1)| 00:00:01 |       |       |
|   5 |      VIEW                                | VM_NWVW_1    |     1 |   135 | 13365   (1)| 00:00:01 |       |       |
|   6 |       HASH GROUP BY                      |              |     1 |   240 | 13365   (1)| 00:00:01 |       |       |
|   7 |        NESTED LOOPS                      |              |     1 |   240 | 13364   (1)| 00:00:01 |       |       |
|   8 |         MERGE JOIN CARTESIAN             |              |     1 |   216 | 10904   (1)| 00:00:01 |       |       |
|   9 |          NESTED LOOPS                    |              |     1 |   201 | 10901   (1)| 00:00:01 |       |       |
|  10 |           NESTED LOOPS                   |              |     1 |   201 | 10901   (1)| 00:00:01 |       |       |
|  11 |            NESTED LOOPS                  |              |     1 |   166 | 10901   (1)| 00:00:01 |       |       |
|  12 |             PARTITION RANGE ITERATOR     |              |     1 |    47 | 10901   (1)| 00:00:01 |     1 |    10 |
|* 13 |              TABLE ACCESS FULL           | POINTS       |     1 |    47 | 10901   (1)| 00:00:01 |     1 |    10 |
|  14 |             TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS |     1 |   119 |     0   (0)| 00:00:01 | ROWID | ROWID |
|* 15 |              INDEX UNIQUE SCAN           | SYS_C009092  |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 16 |            INDEX UNIQUE SCAN             | SYS_C009102  |     1 |       |     0   (0)| 00:00:01 |       |       |
|  17 |           TABLE ACCESS BY INDEX ROWID    | HOUSES       |     1 |    35 |     0   (0)| 00:00:01 |       |       |
|  18 |          BUFFER SORT                     |              |     1 |    15 | 10904   (1)| 00:00:01 |       |       |
|  19 |           TABLE ACCESS FULL              | GRADES_ENUM  |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  20 |         PARTITION LIST ITERATOR          |              |     1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 21 |          TABLE ACCESS FULL               | GRADES       |     1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
-----------------------------------------------------------------------------------------------------------------------------
```

11.3.7 Assign subjects without partitions

```
-----------------------------------------------------------------------------------------------------
| Id  | Operation                  | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT           |                         |  1017 | 52920 |       | 12318   (3)| 00:00:01 |
|   1 |  LOAD TABLE CONVENTIONAL   | STUDENTS_SUBJECTS       |       |       |       |            |          |
|   2 |   HASH UNIQUE              |                         |  1017 | 52920 |       | 12318   (3)| 00:00:01 |
|   3 |    UNION-ALL               |                         |       |       |       |            |          |
|*  4 |     HASH JOIN              |                         |  1016 | 26416 |       |  6155   (3)| 00:00:01 |
|*  5 |      HASH JOIN ANTI        |                         |    93 |  1860 |       |  6152   (3)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL    | STUDENTS                |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|                                                                                                   |
|                                                                                                   |
|   7 |       VIEW                 | VW_NSO_1                | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|*  8 |        FILTER              |                         |       |       |       |            |          |
|   9 |         SORT GROUP BY      |                         | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 10 |          HASH JOIN         |                         |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  11 |           TABLE ACCESS FULL| GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|  12 |           TABLE ACCESS FULL| GRADES                  |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|                                                                                                   |
|* 13 |      TABLE ACCESS FULL     | SUBJECTS                |    76 |   456 |       |     3   (0)| 00:00:01 |
|* 14 |     HASH JOIN ANTI         |                         |     1 |    44 |       |  6158   (3)| 00:00:01 |
|* 15 |      HASH JOIN             |                         |    97 |  3007 |       |    25   (0)| 00:00:01 |
|  16 |       MERGE JOIN CARTESIAN |                         |    97 |  2328 |       |     6   (0)| 00:00:01 |
|* 17 |        TABLE ACCESS FULL   | SUBJECTS                |     1 |    20 |       |     3   (0)| 00:00:01 |
|  18 |        BUFFER SORT         |                         |    97 |   388 |       |     3   (0)| 00:00:01 |
|                                                                                                   |
|  19 |         TABLE ACCESS FULL  | QUIDDITCH_TEAM_MEMBERS  |    97 |   388 |       |     3   (0)| 00:00:01 |
|* 20 |       TABLE ACCESS FULL    | STUDENTS                |  8011 | 56077 |       |    19   (0)| 00:00:01 |
|* 21 |        TABLE ACCESS FULL   | SUBJECTS                |     1 |    20 |       |     3   (0)| 00:00:01 |
|  22 |      VIEW                  | VW_NSO_2                | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|* 23 |       FILTER               |                         |       |       |       |            |          |
|  24 |        SORT GROUP BY       |                         | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 25 |         HASH JOIN          |                         |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  26 |          TABLE ACCESS FULL | GRADES_ENUM             |     6 |    30 |       |     3   (0)| 00:00:01 |
|  27 |          TABLE ACCESS FULL | GRADES                  |  1977K|    16M|       |  2495   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------
```

11.3.8 Assign subjects with partitions

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                    | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT                   |                         |     2 |   302 |  4942   (1)| 00:00:01 |       |       |
|   1 |  LOAD TABLE CONVENTIONAL           | STUDENTS_SUBJECTS       |       |       |            |          |       |       |
|   2 |   HASH UNIQUE                      |                         |     2 |   302 |  4942   (1)| 00:00:01 |       |       |
|   3 |    UNION-ALL                       |                         |       |       |            |          |       |       |
|*  4 |     HASH JOIN                      |                         |     1 |    65 |  2468   (1)| 00:00:01 |       |       |
|   5 |      MERGE JOIN ANTI               |                         |     1 |    39 |  2465   (1)| 00:00:01 |       |       |
|*  6 |       TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS           |     1 |    26 |     0   (0)| 00:00:01 | ROWID | ROWID |
|   7 |        INDEX FULL SCAN             | SYS_C009092             |     1 |       |     0   (0)| 00:00:01 |       |       |
|*  8 |       SORT UNIQUE                  |                         |     1 |    13 |  2465   (1)| 00:00:01 |       |       |
|   9 |        VIEW                        | VW_NSO_1                |     1 |    13 |  2464   (1)| 00:00:01 |       |       |
|* 10 |         FILTER                     |                         |       |       |            |          |       |       |
|  11 |          SORT GROUP BY             |                         |     1 |    43 |  2464   (1)| 00:00:01 |       |       |
|  12 |           NESTED LOOPS             |                         |     1 |    43 |  2463   (1)| 00:00:01 |       |       |
|  13 |            TABLE ACCESS FULL       | GRADES_ENUM             |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  14 |            PARTITION LIST ITERATOR |                         |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 15 |             TABLE ACCESS FULL      | GRADES                  |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 16 |      TABLE ACCESS FULL             | SUBJECTS                |     1 |    26 |     3   (0)| 00:00:01 |       |       |
|  17 |     MERGE JOIN CARTESIAN           |                         |     1 |    86 |     6   (0)| 00:00:01 |       |       |
|  18 |      NESTED LOOPS                  |                         |     1 |    39 |     3   (0)| 00:00:01 |       |       |
|  19 |       NESTED LOOPS                 |                         |     1 |    39 |     3   (0)| 00:00:01 |       |       |
|  20 |        TABLE ACCESS FULL           | QUIDDITCH_TEAM_MEMBERS  |     1 |    13 |     3   (0)| 00:00:01 |       |       |
|* 21 |        INDEX UNIQUE SCAN           | SYS_C009092             |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 22 |       FILTER                       |                         |       |       |            |          |       |       |
|  23 |        HASH GROUP BY               |                         |     1 |    43 |  2464   (1)| 00:00:01 |       |       |
|  24 |         NESTED LOOPS               |                         |     1 |    43 |  2463   (1)| 00:00:01 |       |       |
|  25 |          TABLE ACCESS FULL         | GRADES_ENUM             |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  26 |          PARTITION LIST ITERATOR   |                         |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 27 |           TABLE ACCESS FULL        | GRADES                  |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 28 |      TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS            |     1 |    26 |     0   (0)| 00:00:01 | ROWID | ROWID |
|* 29 |       TABLE ACCESS FULL            | SUBJECTS                |     1 |    47 |     3   (0)| 00:00:01 |       |       |
|  30 |     BUFFER SORT                    |                         |     1 |    47 |     6   (0)| 00:00:01 |       |       |
|* 31 |      TABLE ACCESS FULL             | SUBJECTS                |     1 |    47 |     3   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------------------
```

11.3.9 Remove points without partitions

```
-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT             |              |  2078 | 64418 | 11460   (2)| 00:00:01 |
|   1 |  DELETE                      | POINTS       |       |       |            |          |
|*  2 |   HASH JOIN RIGHT SEMI       |              |  2078 | 64418 | 11460   (2)| 00:00:01 |
|   3 |    VIEW                      | VW_NSO_1     |    13 |   169 |  2569   (5)| 00:00:01 |
|*  4 |     SORT GROUP BY            |              |    13 |   221 |  2569   (5)| 00:00:01 |
|   5 |      VIEW                    |              |   248 |  4216 |  2569   (5)| 00:00:01 |
|                                                                                         |
|                                                                                         |
|*  6 |       FILTER                 |              |       |       |            |          |
|   7 |        HASH GROUP BY         |              |   248 |  5456 |  2569   (5)| 00:00:01 |
|*  8 |         HASH JOIN            |              |  4943 |  106K |  2568   (5)| 00:00:01 |
|   9 |          TABLE ACCESS FULL   | GRADES_ENUM  |     6 |    30 |     3   (0)| 00:00:01 |
|* 10 |          TABLE ACCESS FULL   | GRADES       |  4943 | 84031 |  2565   (5)| 00:00:01 |
|* 11 |    TABLE ACCESS FULL         | POINTS       |  1494K|   25M|  8881   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

11.3.10 Remove points with partitions

```
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT             |              |     1 |    65 | 13365   (1)| 00:00:01 |       |       |
|   1 |  DELETE                      | POINTS       |       |       |            |          |       |       |
|*  2 |   HASH JOIN SEMI             |              |     1 |    65 | 13365   (1)| 00:00:01 |       |       |
|   3 |    PARTITION RANGE ALL       |              |     1 |    52 | 10901   (1)| 00:00:01 |     1 |    11 |
|*  4 |     TABLE ACCESS FULL        | POINTS       |     1 |    52 | 10901   (1)| 00:00:01 |     1 |    11 |
|   5 |    VIEW                      | VW_NSO_1     |     1 |    13 |  2464   (1)| 00:00:01 |       |       |
|*  6 |     SORT GROUP BY            |              |     1 |    26 |  2464   (1)| 00:00:01 |       |       |
|   7 |      VIEW                    |              |     1 |    26 |  2464   (1)| 00:00:01 |       |       |
|*  8 |       FILTER                 |              |       |       |            |          |       |       |
|   9 |        HASH GROUP BY         |              |     1 |    52 |  2464   (1)| 00:00:01 |       |       |
|  10 |         NESTED LOOPS         |              |     1 |    52 |  2463   (1)| 00:00:01 |       |       |
|  11 |          TABLE ACCESS FULL   | GRADES_ENUM  |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  12 |          PARTITION LIST ITERATOR|           |     1 |    37 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 13 |           TABLE ACCESS FULL  | GRADES       |     1 |    37 |  2460   (1)| 00:00:01 |   KEY |   KEY |
-----------------------------------------------------------------------------------------------------------
```

11.4 Experiments

Some experiments were conducted, checking differences in performance between two different types of partitions for each test. For Grades analysis procedure we compared List partition on GRADES.value (default) with Hash partition on GRADES.student_id. For Points analysis we compared Range partition on POINTS.award_date with composite partition - firstly Range partition on POINTS.award_date with Hash subpartition on POINTS.teacher_id.

Procedure name	Min. measured time				Max. measured time				Average measured time				Query plan cost			
	Default	Alt.	Diff.	Profit	Default	Alt.	Diff.	Profit	Default	Alt.	Diff.	Profit	Default	Alt.	Diff.	Profit
Grades analysis	498	1108	-610	-122 %	537	1178	-641	-119 %	514.4	1144.0	-629.6	-122 %	18300	493 M	-493 M	-2694 K %
Points summary	468	806	-338	-72 %	540	857	-317	-59 %	507.8	820.6	-312.8	-62 %	24273	246 K	-222 K	-913 %

Both query plans prove that our initially selected partitions are much more efficient than the alternatives.

11.4.1 Grades analysis with default partition

```
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                 | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                      |     1 |   294 | 18300   (1)| 00:00:01 |       |       |
|   1 |  SORT ORDER BY                           |                      |     1 |   294 | 18300   (1)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                           |                      |     1 |   294 | 18299   (1)| 00:00:01 |       |       |
|   3 |    NESTED LOOPS                          |                      |     1 |   294 | 18299   (1)| 00:00:01 |       |       |
|   4 |     NESTED LOOPS                         |                      |     1 |   247 | 18299   (1)| 00:00:01 |       |       |
|*  5 |      HASH JOIN                           |                      |     1 |   200 | 18299   (1)| 00:00:01 |       |       |
|   6 |       MERGE JOIN CARTESIAN               |                      |     1 |    30 |     6   (0)| 00:00:01 |       |       |
|   7 |        TABLE ACCESS FULL                 | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|   8 |        BUFFER SORT                       |                      |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|   9 |         TABLE ACCESS FULL                | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  10 |       VIEW                               |                      |     1 |   170 | 18293   (1)| 00:00:01 |       |       |
|  11 |        SORT GROUP BY                     |                      |     1 |   280 | 18293   (1)| 00:00:01 |       |       |
|* 12 |         HASH JOIN                        |                      |     1 |   280 | 15829   (1)| 00:00:01 |       |       |
|* 13 |          HASH JOIN OUTER                 |                      |     1 |   228 | 13364   (1)| 00:00:01 |       |       |
|* 14 |           HASH JOIN                      |                      |     1 |   181 | 13361   (1)| 00:00:01 |       |       |
|  15 |            NESTED LOOPS                  |                      |     1 |   146 |  2460   (1)| 00:00:01 |       |       |
|  16 |             NESTED LOOPS                 |                      |     1 |   146 |  2460   (1)| 00:00:01 |       |       |
|  17 |              NESTED LOOPS                |                      |     1 |   123 |  2460   (1)| 00:00:01 |       |       |
|  18 |               PARTITION LIST ALL         |                      |     1 |    13 |  2460   (1)| 00:00:01 |     1 |     7 |
|  19 |                TABLE ACCESS FULL         | GRADES               |     1 |    13 |  2460   (1)| 00:00:01 |     1 |     7 |
|* 20 |               TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS      |     1 |   110 |     0   (0)| 00:00:01 | ROWID | ROWID |
|* 21 |                INDEX UNIQUE SCAN         | SYS_C009092          |     1 |       |     0   (0)| 00:00:01 |       |       |
|  22 |              NESTED LOOPS SEMI           |                      |     1 |    39 |  2463   (1)| 00:00:01 |       |       |
|* 23 |               TABLE ACCESS FULL          | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|* 24 |                TABLE ACCESS FULL         | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|* 25 |               TABLE ACCESS FULL          | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  26 |              PARTITION LIST ITERATOR     |                      |     1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 27 |               TABLE ACCESS FULL          | GRADES               |     1 |    24 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 28 |             INDEX UNIQUE SCAN            | SYS_C009102          |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 29 |            TABLE ACCESS BY INDEX ROWID   | HOUSES               |     1 |    23 |     0   (0)| 00:00:01 |       |       |
|  30 |           PARTITION RANGE ITERATOR       |                      |     1 |    35 | 10901   (1)| 00:00:01 |     1 |    10 |
|* 31 |            TABLE ACCESS FULL             | POINTS               |     1 |    35 | 10901   (1)| 00:00:01 |     1 |    10 |
|  32 |          TABLE ACCESS FULL               | QUIDDITCH_TEAM_MEMBERS |     1 |    47 |     3   (0)| 00:00:01 |       |       |
|  33 |         VIEW                             |                      |     1 |    52 |  2464   (1)| 00:00:01 |       |       |
|  34 |          HASH GROUP BY                   |                      |     1 |    43 |  2464   (1)| 00:00:01 |       |       |
|  35 |           NESTED LOOPS                   |                      |     1 |    43 |  2463   (1)| 00:00:01 |       |       |
|  36 |            TABLE ACCESS FULL             | GRADES_ENUM          |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  37 |            PARTITION LIST ITERATOR       |                      |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 38 |             TABLE ACCESS FULL            | GRADES               |     1 |    28 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|  39 |     TABLE ACCESS BY INDEX ROWID          | SUBJECTS             |     1 |    47 |     0   (0)| 00:00:01 |       |       |
|* 40 |      INDEX UNIQUE SCAN                   | SYS_C009130          |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 41 |    INDEX UNIQUE SCAN                     | SYS_C009130          |     1 |       |     0   (0)| 00:00:01 |       |       |
|  42 |   TABLE ACCESS BY INDEX ROWID            | SUBJECTS             |     1 |    47 |     0   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------------------------
```

11.4.2 Grades analysis with alternative partition

```
-----------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                            |  5764M|  1578G|       |   493M  (1)| 05:21:04 |       |       |
|   1 |  SORT ORDER BY                     |                            |  5764M|  1578G|  1691G|   493M  (1)| 05:21:04 |       |       |
|*  2 |   HASH JOIN                        |                            |  5764M|  1578G|       |   963K (77)| 00:00:38 |       |       |
|   3 |    TABLE ACCESS FULL               | SUBJECTS                   |    76 |  3572 |       |     3   (0)| 00:00:01 |       |       |
|*  4 |    HASH JOIN                       |                            |  5764M|  1325G|       |   924K (76)| 00:00:37 |       |       |
|   5 |     MERGE JOIN CARTESIAN           |                            |  2736 |   205K|       |    64   (0)| 00:00:01 |       |       |
|   6 |      MERGE JOIN CARTESIAN          |                            |    36 |  1080 |       |    13   (0)| 00:00:01 |       |       |
|   7 |       TABLE ACCESS FULL            | GRADES_ENUM                |     6 |    90 |       |     3   (0)| 00:00:01 |       |       |
|   8 |       BUFFER SORT                  |                            |     6 |    90 |       |    10   (0)| 00:00:01 |       |       |
|   9 |        TABLE ACCESS FULL           | GRADES_ENUM                |     6 |    90 |       |     2   (0)| 00:00:01 |       |       |
|  10 |      BUFFER SORT                   |                            |    76 |  3572 |       |    62   (0)| 00:00:01 |       |       |
|  11 |       TABLE ACCESS FULL            | SUBJECTS                   |    76 |  3572 |       |     1   (0)| 00:00:01 |       |       |
|  12 |     VIEW                           |                            |  5764M|   912G|       |   886K (74)| 00:00:35 |       |       |
|  13 |      SORT GROUP BY                 |                            |  5764M|  1513G|       |   886K (74)| 00:00:35 |       |       |
|* 14 |       HASH JOIN RIGHT OUTER        |                            |  5764M|  1513G|       |   326K (30)| 00:00:13 |       |       |
|  15 |        TABLE ACCESS FULL           | QUIDDITCH_TEAM_MEMBERS     |    97 |  4559 |       |     3   (0)| 00:00:01 |       |       |
|* 16 |        HASH JOIN                   |                            |  5764M|  1261G|   97M|   288K (21)| 00:00:12 |       |       |
|  17 |         PART JOIN FILTER CREATE    | :BF0000                    |  1590K|    78M|       |  3418   (5)| 00:00:01 |       |       |
|  18 |          VIEW                      |                            |  1590K|    78M|       |  3418   (5)| 00:00:01 |       |       |
|  19 |           HASH GROUP BY            |                            |  1590K|    65M|       |  3418   (5)| 00:00:01 |       |       |
|* 20 |            HASH JOIN               |                            |  1590K|    65M|       |  3319   (2)| 00:00:01 |       |       |
|  21 |             TABLE ACCESS FULL      | GRADES_ENUM                |     6 |    90 |       |     3   (0)| 00:00:01 |       |       |
|  22 |             PARTITION HASH ALL     |                            |  1590K|    42M|       |  3305   (1)| 00:00:01 |     1 |     4 |
|  23 |              TABLE ACCESS FULL     | GRADES                     |  1590K|    42M|       |  3305   (1)| 00:00:01 |     1 |     4 |
|* 24 |         HASH JOIN                  |                            |    21M|  3690M|   13M| 47048  (41)| 00:00:02 |       |       |
|  25 |          PART JOIN FILTER CREATE   | :BF0001                    |    21M|  3690M|       | 47048  (41)| 00:00:02 |       |       |
|* 26 |           HASH JOIN                |                            | 77571 |    12M|       | 41045  (46)| 00:00:02 |       |       |
|  27 |            NESTED LOOPS ANTI       |                            |   268 | 36180 |       | 11345   (1)| 00:00:01 |       |       |
|* 28 |             HASH JOIN              |                            |  1459 |   189K|       |  1096   (1)| 00:00:01 |       |       |
|* 29 |              TABLE ACCESS FULL     | HOUSES                     |     1 |    23 |       |     3   (0)| 00:00:01 |       |       |
|  30 |              PARTITION HASH JOIN-FILTER|                        |  5835 |   626K|       |  1093   (1)| 00:00:01 |:BF0000|:BF0000|
|* 31 |               TABLE ACCESS FULL    | STUDENTS                   |  5835 |   626K|       |  1093   (1)| 00:00:01 |:BF0000|:BF0000|
|  32 |             VIEW PUSHED PREDICATE  | VW_NSO_1                   |     1 |     2 |       |     7   (0)| 00:00:01 |       |       |
|  33 |              NESTED LOOPS          |                            |     1 |    39 |       |     7   (0)| 00:00:01 |       |       |
|* 34 |               TABLE ACCESS FULL    | GRADES_ENUM                |     1 |    15 |       |     3   (0)| 00:00:01 |       |       |
|* 35 |               TABLE ACCESS FULL    | GRADES_ENUM                |     1 |    15 |       |     3   (0)| 00:00:01 |       |       |
|* 36 |               TABLE ACCESS FULL    | GRADES_ENUM                |     1 |    15 |       |     3   (0)| 00:00:01 |       |       |
|  37 |              PARTITION HASH SINGLE |                            |  2608 | 62592 |       |     4   (0)| 00:00:01 |   KEY |   KEY |
|* 38 |               TABLE ACCESS FULL    | GRADES                     |  2608 | 62592 |       |     4   (0)| 00:00:01 |   KEY |   KEY |
|  39 |           PARTITION RANGE ITERATOR |                            |  2400K|    80M|       | 29684  (64)| 00:00:02 |     1 |    10 |
|  40 |            PARTITION HASH ALL      |                            |  2400K|    80M|       | 29684  (64)| 00:00:02 |     1 |     4 |
|* 41 |             TABLE ACCESS FULL      | POINTS                     |  2400K|    80M|       | 29684  (64)| 00:00:02 |     1 |    40 |
|  42 |          PARTITION HASH JOIN-FILTER|                            |  1590K|    19M|       |  3303   (1)| 00:00:01 |:BF0001|:BF0001|
|  43 |           TABLE ACCESS FULL        | GRADES                     |  1590K|    19M|       |  3303   (1)| 00:00:01 |:BF0001|:BF0001|
-----------------------------------------------------------------------------------------------------------------------------------------
```

11.4.3 Points summary with default partition

```
-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                  | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                       |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   1 |  NESTED LOOPS                            |                       |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                           |                       |     1 |   334 | 24273   (1)| 00:00:01 |       |       |
|   3 |    NESTED LOOPS                          |                       |     1 |   253 | 24273   (1)| 00:00:01 |       |       |
|*  4 |     HASH JOIN                            |                       |     1 |   172 | 24273   (1)| 00:00:01 |       |       |
|   5 |      TABLE ACCESS FULL                   | GRADES_ENUM           |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|   6 |      VIEW                                |                       |     1 |   157 | 24270   (1)| 00:00:01 |       |       |
|   7 |       SORT GROUP BY                      |                       |     1 |   284 | 24270   (1)| 00:00:01 |       |       |
|*  8 |        HASH JOIN                         |                       |     1 |   284 | 24269   (1)| 00:00:01 |       |       |
|*  9 |         HASH JOIN                        |                       |     1 |   245 | 13367   (1)| 00:00:01 |       |       |
|  10 |          NESTED LOOPS                    |                       |     1 |   210 |  2466   (1)| 00:00:01 |       |       |
|  11 |           NESTED LOOPS                   |                       |     1 |   210 |  2466   (1)| 00:00:01 |       |       |
|* 12 |            HASH JOIN                     |                       |     1 |   187 |  2466   (1)| 00:00:01 |       |       |
|  13 |             NESTED LOOPS                 |                       |     1 |   140 |  2463   (1)| 00:00:01 |       |       |
|  14 |              NESTED LOOPS                |                       |     1 |   140 |  2463   (1)| 00:00:01 |       |       |
|  15 |               NESTED LOOPS               |                       |     1 |    30 |  2463   (1)| 00:00:01 |       |       |
|  16 |                TABLE ACCESS FULL         | GRADES_ENUM           |     1 |    15 |     3   (0)| 00:00:01 |       |       |
|  17 |                PARTITION LIST ITERATOR   |                       |     1 |    15 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 18 |                 TABLE ACCESS FULL        | GRADES                |     1 |    15 |  2460   (1)| 00:00:01 |   KEY |   KEY |
|* 19 |               INDEX UNIQUE SCAN          | SYS_C009092           |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 20 |              TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS        |     1 |   110 |     0   (0)| 00:00:01 | ROWID | ROWID |
|  21 |             TABLE ACCESS FULL            | QUIDDITCH_TEAM_MEMBERS|     1 |    47 |     3   (0)| 00:00:01 |       |       |
|* 22 |            INDEX UNIQUE SCAN             | SYS_C009102           |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 23 |           TABLE ACCESS BY INDEX ROWID    | HOUSES                |     1 |    23 |     0   (0)| 00:00:01 |       |       |
|  24 |          PARTITION RANGE OR              |                       |     1 |    35 | 10901   (1)| 00:00:01 |KEY(OR)|KEY(OR)|
|* 25 |           TABLE ACCESS FULL              | POINTS                |     1 |    35 | 10901   (1)| 00:00:01 |KEY(OR)|KEY(OR)|
|  26 |         VIEW                             |                       |     1 |    39 | 10902   (1)| 00:00:01 |       |       |
|  27 |          HASH GROUP BY                   |                       |     1 |    39 | 10902   (1)| 00:00:01 |       |       |
|  28 |           PARTITION RANGE ALL            |                       |     1 |    39 | 10901   (1)| 00:00:01 |     1 |    11 |
|  29 |            TABLE ACCESS FULL             | POINTS                |     1 |    39 | 10901   (1)| 00:00:01 |     1 |    11 |
|  30 |     TABLE ACCESS BY INDEX ROWID          | TEACHERS              |     1 |    81 |     0   (0)| 00:00:01 |       |       |
|* 31 |      INDEX UNIQUE SCAN                   | SYS_C009112           |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 32 |    INDEX UNIQUE SCAN                     | SYS_C009112           |     1 |       |     0   (0)| 00:00:01 |       |       |
|  33 |   TABLE ACCESS BY INDEX ROWID            | TEACHERS              |     1 |    81 |     0   (0)| 00:00:01 |       |       |
-------------------------------------------------------------------------------------------------------------------------
```

11.4.4 Points summary with alternative partition

```
-----------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                       |  1336M|   415G|       |   246K (61)| 00:00:10 |       |       |
|*  1 |  HASH JOIN                               |                       |  1336M|   415G|       |   246K (61)| 00:00:10 |       |       |
|   2 |   TABLE ACCESS FULL                      | TEACHERS              |  1097 | 88857 |       |     5   (0)| 00:00:01 |       |       |
|*  3 |   HASH JOIN                              |                       |  1336M|   315G|       |   237K (59)| 00:00:10 |       |       |
|   4 |    MERGE JOIN CARTESIAN                  |                       |  6582 |   617K|       |    26   (0)| 00:00:01 |       |       |
|   5 |     TABLE ACCESS FULL                    | GRADES_ENUM           |     6 |    90 |       |     3   (0)| 00:00:01 |       |       |
|   6 |     BUFFER SORT                          |                       |  1097 | 88857 |       |    23   (0)| 00:00:01 |       |       |
|   7 |      TABLE ACCESS FULL                   | TEACHERS              |  1097 | 88857 |       |     4   (0)| 00:00:01 |       |       |
|   8 |    VIEW                                  |                       |  1336M|   195G|       |   228K (58)| 00:00:09 |       |       |
|   9 |     SORT GROUP BY                        |                       |  1336M|   353G|       |   228K (58)| 00:00:09 |       |       |
|* 10 |      HASH JOIN                           |                       |  1336M|   353G|  113M|   107K  (9)| 00:00:05 |       |       |
|  11 |       PARTITION RANGE OR                 |                       |  2536K|    84M|       | 11323   (1)| 00:00:01 |KEY(OR)|KEY(OR)|
|  12 |        PARTITION HASH ALL                |                       |  2536K|    84M|       | 11323   (1)| 00:00:01 |     1 |     4 |
|* 13 |         TABLE ACCESS FULL                | POINTS                |  2536K|    84M|       | 11323   (1)| 00:00:01 |   KEY |   KEY |
|* 14 |       HASH JOIN                          |                       |  5170K|  1227M|       | 17321   (3)| 00:00:01 |       |       |
|  15 |        TABLE ACCESS FULL                 | GRADES_ENUM           |     6 |    90 |       |     3   (0)| 00:00:01 |       |       |
|* 16 |        HASH JOIN                         |                       |  5170K|  1153M|  5904K| 17284   (3)| 00:00:01 |       |       |
|  17 |         PART JOIN FILTER CREATE          | :BF0000               |  5170K|  1153M|       | 17284   (3)| 00:00:01 |       |       |
|* 18 |          HASH JOIN                       |                       | 26164 |  5595K|       | 11618   (3)| 00:00:01 |       |       |
|  19 |           JOIN FILTER CREATE             | :BF0001               |    73 | 13140 |       |   103   (0)| 00:00:01 |       |       |
|* 20 |            HASH JOIN                     |                       |    73 | 13140 |       |   103   (0)| 00:00:01 |       |       |
|* 21 |             TABLE ACCESS FULL            | HOUSES                |     3 |    69 |       |     3   (0)| 00:00:01 |       |       |
|  22 |             NESTED LOOPS                 |                       |    97 | 15229 |       |   100   (0)| 00:00:01 |       |       |
|  23 |              NESTED LOOPS                |                       |    97 | 15229 |       |   100   (0)| 00:00:01 |       |       |
|  24 |               TABLE ACCESS FULL          | QUIDDITCH_TEAM_MEMBERS|    97 |  4559 |       |     3   (0)| 00:00:01 |       |       |
|* 25 |               INDEX UNIQUE SCAN          | SYS_C009169           |     1 |       |       |     0   (0)| 00:00:01 |       |       |
|* 26 |              TABLE ACCESS BY GLOBAL INDEX ROWID| STUDENTS        |     1 |   110 |       |     1   (0)| 00:00:01 | ROWID | ROWID |
|  27 |            VIEW                         |                       |  2894K|   107M|       | 11496   (3)| 00:00:01 |       |       |
|  28 |             HASH GROUP BY                |                       |  2894K|   107M|       | 11496   (3)| 00:00:01 |       |       |
|  29 |              JOIN FILTER USE             | :BF0001               |  2894K|   107M|       | 11309   (1)| 00:00:01 |       |       |
|  30 |               PARTITION RANGE ALL        |                       |  2894K|   107M|       | 11309   (1)| 00:00:01 |     1 |    11 |
|  31 |                PARTITION HASH ALL        |                       |  2894K|   107M|       | 11309   (1)| 00:00:01 |     1 |     4 |
|* 32 |                 TABLE ACCESS FULL        | POINTS                |  2894K|   107M|       | 11309   (1)| 00:00:01 |     1 |    44 |
|  33 |         PARTITION HASH JOIN-FILTER       |                       |  1590K|    22M|       |  3303   (1)| 00:00:01 |:BF0000|:BF0000|
|  34 |          TABLE ACCESS FULL               | GRADES                |  1590K|    22M|       |  3303   (1)| 00:00:01 |:BF0000|:BF0000|
-----------------------------------------------------------------------------------------------------------------------------------
```

12 Columnar store - implementation

12.1 Implemented columnar storage

We implemented Oracle In-Memory columnar storage for our databases. We configured it for key tables that are frequently accessed in our workload and benefit from columnar compression and parallel processing. These tables are often subject to aggregate functions, so the columnar storage should improve the efficiency of our workload.

First, we selected the in-memory size to be 250 MB using:

```
ALTER SYSTEM SET inmemory_size=250M SCOPE=SPFILE;
```

We selected tables for columnar storage based on the usage of these tables in our workload. For all of the selected tables, we used MEMCOMPRESS FOR QUERY HIGH, because experiments showed that it is usually the best option for us. Analysis of the workload made us select the following tables:

- **GRADES and POINTS** (Critical Priority): Large fact tables with millions of records. Heavy analytical processing with aggregations, filtering, and joins. Core tables for most workload queries. However, column POINTS.description should be excluded from columnar storage as it is a column with relatively big textual data that is not being used in any query.

- **STUDENTS** (High Priority): Central dimensional table frequently joined with fact tables. Medium size but high join frequency across all analytical queries.
- **SUBJECTS and STUDENTS_SUBJECTS** (High Priority): Key dimensional and junction tables essential for query performance. Moderate size but critical for join operations.

Small lookup tables (HOUSES, TEACHERS, GRADES_ENUM) remained in traditional row storage due to their size and access patterns being more suitable for OLTP operations.

```
ALTER TABLE Points
    INMEMORY MEMCOMPRESS FOR QUERY HIGH PRIORITY CRITICAL
    NO INMEMORY (description);

ALTER TABLE Grades
    INMEMORY MEMCOMPRESS FOR QUERY HIGH PRIORITY CRITICAL;

ALTER TABLE Students
    INMEMORY MEMCOMPRESS FOR QUERY HIGH PRIORITY HIGH;

ALTER TABLE Subjects
    INMEMORY MEMCOMPRESS FOR QUERY HIGH PRIORITY HIGH;

ALTER TABLE Students_Subjects
    INMEMORY MEMCOMPRESS FOR QUERY HIGH PRIORITY HIGH;
```

12.2 Cost comparison

We ran the whole workload on the database with columnar store enabled and disabled, **10 times** in each case. Enabling columnar store influenced positively 5 out of 6 queries in our workload in terms of query plan cost. In terms of actual running times, the improvement is only visible in 3 out of 6 queries. For the whole workload, however, both running time and query plan cost were overall improved, so the usage of columnar store in our database is completely justified.

Proc. name	**Min. measured time**				**Max. measured time**				**Average measured time**				**Query plan cost**			
	Disa.	Ena.	Gain	Profit	Disa.	Ena.	Gain	Profit	Disa.	Ena.	Gain	Profit	Disa.	Ena.	Gain	Profit
Grades analysis	564	517	-47	-8.3 %	813	685	-128	-15.7 %	677.0	565.8	-111.2	-16.4 %	3628 K	816 K	-2812 K	-77.5 %
Points summary	377	492	115	30.5 %	556	659	103	18.5 %	468.6	558.7	90.1	19.2 %	39 M	260 K	-38740 K	-99.3 %
Best students	221	291	70	31.7 %	297	357	60	20.2 %	250.0	319.9	69.9	28.0 %	22 M	38839	-21961 K	-99.8 %
Raise grades	26694	1767	-24927	-93.4 %	33966	9302	-24664	-72.6 %	30823.8	7923.3	-22900.5	-74.3 %	11460	833 K	821540	7168.8 %
Assign subjects	2735	2232	-503	-18.4 %	3932	3250	-682	-17.3 %	3403.8	2571.8	-832	-24.4 %	12318	2095	-10223	-83.0 %
Remove points	26694	41439	14745	55.2 %	33966	56561	22595	66.5 %	30823.8	44774.3	13950.5	45.3 %	11460	6819	-4641	-40.5 %
Whole workload	57285	46738	-10547	-18.4 %	73530	70814	-2716	-3.7 %	66447	56713.8	-9733.2	-14.6 %	64663 K	1957 K	-62706 K	-97.0 %

12.3 Query plan comparison

Enabling columnar store significantly changed all the query plans. Below we present their comparison.

12.3.1 Grades analysis without columnar store

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name                      | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                           |    47M|  9981M|       |  3628K  (1)| 00:02:22 |
|   1 |  SORT ORDER BY                 |                           |    47M|  9981M|    10G|  3628K  (1)| 00:02:22 |
|*  2 |   HASH JOIN                    |                           |    47M|  9981M|       |  1359K  (1)| 00:00:54 |
|   3 |    TABLE ACCESS FULL           | SUBJECTS                  |    76 |  1520 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                   |                           |    47M|  9074M|       |  1359K  (1)| 00:00:54 |
|   5 |     MERGE JOIN CARTESIAN       |                           |  2736 | 82080 |       |    64   (0)| 00:00:01 |
|   6 |      MERGE JOIN CARTESIAN      |                           |    36 |   360 |       |    13   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL        | GRADES_ENUM               |     6 |    30 |       |     3   (0)| 00:00:01 |
|   8 |       BUFFER SORT              |                           |     6 |    30 |       |    10   (0)| 00:00:01 |
|   9 |        TABLE ACCESS FULL       | GRADES_ENUM               |     6 |    30 |       |     2   (0)| 00:00:01 |
|  10 |      BUFFER SORT               |                           |    76 |  1520 |       |    62   (0)| 00:00:01 |
|  11 |       TABLE ACCESS FULL        | SUBJECTS                  |    76 |  1520 |       |     1   (0)| 00:00:01 |
|  12 |     VIEW                       |                           |    47M|  7712M|       |  1358K  (1)| 00:00:54 |
|  13 |      SORT GROUP BY             |                           |    47M|  5671M|  6092M|  1358K  (1)| 00:00:54 |
|* 14 |       HASH JOIN                |                           |    47M|  5671M|    28M| 24853   (3)| 00:00:01 |
|* 15 |        HASH JOIN               |                           |   224K|    25M|       | 19118   (2)| 00:00:01 |
|* 16 |         HASH JOIN              |                           |   711 | 74655 |       | 10204   (3)| 00:00:01 |
|  17 |          JOIN FILTER CREATE    | :BF0000                   |    13 |   806 |       |  2548   (3)| 00:00:01 |
|* 18 |           HASH JOIN OUTER      |                           |    13 |   806 |       |  2548   (3)| 00:00:01 |
|* 19 |            HASH JOIN ANTI      |                           |    13 |   585 |       |  2545   (3)| 00:00:01 |
|* 20 |             HASH JOIN          |                           |  1323 | 54243 |       |    22   (0)| 00:00:01 |
|* 21 |              TABLE ACCESS FULL | HOUSES                    |     1 |    14 |       |     3   (0)| 00:00:01 |
|* 22 |              TABLE ACCESS FULL | STUDENTS                  |  5291 |  139K |       |    19   (0)| 00:00:01 |
|  23 |             VIEW               | VW_NSO_1                  |  190K |  745K |       |  2521   (3)| 00:00:01 |
|* 24 |              HASH JOIN         |                           |  190K | 3541K |       |  2521   (3)| 00:00:01 |
|* 25 |               TABLE ACCESS FULL| GRADES_ENUM               |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 26 |               TABLE ACCESS FULL| GRADES_ENUM               |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 27 |               TABLE ACCESS FULL| GRADES_ENUM               |     1 |     5 |       |     3   (0)| 00:00:01 |
|* 28 |              TABLE ACCESS FULL | GRADES                    |  1957K|    26M|       |  2505   (2)| 00:00:01 |
|  29 |            TABLE ACCESS FULL   | QUIDDITCH_TEAM_MEMBERS    |    97 |  1649 |       |     3   (0)| 00:00:01 |
|  30 |          VIEW                  |                           |  502K |    20M|       |  7653   (3)| 00:00:01 |
|  31 |           HASH GROUP BY        |                           |  502K |  6866K|    45M|  7653   (3)| 00:00:01 |
|* 32 |            HASH JOIN           |                           | 1977K |    26M|       |  2511   (2)| 00:00:01 |
|  33 |             TABLE ACCESS FULL  | GRADES_ENUM               |     6 |    30 |       |     3   (0)| 00:00:01 |
|  34 |             JOIN FILTER USE    | :BF0000                   | 1977K |    16M|       |  2495   (2)| 00:00:01 |
|* 35 |              TABLE ACCESS FULL | GRADES                    | 1977K |    16M|       |  2495   (2)| 00:00:01 |
|* 36 |         TABLE ACCESS FULL      | POINTS                    | 2956K |    45M|       |  8895   (1)| 00:00:01 |
|  37 |        TABLE ACCESS FULL       | GRADES                    | 1977K |  7723K|       |  2492   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------
```

12.3.2 Grades analysis with columnar store

```
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name                      | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                           |   110 | 32340 |       |   816K (74)| 00:00:32 |
|   1 |  SORT ORDER BY                 |                           |   110 | 32340 |       |   816K (74)| 00:00:32 |
|*  2 |   HASH JOIN                    |                           |   110 | 32340 |       |   816K (74)| 00:00:32 |
|   3 |    TABLE ACCESS INMEMORY FULL  | SUBJECTS                  |    76 |  3572 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                   |                           |   110 | 27170 |       |   816K (74)| 00:00:32 |
|   5 |     TABLE ACCESS INMEMORY FULL | SUBJECTS                  |    76 |  3572 |       |     3   (0)| 00:00:01 |
|*  6 |     HASH JOIN                  |                           |   110 | 22000 |       |   816K (74)| 00:00:32 |
|   7 |      TABLE ACCESS FULL         | GRADES_ENUM               |     6 |    90 |       |     3   (0)| 00:00:01 |
|*  8 |      HASH JOIN                 |                           |   110 | 20350 |       |   816K (74)| 00:00:32 |
|   9 |       TABLE ACCESS FULL        | GRADES_ENUM               |     6 |    90 |       |     3   (0)| 00:00:01 |
|  10 |       VIEW                     |                           |   110 | 18700 |       |   816K (74)| 00:00:32 |
|  11 |        SORT GROUP BY           |                           |   110 | 32230 |       |   816K (74)| 00:00:32 |
|* 12 |         HASH JOIN RIGHT OUTER  |                           |  5468M|  1492G|       |   287K (26)| 00:00:12 |
|  13 |          TABLE ACCESS FULL     | QUIDDITCH_TEAM_MEMBERS    |    97 |  4559 |       |     3   (0)| 00:00:01 |
|* 14 |          HASH JOIN             |                           |  5468M|  1252G|    39M|   250K (15)| 00:00:10 |
|  15 |           TABLE ACCESS INMEMORY FULL | GRADES                |  1656K|    20M|       |   100   (7)| 00:00:01 |
|* 16 |           HASH JOIN            |                           |    17M|  3939M|    18M|  6856   (6)| 00:00:01 |
|  17 |            JOIN FILTER CREATE  | :BF0000                   |   101K|    17M|       |   556  (23)| 00:00:01 |
|* 18 |             HASH JOIN          |                           |   101K|    17M|       |   556  (23)| 00:00:01 |
|  19 |              JOIN FILTER CREATE| :BF0001                   |   315 | 45990 |       |   136  (27)| 00:00:01 |
|* 20 |               HASH JOIN ANTI   |                           |   315 | 45990 |       |   136  (27)| 00:00:01 |
|  21 |                JOIN FILTER CREATE| :BF0002                   |  1343 |  174K|       |     4   (0)| 00:00:01 |
|* 22 |                 HASH JOIN      |                           |  1343 |  174K|       |     4   (0)| 00:00:01 |
|  23 |                  JOIN FILTER CREATE| :BF0003                   |     1 |    23 |       |     3   (0)| 00:00:01 |
|* 24 |                   TABLE ACCESS FULL| HOUSES                    |     1 |    23 |       |     3   (0)| 00:00:01 |
|  25 |                  JOIN FILTER USE| :BF0003                   |  5370 |  576K|       |     1   (0)| 00:00:01 |
|* 26 |                   TABLE ACCESS INMEMORY FULL | STUDENTS                  |  5370 |  576K|       |     1   (0)| 00:00:01 |
|  27 |                VIEW            | VW_NSO_1                  |   158K|  2012K|       |   130  (27)| 00:00:01 |
|* 28 |                 HASH JOIN      |                           |   158K|  6038K|       |   130  (27)| 00:00:01 |
|  29 |                  JOIN FILTER CREATE| :BF0004                   |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 30 |                   TABLE ACCESS FULL| GRADES_ENUM               |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 31 |                   TABLE ACCESS FULL| GRADES_ENUM               |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 32 |                   TABLE ACCESS FULL| GRADES_ENUM               |     1 |    15 |       |     3   (0)| 00:00:01 |
|  33 |                  JOIN FILTER USE| :BF0002                   |  1626K|    37M|       |   116  (20)| 00:00:01 |
|  34 |                   JOIN FILTER USE| :BF0004                   |  1626K|    37M|       |   116  (20)| 00:00:01 |
|* 35 |                    TABLE ACCESS INMEMORY FULL| GRADES                    |  1626K|    37M|       |   116  (20)| 00:00:01 |
|  36 |              JOIN FILTER USE   | :BF0001                   |  2958K|    98M|       |   401  (19)| 00:00:01 |
|* 37 |               TABLE ACCESS INMEMORY FULL | POINTS                    |  2958K|    98M|       |   401  (19)| 00:00:01 |
|  38 |            VIEW                |                           |  1656K|    82M|       |   229  (59)| 00:00:01 |
|  39 |             HASH GROUP BY      |                           |  1656K|   170M|       |   229  (59)| 00:00:01 |
|* 40 |              HASH JOIN         |                           |  1656K|   170M|       |   126  (24)| 00:00:01 |
|  41 |               VIEW             | VW_GBC_8                  |     6 |   168 |       |     4  (25)| 00:00:01 |
|  42 |                HASH GROUP BY   |                           |     6 |    90 |       |     4  (25)| 00:00:01 |
|  43 |                 TABLE ACCESS FULL | GRADES_ENUM               |     6 |    90 |       |     3   (0)| 00:00:01 |
|  44 |               VIEW             | VW_GBF_9                  |  1656K|   126M|       |   111  (17)| 00:00:01 |
|  45 |                HASH GROUP BY   |                           |  1656K|    44M|       |   111  (17)| 00:00:01 |
|  46 |                 JOIN FILTER USE| :BF0000                   |  1656K|    44M|       |   110  (16)| 00:00:01 |
|* 47 |                  TABLE ACCESS INMEMORY FULL | GRADES                    |  1656K|    44M|       |   110  (16)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------
```

12.3.3 Points summary without columnar store

```
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                       |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|*  1 |  HASH JOIN                   |                       |  1105M|   205G|       |    39M  (1)| 00:25:29 |
|   2 |   TABLE ACCESS FULL          | TEACHERS              |  1097 | 20843 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                       |  1105M|   186G|       |    39M  (1)| 00:25:29 |
|   4 |    MERGE JOIN CARTESIAN      |                       |  6582 |   154K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                       |  1097 | 20843 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS              |  1097 | 20843 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                       |  1105M|   161G|       |    39M  (1)| 00:25:29 |
|   9 |     SORT GROUP BY            |                       |  1105M|   118G|   135G|    39M  (1)| 00:25:29 |
|* 10 |      HASH JOIN               |                       |  1120M|   120G|   119M| 65365  (13)| 00:00:03 |
|  11 |       VIEW                   |                       |  2986K|    85M|       | 22617   (2)| 00:00:01 |
|  12 |        HASH GROUP BY         |                       |  2986K|    34M|    68M| 22617   (2)| 00:00:01 |
|  13 |         TABLE ACCESS FULL    | POINTS                |  2986K|    34M|       |  8879   (1)| 00:00:01 |
|* 14 |       HASH JOIN              |                       |  3506K|   284M|       | 13234   (2)| 00:00:01 |
|  15 |        TABLE ACCESS FULL     | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 16 |        HASH JOIN             |                       |  3506K|   267M|  1400K| 13207   (2)| 00:00:01 |
|* 17 |         HASH JOIN            |                       | 16576 |  1197K|       |  8928   (1)| 00:00:01 |
|* 18 |          HASH JOIN           |                       |    73 |  4234 |       |    25   (0)| 00:00:01 |
|* 19 |           HASH JOIN          |                       |    97 |  4268 |       |    22   (0)| 00:00:01 |
|  20 |            TABLE ACCESS FULL | QUIDDITCH_TEAM_MEMBERS|    97 |  1649 |       |     3   (0)| 00:00:01 |
|* 21 |            TABLE ACCESS FULL | STUDENTS              |  5504 |   145K|       |    19   (0)| 00:00:01 |
|* 22 |          TABLE ACCESS FULL   | HOUSES                |     3 |    42 |       |     3   (0)| 00:00:01 |
|* 23 |         TABLE ACCESS FULL    | POINTS                |  2129K|    32M|       |  8888   (1)| 00:00:01 |
|  24 |        TABLE ACCESS FULL     | GRADES                |  1977K|    11M|       |  2492   (2)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------
```

12.3.4 Points summary with columnar store

```
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                       |  1668M|   518G|       |   260K (72)| 00:00:11 |
|*  1 |  HASH JOIN                   |                       |  1668M|   518G|       |   260K (72)| 00:00:11 |
|   2 |   TABLE ACCESS FULL          | TEACHERS              |  1097 | 88857 |       |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN                  |                       |  1668M|   393G|       |   249K (71)| 00:00:10 |
|   4 |    MERGE JOIN CARTESIAN      |                       |  6582 |   617K|       |    26   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL        | GRADES_ENUM           |     6 |    90 |       |     3   (0)| 00:00:01 |
|   6 |     BUFFER SORT              |                       |  1097 | 88857 |       |    23   (0)| 00:00:01 |
|   7 |      TABLE ACCESS FULL       | TEACHERS              |  1097 | 88857 |       |     4   (0)| 00:00:01 |
|   8 |    VIEW                      |                       |  1668M|   243G|       |   238K (69)| 00:00:10 |
|   9 |     SORT GROUP BY            |                       |  1668M|   441G|       |   238K (69)| 00:00:10 |
|* 10 |      HASH JOIN               |                       |  1668M|   441G|   146M| 85682  (14)| 00:00:04 |
|  11 |       VIEW                   |                       |  3005K|   111M|       |   388  (16)| 00:00:01 |
|  12 |        HASH GROUP BY         |                       |  3005K|   111M|       |   388  (16)| 00:00:01 |
|  13 |         TABLE ACCESS INMEMORY FULL | POINTS          |  3005K|   111M|       |   387  (16)| 00:00:01 |
|* 14 |       HASH JOIN              |                       |  5260K|  1229M|       |  2967   (7)| 00:00:01 |
|  15 |        TABLE ACCESS FULL     | GRADES_ENUM           |     6 |    90 |       |     3   (0)| 00:00:01 |
|* 16 |        HASH JOIN             |                       |  5260K|  1153M|  4744K|  2929   (6)| 00:00:01 |
|  17 |         JOIN FILTER CREATE   | :BF0000               | 21400 |  4493K|       |   432  (23)| 00:00:01 |
|* 18 |          HASH JOIN           |                       | 21400 |  4493K|       |   432  (23)| 00:00:01 |
|  19 |           JOIN FILTER CREATE | :BF0001               |    73 | 13140 |       |     8  (13)| 00:00:01 |
|* 20 |            HASH JOIN         |                       |    73 | 13140 |       |     8  (13)| 00:00:01 |
|  21 |             JOIN FILTER CREATE | :BF0002             |     3 |    69 |       |     3   (0)| 00:00:01 |
|* 22 |              TABLE ACCESS FULL | HOUSES              |     3 |    69 |       |     3   (0)| 00:00:01 |
|* 23 |             HASH JOIN        |                       |    97 | 15229 |       |     5  (20)| 00:00:01 |
|  24 |              JOIN FILTER CREATE | :BF0003             |    97 |  4559 |       |     3   (0)| 00:00:01 |
|  25 |               TABLE ACCESS FULL | QUIDDITCH_TEAM_MEMBERS | 97 |  4559 |       |     3   (0)| 00:00:01 |
|  26 |              JOIN FILTER USE | :BF0002               |  6739 |   723K|       |     1   (0)| 00:00:01 |
|  27 |               JOIN FILTER USE | :BF0003              |  6739 |   723K|       |     1   (0)| 00:00:01 |
|* 28 |                TABLE ACCESS INMEMORY FULL | STUDENTS | 6739 |   723K|       |     1   (0)| 00:00:01 |
|  29 |           JOIN FILTER USE    | :BF0001               |  2704K|    90M|       |   406  (20)| 00:00:01 |
|* 30 |            TABLE ACCESS INMEMORY FULL | POINTS       |  2704K|    90M|       |   406  (20)| 00:00:01 |
|  31 |         JOIN FILTER USE      | :BF0000               |  1656K|    23M|       |   105  (12)| 00:00:01 |
|* 32 |          TABLE ACCESS INMEMORY FULL | GRADES         |  1656K|    23M|       |   105  (12)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------
```

12.3.5 Best students without columnar store

```
-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                            |    10 |  1330 |       | 93493  (69)| 00:00:04 |
|   1 |  SORT ORDER BY                           |                            |    10 |  1330 |       | 93493  (69)| 00:00:04 |
|*  2 |   VIEW                                   |                            |    10 |  1330 |       | 93492  (69)| 00:00:04 |
|   3 |    TEMP TABLE TRANSFORMATION             |                            |       |       |       |            |          |
|   4 |     LOAD AS SELECT (CURSOR DURATION MEMORY)| SYS_TEMP_0FD9D664F_95BC7EA |       |       |       |            |          |
|   5 |      HASH GROUP BY                       |                            |  3174 |   133K|       |    24   (9)| 00:00:01 |
|   6 |       KEY VECTOR CREATE BUFFERED         | :KV0000                    |  3174 |   133K|       |    22   (0)| 00:00:01 |
|*  7 |        HASH JOIN                         |                            |  9346 |   355K|       |    22   (0)| 00:00:01 |
|   8 |         TABLE ACCESS FULL                | HOUSES                     |     4 |    56 |       |     3   (0)| 00:00:01 |
|   9 |         TABLE ACCESS FULL                | STUDENTS                   |  9346 |   228K|       |    19   (0)| 00:00:01 |
|* 10 |     WINDOW SORT PUSHED RANK              |                            |  9346 |   857K|  1048K| 93468  (69)| 00:00:04 |
|  11 |      HASH GROUP BY                       |                            |  9346 |   857K|  1048K| 93468  (69)| 00:00:04 |
|* 12 |       HASH JOIN                          |                            |  9346 |   857K|       | 93059  (69)| 00:00:04 |
|  13 |        TABLE ACCESS FULL                 | SYS_TEMP_0FD9D664F_95BC7EA |  3174 |   130K|       |     7   (0)| 00:00:01 |
|  14 |        VIEW                              | VW_VT_9E908A56             |  9346 |   474K|       | 93052  (69)| 00:00:04 |
|  15 |         VECTOR GROUP BY                  |                            |  9346 |   282K|       | 93052  (69)| 00:00:04 |
|  16 |          HASH GROUP BY                   |                            |  9346 |   282K|       | 93052  (69)| 00:00:04 |
|  17 |           KEY VECTOR USE                 | :KV0000                    |   631M|    18G|       | 33571  (14)| 00:00:02 |
|* 18 |            HASH JOIN                     |                            |   631M|    15G|    43M| 33466  (14)| 00:00:02 |
|  19 |             VIEW                         |                            |  1977K|    20M|       | 14229   (2)| 00:00:01 |
|  20 |              HASH GROUP BY               |                            |  1977K|    35M|    60M| 14229   (2)| 00:00:01 |
|* 21 |               HASH JOIN                  |                            |  1977K|    35M|       |  2521   (3)| 00:00:01 |
|  22 |                TABLE ACCESS FULL         | GRADES_ENUM                |     6 |    30 |       |     3   (0)| 00:00:01 |
|* 23 |                TABLE ACCESS FULL         | GRADES                     |  1977K|    26M|       |  2505   (2)| 00:00:01 |
|* 24 |              TABLE ACCESS FULL           | POINTS                     |  2986K|    45M|       |  8895   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------------
```

12.3.6 Best students with columnar store

```
-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                            |    10 |  1330 |       | 38839  (33)| 00:00:02 |
|   1 |  SORT ORDER BY                           |                            |    10 |  1330 |       | 38839  (33)| 00:00:02 |
|*  2 |   VIEW                                   |                            |    10 |  1330 |       | 38838  (33)| 00:00:02 |
|   3 |    TEMP TABLE TRANSFORMATION             |                            |       |       |       |            |          |
|   4 |     LOAD AS SELECT (CURSOR DURATION MEMORY)| SYS_TEMP_0FD9D6640_95BC7EA |       |       |       |            |          |
|   5 |      HASH GROUP BY                       |                            |  5310 |   694K|       |     6  (34)| 00:00:01 |
|   6 |       KEY VECTOR CREATE BUFFERED         | :KV0000                    |  5310 |   694K|       |     4   (0)| 00:00:01 |
|*  7 |        HASH JOIN                         |                            |  5310 |   674K|       |     4   (0)| 00:00:01 |
|   8 |         TABLE ACCESS FULL                | HOUSES                     |   409 |  9407 |       |     3   (0)| 00:00:01 |
|   9 |         TABLE ACCESS INMEMORY FULL       | STUDENTS                   |  5310 |   554K|       |     1   (0)| 00:00:01 |
|* 10 |     WINDOW SORT PUSHED RANK              |                            |   352K|    55M|    62M| 38833  (33)| 00:00:02 |
|  11 |      HASH GROUP BY                       |                            |   352K|    55M|    62M| 38833  (33)| 00:00:02 |
|* 12 |       HASH JOIN                          |                            |   352K|    55M|       | 13191  (97)| 00:00:01 |
|  13 |        VIEW                              | VW_VT_9E908A56             |  6644 |   395K|       | 13160  (97)| 00:00:01 |
|  14 |         VECTOR GROUP BY                  |                            |  6644 |   382K|       | 13160  (97)| 00:00:01 |
|  15 |          HASH GROUP BY                   |                            |  6644 |   382K|       | 13160  (97)| 00:00:01 |
|  16 |           KEY VECTOR USE                 | :KV0000                    |  6644 |   382K|       | 13159  (97)| 00:00:01 |
|  17 |            MERGE JOIN                    |                            |  6644 |   356K|       |   634  (33)| 00:00:01 |
|  18 |             SORT JOIN                    |                            |  1644K|    31M|       |   233  (59)| 00:00:01 |
|  19 |              VIEW                        |                            |  1644K|    31M|       |   233  (59)| 00:00:01 |
|  20 |               HASH GROUP BY              |                            |  1644K|    61M|       |   233  (59)| 00:00:01 |
|* 21 |                HASH JOIN                 |                            |  1644K|    61M|       |   130  (27)| 00:00:01 |
|  22 |                 TABLE ACCESS FULL        | GRADES_ENUM                |     6 |    90 |       |     3   (0)| 00:00:01 |
|* 23 |                 TABLE ACCESS INMEMORY FULL | GRADES                   |  1644K|    37M|       |   116  (20)| 00:00:01 |
|* 24 |               SORT JOIN                  |                            |  6644 |   227K|       |   401  (19)| 00:00:01 |
|* 25 |                TABLE ACCESS INMEMORY FULL | POINTS                    |  6644 |   227K|       |   399  (18)| 00:00:01 |
|  26 |         TABLE ACCESS FULL                | SYS_TEMP_0FD9D6640_95BC7EA |  5310 |   539K|       |    28   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------------
```

12.3.7 Raise grades without columnar store

```
-------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | MERGE STATEMENT               |              |  9346 | 37384 |       | 21196   (2)| 00:00:01 |
|   1 |  MERGE                        | GRADES       |       |       |       |            |          |
|   2 |   VIEW                        |              |       |       |       |            |          |
|*  3 |    HASH JOIN                  |              |  9346 |   520K|       | 21196   (2)| 00:00:01 |
|*  4 |     VIEW                      |              |  9346 |   255K|       | 18686   (2)| 00:00:01 |
|*  5 |      WINDOW SORT PUSHED RANK  |              |  1115K|    22M|    34M| 18686   (2)| 00:00:01 |
|*  6 |       HASH JOIN               |              |  1115K|    22M|       | 11451   (2)| 00:00:01 |
|*  7 |        TABLE ACCESS FULL      | GRADES_ENUM  |     6 |    30 |       |     3   (0)| 00:00:01 |
|*  8 |        HASH JOIN              |              |  1115K|    17M|       | 11441   (2)| 00:00:01 |
|   9 |         VIEW                  |              |  7467 | 29868 |       |  8926   (2)| 00:00:01 |
|* 10 |          FILTER               |              |       |       |       |            |          |
|  11 |           SORT GROUP BY       |              |  7467 |   320K|       |  8926   (2)| 00:00:01 |
|* 12 |            HASH JOIN OUTER    |              |   149K|  6416K|       |  8917   (1)| 00:00:01 |
|* 13 |             HASH JOIN         |              |   467 | 16812 |       |    22   (0)| 00:00:01 |
|* 14 |              TABLE ACCESS FULL| HOUSES       |     1 |    14 |       |     3   (0)| 00:00:01 |
|  15 |              TABLE ACCESS FULL| STUDENTS     |  9346 |   200K|       |    19   (0)| 00:00:01 |
|  16 |             TABLE ACCESS FULL | POINTS       |  2986K|    22M|       |  8875   (1)| 00:00:01 |
|* 17 |           TABLE ACCESS FULL   | GRADES       |  1396K|    15M|       |  2506   (2)| 00:00:01 |
|  18 |      TABLE ACCESS FULL        | GRADES       |  1977K|    54M|       |  2497   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

12.3.8 Raise grades with columnar store

```
-------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | MERGE STATEMENT               |              |   110 |   440 |       |   833K  (1)| 00:00:33 |
|   1 |  MERGE                        | GRADES       |       |       |       |            |          |
|   2 |   VIEW                        |              |       |       |       |            |          |
|   3 |    NESTED LOOPS               |              |   110 | 11330 |       |   833K  (1)| 00:00:33 |
|   4 |     NESTED LOOPS              |              |   110 | 11330 |       |   833K  (1)| 00:00:33 |
|*  5 |      VIEW                     |              |   110 |  3080 |       |   833K  (1)| 00:00:33 |
|*  6 |       WINDOW SORT PUSHED RANK |              |    60M|  3237M|  3728M|   833K  (1)| 00:00:33 |
|*  7 |        HASH JOIN              |              |    60M|  3237M|    17M|  3868  (15)| 00:00:01 |
|   8 |         VIEW                  |              |   751K|  9541K|       |   438  (24)| 00:00:01 |
|*  9 |          FILTER               |              |       |       |       |            |          |
|  10 |           SORT GROUP BY       |              |   751K|   102M|       |   438  (24)| 00:00:01 |
|* 11 |            HASH JOIN OUTER    |              |   751K|   102M|       |   393  (16)| 00:00:01 |
|  12 |             JOIN FILTER CREATE| :BF0000      |  2370 |   270K|       |     4   (0)| 00:00:01 |
|* 13 |              HASH JOIN        |              |  2370 |   270K|       |     4   (0)| 00:00:01 |
|  14 |               JOIN FILTER CREATE| :BF0001    |     1 |    23 |       |     3   (0)| 00:00:01 |
|* 15 |                TABLE ACCESS FULL| HOUSES     |     1 |    23 |       |     3   (0)| 00:00:01 |
|  16 |               JOIN FILTER USE | :BF0001      |  9480 |   870K|       |     1   (0)| 00:00:01 |
|* 17 |                TABLE ACCESS INMEMORY FULL| STUDENTS | 9480 | 870K|       |     1   (0)| 00:00:01 |
|  18 |             JOIN FILTER USE   | :BF0000      |  3005K|    74M|       |   369  (12)| 00:00:01 |
|* 19 |              TABLE ACCESS INMEMORY FULL | POINTS | 3005K|    74M|       |   369  (12)| 00:00:01 |
|* 20 |         HASH JOIN             |              |   764K|    31M|       |   136  (30)| 00:00:01 |
|  21 |          JOIN FILTER CREATE   | :BF0002      |     4 |    60 |       |     3   (0)| 00:00:01 |
|* 22 |           TABLE ACCESS FULL   | GRADES_ENUM  |     4 |    60 |       |     3   (0)| 00:00:01 |
|  23 |          JOIN FILTER USE      | :BF0002      |  1146K|    30M|       |   126  (27)| 00:00:01 |
|* 24 |           TABLE ACCESS INMEMORY FULL | GRADES |  1146K|    30M|       |   126  (27)| 00:00:01 |
|* 25 |      INDEX UNIQUE SCAN        | SYS_C009281  |     1 |       |       |     0   (0)| 00:00:01 |
|  26 |     TABLE ACCESS BY INDEX ROWID | GRADES     |     1 |    75 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------
```

12.3.9 Assign subjects without columnar store

```
-------------------------------------------------------------------------------------------------------
| Id  | Operation                   | Name                  | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT            |                       |  1017 | 52920 |       | 12318   (3)| 00:00:01 |
|   1 |  LOAD TABLE CONVENTIONAL    | STUDENTS_SUBJECTS     |       |       |       |            |          |
|   2 |   HASH UNIQUE               |                       |  1017 | 52920 |       | 12318   (3)| 00:00:01 |
|   3 |    UNION-ALL                |                       |       |       |       |            |          |
|*  4 |     HASH JOIN               |                       |  1016 | 26416 |       |  6155   (3)| 00:00:01 |
|*  5 |      HASH JOIN ANTI         |                       |    93 |  1860 |       |  6152   (3)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL     | STUDENTS              |  9346 | 65422 |       |    19   (0)| 00:00:01 |
|   7 |       VIEW                  | VW_NSO_1              | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|*  8 |        FILTER               |                       |       |       |       |            |          |
|   9 |         SORT GROUP BY       |                       | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 10 |          HASH JOIN          |                       |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  11 |           TABLE ACCESS FULL | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|  12 |           TABLE ACCESS FULL | GRADES                |  1977K|    16M|       |  2495   (2)| 00:00:01 |
|* 13 |      TABLE ACCESS FULL      | SUBJECTS              |    76 |   456 |       |     3   (0)| 00:00:01 |
|* 14 |     HASH JOIN ANTI          |                       |     1 |    44 |       |  6158   (3)| 00:00:01 |
|* 15 |      HASH JOIN              |                       |    97 |  3007 |       |    25   (0)| 00:00:01 |
|  16 |       MERGE JOIN CARTESIAN  |                       |    97 |  2328 |       |     6   (0)| 00:00:01 |
|* 17 |        TABLE ACCESS FULL    | SUBJECTS              |     1 |    20 |       |     3   (0)| 00:00:01 |
|  18 |        BUFFER SORT          |                       |    97 |   388 |       |     3   (0)| 00:00:01 |
|  19 |         TABLE ACCESS FULL   | QUIDDITCH_TEAM_MEMBERS|    97 |   388 |       |     3   (0)| 00:00:01 |
|* 20 |       TABLE ACCESS FULL     | STUDENTS              |  8011 | 56077 |       |    19   (0)| 00:00:01 |
|* 21 |        TABLE ACCESS FULL    | SUBJECTS              |     1 |    20 |       |     3   (0)| 00:00:01 |
|  22 |      VIEW                   | VW_NSO_2              | 25113 |   318K|       |  6133   (3)| 00:00:01 |
|* 23 |       FILTER                |                       |       |       |       |            |          |
|  24 |        SORT GROUP BY        |                       | 25113 |   343K|    45M|  6133   (3)| 00:00:01 |
|* 25 |         HASH JOIN           |                       |  1977K|    26M|       |  2511   (2)| 00:00:01 |
|  26 |          TABLE ACCESS FULL  | GRADES_ENUM           |     6 |    30 |       |     3   (0)| 00:00:01 |
|  27 |          TABLE ACCESS FULL  | GRADES                |  1977K|    16M|       |  2495   (2)| 00:00:01 |
-------------------------------------------------------------------------------------------------------
```

12.3.10 Assign subjects with columnar store

```
-------------------------------------------------------------------------------------------------------
| Id  | Operation                   | Name                  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------
|   0 | INSERT STATEMENT            |                       |   102K|    12M|  2095  (14)| 00:00:01 |
|   1 |  LOAD TABLE CONVENTIONAL    | STUDENTS_SUBJECTS     |       |       |            |          |
|   2 |   HASH UNIQUE               |                       |   102K|    12M|  2095  (14)| 00:00:01 |
|   3 |    UNION-ALL                |                       |       |       |            |          |
|*  4 |     HASH JOIN               |                       |   102K|  6533K|   243  (59)| 00:00:01 |
|*  5 |      TABLE ACCESS INMEMORY FULL | SUBJECTS          |    76 |  1976 |     3   (0)| 00:00:01 |
|*  6 |      HASH JOIN ANTI         |                       |  9480 |   361K|   239  (60)| 00:00:01 |
|*  7 |       TABLE ACCESS INMEMORY FULL | STUDENTS         |  9480 |   240K|     1   (0)| 00:00:01 |
|   8 |       VIEW                  | VW_NSO_1              |  1656K|    20M|   227  (58)| 00:00:01 |
|*  9 |        FILTER               |                       |       |       |            |          |
|  10 |         SORT GROUP BY       |                       |  1656K|    67M|   227  (58)| 00:00:01 |
|* 11 |          HASH JOIN          |                       |  1656K|    67M|   124  (23)| 00:00:01 |
|  12 |           TABLE ACCESS FULL | GRADES_ENUM           |     6 |    90 |     3   (0)| 00:00:01 |
|  13 |           TABLE ACCESS INMEMORY FULL | GRADES       |  1656K|    44M|   110  (16)| 00:00:01 |
|* 14 |      HASH JOIN ANTI         |                       |     1 |    99 |   245  (58)| 00:00:01 |
|* 15 |       HASH JOIN RIGHT SEMI  |                       |    97 |  8342 |     7   (0)| 00:00:01 |
|  16 |        TABLE ACCESS FULL    | QUIDDITCH_TEAM_MEMBERS|    97 |  1261 |     3   (0)| 00:00:01 |
|  17 |        MERGE JOIN CARTESIAN |                       |   474 | 34602 |     4   (0)| 00:00:01 |
|* 18 |         TABLE ACCESS INMEMORY FULL | SUBJECTS       |     1 |    47 |     3   (0)| 00:00:01 |
|  19 |         BUFFER SORT         |                       |   474 | 12324 |     1   (0)| 00:00:01 |
|* 20 |          TABLE ACCESS INMEMORY FULL | STUDENTS       |   474 | 12324 |     1   (0)| 00:00:01 |
|* 21 |           TABLE ACCESS INMEMORY FULL | SUBJECTS      |     1 |    47 |     3   (0)| 00:00:01 |
|  22 |       VIEW                  | VW_NSO_2              |  1656K|    20M|   227  (58)| 00:00:01 |
|* 23 |        FILTER               |                       |       |       |            |          |
|  24 |         SORT GROUP BY       |                       |  1656K|    67M|   227  (58)| 00:00:01 |
|* 25 |          HASH JOIN          |                       |  1656K|    67M|   124  (23)| 00:00:01 |
|  26 |           TABLE ACCESS FULL | GRADES_ENUM           |     6 |    90 |     3   (0)| 00:00:01 |
|  27 |           TABLE ACCESS INMEMORY FULL | GRADES       |  1656K|    44M|   110  (16)| 00:00:01 |
-------------------------------------------------------------------------------------------------------
```

12.3.11 Remove points without columnar store

```
-------------------------------------------------------------------------------------
| Id  | Operation                | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT         |              |  2078 | 64418 | 11495   (2)| 00:00:01 |
|   1 |  DELETE                  | POINTS       |       |       |            |          |
|*  2 |   HASH JOIN RIGHT SEMI   |              |  2078 | 64418 | 11495   (2)| 00:00:01 |
|   3 |    VIEW                  | VW_NSO_1     |    13 |   169 |  2604   (6)| 00:00:01 |
|*  4 |     SORT GROUP BY        |              |    13 |   221 |  2604   (6)| 00:00:01 |
|   5 |      VIEW                |              |   248 |  4216 |  2604   (6)| 00:00:01 |
|*  6 |       FILTER             |              |       |       |            |          |
|   7 |        HASH GROUP BY     |              |   248 |  5456 |  2604   (6)| 00:00:01 |
|*  8 |         HASH JOIN        |              |  4943 |  106K |  2603   (6)| 00:00:01 |
|   9 |          TABLE ACCESS FULL| GRADES_ENUM  |     6 |    30 |     3   (0)| 00:00:01 |
|* 10 |          TABLE ACCESS FULL| GRADES       |  4943 | 84031 |  2600   (6)| 00:00:01 |
|* 11 |    TABLE ACCESS FULL     | POINTS       |  1494K|   25M|  8881   (1)| 00:00:01 |
-------------------------------------------------------------------------------------
```

12.3.12 Remove points with columnar store

```
-----------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT                   |              |  1487K|    92M|       |  6819   (5)| 00:00:01 |
|   1 |  DELETE                            | POINTS       |       |       |       |            |          |
|*  2 |   HASH JOIN RIGHT SEMI             |              |  1487K|    92M|    31M|  6819   (5)| 00:00:01 |
|   3 |    VIEW                            | VW_NSO_1     |  1323K|    16M|       |   300  (68)| 00:00:01 |
|*  4 |     SORT GROUP BY                  |              |  1323K|    32M|       |   300  (68)| 00:00:01 |
|   5 |      VIEW                          |              |  1323K|    32M|       |   300  (68)| 00:00:01 |
|*  6 |       FILTER                       |              |       |       |       |            |          |
|   7 |        HASH GROUP BY               |              |  1323K|    65M|       |   300  (68)| 00:00:01 |
|*  8 |         HASH JOIN                  |              |  1323K|    65M|       |   219  (57)| 00:00:01 |
|   9 |          TABLE ACCESS FULL         | GRADES_ENUM  |     6 |    90 |       |     3   (0)| 00:00:01 |
|* 10 |          TABLE ACCESS INMEMORY FULL| GRADES       |  1323K|    46M|       |   207  (56)| 00:00:01 |
|* 11 |    TABLE ACCESS INMEMORY FULL      | POINTS       |  1487K|    73M|       |   416  (22)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------
```

12.4 Experiment 1: In-Memory Buffer Size Impact Analysis

We conducted the first experiment to evaluate the impact of different In-Memory buffer sizes on columnar storage performance. The experiment compared two configurations: **256MB In-Memory buffer**, and **1GB In-Memory buffer**, measuring performance **10 times** for each configuration to ensure statistical reliability.

12.4.1 Experiment 1 - System Configuration Parameters

Memory Component	256MB In-Memory (bytes)	1GB In-Memory (bytes)
Total System Global Area	2,147,482,608	2,147,480,608
Fixed Size	9,858,032	9,876,512
Variable Size	1,358,954,496	721,420,288
Database Buffers	503,316,480	339,738,624
Redo Buffers	6,918,144	2,703,360
In-Memory Area	268,435,456	1,073,741,824
In-Memory Size (MB)	**256**	**1024**

12.4.2 Experiment 1 - Performance Results

As it turns out - a bigger in-memory buffer doesn't always provide improved performance. In fact, usually both running time and query plan cost are better for the smaller buffer, at least on our computers. Below we show the results of our experiment (we excluded two queries for which the difference was too small).

Procedure name	Min execution time (ms)				Max execution time (ms)				Avg execution time (ms)				Query plan cost			
	256 MB	1 GB	Gain	Profit	256 MB	1 GB	Gain	Profit	256 MB	1 GB	Gain	Profit	256 MB	1 GB	Gain	Profit
Grades analysis	723	664	-59	-8.2 %	868	1598	730	84.1 %	778.6	1194.8	416.2	53.5 %	2446 K	3347 K	901 K	36.8 %
Best students	438	487	49	11.2 %	548	836	288	52.6 %	497.8	672.9	175.1	35.2 %	44600	42704	-1896	-4.3 %
Raise grades	2206	1469	-737	-33.4 %	2442	5376	2934	120.1 %	2306.6	4162.9	1856.3	80.5 %	498 K	746 K	248 K	49.8 %
Remove points	3587	5414	1827	50.9 %	5910	6985	1075	18.2 %	4046.1	6399.2	2353.1	58.2 %	6479	559	-5920	-91.4 %
Whole workload	6954	8034	1080	15.5 %	9768	14795	5027	51.5 %	7629.1	12429.8	4800.7	62.9 %	2995 K	4136 K	1141 K	38.1 %

12.4.3 Experiment 1 - Performance Analysis

The results demonstrate that Oracle's In-Memory optimization effectiveness varies significantly with buffer size allocation:

- **256 MB Configuration**: Overall better than 1 GB buffer, both in running time and in query plan cost. Better choice for our scenario.
- **1 GB Configuration**: Significantly cheaper query plan only for query that involves deleting many rows. For the whole workload however on average it runs 62.9% slower and has 38.1% bigger cost.

12.5 Query plans comparison

Implementing columnar storage dramatically changed query execution plans. The Oracle optimizer leveraged in-memory columnar operations, vector processing, and specialized algorithms for analytical workloads.

12.5.1 Grades analysis - 256MB In-Memory

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                     | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                          |   106 | 37948 |       |  2446K (66)| 00:01:36 |
|   1 |  SORT ORDER BY                           |                          |   106 | 37948 |       |  2446K (66)| 00:01:36 |
|*  2 |   HASH JOIN                              |                          |   106 | 37948 |       |  2446K (66)| 00:01:36 |
|   3 |    TABLE ACCESS INMEMORY FULL            | SUBJECTS                 |    76 |  6004 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                          |   106 | 29574 |       |  2446K (66)| 00:01:36 |
|   5 |     TABLE ACCESS INMEMORY FULL           | SUBJECTS                 |    76 |  6004 |       |     3   (0)| 00:00:01 |
|*  6 |     HASH JOIN                            |                          |   106 | 21200 |       |  2446K (66)| 00:01:36 |
|   7 |      TABLE ACCESS FULL                   | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|*  8 |      HASH JOIN                           |                          |   106 | 19610 |       |  2446K (66)| 00:01:36 |
|   9 |       TABLE ACCESS FULL                  | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  10 |       VIEW                               |                          |   106 | 18020 |       |  2446K (66)| 00:01:36 |
|  11 |        SORT GROUP BY                     |                          |   106 | 33602 |       |  2446K (66)| 00:01:36 |
|* 12 |         HASH JOIN RIGHT OUTER            |                          |    13G|  4118G|       |  1039K (19)| 00:00:41 |
|  13 |          TABLE ACCESS FULL               | QUIDDITCH_TEAM_MEMBERS   |     1 |    47 |       |     3   (0)| 00:00:01 |
|* 14 |          HASH JOIN                       |                          |    13G|  3508G|   101M|   946K (10)| 00:00:37 |
|  15 |           VIEW                           |                          |  1666K|    82M|       |   231   (58)| 00:00:01 |
|  16 |            HASH GROUP BY                 |                          |  1666K|   151M|       |   231   (58)| 00:00:01 |
|* 17 |             HASH JOIN                    |                          |  1666K|   151M|       |   127   (23)| 00:00:01 |
|  18 |              TABLE ACCESS FULL           | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  19 |              VIEW                        | VW_GBF_7                 |  1666K|   127M|       |   113   (16)| 00:00:01 |
|  20 |               HASH GROUP BY              |                          |  1666K|    44M|       |   113   (16)| 00:00:01 |
|  21 |                TABLE ACCESS INMEMORY FULL| GRADES                   |  1666K|    44M|       |   112   (16)| 00:00:01 |
|* 22 |           HASH JOIN                      |                          |    76M|   15G|    39M| 13423    (5)| 00:00:01 |
|  23 |            TABLE ACCESS INMEMORY FULL    | GRADES                   |  1666K|    20M|       |   102    (7)| 00:00:01 |
|* 24 |            HASH JOIN                     |                          |   422K|    82M|       |  6489    (2)| 00:00:01 |
|  25 |             JOIN FILTER CREATE           | :BF0000                  |  2292 |   380K|       |   138   (27)| 00:00:01 |
|* 26 |              HASH JOIN ANTI              |                          |  2292 |   380K|       |   138   (27)| 00:00:01 |
|  27 |               JOIN FILTER CREATE         | :BF0001                  |  2292 |   351K|       |     5   (20)| 00:00:01 |
|* 28 |                HASH JOIN                 |                          |  2292 |   351K|       |     5   (20)| 00:00:01 |
|  29 |                 JOIN FILTER CREATE       | :BF0002                  |     1 |    47 |       |     3   (0)| 00:00:01 |
|* 30 |                  TABLE ACCESS FULL       | HOUSES                   |     1 |    47 |       |     3   (0)| 00:00:01 |
|  31 |                 JOIN FILTER USE          | :BF0002                  |  9167 |   984K|       |     1   (0)| 00:00:01 |
|* 32 |                  TABLE ACCESS INMEMORY FULL | STUDENTS                 |  9167 |   984K|       |     1   (0)| 00:00:01 |
|  33 |               VIEW                       | VW_NSO_1                 |   159K|  2028K|       |   133   (27)| 00:00:01 |
|* 34 |                HASH JOIN                 |                          |   159K|  6085K|       |   133   (27)| 00:00:01 |
|  35 |                 JOIN FILTER CREATE       | :BF0003                  |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 36 |                  TABLE ACCESS FULL       | GRADES_ENUM              |     1 |    15 |       |     3   (0)| 00:00:01 |
|  37 |                   TABLE ACCESS BY INDEX ROWID| GRADES_ENUM              |     1 |    15 |       |     2   (0)| 00:00:01 |
|* 38 |                    INDEX UNIQUE SCAN     | SYS_C009232              |     1 |       |       |     1   (0)| 00:00:01 |
|  39 |                  TABLE ACCESS BY INDEX ROWID| GRADES_ENUM              |     1 |    15 |       |     2   (0)| 00:00:01 |
|* 40 |                   INDEX UNIQUE SCAN      | SYS_C009232              |     1 |       |       |     1   (0)| 00:00:01 |
|  41 |                 JOIN FILTER USE          | :BF0001                  |  1638K|    37M|       |   119   (21)| 00:00:01 |
|  42 |                  JOIN FILTER USE         | :BF0003                  |  1638K|    37M|       |   119   (21)| 00:00:01 |
|* 43 |                   TABLE ACCESS INMEMORY FULL | GRADES                   |  1638K|    37M|       |   119   (21)| 00:00:01 |
|  44 |            JOIN FILTER USE               | :BF0000                  |  1755K|    58M|       |  6339    (1)| 00:00:01 |
|* 45 |             TABLE ACCESS INMEMORY FULL   | POINTS                   |  1755K|    58M|       |  6339    (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

12.5.2 Grades analysis - 1GB In-Memory

```
-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                     | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                          |   106 | 37948 |       |  3347K(100)| 00:02:11 |
|   1 |  SORT ORDER BY                           |                          |   106 | 37948 |       |  3347K(100)| 00:02:11 |
|*  2 |   HASH JOIN                              |                          |   106 | 37948 |       |  3347K(100)| 00:02:11 |
|   3 |    TABLE ACCESS INMEMORY FULL            | SUBJECTS                 |    76 |  6004 |       |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN                             |                          |   106 | 29574 |       |  3347K(100)| 00:02:11 |
|   5 |     TABLE ACCESS INMEMORY FULL           | SUBJECTS                 |    76 |  6004 |       |     3   (0)| 00:00:01 |
|*  6 |     HASH JOIN                            |                          |   106 | 21200 |       |  3347K(100)| 00:02:11 |
|   7 |      TABLE ACCESS FULL                   | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|*  8 |      HASH JOIN                           |                          |   106 | 19610 |       |  3347K(100)| 00:02:11 |
|   9 |       TABLE ACCESS FULL                  | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  10 |       VIEW                               |                          |   106 | 18020 |       |  3347K(100)| 00:02:11 |
|  11 |        SORT GROUP BY                     |                          |   106 | 33602 |       |  3347K(100)| 00:02:11 |
|* 12 |         HASH JOIN                        |                          |    21G|  6277G|       |  1163K (98)| 00:00:46 |
|* 13 |          TABLE ACCESS FULL               | HOUSES                   |     1 |    47 |       |     3   (0)| 00:00:01 |
|  14 |          MERGE JOIN                      |                          |    85G|   20T|       |   598K (95)| 00:00:24 |
|  15 |           MERGE JOIN                     |                          |   467M|   111G|       | 28849  (13)| 00:00:02 |
|  16 |            MERGE JOIN ANTI               |                          |  1666K|   352M|       |  1367  (14)| 00:00:01 |
|  17 |             MERGE JOIN OUTER             |                          |  1666K|   332M|       |   468  (30)| 00:00:01 |
|  18 |              MERGE JOIN                  |                          |  1666K|   257M|       |   464  (30)| 00:00:01 |
|  19 |               SORT JOIN                  |                          |  1666K|    82M|       |   231   (58)| 00:00:01 |
|  20 |                VIEW                      |                          |  1666K|    82M|       |   231   (58)| 00:00:01 |
|  21 |                 HASH GROUP BY            |                          |  1666K|   151M|       |   231   (58)| 00:00:01 |
|* 22 |                  HASH JOIN               |                          |  1666K|   151M|       |   127   (23)| 00:00:01 |
|  23 |                   TABLE ACCESS FULL      | GRADES_ENUM              |     6 |    90 |       |     3   (0)| 00:00:01 |
|  24 |                   VIEW                   | VW_GBF_7                 |  1666K|   127M|       |   113   (16)| 00:00:01 |
|  25 |                    HASH GROUP BY         |                          |  1666K|    44M|       |   113   (16)| 00:00:01 |
|  26 |                     TABLE ACCESS INMEMORY FULL| GRADES                   |  1666K|    44M|       |   112   (16)| 00:00:01 |
|* 27 |               SORT JOIN                  |                          |  9167 |   984K|  2344K|   233    (1)| 00:00:01 |
|* 28 |                TABLE ACCESS INMEMORY FULL| STUDENTS                 |  9167 |   984K|       |     1   (0)| 00:00:01 |
|* 29 |              SORT JOIN                   |                          |    97 |  4559 |       |     4  (25)| 00:00:01 |
|  30 |               TABLE ACCESS FULL          | QUIDDITCH_TEAM_MEMBERS   |    97 |  4559 |       |     3   (0)| 00:00:01 |
|* 31 |             SORT UNIQUE                  |                          |   159K|  2028K|  6280K|   899   (6)| 00:00:01 |
|  32 |              VIEW                        | VW_NSO_1                 |   159K|  2028K|       |   133   (27)| 00:00:01 |
|* 33 |               HASH JOIN                  |                          |   159K|  6085K|       |   133   (27)| 00:00:01 |
|  34 |                JOIN FILTER CREATE        | :BF0000                  |     1 |    15 |       |     3   (0)| 00:00:01 |
|* 35 |                 TABLE ACCESS FULL        | GRADES_ENUM              |     1 |    15 |       |     3   (0)| 00:00:01 |
|  36 |                  TABLE ACCESS BY INDEX ROWID| GRADES_ENUM              |     1 |    15 |       |     2   (0)| 00:00:01 |
|* 37 |                   INDEX UNIQUE SCAN      | SYS_C009232              |     1 |       |       |     1   (0)| 00:00:01 |
|  38 |                  TABLE ACCESS BY INDEX ROWID| GRADES_ENUM              |     1 |    15 |       |     2   (0)| 00:00:01 |
|* 39 |                   INDEX UNIQUE SCAN      | SYS_C009232              |     1 |       |       |     1   (0)| 00:00:01 |
|  40 |                JOIN FILTER USE           | :BF0000                  |  1638K|    37M|       |   119   (21)| 00:00:01 |
|* 41 |                 TABLE ACCESS INMEMORY FULL | GRADES                   |  1638K|    37M|       |   119   (21)| 00:00:01 |
|* 42 |              SORT JOIN                   |                          |  2572K|    85M|   236M| 24391    (2)| 00:00:01 |
|* 43 |               TABLE ACCESS INMEMORY FULL | POINTS                   |  2572K|    85M|       |   404  (18)| 00:00:01 |
|* 44 |             SORT JOIN                    |                          |  1666K|    20M|    63M|  8082    (2)| 00:00:01 |
|  45 |              TABLE ACCESS INMEMORY FULL  | GRADES                   |  1666K|    20M|       |   102   (7)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------
```

12.5.3 Best students display - 256MB In-Memory

```
===============Best students display================
Plan hash value: 3211729349
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                            |    10 |  1570 |       | 44600  (29)| 00:00:02 |
|   1 |  SORT ORDER BY                           |                            |    10 |  1570 |       | 44600  (29)| 00:00:02 |
|*  2 |   VIEW                                   |                            |    10 |  1570 |       | 44599  (29)| 00:00:02 |
|   3 |    TEMP TABLE TRANSFORMATION             |                            |       |       |       |            |          |
|   4 |     LOAD AS SELECT (CURSOR DURATION MEMORY)| SYS_TEMP_0FD9D6611_1836B28 |       |       |       |            |          |
|   5 |      HASH GROUP BY                       |                            |  5310 |   819K|       |     6  (34)| 00:00:01 |
|   6 |       KEY VECTOR CREATE BUFFERED         | :KV0000                    |  5310 |   819K|       |     4   (0)| 00:00:01 |
|*  7 |        HASH JOIN                         |                            |  5310 |   798K|       |     4   (0)| 00:00:01 |
|   8 |         TABLE ACCESS FULL                | HOUSES                     |   409 | 19223 |       |     3   (0)| 00:00:01 |
|   9 |         TABLE ACCESS INMEMORY FULL       | STUDENTS                   |  5310 |   554K|       |     1   (0)| 00:00:01 |
|* 10 |     WINDOW SORT PUSHED RANK              |                            |   319K|    57M|   64M| 44594  (29)| 00:00:02 |
|  11 |      HASH GROUP BY                       |                            |   319K|    57M|   64M| 44594  (29)| 00:00:02 |
|* 12 |       HASH JOIN                          |                            |   319K|    57M|       | 18223  (69)| 00:00:01 |
|  13 |        VIEW                              | VW_VT_9E908A56             |  6019 |   358K|       | 18188  (69)| 00:00:01 |
|  14 |         VECTOR GROUP BY                  |                            |  6019 |   346K|       | 18188  (69)| 00:00:01 |
|  15 |          HASH GROUP BY                   |                            |  6019 |   346K|       | 18188  (69)| 00:00:01 |
|  16 |           KEY VECTOR USE                 | :KV0000                    |  6018 |   346K|       | 18186  (69)| 00:00:01 |
|  17 |            MERGE JOIN                    |                            |  6019 |   323K|       |  5924   (4)| 00:00:01 |
|  18 |             SORT JOIN                    |                            |  1655K|    31M|       |   236  (59)| 00:00:01 |
|  19 |              VIEW                        |                            |  1655K|    31M|       |   236  (59)| 00:00:01 |
|  20 |               HASH GROUP BY              |                            |  1655K|    61M|       |   236  (59)| 00:00:01 |
|* 21 |                HASH JOIN                 |                            |  1655K|    61M|       |   133  (27)| 00:00:01 |
|  22 |                 TABLE ACCESS FULL        | GRADES_ENUM                |     6 |    90 |       |     3   (0)| 00:00:01 |
|* 23 |                 TABLE ACCESS INMEMORY FULL | GRADES                   |  1655K|    37M|       |   119  (21)| 00:00:01 |
|* 24 |             SORT JOIN                    |                            |  6019 |   205K|       |  5688   (2)| 00:00:01 |
|* 25 |              TABLE ACCESS INMEMORY FULL  | POINTS                     |  6019 |   205K|       |  5687   (2)| 00:00:01 |
|  26 |       TABLE ACCESS FULL                  | SYS_TEMP_0FD9D6611_1836B28 |  5310 |   663K|       |    33   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("RANK"<=10)
   7 - access("S"."HOUSE_ID"="H"."ID")
  10 - filter(ROW_NUMBER() OVER ( ORDER BY ROUND(DECODE(NVL(SUM("ITEM_5"),0),0,TO_NUMBER(NULL),SUM("ITEM_4")/NVL(SUM("ITE
              M_5"),0)),2) DESC ,NVL(SUM("ITEM_3"),0) DESC )<=10)
  12 - access("ITEM_11"=INTERNAL_FUNCTION("CO"))
  21 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  23 - inmemory("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  24 - access("P"."STUDENT_ID"="MA"."STUDENT_ID")
       filter("P"."STUDENT_ID"="MA"."STUDENT_ID")
  25 - inmemory("P"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "P"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter("P"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "P"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - vector transformation used for this statement
```

12.5.4 Best students display - 1GB In-Memory

```
===============Best students display================
Plan hash value: 355475729
-----------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                       | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                            |    10 |  1570 |       | 42704  (31)| 00:00:02 |
|   1 |  SORT ORDER BY                           |                            |    10 |  1570 |       | 42704  (31)| 00:00:02 |
|*  2 |   VIEW                                   |                            |    10 |  1570 |       | 42703  (31)| 00:00:02 |
|   3 |    TEMP TABLE TRANSFORMATION             |                            |       |       |       |            |          |
|   4 |     LOAD AS SELECT (CURSOR DURATION MEMORY)| SYS_TEMP_0FD9D6610_187B43B |       |       |       |            |          |
|   5 |      HASH GROUP BY                       |                            |  5310 |   819K|       |     6  (34)| 00:00:01 |
|   6 |       KEY VECTOR CREATE BUFFERED         | :KV0000                    |  5310 |   819K|       |     4   (0)| 00:00:01 |
|*  7 |        HASH JOIN                         |                            |  5310 |   798K|       |     4   (0)| 00:00:01 |
|   8 |         TABLE ACCESS FULL                | HOUSES                     |   409 | 19223 |       |     3   (0)| 00:00:01 |
|   9 |         TABLE ACCESS INMEMORY FULL       | STUDENTS                   |  5310 |   554K|       |     1   (0)| 00:00:01 |
|* 10 |     WINDOW SORT PUSHED RANK              |                            |   356K|    64M|   71M| 42697  (31)| 00:00:02 |
|  11 |      HASH GROUP BY                       |                            |   356K|    64M|   71M| 42697  (31)| 00:00:02 |
|* 12 |       HASH JOIN                          |                            |   356K|    64M|       | 13246  (97)| 00:00:01 |
|  13 |        VIEW                              | VW_VT_9E908A56             |  6722 |   400K|       | 13211  (97)| 00:00:01 |
|  14 |         VECTOR GROUP BY                  |                            |  6722 |   387K|       | 13211  (97)| 00:00:01 |
|  15 |          HASH GROUP BY                   |                            |  6722 |   387K|       | 13211  (97)| 00:00:01 |
|  16 |           KEY VECTOR USE                 | :KV0000                    |  6721 |   387K|       | 13209  (97)| 00:00:01 |
|* 17 |            HASH JOIN                     |                            |  6722 |   361K|       |   651  (34)| 00:00:01 |
|  18 |             JOIN FILTER CREATE           | :BF0000                    |  6722 |   229K|       |   404  (18)| 00:00:01 |
|* 19 |              TABLE ACCESS INMEMORY FULL  | POINTS                     |  6722 |   229K|       |   404  (18)| 00:00:01 |
|  20 |             VIEW                         |                            |  1655K|    31M|       |   236  (59)| 00:00:01 |
|  21 |              HASH GROUP BY               |                            |  1655K|    61M|       |   236  (59)| 00:00:01 |
|* 22 |               HASH JOIN                  |                            |  1655K|    61M|       |   133  (27)| 00:00:01 |
|  23 |                TABLE ACCESS FULL         | GRADES_ENUM                |     6 |    90 |       |     3   (0)| 00:00:01 |
|  24 |                JOIN FILTER USE           | :BF0000                    |  1655K|    37M|       |   119  (21)| 00:00:01 |
|* 25 |                 TABLE ACCESS INMEMORY FULL | GRADES                   |  1655K|    37M|       |   119  (21)| 00:00:01 |
|  26 |       TABLE ACCESS FULL                  | SYS_TEMP_0FD9D6610_187B43B |  5310 |   663K|       |    33   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("RANK"<=10)
   7 - access("S"."HOUSE_ID"="H"."ID")
  10 - filter(ROW_NUMBER() OVER ( ORDER BY ROUND(DECODE(NVL(SUM("ITEM_5"),0),0,TO_NUMBER(NULL),SUM("ITEM_4")/NVL(SUM("ITE
              M_5"),0)),2) DESC ,NVL(SUM("ITEM_3"),0) DESC )<=10)
  12 - access("ITEM_11"=INTERNAL_FUNCTION("CO"))
  17 - access("P"."STUDENT_ID"="MA"."STUDENT_ID")
  19 - inmemory("P"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "P"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
       filter("P"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "P"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  22 - access("GRADES"."VALUE"="GRADES_ENUM"."SYMBOL")
  25 - inmemory("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              SYS_OP_BLOOM_FILTER(:BF0000,"GRADES"."STUDENT_ID"))
       filter("GRADES"."AWARD_DATE">=TO_DATE(' 2023-09-02 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              "GRADES"."AWARD_DATE"<=TO_DATE(' 2024-06-28 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              SYS_OP_BLOOM_FILTER(:BF0000,"GRADES"."STUDENT_ID"))

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - vector transformation used for this statement
```

12.6 Experiment 2: Compression Methods and Storage Strategy Comparison

We conducted a comprehensive second experiment to evaluate different Oracle In-Memory compression methods and storage strategies. This experiment compared five distinct configurations: **clean database** (no In-Memory), **FOR CAPACITY HIGH** (maximum compression), **FOR QUERY HIGH** (query-optimized), **FOR CAPACITY LOW** (balanced approach), and **5-columnar storage** (full In-Memory implementation), measuring performance **10 times** for each configuration.

12.6.1 Experiment 2 - In-Memory Configuration Details

In this experiment, we applied In-Memory columnar storage **selectively** to the same three critical analytical tables across all test variants, while keeping remaining tables in traditional row storage. The target tables for In-Memory optimization were:

- **GRADES** - Primary fact table containing millions of grade records with high analytical query frequency
- **POINTS** - Secondary fact table with student achievement data requiring frequent aggregations
- **STUDENTS** - Central dimensional table serving as the primary join point for analytical operations

We tested different combinations of Oracle In-Memory parameters to evaluate compression effectiveness and query performance trade-offs:

1. **FOR CAPACITY HIGH Configuration**:

```
CREATE TABLE students (...)
INMEMORY PRIORITY HIGH MEMCOMPRESS FOR CAPACITY HIGH;

CREATE TABLE grades (...)
INMEMORY PRIORITY CRITICAL MEMCOMPRESS FOR CAPACITY HIGH;

CREATE TABLE points (...)
INMEMORY PRIORITY CRITICAL MEMCOMPRESS FOR CAPACITY HIGH;
```

2. **FOR QUERY HIGH Configuration**:

```
CREATE TABLE students (...)
INMEMORY PRIORITY HIGH MEMCOMPRESS FOR QUERY HIGH;

CREATE TABLE grades (...)
INMEMORY PRIORITY CRITICAL MEMCOMPRESS FOR QUERY HIGH;

CREATE TABLE points (...)
INMEMORY PRIORITY CRITICAL MEMCOMPRESS FOR QUERY HIGH;
```

3. **FOR CAPACITY LOW Configuration**:

```
CREATE TABLE students (...)
INMEMORY PRIORITY LOW MEMCOMPRESS FOR CAPACITY LOW;
```

```
CREATE TABLE grades (...)
INMEMORY PRIORITY MEDIUM MEMCOMPRESS FOR CAPACITY LOW;

CREATE TABLE points (...)
INMEMORY PRIORITY MEDIUM MEMCOMPRESS FOR CAPACITY LOW;
```

The parameter combinations control two critical aspects of In-Memory behavior:

- **PRIORITY** (HIGH/CRITICAL/MEDIUM/LOW): Determines loading sequence and memory allocation precedence during population
- **MEMCOMPRESS** (FOR CAPACITY HIGH/FOR QUERY HIGH/FOR CAPACITY LOW): Controls compression algorithm selection balancing memory usage versus query performance

12.6.2 Experiment 2 - System Configuration Parameters

All configurations in this experiment utilized a consistent memory allocation optimized for compression method comparison:

Memory Component	Configuration (bytes)
Total System Global Area	1,333,788,432
Fixed Size	9,867,024
Variable Size	515,899,392
Database Buffers	268,435,456
In-Memory Area	536,870,912
In-Memory Size (MB)	**512**
Total SGA Size (GB)	**1.24**

12.6.3 Experiment 2 - Performance Results

The compression method comparison revealed significant performance variations across different In-Memory optimization strategies. All 25 test cases were executed 10 times each to ensure statistical reliability. Results are grouped by procedure for easy comparison across columnar storage technologies:

Procedure	Columnar Technology	Min (ms)	Max (ms)	Avg (ms)	vs Clean (%)	Query Cost	Rank
Grades analysis	Clean Database (baseline)	752	1011	851.4	*baseline*	4851K	*baseline*
	FOR CAPACITY HIGH	1469	1801	1638.2	+92.4%	3801K	-21.6%
	FOR QUERY HIGH	752	882	818.8	-3.8%	1267K	-73.9%
	FOR CAPACITY LOW	759	909	832.2	-2.3%	6875K	+41.7%
Best students	Clean Database (baseline)	492	571	535.8	*baseline*	53205	*baseline*
	FOR CAPACITY HIGH	519	612	576.8	+7.6%	45698	-14.1%
	FOR QUERY HIGH	410	540	476.9	-11.0%	25348	-52.4%
	FOR CAPACITY LOW	471	561	524.4	-2.1%	44668	-16.0%
Raise grades	Clean Database (baseline)	2200	2745	2459.1	*baseline*	1243K	*baseline*
	FOR CAPACITY HIGH	2443	2734	2619.3	+6.5%	1124K	-9.6%
	FOR QUERY HIGH	2346	2754	2587.4	+5.2%	532K	-57.2%
	FOR CAPACITY LOW	2465	2786	2581.4	+5.0%	1662K	+33.7%
Assign subjects	Clean Database (baseline)	3587	3987	3853.3	*baseline*	7421	*baseline*
	FOR CAPACITY HIGH	3447	3897	3636.1	-5.6%	2691	-63.7%
	FOR QUERY HIGH	3687	4100	3847.3	-0.2%	7181	-3.2%
	FOR CAPACITY LOW	3568	3971	3765.2	-2.3%	7443	+0.3%
Remove points	Clean Database (baseline)	3492	5368	4340.5	*baseline*	15863	*baseline*
	FOR CAPACITY HIGH	3683	4333	4004.2	-7.7%	9021	-43.1%
	FOR QUERY HIGH	3695	5397	4342.6	+0.05%	11347	-28.5%
	FOR CAPACITY LOW	3811	4458	4077.9	-6.1%	16329	+2.9%

12.6.4 Experiment 2 - Results Summary

Performance comparison results:

- **FOR QUERY HIGH**: Best overall - 73.9% cost reduction for Grades analysis, minimal execution time impact
- **FOR CAPACITY HIGH**: Excellent for DML operations - 63.7% cost reduction for Assign subjects
- **FOR CAPACITY LOW**: Balanced approach with consistent modest improvements