

```
CREATE TABLE IF NOT EXISTS public.classe
(
    name character varying(5) PRIMARY KEY
);
```

```
INSERT INTO public.classe (name) VALUES
('DIA3'),
('DIA5');
```

```
CREATE TABLE IF NOT EXISTS public.cours
(
    title character varying(255) PRIMARY KEY,
    description text,
    start_date date,
    end_date date,
    number_hours integer NOT NULL
);
```

```
INSERT INTO public.cours (title, description, start_date, end_date, number_hours) VALUES
('Software Engineering', 'Le génie logiciel est l'étude des principes et pratiques de conception, développement, test et maintenance de logiciels de qualité.', '2023-09-02', '2023-12-22', 30),
('Mathematics For Deep Learning', 'Les fondements mathématiques essentiels à la compréhension et à la mise en œuvre de techniques d'apprentissage profond', '2023-09-02', '2023-12-22', 20) ;
```

```
CREATE SEQUENCE IF NOT EXISTS public.prof_id_seq
INCREMENT 1
START 1000
MINVALUE 1000
MAXVALUE 9999
CACHE 1;
```

```
CREATE TABLE IF NOT EXISTS public.prof
(
    email character varying(255) PRIMARY KEY,
    nom character varying(255) NOT NULL,
    prenom character varying(255) NOT NULL,
    cours_title character varying(255) REFERENCES public.cours(title)
);
```

```

INSERT INTO public.prof (email, nom, prenom, cours_title)
VALUES ('lisapignon@gmail.com', 'Pignon', 'Lisa', 'Software Engineering'),
('lamarmichel@gmail.com', 'Lamar', 'Michel', 'Software Engineering'),
('mariedurand@gmail.com', 'Durand', 'Marie', 'Mathematics For Deep Learning'),
('jeandupuis@gmail.com', 'Dupuis', 'Jean', 'Mathematics For Deep Learning');

```

```

CREATE SEQUENCE IF NOT EXISTS public.etudiants_id_seq
INCREMENT 1
START 70000
MINVALUE 70000
MAXVALUE 99999
CACHE 1;

```

```

CREATE OR REPLACE FUNCTION generate_random_password()
RETURNS text AS $$
DECLARE
    chars text[] :=
ARRAY['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u',
'v','w','x','y','z','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'
];
    password text := "";
    i integer;
BEGIN
    FOR i IN 1..10 LOOP
        password := password || chars[1 + floor(random() * 62)];
    END LOOP;
    RETURN password;
END;
$$ LANGUAGE plpgsql VOLATILE;

```

```

CREATE TABLE IF NOT EXISTS public.etudiants
(
    id integer NOT NULL DEFAULT nextval('public.etudiants_id_seq') PRIMARY KEY,
    nom character varying(255) NOT NULL,
    prenom character varying(255) NOT NULL,
    date_naissance date NOT NULL,
    ville_naissance character varying(255),
    adresse_postale character varying(255),
    numero_telephone character varying(15),
    contact_urgence_nom character varying(255),

```

```

contact_urgence_telephone character varying(15),
classe character varying(5) REFERENCES public.classe(name),
specialisation character varying(255),
mot_de_passe character varying(255) COLLATE pg_catalog."default" DEFAULT
generate_random_password()
);

```

```

INSERT INTO public.etudiants
(nom, prenom, date_naissance, ville_naissance, adresse_postale, numero_telephone,
contact_urgence_nom, contact_urgence_telephone, classe, specialisation) VALUES
('Martin', 'Clara', '2002-06-15', 'Marseille', '25 avenue du Prado, 13008 Marseille',
'0623456789', 'Martin', '0676543210', 'DIA5', 'DIA'),
('Lefebvre', 'Émilie', '2003-11-30', 'Bordeaux', '78 cours de l'Intendance, 33000 Bordeaux',
'0634567890', 'Lefebvre', '0687654321', 'DIA3', 'DIA'),
('Bernard', 'Julien', '2003-09-05', 'Nice', '14 boulevard Victor Hugo, 06000 Nice', '0645678901',
'Bernard', '0699876543', 'DIA3', 'DIA'),
('Dupont', 'Alexandre', '2001-04-23', 'Lyon', '3 rue de la République, 75001 Paris',
'0612345678', 'Dupont', '0698765432', 'DIA5', 'DIA') ;

```

```

DO $$
BEGIN
    IF NOT EXISTS (SELECT 1 FROM pg_sequences WHERE sequencename =
'numero_controle_seq') THEN
        CREATE SEQUENCE numero_controle_seq START 1;
    END IF;
END $$;

```

```

CREATE TABLE IF NOT EXISTS public.notes
(
    id serial PRIMARY KEY,
    etudiant_id integer,
    cours_title character varying(255) COLLATE pg_catalog."default",
    note numeric(5,2) NOT NULL,
    coefficient numeric(3,1) NOT NULL,
    numero_controle integer DEFAULT nextval('numero_controle_seq'::regclass),
    CONSTRAINT notes_note_check CHECK (note >= 0::numeric AND note <= 20::numeric),
    CONSTRAINT notes_unique UNIQUE (cours_title, numero_controle)
);

```

```

INSERT INTO public.notes (etudiant_id, cours_title, note, coefficient) VALUES
(70000, 'Software Engineering', 15, 3),
(70000, 'Software Engineering', 16, 1),
(70000, 'Software Engineering', 14, 1),

```

(70000, 'Mathematics For Deep Learning', 18, 2),  
(70000, 'Mathematics For Deep Learning', 13, 5),  
(70000, 'Mathematics For Deep Learning', 20, 1),  
(70001, 'Software Engineering', 11, 3),  
(70001, 'Software Engineering', 17, 1),  
(70001, 'Software Engineering', 9, 1),  
(70001, 'Mathematics For Deep Learning', 16, 2),  
(70001, 'Mathematics For Deep Learning', 16, 5),  
(70001, 'Mathematics For Deep Learning', 18, 1),  
(70002, 'Software Engineering', 19, 3),  
(70002, 'Software Engineering', 12, 1),  
(70002, 'Software Engineering', 15, 1),  
(70002, 'Mathematics For Deep Learning', 9, 2),  
(70002, 'Mathematics For Deep Learning', 13, 5),  
(70002, 'Mathematics For Deep Learning', 14, 1),  
(70003, 'Software Engineering', 15, 3),  
(70003, 'Software Engineering', 17, 1),  
(70003, 'Software Engineering', 20, 1),  
(70003, 'Mathematics For Deep Learning', 12, 2),  
(70003, 'Mathematics For Deep Learning', 12, 5),  
(70003, 'Mathematics For Deep Learning', 13, 1);

**CREATE TABLE** IF NOT EXISTS public.events

(  
    id serial PRIMARY KEY,  
    title character varying(255) NOT NULL,  
    start\_date date NOT NULL,  
    end\_date date NOT NULL,  
    description text  
);

**INSERT INTO** public.events (title, start\_date, end\_date, description) VALUES  
('Career Fair', '2024-01-22', '2024-01-26', 'Rencontrer des entreprises depuis votre ordinateur  
ou téléphone');

```
CREATE TABLE IF NOT EXISTS public.holidays
(
    name character varying(255) PRIMARY KEY,
    start_date date NOT NULL,
    end_date date NOT NULL
);
```

```
ALTER TABLE IF EXISTS public.holidays
    OWNER to postgres;
```

```
INSERT INTO public.holidays (name, start_date, end_date) VALUES
('Vacances de Noël', '2023-12-22', '2024-01-07');
```

```
CREATE TABLE IF NOT EXISTS public.pdf_documents (
    id serial PRIMARY KEY,
    etudiant_id integer REFERENCES public.etudiants(id),
    filename character varying(255),
    file_path character varying(255)
);
```

```
INSERT INTO public.pdf_documents (etudiant_id, filename, file_path) VALUES
(70000, 'certificat_scolarite.pdf', '/Users/anoukleyris/Documents/4ème année ESILV
/Software engineering/Projet/Documents élèves /certificat_scolarite.pdf');
```

```
CREATE TABLE IF NOT EXISTS public.avis_course
(
    id serial PRIMARY KEY,
    avis text NOT NULL,
    cours character varying(255) REFERENCES public.cours(title)
);
```

```
CREATE TABLE IF NOT EXISTS public.avis_prof
(
    id serial PRIMARY KEY,
    avis text NOT NULL,
    email character varying(255) REFERENCES public.prof(email)
);
```

```
CREATE TABLE IF NOT EXISTS public.certificats
(
    id SERIAL PRIMARY KEY,
    student_id INTEGER,
    filename VARCHAR(255),
    filepath VARCHAR(255),
    CONSTRAINT certificats_student_id_fkey FOREIGN KEY (student_id)
        REFERENCES public.etudiants (id)
        ON UPDATE NO ACTION
        ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS public.document_eleve
(
    id SERIAL PRIMARY KEY,
    student_id INTEGER,
    filename VARCHAR(255),
    filepath VARCHAR(255),
    CONSTRAINT document_eleve_id_fkey FOREIGN KEY (student_id)
        REFERENCES public.etudiants (id)
        ON UPDATE NO ACTION
        ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS public.parent
(
    student_id integer,
    nom character varying(255) NOT NULL,
```

```

    mot_de_passe character varying(255) COLLATE pg_catalog."default" DEFAULT
generate_random_password(),
    PRIMARY KEY (student_id),
    FOREIGN KEY (student_id) REFERENCES public.etudiants(id),
    FOREIGN KEY (nom) REFERENCES public.etudiants(contact_urgence_nom)
);

```

```

INSERT INTO public.parent (student_id, nom)
SELECT id, contact_urgence_nom
FROM public.etudiants;

```

```

CREATE TABLE IF NOT EXISTS public.absence_reports
(
    report_id SERIAL PRIMARY KEY,
    student_id integer,
    report_date date,
    absence_reason text,
    FOREIGN KEY (student_id) REFERENCES public.etudiants(id)
);

```

```

INSERT INTO public.absence_reports
(student_id, report_date, absence_reason) VALUES
(70000, '2023-09-18', 'Maladie'),
(70000, '2023-11-30', 'Maladie'),
(70002, '2023-10-21', 'Journée d'intégration')

```

```

INSERT INTO public.absence_reports
(student_id, report_date, absence_reason) VALUES
(70000, '2023-10-01', 'Journée d'intégration'),
(70000, '2023-12-21', ' ')

```

```

CREATE TABLE IF NOT EXISTS public.message_admin
(
    id serial PRIMARY KEY,
    message text COLLATE pg_catalog."default" NOT NULL,
    classe character varying(5),
    date_envoi timestamp DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (classe) REFERENCES public.classe (name) ON DELETE CASCADE
)

```