**1.Chosen Case Study**

**Case Study: Database design for a high school management system**

**2. Description of the Case Study**

**A database will be designed for a high school management system. This system will be used to manage students, teachers, courses, classes, and club memberships. The objective is to systematically manage student enrollments, teacher assignments, course details, and club memberships.**

**3. Requirements Analysis and Collection**

**General Requirements:**

**Recording student information, class numbers, and enrolled courses.**

**Recording teacher information, subjects, and assigned courses.**

**Managing each course’s assigned teacher and enrolled students.**

**Recording student grades and attendance data.**

**Managing club details and club memberships.**

**Specific Requirements:**

**A student can enroll in multiple courses.**

**A teacher can teach multiple courses.**

**Each course will have one assigned teacher.**

**Each class will have one class president.**

**Students will be able to join different clubs.**

**4. Structuring of Requirements**

**Student: StudentID, Name, SurName, BirthDate, Gender, Register state, Classno**

**Teacher: TeacherID, Name, Surname, Branch, Phone, Email**

**Lessons: LessonName, LessonCode, Credits**

**Class: ClassNo, Number of Student,Lesson Code,Minister of Class Id**

**Registration: RegisterID, StudentID, LessonCode, Score, Discontinutiy**

**Assignment: AssignID, LessonCode, TeacherID**

**Club: ClubID, ClubName, Advisor**

**Club Membership: MembershipID, StudentID, ClubID**

**5. Glossary of Terms**

**Student: Individuals enrolled at the high school.**

**Teacher: Members of the high school teaching staff.**

**Course: Subjects taught at the high school.**

**Class: Educational group to which students belong.**

**Registration: The process of a student registering for a course.**

***Assignme*nt: The process of a teacher being assigned to a course.**

**Club: Organizations where students participate in social and academic activities.**

**Club Membership: The participation status of students in various clubs**

**6.Logical Design**

**İt is in attechment**

**7.Relational Algebra Queries**

**1. List of courses a student is enrolled in**

* **Query:** σRegistration.StudentID=1(Registration⋈Lessons)\
* **Explanation:**
  + Joins the Registration and Lessons tables on LessonCode.
  + Filters the results to include only rows where StudentID = 1.

**2. List of courses taught by a teacher**

* **Query:** σAssignment.TeacherID=1(Assignment⋈Lessons)
* **Explanation:**
  + Joins the Assignment and Lessons tables on LessonCode.
  + Filters the results for TeacherID = 1.

**3. List of all students in a specific class**

* **Query:** σStudent.ClassNo=101(Student)
* **Explanation:**
  + Filters the Student table to include only those with ClassNo = 101.

**4. List of students in a specific club**

* **Query:** πStudentID(σClubMembership.ClubID=5(ClubMembership))
* **Explanation:**
  + Filters the ClubMembership table for ClubID = 5.
  + Projects only the StudentID column.

**5. Number of students in each course**

* **Query:** γLessonCode,count(StudentID)(Registration)
* **Explanation:**
  + Groups the Registration table by LessonCode.
  + Counts the number of StudentID entries for each course.

**6. List of grades and attendance of a specific student**

* **Query:** πLessonCode,Score,Discontinuity(σRegistration.StudentID=1(Registration))
* **Explanation:**
  + Filters the Registration table for StudentID = 1.
  + Projects only the LessonCode, Score, and Discontinuity columns.

**7. List of clubs and their advisors**

* **Query:** πClubName,Advisor(Club)
* **Explanation:**
  + Projects the ClubName and Advisor columns from the Club table.

**8. Identify the class president for each class**

* **Query:** πName,SurName(σStudent.StudentID=Class.MinisterOfClassID(Student⋈Class))
* **Explanation:**
  + Joins the Student and Class tables.
  + Filters the results where StudentID matches MinisterOfClassID.

**9. List of students in a club advised by a specific teacher**

* **Query:** πStudentID((σClub.Advisor=1(Club)⋈ClubMembership))
* **Explanation:**
  + Filters the Club table for Advisor = 1.
  + Joins the result with ClubMembership to list all StudentID entries.

**10. List of lessons assigned to a specific class**

* **Query:** πLessonCode(σClass.ClassNo=101(Lessons⋈Class))
* **Explanation:**
  + Joins the Lessons and Class tables on LessonCode.
  + Filters the results for ClassNo = 101.
  + Projects only the LessonCode.