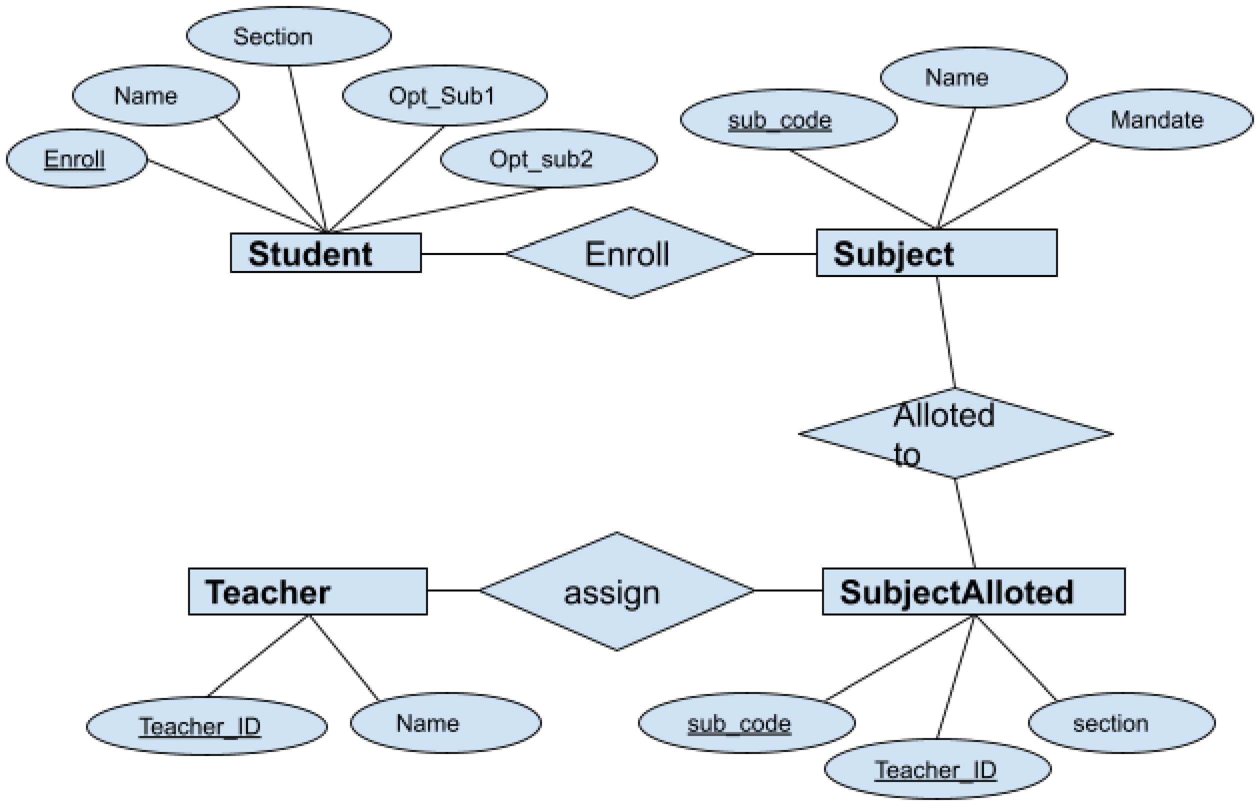
E-R Diagram



Student

Enr_no	Std_name	SECTION	Opt_sub1	Opt_sub2

Subject_

sub_code	sub_name	Opt

Subject_allot

sub_code	teacher_id	SECTION

Teacher

teacher_id	teacher_name

```
	CREATE TABLE Student (
		Enr_noINTEGER PRIMARY KEY,
		std_name TEXT,
		section	TEXT,
		opt_sub1 TEXT,
		opt_sub2 TEXT

	);
	CREATE TABLE Subject_ (
		sub_code	INTEGER PRIMARY KEY,
		sub_name	TEXT,
		opt	TEXT

	);
	CREATE TABLE Subject_allot (
		sub_code	INTEGER ,
		teacher_id	INTEGER ,
		section	TEXT ,
		PRIMARY KEY (sub_code,teacher_id)
	);
	CREATE TABLE Teacher (
		teacher_id	INTEGER,
		teacher_name	TEXT,
		PRIMARY KEY (teacher_id)
	);
```
==

Schema:

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.1 (Debian 14.1-1.pgdg110+1)
-- Dumped by pg_dump version 14.1 (Debian 14.1-1.pgdg110+1)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: demo; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.demo (
    id integer NOT NULL,
    name character varying(200) DEFAULT ''::character varying NOT NULL,
    hint text DEFAULT ''::text NOT NULL
);


--
-- Name: demo_id_seq; Type: SEQUENCE; Schema: public; Owner: -
--

CREATE SEQUENCE public.demo_id_seq
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


--
-- Name: demo_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -
--

ALTER SEQUENCE public.demo_id_seq OWNED BY public.demo.id;


--
-- Name: student; Type: TABLE; Schema: public; Owner: -
```

```
--

CREATE TABLE public.student (
    enr_no integer NOT NULL,
    std_name text,
    section text,
    opt_sub1 text,
    opt_sub2 text
);


--
-- Name: subject_; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.subject_ (
    sub_code integer NOT NULL,
    sub_name text,
    opt text
);


--
-- Name: subject_allot; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.subject_allot (
    sub_code integer NOT NULL,
    teacher_id integer NOT NULL,
    section text
);


--
-- Name: teacher; Type: TABLE; Schema: public; Owner: -
--

CREATE TABLE public.teacher (
    teacher_id integer NOT NULL,
    teacher_name text
);


--
```

```
-- Name: demo id; Type: DEFAULT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.demo ALTER COLUMN id SET DEFAULT
nextval('public.demo_id_seq'::regclass);


--
-- Name: demo demo_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.demo
    ADD CONSTRAINT demo_pkey PRIMARY KEY (id);


--
-- Name: student student_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.student
    ADD CONSTRAINT student_pkey PRIMARY KEY (enr_no);


--
-- Name: subject_ subject__pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.subject_
    ADD CONSTRAINT subject__pkey PRIMARY KEY (sub_code);


--
-- Name: subject_allot subject_allot_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.subject_allot
    ADD CONSTRAINT subject_allot_pkey PRIMARY KEY (sub_code, teacher_id);


--
-- Name: teacher teacher_pkey; Type: CONSTRAINT; Schema: public; Owner: -
--

ALTER TABLE ONLY public.teacher
```

```
    ADD CONSTRAINT teacher_pkey PRIMARY KEY (teacher_id);


--
-- PostgreSQL database dump complete
--
```

==

Queries:

1. **List all the subjects taught by given teacher (section wise)**

   ```
   Select sub.Sub_name form subjectAlloted subA join subject sub on
   subA.subj_code= sub.sub_code where Teacher_ID=(Select Teacher_ID from Teacher
   where Name='name')
   ```

2. **List of all the students who have selected a given optional subject (section wise)**

   ```
   SELECT std_name FROM student WHERE opt_sub1='Hindi' order by section;
   ```

3. **List teacher who teaches given section**

   ```
   SELECT teacher_name FROM teacher teach INNER JOIN subject_allot sub on
   teach.teacher_id=sub.teacher_id where sub.section='C';
   ```

4. **List of subject taught by given teacher**

   ```
   SELECT sub_code,sub_name FROM subject_ WHERE sub_code IN (SELECT
   S.sub_code from Teacher te INNER JOIN Subject_allot S ON
   te.teacher_id=S.teacher_id where te.teacher_name = 'Mr. X');
   ```

5. **List all the Optional Subjects**

   ```
   Select sub_code,sub_name from Subject_ where opt='Y';
   ```

==

API

- Creating Connection with MySQL
- Creating API using Flask

app.py

```
from flask import Flask,jsonify
import mysql.connector


mydb =
mysql.connector.connect(host='localhost',user='root',password='sam',db='sc
hool')
cur=mydb.cursor()

app=Flask(__name__)


@app.route('/student') #retrive all data from student
http://127.0.0.1:5000/student
def get_student_info():

   '''
   query="Select sub.Sub_name form subjectAlloted subA join subject sub on
subA.subj_code= sub.sub_code where Teacher_ID=(Select Teacher_ID from
Teacher where Name=’name’)"
   query="SELECT std_name FROM student WHERE opt_sub1='Hindi' order by
section"
   query="SELECT teacher_name FROM teacher teach INNER JOIN subject_allot
sub on teach.teacher_id=sub.teacher_id where sub.section='C'"
   query="SELECT sub_code,sub_name FROM subject_ WHERE sub_code IN (SELECT
S.sub_code from Teacher te INNER JOIN Subject_allot S ON
te.teacher_id=S.teacher_id where te.teacher_name = 'Mr. X')"
   query="Select sub_code,sub_name from Subject_ where opt='Y'"
   '''
   query='SELECT * FROM student'

   result=cur.fetchall()
   for rec in result:
       print(rec)

   #suppose data from student Table
```

```
student_tab=[{"enr_no":"102","std_name":"Shubham","section":"A","opt_sub1":"Hindi","opt_sub2":"Eng"},

{"enr_no":"103","std_name":"Deepak","section":"B","opt_sub1":"Marathi","opt_sub2":"Eng"},

{"enr_no":"104","std_name":"Shashank","section":"C","opt_sub1":"Eng","opt_sub2":"Hindi"}]

   return jsonify({'student':student_tab})

app.run(port=5000)
```

Link: GitHub