**School Database Management Project**

**Elaborated By:**

Imen BEN DAI

Contents

[General introduction 3](#_Toc123565459)

[Database Overview 4](#_Toc123565460)

[I- Introduction 4](#_Toc123565461)

[II- Database schema 5](#_Toc123565462)

[III- Tables 5](#_Toc123565463)

[1- Subject table 5](#_Toc123565464)

[2- Teacher Table 6](#_Toc123565465)

[3- Field Table 6](#_Toc123565466)

[4- Student Table 6](#_Toc123565467)

[5- Enrollment Table 7](#_Toc123565468)

[6- Teaching Table 7](#_Toc123565469)

[7- Subject\_field Table 8](#_Toc123565470)

[IV- Functions and Procedures 8](#_Toc123565471)

[1- Functions 8](#_Toc123565472)

[2- Procedures 9](#_Toc123565473)

[V- Triggers 10](#_Toc123565474)

[Conclusion 11](#_Toc123565475)

General introduction

A database is a collection of data that is organized and stored in a structured format, allowing for easy access and manipulation. In this report, we will be discussing a database that aims to manage the database of a school.

In this project, a database has been developed to store and manage information about students, teachers, fields of study, and subjects, as well as the relationships between these entities. This database is an important tool for organizing and managing educational data at a school.

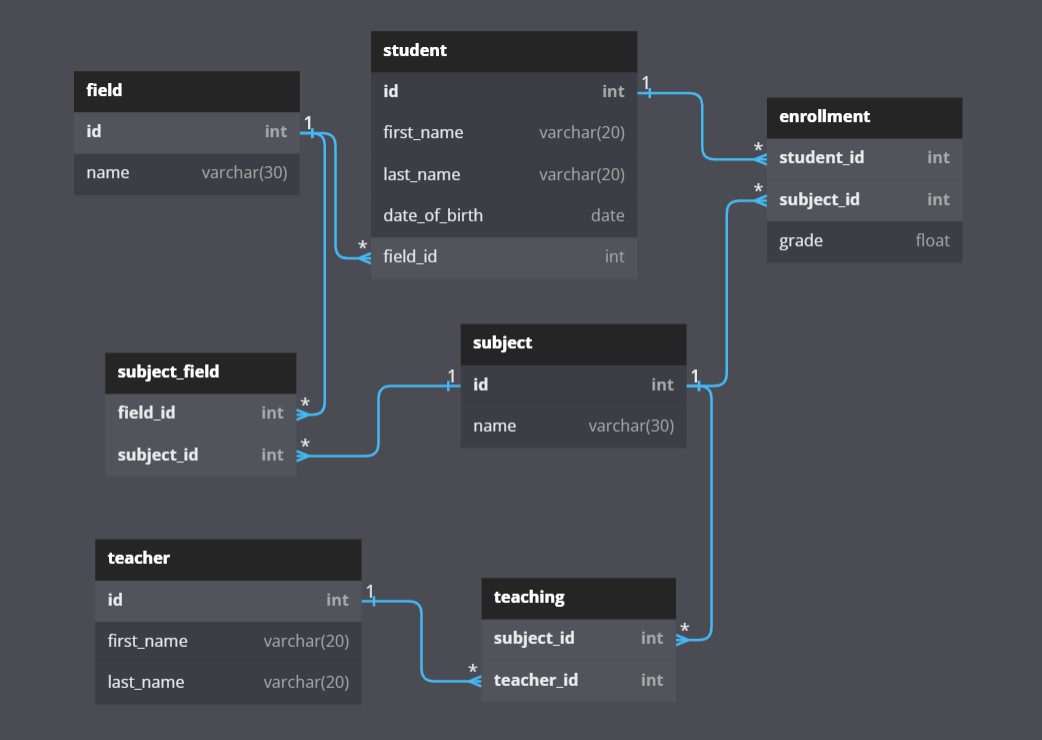
The purpose of this project is to design and implement a database that can be used to manage educational data at a school. In the following sections of this report, a detailed overview of the structure and capabilities of the database will be provided. Each table will be described in detail, including the fields and data types, as well as the relationships between the tables. Any potential future developments for the database will also be discussed.

Database Overview

1. Introduction:

The database consists of four tables: student, teacher, field, subject, and three intermediate tables: subject\_field, enrollment and teaching. The student table contains information about each student, including their name, age, and field of study. The field table contains a list of fields of study that are available at the school. The teacher table contains information about each teacher, including their name. The subject table contains a list of subjects that are offered at the school. The enrollment table is used to track which subjects each student is enrolled in, and their grades in those subjects. The teaching table is used to track which subjects each teacher is responsible for teaching. The subject\_field table is used to track which fields each subject belongs to.

1. Database schema



1. Tables:
2. **Subject table**:

The subject table stores information about the subjects offered at the school.

It has two fields: id and name. The id field is an integer and serves as the primary key for the table. It is also inserted using a sequence. The name field is a string of up to 30 characters and stores the name of the subject.

1. Teacher Table:

The teacher table stores information about the teachers at the school.

It has three fields: id, first\_name, and last\_name. The id field is an integer and serves as the primary key for the table. It is also populated using a sequence, with the values being generated automatically starting at 100 when new teachers are added to the table. The first\_name and last\_name fields are strings of up to 20 characters each, and store the first and last names of the teacher, respectively.

1. Field Table:

The field table stores information about the fields of study that are available at the school.

It has two fields: id and name. The id field is an integer and serves as the primary key for the table. It is also populated using a sequence, with the values being generated automatically starting at 200 when new fields are added to the table. The name field is a string of up to 30 characters and stores the name of the field of study.

1. Student Table:

The student table stores information about the students at the school.

It has five fields: id, first\_name, last\_name, date\_of\_birth, and field\_id. The id is an integer and serves as the primary key for the table. It is also populated using a sequence, with the values being generated automatically when new students are added to the table. The first\_name and last\_name fields are strings of up to 20 characters each, and store the first and last names of the student, respectively. The date\_of\_birth field is a date data type and stores the student's date of birth. The field\_id field is an integer and stores the ID of the field of study that the student is enrolled in. There is a foreign key relationship between the student and field tables, with the field\_id field referencing the id in the field table.

1. Enrollment Table:

The enrollment table stores information about the subjects that students are enrolled in, and their grades in those subjects.

It has three fields: student\_id, subject\_id, and grade. The student\_id and subject\_id fields are integers and store the IDs of the student and subject, respectively. The grade field is a float data type and stores the student's grade in the subject. The value for the grade field is generated using the TRUNC function, which truncates the decimal part of a random value generated by the DBMS\_RANDOM package, and divides the result by 100. The combination of the student\_id and subject\_id fields serves as the primary key for the table. There are foreign key relationships between the enrollment table and the student and subject tables, with the student\_id and subject\_id fields referencing the id fields in the student and subject tables, respectively.

1. Teaching Table:

The teaching table stores information about the subjects that teachers are responsible for teaching.

It has two fields: subject\_id and teacher\_id. The subject\_id and teacher\_id fields are integers and store the IDs of the subject and teacher, respectively. The combination of the subject\_id and teacher\_id fields serves as the primary key for the table. There are foreign key relationships between the teaching table and the subject and teacher tables, with the subject\_id and teacher\_id fields referencing the id fields in the subject and teacher tables, respectively.

1. Subject\_field Table:

The subject\_field table stores information about the fields that subjects belong to.

It has two fields: subject\_id and field\_id. The subject\_id and field\_id fields are integers and store the IDs of the subject and field, respectively. The combination of the subject\_id and field\_id fields serves as the primary key for the table. There are foreign key relationships between the subject\_field table and the subject and field tables, with the subject\_id and field\_id fields referencing the id fields in the subject and field tables, respectively.

1. Functions and Procedures
2. Functions:

The **mean\_subject** function calculates and returns the mean grade of all students enrolled in a given subject. It takes a subject ID as input and uses a SELECT statement to retrieve the necessary data from the enrollment table.

The **mean\_field** function calculates and returns the mean grade of all students enrolled in subjects belonging to a given field of study. It takes a field ID as input and uses a cursor and the mean\_subject function to iterate over the subject\_field table and calculate the mean grades for each subject.

The **mean\_student** function calculates and returns the mean grade of a given student across all subjects that they are enrolled in. It takes a student ID as input and uses a SELECT statement to retrieve the necessary data from the enrollment table.

The **highest\_mean** function calculates and returns the highest mean grade of all students in the database. It uses a cursor and the mean\_student function to iterate over the student table and compare the mean grades of each student to find the highest value.

1. **Procedures** :

The **repeating\_student\_per\_field** procedure outputs the names and mean grades of all students in a given field of study who have a mean grade below 10. It takes a field ID as input and uses a cursor to iterate over the student table, filtering the results by field ID. For each student, it calls the mean\_student function to calculate the mean grade and outputs the results if the mean is below 10. If the procedure is unable to find any data in the student table with the specified field ID, it will raise a NO\_DATA\_FOUND exception and output a message indicating that there is no repeating students in that field.

The **top\_ten\_students** procedure outputs the names and mean grades of the top 10 students in the database, ranked in descending order of mean grade. It uses a cursor and the mean\_student function to iterate over the student table, sorting the results by mean grade and outputting the top 10. This procedure does not have any exceptions defined.

The top\_three\_students\_field procedure outputs the names and mean grades of the top 3 students in a given field of study, ranked in descending order of mean grade. It takes a field ID as input and uses a cursor and the mean\_student function to iterate over the student table, filtering the results by field ID and outputting the top 3. If the procedure is unable to find any data in the student table with the specified field ID, it will raise a NO\_DATA\_FOUND exception and output a message indicating that the field ID does not exist.

The delete\_student procedure deletes a student from the database, given their ID. It first deletes any records in the enrollment table that have the student ID as a foreign key, and then deletes the student's record from the student table. If the procedure is unable to find a student with the specified ID in the student table, it will raise a NO\_DATA\_FOUND exception and output a message indicating that the student ID does not exist.

1. **Triggers** :

The student\_trigger prevents ID values outside the range of 1 to 99 from being inserted or updated in the student table.

The teacher\_trigger prevents ID values outside the range of 100 to 199 from being inserted or updated in the teacher table.

The field\_trigger prevents ID values outside the range of 200 to 299 from being inserted or updated in the field table.

The subject\_trigger prevents ID values outside the range of 300 to 399 from being inserted or updated in the subject table.

These triggers execute before an insert or update operation on their corresponding table, and operate on each row of the table separately. If an invalid ID value is encountered, an application error is raised to prevent it from being inserted or updated in the table.

These triggers were created by another user 'imen'.

Code

Database :

Connected with: system

CREATE TABLE subject (

id INTEGER PRIMARY KEY,

name VARCHAR2(30) NOT NULL);

CREATE TABLE teacher (

id INTEGER PRIMARY KEY,

first\_name VARCHAR2(20) NOT NULL,

last\_name VARCHAR2(20) NOT NULL);

CREATE TABLE field (

id INTEGER PRIMARY KEY,

name VARCHAR2(30) NOT NULL);

CREATE TABLE student (

id INTEGER PRIMARY KEY,

first\_name VARCHAR2(20) NOT NULL,

last\_name VARCHAR2(20) NOT NULL,

date\_of\_birth DATE NOT NULL,

field\_id INTEGER NOT NULL,

FOREIGN KEY (field\_id) REFERENCES field(id));

CREATE TABLE enrollment (

student\_id INTEGER NOT NULL,

subject\_id INTEGER NOT NULL,

grade FLOAT NOT NULL,

PRIMARY KEY (student\_id, subject\_id),

FOREIGN KEY (student\_id) REFERENCES student(id),

FOREIGN KEY (subject\_id) REFERENCES subject(id));

CREATE TABLE teaching (

subject\_id INTEGER NOT NULL,

teacher\_id INTEGER NOT NULL,

PRIMARY KEY (subject\_id, teacher\_id),

FOREIGN KEY (subject\_id) REFERENCES subject(id),

FOREIGN KEY (teacher\_id) REFERENCES teacher(id));

CREATE TABLE subject\_field (

subject\_id INTEGER NOT NULL,

field\_id INTEGER NOT NULL,

PRIMARY KEY (subject\_id, field\_id),

FOREIGN KEY (subject\_id) REFERENCES subject(id),

FOREIGN KEY (field\_id) REFERENCES field(id));

CREATE SEQUENCE seq\_student

INCREMENT BY 1;

CREATE SEQUENCE seq\_teacher

START WITH 100

INCREMENT BY 1;

CREATE SEQUENCE seq\_field

START WITH 200

INCREMENT BY 1;

CREATE SEQUENCE seq\_subject

START WITH 300

INCREMENT BY 1;

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Fahd', 'Jbeli');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Sara', 'louati');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Mohamed', 'Karoui');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Yasmin', 'Azizi');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Imen', 'Ben mahmoud');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Aisha', 'Ben sliman');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Soumaya', 'Trabelsi');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Hichem', 'Mrabet');

INSERT INTO teacher (id, first\_name, last\_name) VALUES (seq\_teacher.nextval, 'Samia', 'Zidi');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Business Law');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Operations Management');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Human Resource Management');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Organizational Behavior');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'International Business');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Data Mining');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Data Visualization');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Predictive Analytics');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Business Statistics');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Strategic Management');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Marketing Research');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Web Development');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Project Management');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Mobile Development');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'System Architecture');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Taxation');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Financial Management');

INSERT INTO subject (id, name) VALUES (seq\_subject.nextval, 'Financial Markets');

INSERT INTO field (id, name) VALUES (seq\_field.nextval, 'Marketing');

INSERT INTO field (id, name) VALUES (seq\_field.nextval, 'Management');

INSERT INTO field (id, name) VALUES (seq\_field.nextval, 'Business Intelligence');

INSERT INTO field (id, name) VALUES (seq\_field.nextval, 'Finance');

INSERT INTO field (id, name) VALUES (seq\_field.nextval, 'Business Computing');

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id) VALUES (seq\_student.nextval, 'Hana', 'Ben Salem', '2000-01-01', 200);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id) VALUES (seq\_student.nextval, 'Mohammed', 'Khemiri', '1999-02-15', 201);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Fatma', 'Zouaoui', '1998-03-31', 202);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Ali', 'Toumi', '2001-04-30', 203);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Leila', 'Karoui', '1998-05-15', 201);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Ahmed', 'Slim', '2002-06-30', 202);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Marwa', 'Ben Said', '2001-11-01', 200);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Mohammed', 'Melliti', '1998-03-15', 204);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Faycel', 'Sehli', '2000-03-01', 202);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Ali', 'Trabelsi', '2000-05-30', 203);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Leila', 'Neffati', '1999-06-25', 201);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Ahmed', 'Sliti', '2000-06-23', 202);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Hazem', 'Akrimi', '1999-08-15', 204);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Seif', 'Jabri', '2001-03-21', 200);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Mohamed', 'Hachicha', '2002-05-20', 203);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Marwa', 'Abed', '1999-09-15', 201);

INSERT INTO student (id, first\_name, last\_name, date\_of\_birth, field\_id)

VALUES (seq\_student.nextval, 'Ahmed', 'Ben Amor', '2000-07-13', 202);

insert into subject\_field (subject\_id ,field\_id)

values (300,200);

insert into subject\_field (subject\_id ,field\_id)

values (302,200);

insert into subject\_field (subject\_id ,field\_id)

values (304,200);

insert into subject\_field (subject\_id ,field\_id)

values (308,200);

insert into subject\_field (subject\_id ,field\_id)

values (310,200);

insert into subject\_field (subject\_id ,field\_id)

values (312,200);

insert into subject\_field (subject\_id ,field\_id)

values (300,201);

insert into subject\_field (subject\_id ,field\_id)

values (301,201);

insert into subject\_field (subject\_id ,field\_id)

values (302,201);

insert into subject\_field (subject\_id ,field\_id)

values (303,201);

insert into subject\_field (subject\_id ,field\_id)

values (304,201);

insert into subject\_field (subject\_id ,field\_id)

values (308,201);

insert into subject\_field (subject\_id ,field\_id)

values (309,201);

insert into subject\_field (subject\_id ,field\_id)

values (312,201);

insert into subject\_field (subject\_id ,field\_id)

values (305,202);

insert into subject\_field (subject\_id ,field\_id)

values (306,202);

insert into subject\_field (subject\_id ,field\_id)

values (307,202);

insert into subject\_field (subject\_id ,field\_id)

values (308,202);

insert into subject\_field (subject\_id ,field\_id)

values (311,202);

insert into subject\_field (subject\_id ,field\_id)

values (312,202);

insert into subject\_field (subject\_id ,field\_id)

values (313,202);

insert into subject\_field (subject\_id ,field\_id)

values (300,203);

insert into subject\_field (subject\_id ,field\_id)

values (303,203);

insert into subject\_field (subject\_id ,field\_id)

values (304,203);

insert into subject\_field (subject\_id ,field\_id)

values (308,203);

insert into subject\_field (subject\_id ,field\_id)

values (315,203);

insert into subject\_field (subject\_id ,field\_id)

values (316,203);

insert into subject\_field (subject\_id ,field\_id)

values (317,203);

insert into subject\_field (subject\_id ,field\_id)

values (304,204);

insert into subject\_field (subject\_id ,field\_id)

values (308,204);

insert into subject\_field (subject\_id ,field\_id)

values (311,204);

insert into subject\_field (subject\_id ,field\_id)

values (312,204);

insert into subject\_field (subject\_id ,field\_id)

values (313,204);

insert into subject\_field (subject\_id ,field\_id)

values (314,204);

insert into teaching (subject\_id ,teacher\_id)

values (300,100);

insert into teaching (subject\_id ,teacher\_id)

values (315,100);

insert into teaching (subject\_id ,teacher\_id)

values (301,101);

insert into teaching (subject\_id ,teacher\_id)

values (302,101);

insert into teaching (subject\_id ,teacher\_id)

values (303,101);

insert into teaching (subject\_id ,teacher\_id)

values (304,101);

insert into teaching (subject\_id ,teacher\_id)

values (309,101);

insert into teaching (subject\_id ,teacher\_id)

values (312,101);

insert into teaching (subject\_id ,teacher\_id)

values (301,102);

insert into teaching (subject\_id ,teacher\_id)

values (302,102);

insert into teaching (subject\_id ,teacher\_id)

values (303,102);

insert into teaching (subject\_id ,teacher\_id)

values (304,102);

insert into teaching (subject\_id ,teacher\_id)

values (309,102);

insert into teaching (subject\_id ,teacher\_id)

values (312,102);

insert into teaching (subject\_id ,teacher\_id)

values (305,103);

insert into teaching (subject\_id ,teacher\_id)

values (306,103);

insert into teaching (subject\_id ,teacher\_id)

values (307,103);

insert into teaching (subject\_id ,teacher\_id)

values (308,103);

insert into teaching (subject\_id ,teacher\_id)

values (305,104);

insert into teaching (subject\_id ,teacher\_id)

values (306,104);

insert into teaching (subject\_id ,teacher\_id)

values (307,104);

insert into teaching (subject\_id ,teacher\_id)

values (308,104);

insert into teaching (subject\_id ,teacher\_id)

values (314,105);

insert into teaching (subject\_id ,teacher\_id)

values (313,105);

insert into teaching (subject\_id ,teacher\_id)

values (311,105);

insert into teaching (subject\_id ,teacher\_id)

values (314,106);

insert into teaching (subject\_id ,teacher\_id)

values (313,106);

insert into teaching (subject\_id ,teacher\_id)

values (311,106);

insert into teaching (subject\_id ,teacher\_id)

values (310,107);

insert into teaching (subject\_id ,teacher\_id)

values (316,108);

insert into teaching (subject\_id ,teacher\_id)

values (317,108);

insert into enrollment(subject\_id, student\_id, grade) values (300,1,15);

insert into enrollment(subject\_id, student\_id, grade) values (304,1,10);

insert into enrollment(subject\_id, student\_id, grade) values (308,1,5);

insert into enrollment(subject\_id, student\_id, grade) values (310,1,16);

insert into enrollment(subject\_id, student\_id, grade) values (302,1,12);

insert into enrollment(subject\_id, student\_id, grade) values (312,1,6);

insert into enrollment(subject\_id, student\_id, grade) values (300,7,6);

insert into enrollment(subject\_id, student\_id, grade) values (304,7,20);

insert into enrollment(subject\_id, student\_id, grade) values (308,7,15);

insert into enrollment(subject\_id, student\_id, grade) values (310,7,10);

insert into enrollment(subject\_id, student\_id, grade) values (302,7,8.5);

insert into enrollment(subject\_id, student\_id, grade) values (312,7,16);

insert into enrollment(subject\_id, student\_id, grade) values (300,14,7.5);

insert into enrollment(subject\_id, student\_id, grade) values (304,14,17);

insert into enrollment(subject\_id, student\_id, grade) values (308,14,5.75);

insert into enrollment(subject\_id, student\_id, grade) values (310,14,11);

insert into enrollment(subject\_id, student\_id, grade) values (302,14,2);

insert into enrollment(subject\_id, student\_id, grade) values (312,14,14);

insert into enrollment(subject\_id, student\_id, grade) values (304,2,6);

insert into enrollment(subject\_id, student\_id, grade) values (303,2,20);

insert into enrollment(subject\_id, student\_id, grade) values (302,2,15);

insert into enrollment(subject\_id, student\_id, grade) values (301,2,10);

insert into enrollment(subject\_id, student\_id, grade) values (300,2,8.5);

insert into enrollment(subject\_id, student\_id, grade) values (312,2,16);

insert into enrollment(subject\_id, student\_id, grade) values (308,2,10);

insert into enrollment(subject\_id, student\_id, grade) values (309,2,8.5);

insert into enrollment(subject\_id, student\_id, grade) values (304,5,6.75);

insert into enrollment(subject\_id, student\_id, grade) values (303,5,2.5);

insert into enrollment(subject\_id, student\_id, grade) values (302,5,11);

insert into enrollment(subject\_id, student\_id, grade) values (301,5,15);

insert into enrollment(subject\_id, student\_id, grade) values (300,5,12.5);

insert into enrollment(subject\_id, student\_id, grade) values (312,5,8);

insert into enrollment(subject\_id, student\_id, grade) values (308,5,20);

insert into enrollment(subject\_id, student\_id, grade) values (309,5,18.5);

insert into enrollment(subject\_id, student\_id, grade) values (304,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (303,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (302,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (301,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (300,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (309,11,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (303,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (302,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (301,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (300,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (309,16,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (305,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (306,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (307,3,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (305,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (306,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (307,6,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (305,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (306,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (307,9,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (305,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (306,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (307,17,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (316,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (317,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (300,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (303,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (315,4,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (316,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (317,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (300,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (303,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (315,10,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (316,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (317,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (300,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (303,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (315,15,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (314,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,8,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (314,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,12,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (313,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (312,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (308,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (314,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (311,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

insert into enrollment(subject\_id, student\_id, grade) values (304,13,TRUNC(DBMS\_RANDOM.VALUE(0, 2000)) / 100);

Package:

create or replace package pkg\_school as

function mean\_subject(id\_sub subject.id%type) return float;

function mean\_field(id\_f field.id%type)return FLOAT;

function mean\_student(id\_s student.id%type)return float;

function highest\_mean return float;

procedure repeating\_student\_per\_field(id\_f field.id%type);

procedure top\_ten\_students;

procedure top\_three\_students\_field(id\_f field.id%type);

procedure delete\_student(id\_s student.id%type);

end pkg\_school;

/

create or replace package body pkg\_school as

function mean\_subject(id\_sub subject.id%type)return float IS

sum\_grade float;

student\_count int;

moy float;

begin

select count(student\_id), sum(grade) into student\_count, sum\_grade from enrollment where subject\_id=id\_sub;

moy := sum\_grade/student\_count;

return(moy);

end mean\_subject;

function mean\_field(id\_f field.id%type)return float is

grade\_total float;

count\_sub int;

mean\_s float;

mean\_f float;

id\_sub subject.id%type;

cursor c\_field is

select subject\_id from subject\_field where field\_id=id\_f;

begin

select count(subject\_id) into count\_sub from subject\_field where field\_id=id\_f;

mean\_f := 0;

mean\_s := 0;

grade\_total := 0;

open c\_field;

loop

fetch c\_field into id\_sub;

exit when c\_field%notfound;

mean\_s := mean\_subject(id\_sub);

grade\_total:= grade\_total + mean\_s;

end loop;

close c\_field;

mean\_f:=grade\_total/count\_sub;

return mean\_f;

Exception

When no\_data\_found then dbms\_output.put\_line ('this subject id does not exist');

end mean\_field;

function mean\_student(id\_s student.id%type)return float is

sum\_grade float;

sub\_count int;

moy float;

begin

select count(subject\_id), sum(grade) into sub\_count, sum\_grade from enrollment where student\_id=id\_s;

moy := sum\_grade/sub\_count;

return(moy);

Exception

When no\_data\_found then dbms\_output.put\_line ('this student id does not exist');

end mean\_student;

function highest\_mean return float is

count\_stu int;

id\_stu int;

mean\_s float;

cursor c\_moy is

select id from student;

begin

select count(id) into count\_stu from student;

mean\_s := 0;

open c\_moy;

loop

fetch c\_moy into id\_stu;

exit when c\_moy%notfound;

if mean\_s < mean\_student(id\_stu) then

mean\_s := mean\_student(id\_stu);

end if;

end loop;

close c\_moy;

return mean\_s;

end highest\_mean;

procedure repeating\_student\_per\_field(id\_f field.id%type) is

cursor c\_stu is

select id from student where field\_id=id\_f;

nom varchar2(30);

id\_stu int ;

prenom varchar2(30);

begin

open c\_stu;

loop

fetch c\_stu into id\_stu;

exit when c\_stu%notfound;

if mean\_student(id\_stu)<10 then

select first\_name into prenom from student where id=id\_stu;

select last\_name into nom from student where id=id\_stu;

dbms\_output.put\_line('Prénom : '||prenom||' Nom : '||nom|| ' Moyenne : '||mean\_student(id\_stu));

else dbms\_output.put\_line('Aucun élève n a redoublé dans cette section');

end if;

end loop;

close c\_stu;

Exception

When no\_data\_found then dbms\_output.put\_line ('this field id does not exist');

end repeating\_student\_per\_field;

procedure top\_ten\_students is

id\_stu int ;

cursor c\_stu is

select id into id\_stu from student order by mean\_student(id) desc fetch first 10 rows only;

nom varchar2(30);

prenom varchar2(30);

i int;

begin

open c\_stu;

i:=0;

loop

fetch c\_stu into id\_stu;

exit when c\_stu%notfound;

select first\_name into prenom from student where id=id\_stu;

select last\_name into nom from student where id=id\_stu;

i:=i+1;

dbms\_output.put\_line('Prénom : '||prenom||' Nom : '||nom|| ' Moyenne : '||round(mean\_student(id\_stu),2)||' Classement : '||i);

end loop;

close c\_stu;

end top\_ten\_students;

procedure top\_three\_students\_field(id\_f field.id%type) is

id\_stu int ;

cursor c\_stu is

select id into id\_stu from student where field\_id=id\_f order by mean\_student(id) desc fetch first 3 rows only;

nom varchar2(30);

prenom varchar2(30);

i int;

begin

open c\_stu;

i:=0;

loop

fetch c\_stu into id\_stu;

exit when c\_stu%notfound;

select first\_name into prenom from student where id=id\_stu;

select last\_name into nom from student where id=id\_stu;

i:=i+1;

dbms\_output.put\_line('Prénom : '||prenom||' Nom : '||nom|| ' Moyenne : '||round(mean\_student(id\_stu),2)||' Classement : '||i);

end loop;

close c\_stu;

Exception

When no\_data\_found then dbms\_output.put\_line ('this field id does not exist');

end top\_three\_students\_field;

procedure delete\_student(id\_s student.id%type) is

begin

delete from enrollment where student\_id=id\_s;

delete from student where id=id\_s;

Exception

When no\_data\_found then dbms\_output.put\_line ('this student id does not exist');

end delete\_student;

end pkg\_school;

Triggers:

Connected with: imen, password: 0000

create or replace trigger student\_trigger

before insert or update on student

for each row

begin

if :NEW.id not between 1 and 99 then

RAISE\_APPLICATION\_ERROR (-20001, 'You can only insert ids between 1 and 99');

end if;

end student\_id;

create or replace trigger teacher\_trigger

before insert or update on teacher

for each row

begin

if :NEW.id not between 100 and 199 then

RAISE\_APPLICATION\_ERROR (-20002, 'You can only insert ids between 100 and 199');

end if;

end teacher\_id;

create or replace trigger field\_trigger

before insert or update on field

for each row

begin

if :NEW.id not between 200 and 299 then

RAISE\_APPLICATION\_ERROR (-20003, 'You can only insert ids between 200 and 299');

end if;

end field\_id;

create or replace trigger subject\_trigger

before insert or update on subject

for each row

begin

if :NEW.id not between 300 and 399 then

RAISE\_APPLICATION\_ERROR (-20004, 'You can only insert ids between 300 and 399');

end if;

end subject\_id;

Conclusion

Overall, the database is designed to store and manage a wide range of educational data, including information about students, teachers, subjects, and fields of study, as well as the relationships between these entities. It is structured in a way that allows for efficient data retrieval and manipulation, and helps to ensure the integrity and consistency of the data. With its various tables, procedures, functions and triggers, the database is well-suited to support the management of educational data at a school.