**Database and Application Design Report**

**1. Case Study**

**Selected Case Study:** High School Management System  
**Description:**  
This system is designed to manage high school students, teachers, courses, classes, and clubs. The goal is to organize student enrollments, teacher assignments, and club memberships systematically.

**2. Requirements Analysis**

**General Requirements:**

* Recording student details (name, surname, birthdate, etc.) and their class enrollments.
* Managing teacher details and their course assignments.
* Recording course details and the students enrolled in each course.
* Tracking student grades and attendance records.
* Managing club details and memberships.

**Specific Requirements:**

* A student can enroll in multiple courses.
* A teacher can teach multiple courses.
* Each course is assigned one teacher.
* Each class has one class president.
* Students can join different clubs.

**3. Database Design**

The following table structures were created:

**Student Table:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| student\_id | Integer | Primary Key |
| name | String | Student's first name |
| surname | String | Student's last name |
| birth\_date | Date | Date of birth |
| gender | String | Gender |
| register\_state | String | Enrollment status |
| class\_no | Integer | Associated class number |

**Teacher Table:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| teacher\_id | Integer | Primary Key |
| name | String | Teacher's first name |
| surname | String | Teacher's last name |
| branch | String | Subject specialization |
| phone | String | Contact number |
| email | String | Email address |

**Course Table:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| lesson\_code | String | Primary Key |
| lesson\_name | String | Course name |
| credits | Integer | Number of credits |

**4. Implementation and Technologies**

* **Python:** Used as the core programming language.
* **SQLAlchemy:** For Object-Relational Mapping (ORM).
* **SQLite:** Used as the lightweight database management system.

**Example CRUD Operation**

python

Kodu kopyala

# Adding a new student

new\_student = Student(name="Ali", surname="Veli", birth\_date="2005-06-15", gender="M", register\_state="Active", class\_no=101)

session.add(new\_student)

session.commit()

**5. Example SQL Queries**

**Query 4: List of students in a specific club**  
**Query:**

sql

Kodu kopyala

SELECT StudentID FROM ClubMembership WHERE ClubID = 5;

**SQLAlchemy Implementation:**

python

Kodu kopyala

students\_in\_club = session.query(ClubMembership.student\_id).filter(ClubMembership.club\_id == 5).all()

**6. Conclusion and Evaluation**

This report covers the design of a high school management system's database and a corresponding Python application. The system's functionality has been implemented and tested using CRUD operations and SQL queries.