

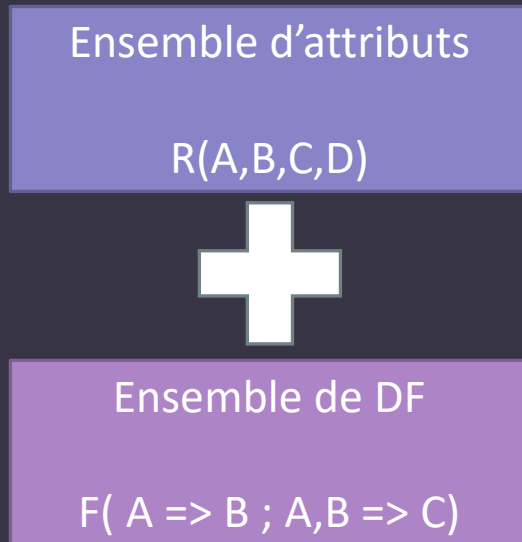
# Présentation de début de projet

---

SIMULATEUR DE DÉPENDANCES FONCTIONNELLES

# Concept

---



- Couverture Minimale
- Forme Normale
- Clé minimale
- Etc.

Compute Attribute Closure

Find Minimal Cover

Find Candidate Keys

3NF

BCNF

Chase Test

STUDY SECTION

Send feedback

Why this ad?

They were a regular

MINDOM LINES

EARLY BOOKING

-20%

VALUABLE FOUR LEGS LINED ADAPTATIONS

#EarlyBookingDiscount

### CalculateFDClosure(F)

Closure of a Set of Functional Dependencies Algorithm to compute the closure  $F^+$ , given  $F$

input : Set  $F$  of Functional Dependencies

output : Closure  $F^+$  of  $F$

### CalculateAttributeClosure(F, A)

The algorithm for calculating attributes closure  $A^+$  is based on the repeated use of the transitivity rule and has a runtime complexity that is quadratic in the size of  $F$

input : A set  $F$  of Functional Dependencies over a relation schema  $R$  and a set  $A \subseteq R$  of attributes

Griffith UNIVERSITY

Normalization Tool

☒ EDIT ATTRIBUTES

LEARNING RESOURCES

LOAD EXAMPLE

Functions

FIND A MINIMAL COVER

FIND ALL CANDIDATE KEYS

CHECK NORMAL FORM

NORMALIZE TO 2NF

NORMALIZE TO 3NF METHOD 1

NORMALIZE TO 3NF METHOD 2

NORMALIZE TO BCNF

About this tool

Attributes in Table

Separate attributes using a comma ( , )

a, b, c, d

Functional Dependencies

→

Delete

→

Delete

→

Delete

Add Another Dependency

Save This Table

Save this table to your PC and you can use it next time.

Relational Database Tools

Attribute closure calculator, Candidate key calculator, Minimum (Canonical) cover calculator, Functional dependency calculator and Normal form calculator

Currently supported functionality: (July 31, 2017)

- Correctly parses user input for relation schema, functional dependencies, and multivalued dependencies.
- Calculates the closure of each element in the power set of given attributes.
- Calculates minimum-set (candidate) keys and superkeys.
- Calculates a minimum (canonical) cover of functional dependencies on the given set of functional dependencies.
- Derives complete set of functional dependencies based on input set.
- Checks if given relation schema is in First (1NF), Second (2NF), Third (3NF), Boyce-Codd (BCNF), and Fourth (4NF) normal forms.
- If given relation schema is not in 3NF, will decompose input relation in a lossless and functional dependency preserving manner.
- If given relation schema is not in BCNF, will decompose input relation in a lossless but not necessarily functional dependency preserving manner.

To-do in the future:

- Improve output interface.
- If given relation schema is not in Fourth Normal form, decompose into 4NF relations.

Link to source code on GitHub

Anonymous report a bug

This is a personal project and not a professionally managed service. If you find a programming mistake please report it so it can be fixed.

Reset

Enter the relation schema in form R(A,B,C,AB,ABC)

Use commas to separate attributes. Spaces are optional. Inputs are case-insensitive.

Enter all given functional dependencies in form A -> B; AB -> C; B,C -> A

Use commas to separate attributes that belong to the same side of the same functional dependency.

Use semi-colons to separate different functional dependencies.

Enter all given multivalued dependencies in form A ->> B; AB ->> C; B,C ->> A (same as functional dependencies)

# Support

## APPLICATION WEB:

# Question: Comment faire tout cela ?

---

On dispose d'un objectif et d'un support

On doit donc découper ces deux éléments en bloc de tâches que l'on doit effectuer pour répondre aux critères

Avant cela on doit détailler un maximum pour déduire du sujet, une sorte de **Cahier des charges**

=> Ce cahier va être découpé en 3 types de charges: **Obligatoire**, **Intéressant**, **Optionnel** afin de représenter l'ensemble de ce que notre application doit pouvoir fournir

# Cahier des charges

---

- L'application permet de calculer une couverture minimale à partir d'un ensemble d'attributs et de DF (ADF)
- L'application permet de calculer la 1<sup>ère</sup> forme normale à partir ADF
- L'application permet de calculer la fermeture d'une clé
- L'application dispose d'explication pédagogique sur la méthodologie de calcul utilisée
- L'application est disponible en Français et en Anglais
- L'application dispose d'un mode « pro » qui calcul directement ce qui est demandé sans détaillé les calculs
- L'application dispose d'un mode « débutant »
- etc etc ceci n'est qu'un exemple le cahier complet et modifiable au fur et à mesure et disponible ici

# Organisation

---

Pour mener à bien ce projet on doit travailler de manière coordonnée et efficace

Pour cela: On a besoin: d'outils et d'attribuer des tâches

Outils obligatoires: Git (décentralisation du travail) / Overleaf (Latex)

Proposition d'outils:

- **Wampserver** (serveur local pour gérer la partie web et bdd)
- Trello (Gestion des tâches)/ Bitrix24 (Organisateur) => Globalement outils de gestion
- Support de communication (Améliorer discord ?)

# Attribution des tâches

---

Le cahier des charges comme dis précédemment pourra être découpé en blocs à ces bloc se rajoutera un bloc organisationnel (gestion du latex, architecture, modele EA )

Ainsi on pourrait découper tout cela en 3 blocs + 1:

- **Partie Algo/Fonctionnelle(backend)** (A): Fournir les algorithmes liés aux fonctionnalités du projet
- **Partie Design/Développement du site (B)**(frontend): Préparer le design du site prêt à accueillir les parties fonctionnelles(Html, Css)
- **Partie Bdd(backend)** ©: Création et gestion de la base de donnée
- **Partie Organisation et Conception (D)**: Gestion de l'architecture, modélisations et gestion du projet

# KIFEKOI ?

---

Objectif: Le projet doit être fait le plus rapidement possible pour pouvoir avoir de la marge

=> Maître mot: efficacité

On doit donc chercher à travailler en //

Plusieurs options:



# Travail en binôme

---

2x2 Binômes : 1 sur le bloc Algo et Orga (BA) et 1 sur le bloc Design et bdd (BB)

## Avantages:

- Support d'une seconde personne sur les tâches attribuer (en plus de moi)
- Avancement indépendant du 2ème binôme
- (BB) n'a pas à faire autre chose que du code

## Inconvénient:

- Avancement moins rapide si les 2 membres du binôme ne s'organisent pas en interne
- Enorme dépendance entre les 2 groupes, si l'un ne fait pas son travail, l'autre peut se retrouver rapidement bloquer

# Chacun son bloc

---

1 responsable par bloc, qui doit fournir à chaque réunion l'avancée sur son travail

## Avantage:

- Chacun travail dans son coin à son rythme
- Efficacité maximale (car tout le monde travail sur un axe différent en // )

## Inconvénient:

- S'écroule si retard trop important dans l'un des blocs
- La communication doit être parfaite => Réunions hebdomadaire obligatoire pour s'assurer de cela
- Demande une grande homogénéité dans les rendus

# Travail Mixte

---

1 Référent par bloc qui fait appel à des « assistants » pour l'assister dans ses tâches

## Avantage:

- Responsabilité de chacun (Car le bon fonctionnement de l'avancement du bloc dépend de votre gestion/ communication et capacité à anticiper les tâches couteuse en temps)
- Tout le monde touche à tout (Car tout le monde peut-être appelé par tout le monde pour demander de l'aide)

## Inconvénient:

- Demande une grande homogénéité des connaissances
- Tout s'effondre si un référent ne communique pas assez (retard + absence potentielle d'information sur les parties en court de développement)

# Votre solution ?

---

Si vous avez d'autres idées en tête je prend !

Sinon on doit faire un choix, pour cela:

On regarde:

- les compétences de chacun
- La motivation pour un bloc en particulier
- Les capacités (Ce que la personne sait faire et sa marge de progression)
- Avis du reste du groupe

# Détails des blocs

---

## Bloc Algo/Fonctionnalité (A)

Tout ce qui concerne la conception et l'efficacité des algorithmes utilisés pour réaliser les fonctionnalités.

Cette personne sera une sorte de vendeur, on lui demande un algo avec ce qu'il prend et ce qu'il sort et cette personne sera chargée de fournir cet algo

Technologies principales: php , Javascript , Python, SQL (collaboration avec bloc C)

(Sur le modèle vue-contrôleur) cette personne serai en charge de la partie contrôleur

# Détails des blocs

---

## **Bloc Design (Frontend):**

Bloc responsable de ce à quoi le site ressemblera, il faut de l'imagination et de l'organisation pour mettre en place des interfaces exploitables facilement dans lesquelles on n'aura juste à implémenter les algos pour rentrer les fonctionnalités

Utilisations d'outils type bootstrap/framework fortement recommandé pour gagner du temps et aller plus rapidement vers des détails design moderne/sympathique

Technologies principales: Html, CSS, Php, Javascript

Sur le MVC ce bloc gère la partie vue utilisateur

# Détails des blocs

---

## Bloc bdd:

Conception du modèle EA de l'application

Gestionnaire de la construction et des interactions entre l'application et la bdd, doit mettre à disposition les outils permettant les bons transferts de données au bon moment

Le bloc bdd est un bloc assez redondant, mais demandant beaucoup de rigueur

Technologies: SQL, php

Partie modèle du mvc

# Détail des blocs

---

## Bloc organisationnel et conception:

S'occupe dans un premier temps de faire les choix d'architecture de l'application, il est le garant du mvc et s'occupe de récupérer chaque parties du code et de les implémenter ensemble

S'occupe aussi du debug / vérification de l'application globale

S'occupe de la conception du modèle EA en collaboration avec le bloc bdd

Aussi, ce bloc est celui qui s'occupera du rapport (et des rapports intermédiaire de réunions)

Ce bloc gere globalement l'organisation et la vérification de l'avancement pour assurer la communication entre les différents blocs et définir les objectifs/besoins du projet à chaque étapes

Gestion du cahier des charges

Technologies: Php, sql, Latex, outils de gestion



# A faire aujourd'hui

---

Décider:

- Organisation du groupe
- Faire test maitrise git
- Faire test maitrise overleaf
- Définir date pour 1<sup>ère</sup> réunion avec professeur
- Idée de comment chacun imagine le projet

# Les étages du projet:

---

## **Travail à réaliser :**

Les différentes étapes pour mener à bien ce TER sont les suivantes :

- 1 – Réaliser une première application Web dans laquelle il est possible de pouvoir saisir les attributs et les dépendances fonctionnelles d'une relation R
- 2 – Intégrer dans la plateforme différentes fonctions : Calcul de la fermeture des dépendances fonctionnelles, calcul de la fermeture des attributs, calcul d'une couverture minimale, algorithme de synthèse, vérification des formes normales, transformation des relations pour respecter les formes normales, algorithmes de Chase, etc.
- 3 – Améliorer la plateforme pour qu'elle explique bien les étapes des différents algorithmes
- 4 – Génération du code SQL correspondant aux relations.
- 5 – En fonction du temps et de l'intérêt il est possible de s'intéresser à la partie modélisation à partir d'un schéma E/A, i.e. pouvoir dessiner le schéma E/A [3], générer les relations, vérifier qu'elles sont bien en 3NF/BCNF, etc.