E-Attendance System

Introduction :

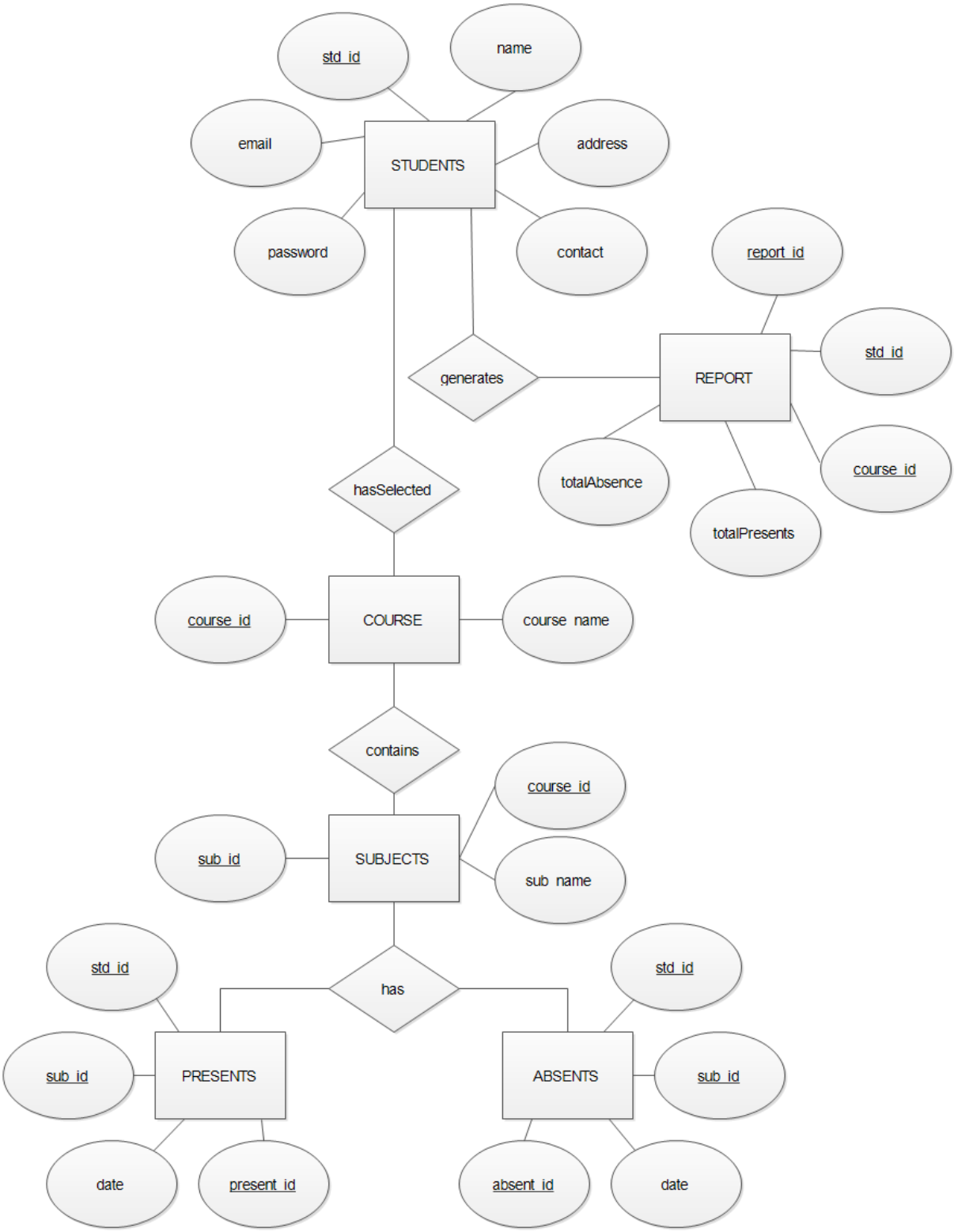
E-Attendance System will used to take attendace of student and manage their attendace data.

System will take details of students like name, contact, email ,passwords address to login in system and information purpose.

System will take attendance and with procedures it will generates presents and absents details and also update report table for generating overall report of sudent that are uder particular cource and attendance will taken upon subject that allocated to course and student that assigned to course.

System will also track students if student leave the system permanantly it will automatically deletes their reports.s

ER – Diagram:


std_id
name
email
STUDENTS
address
password
contact
report_id
std_id
generates
REPORT
course_id
totalAbsence
totalPresents
hasSelected
course_id
COURSE
course_name
contains
course_id
sub_id
SUBJECTS
sub_name
std_id
has
std_id
sub_id
PRESENTS
ABSENTS
sub_id
date
present_id
absent_id
date

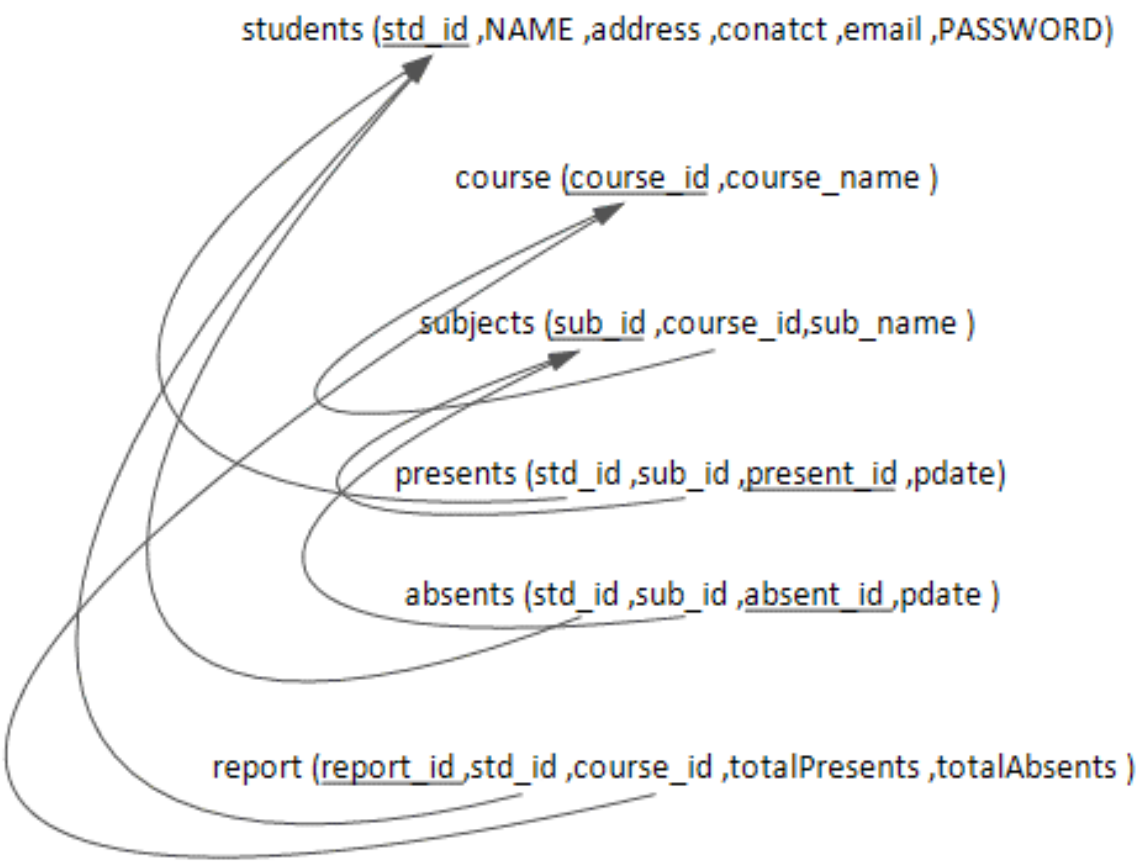
Relations :

Students has course details that will use to get presents and absents and also generates a report.

Course has subject details

Subjects subject detail wise present and absents are evaluated

Relational Schema :



Students : this table will store all students information.

CREATE TABLE students (std_id INT PRIMARY KEY,NAME VARCHAR(20),address VARCHAR(30),conatct VARCHAR(11),email VARCHAR(10),PASSWORD VARCHAR(11));

std_id	name	address	conatct	email	password
1	Max	surat	9145873587	max@gmail.	max@123
2	Roy	mumbai	9145951287	roy@gmail.	roy@123
3	Mia	delhi	9185233587	mia@gmail.	mia@123
4	Jenna	surat	9145626523	jenna@gmai	jenna@123
5	Seina	surat	9145628597	seina@gmai	seina@123

Course : this table store a basic details of course.

CREATE TABLE course (course_id INT PRIMARY KEY,course_name VARCHAR(20));

course_id	course_name
101	Computer
102	Civil
103	Mech

Subjects : this table stores the details of subject that assigned to course.

CREATE TABLE subjects (sub_id INT PRIMARY KEY,course_id INT REFERENCES course (course_id),sub_name VARCHAR(20));

sub_id	course_id	sub_name
1101	101	Flutter Devs
1102	101	Ruby Codes
1103	101	Advanced Algorithm T
1201	101	Group Handle soft
1301	101	Interview Spoils

Presents : this table stores a present data of student of particular subject of particular course.

CREATE TABLE presents (std_id INT REFERENCES students (std_id),sub_id INT REFERENCES subjects (sub_id),present_id INT PRIMARY KEY,pdate DATE);

std_id	sub_id	present_id	pdate

Absents : this table stores a absent data of student of particular subject of particular course.

CREATE TABLE absents (std_id INT REFERENCES students (std_id),sub_id INT REFERENCES subjects (sub_id),absent_id INT PRIMARY KEY,pdate DATE);

std_id	sub_id	absent_id	pdate

Report : this table generates overall attendance report of student.

CREATE TABLE report (report_id INT PRIMARY KEY,std_id INT REFERENCES students (std_id) ,course_id INT REFERENCES course (course_id),totalPresents INT,totalAbsents INT);

report_id	std_id	course_id	totalPresents	totalAbsents
1	1	101	0	0

Procedure to present add and update report's total present

```
DELIMITER $$

CREATE PROCEDURE `presentStd`(IN stdId INT,IN subId INT)

BEGIN

        INSERT INTO presents VALUES (stdId,subId,(stdId+RAND()),SYSDATE());

        UPDATE report SET totalPresents = totalPresents + 1 WHERE std_id = stdId;

    END$$

DELIMITER ;
```

Presents table :

std_id	sub_id	present_id	pdate
1	1102	1	2023-05-07

Report table :

report_id	std_id	course_id	totalPresents	totalAbsents
1	1	101	1	0

Procedure to absent add and update report's total absents

```
DELIMITER $$

CREATE PROCEDURE `absentStd`(IN stdId INT,IN subId INT)

BEGIN

        INSERT INTO absents VALUES (stdId,subId,(stdId+RAND()),SYSDATE());

        UPDATE report SET totalAbsents = totalPresents + 1 WHERE std_id = stdId;

    END$$

DELIMITER ;
```

Absents table :

std_id	sub_id	absent_id	pdate
1	1101	1	2023-05-07

Report table :

report_id	std_id	course_id	totalPresents	totalAbsents
1	1	101	1	2

Trigger that delete whole report data of particular student when student is deleted from table (After deleting data report deleted)

```
DELIMITER $$

CREATE

   TRIGGER `deleteStd` AFTER DELETE ON `students`

   FOR EACH ROW BEGIN

        DECLARE sId INT;

        SET sId = old.std_id;

        DELETE FROM report WHERE std_id = sId;

   END;

$$

DELIMITER ;
```

Students table :

std_id	name	address	conatct	email	password
1	Max	surat	9145873587	max@gmail.	max@123
2	Roy	mumbai	9145951287	roy@gmail.	roy@123
3	Mia	delhi	9185233587	mia@gmail.	mia@123
4	Jenna	surat	9145626523	jenna@gmai	jenna@123
5	Seina	surat	9145628597	seina@gmai	seina@123

(performing : DELETE FROM students WHERE std_id = 1;)

std_id	name	address	conatct	email	password
2	Roy	mumbai	9145951287	roy@gmail.	roy@123
3	Mia	delhi	9185233587	mia@gmail.	mia@123
4	Jenna	surat	9145626523	jenna@gmai	jenna@123
5	Seina	surat	9145628597	seina@gmai	seina@123

Report table :

report_id	std_id	course_id	totalPresents	totalAbsents