

Projet : Plateforme d'Optimisation des Emplois du Temps d'Examens Universitaires

Contexte et problématique

Dans une faculté de plus de 13 000 étudiants répartis sur 7 départements et plus de 200 offres de formation (6-9 modules par formation), l'élaboration manuelle des emplois du temps d'examens génère fréquemment des conflits : surcharge des amphithéâtres (capacités variables), salles limitées à 20 étudiants max (en période d'examen), chevauchements étudiants/professeurs, contraintes d'équipements. Ce projet vise à concevoir une base de données relationnelle couplée à un algorithme d'optimisation automatique pour générer des plannings optimaux en moins de 45 secondes (idéal).

Objectifs pédagogiques

- Maîtriser la modélisation relationnelle complexe avec contraintes multiples
- Implémenter des requêtes analytiques avancées (détection de conflits, optimisation).
- Optimiser les performances sur datasets volumineux (130k inscriptions estimées).
- Développer une interface web fonctionnelle pour démonstration.

Acteurs et fonctionnalités

Acteur	Fonctionnalités principales
Vice-doyen et doyen	Vue stratégique globale : occupation globale des amphithéâtres et salles, taux conflits par département, validation finale EDT, KPIs académiques (heures profs, taux salles utilisées, ...etc.)
Administrateur examens (service planification)	Génération automatique EDT, détection conflits, optimisation des ressources
Chef de département	Validation par département, statistiques, conflits par formation
Étudiants/Professeurs	Consultation planning personnalisé (filtrage département/formation)

Idée sur les tables principales à implémenter :

```
-- Structure adaptée à l'échelle réelle  
departements (id, nom);  
formations (id, nom, dept_id, nb_modules);  
etudiants (id, nom, prenom, formation_id, promo);  
modules (id, nom, credits, formation_id, pre_req_id);  
lieu_examen (id, nom, capacite, type, batiment);  
professeurs (id, nom, dept_id, specialite);  
inscriptions (etudiant_id, module_id, note);  
examens (id, module_id, prof_id, salle_id, date_heure, duree_minutes);
```

Remarque: la liste des tables n'est qu'un exemple, il faudra essayer d'adapter avec l'existant.

Contraintes critiques à modéliser

- Étudiants : Maximum 1 examen par jour
- Professeurs : Maximum 3 examens par jour
- Salles et amphis: Respect capacité réelle
- Priorités: Examens du département sont priorisés (un enseignant surveille les examens de son département en priorité)
- Tous les enseignants doivent avoir le même nombre de surveillances..
- ... etc.

Technologies

- SGBD : MySQL ou PostgreSQL
- Backend : Python
- Frontend : Streamlit + Bootstrap
- Optimisation : Procédures PL/pgSQL + index partiels

Livrables obligatoires (évaluation 20/20)

- Scripts SQL complets (création BD + requêtes utilisées dans le dashboard)
- Dataset réaliste (la base de données doit être alimentée pour pouvoir la tester)
- Prototype fonctionnel (génération EDT + interface multi-départements)
- Rapport technique (10-15 pages PDF) :
- Benchmarks performances (temps d'exécution des requêtes)

Ce que vous devez envoyer sur classroom

- Le lien youtube d'une vidéo expliquant votre solution (entre 5 et 10 mn).
- Le lien vers la plateforme développée (il faudra l'héberger en ligne.)

Informations importante

- Date de début du projet: 20/10/2025
- Date de remise des liens: 19/01/2026 23:59
- Travail par trinômes
- Aucun retard ne sera accepté (CC2=0)
- Tout plagiat sera sanctionné (CC2=0)

La procédure à suivre

1. Veuillez vous inscrire sur la liste disponible dans classroom (obligatoire)
2. Le chef du trinôme déposera les liens sur son compte.