

Plan

- 1 Introduction
- 2 *Patterns* et technologies
- 3 Implémentations
- 4 Récapitulatif et synthèse
- 5 Conclusion

2010年12月10日

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

- Étudier les technologies, les moteurs de requêtes et les *patterns* existants mettant à disposition un *event store*

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

- Étudier les technologies, les moteurs de requêtes et les *patterns* existants mettant à disposition un *event store*
- Proposer un *use case* et des implémentations selon les différents *patterns* et technologies

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

- Étudier les technologies, les moteurs de requêtes et les *patterns* existants mettant à disposition un *event store*
- Proposer un *use case* et des implémentations selon les différents *patterns* et technologies
- Montrer la transformation d'un modèle relationnel en un modèle orienté événements

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

- Étudier les technologies, les moteurs de requêtes et les *patterns* existants mettant à disposition un *event store*
- Proposer un *use case* et des implémentations selon les différents *patterns* et technologies
- Montrer la transformation d'un modèle relationnel en un modèle orienté événements
- Analyser la cohérence des données et les transactions qui permettent de garder une source de données en tant que source de confiance

Motivations

- Besoin de conserver l'historique des changements
- Modèle relationnel (seul) non adapté (CRUD "classique")

Buts

- Étudier les technologies, les moteurs de requêtes et les *patterns* existants mettant à disposition un *event store*
- Proposer un *use case* et des implémentations selon les différents *patterns* et technologies
- Montrer la transformation d'un modèle relationnel en un modèle orienté événements
- Analyser la cohérence des données et les transactions qui permettent de garder une source de données en tant que source de confiance
- Souligner les avantages et limitations des implémentations en les comparant

Relationnel, CDC et *Event Sourcing*

Relationnel, CDC et *Event Sourcing*

| NoEtu | Prénom | Nom | ISBN | Titre | Echéance | Thème |
|-------|--------|------------|---------------|-------------------------|------------|--------------|
| 1 | Bill | Torvalds | 2-234432-54-3 | Les Aqueducs | 18.06.1999 | Architecture |
| 2 | Linus | Gates | 2-234432-54-3 | Les Aqueducs | 23.04.2012 | Architecture |
| 2 | Linus | Gates | 1-657879-34-6 | Python pour les nuls | 19.07.2013 | Informatique |
| 3 | Mark | Jobs | 4-493828-43-2 | Base de données avancée | 01.05.2011 | Informatique |
| 4 | Steve | Zuckerberg | 8-098384-98-7 | Algèbre et géométrie | 05.12.2005 | Mathématique |
| 4 | Steve | Zuckerberg | 4-493828-43-2 | Base de données avancée | 06.08.2013 | Informatique |



| NoEtu | Prénom | Nom |
|-------|--------|------------|
| 1 | Bill | Torvalds |
| 2 | Linus | Gates |
| 3 | Mark | Jobs |
| 4 | Steve | Zuckerberg |

| ISBN | Titre | NoThm |
|---------------|-------------------------|-------|
| 2-234432-54-3 | Les Aqueducs | 1 |
| 1-657879-34-6 | Python pour les nuls | 2 |
| 4-493828-43-2 | Base de données avancée | 2 |
| 8-098384-98-7 | Algèbre et géométrie | 3 |

| NoThm | Thème |
|-------|--------------|
| 1 | Architecture |
| 2 | Informatique |
| 3 | Mathématique |

| Echéance | NoEtu | ISBN |
|------------|-------|---------------|
| 18.06.1999 | 1 | 2-234432-54-3 |
| 23.04.2012 | 2 | 2-234432-54-3 |
| 19.07.2013 | 2 | 1-657879-34-6 |
| 01.05.2011 | 3 | 4-493828-43-2 |
| 05.12.2005 | 4 | 2-234432-54-3 |
| 06.08.2013 | 4 | 4-493828-43-2 |

FIGURE – Séparation de l'information en plusieurs tables - Joël Cavat

Relationnel, CDC et *Event Sourcing*

| NoEtu | Prénom | Nom | ISBN | Titre | Echéance | Thème |
|-------|--------|------------|---------------|-------------------------|------------|--------------|
| 1 | Bill | Torvalds | 2-234432-54-3 | Les Aqueducs | 18.06.1999 | Architecture |
| 2 | Linus | Gates | 2-234432-54-3 | Les Aqueducs | 23.04.2012 | Architecture |
| 2 | Linus | Gates | 1-657879-34-6 | Python pour les nuls | 19.07.2013 | Informatique |
| 3 | Mark | Jobs | 4-493828-43-2 | Base de données avancée | 01.05.2011 | Informatique |
| 4 | Steve | Zuckerberg | 8-098384-98-7 | Algèbre et géométrie | 05.12.2005 | Mathématique |
| 4 | Steve | Zuckerberg | 4-493828-43-2 | Base de données avancée | 06.08.2013 | Informatique |



| NoEtu | Prénom | Nom |
|-------|--------|------------|
| 1 | Bill | Torvalds |
| 2 | Linus | Gates |
| 3 | Mark | Jobs |
| 4 | Steve | Zuckerberg |

| ISBN | Titre | NoThm |
|---------------|-------------------------|-------|
| 2-234432-54-3 | Les Aqueducs | 1 |
| 1-657879-34-6 | Python pour les nuls | 2 |
| 4-493828-43-2 | Base de données avancée | 2 |
| 8-098384-98-7 | Algèbre et géométrie | 3 |

| NoThm | Thème |
|-------|--------------|
| 1 | Architecture |
| 2 | Informatique |
| 3 | Mathématique |

| Echéance | NoEtu | ISBN |
|------------|-------|---------------|
| 18.06.1999 | 1 | 2-234432-54-3 |
| 23.04.2012 | 2 | 2-234432-54-3 |
| 19.07.2013 | 2 | 1-657879-34-6 |
| 01.05.2011 | 3 | 4-493828-43-2 |
| 05.12.2005 | 4 | 2-234432-54-3 |
| 06.08.2013 | 4 | 4-493828-43-2 |

Change Data Capture

- Détecter et capturer les changements survenus
- Timestamps, versions, statuts ou combinaison des trois
- Ajout de table par attribut changeant / à surveiller

FIGURE – Séparation de l'information en plusieurs tables - Joël Cavat

Relationnel, CDC et *Event Sourcing*

| NoEtu | Prénom | Nom | ISBN | Titre | Echéance | Thème |
|-------|--------|------------|---------------|-------------------------|------------|--------------|
| 1 | Bill | Torvalds | 2-234432-54-3 | Les Aqueducs | 18.06.1999 | Architecture |
| 2 | Linus | Gates | 2-234432-54-3 | Les Aqueducs | 23.04.2012 | Architecture |
| 2 | Linus | Gates | 1-657879-34-6 | Python pour les nuls | 19.07.2013 | Informatique |
| 3 | Mark | Jobs | 4-493828-43-2 | Base de données avancée | 01.05.2011 | Informatique |
| 4 | Steve | Zuckerberg | 8-098384-98-7 | Algèbre et géométrie | 05.12.2005 | Mathématique |
| 4 | Steve | Zuckerberg | 4-493828-43-2 | Base de données avancée | 06.08.2013 | Informatique |



| NoEtu | Prénom | Nom |
|-------|--------|------------|
| 1 | Bill | Torvalds |
| 2 | Linus | Gates |
| 3 | Mark | Jobs |
| 4 | Steve | Zuckerberg |

| ISBN | Titre | NoThm |
|---------------|-------------------------|-------|
| 2-234432-54-3 | Les Aqueducs | 1 |
| 1-657879-34-6 | Python pour les nuls | 2 |
| 4-493828-43-2 | Base de données avancée | 2 |
| 8-098384-98-7 | Algèbre et géométrie | 3 |

| NoThm | Thème |
|-------|--------------|
| 1 | Architecture |
| 2 | Informatique |
| 3 | Mathématique |

| Echéance | NoEtu | ISBN |
|------------|-------|---------------|
| 18.06.1999 | 1 | 2-234432-54-3 |
| 23.04.2012 | 2 | 2-234432-54-3 |
| 19.07.2013 | 2 | 1-657879-34-6 |
| 01.05.2011 | 3 | 4-493828-43-2 |
| 05.12.2005 | 4 | 2-234432-54-3 |
| 06.08.2013 | 4 | 4-493828-43-2 |

FIGURE – Séparation de l'information en plusieurs tables - Joël Cavat

Change Data Capture

- Détecter et capturer les changements survenus
- Timestamps, versions, statuts ou combinaison des trois
- Ajout de table par attribut changeant / à surveiller

Event Sourcing

- Séquence de changements d'états amenant au résultat actuel
- Suivre le chemin parcouru
- Exemple du compte en banque

Message broker et Kafka

Message broker et Kafka

- Unité de base : message (clé, valeur, timestamp, offset), format libre

Message broker et Kafka

- Unité de base : message (clé, valeur, timestamp, offset), format libre
- Persistance sur le disque, rétention par défaut "infinie"

Message broker et Kafka

- Unité de base : message (clé, valeur, timestamp, offset), format libre
- Persistance sur le disque, rétention par défaut "infinie"
- *Topics* : regroupement de messages selon un même thème

Message broker et Kafka

- Unité de base : message (clé, valeur, timestamp, offset), format libre
- Persistance sur le disque, rétention par défaut "infinie"
- *Topics* : regroupement de messages selon un même thème
- Producteurs : émissions de messages dans *topics* (envoi batch)

Debezium

- Plateforme de CDC

Debezium

- Plateforme de CDC
- Écoute sur MySQL (PostgreSQL / MongoDB) et publie dans Kafka

100

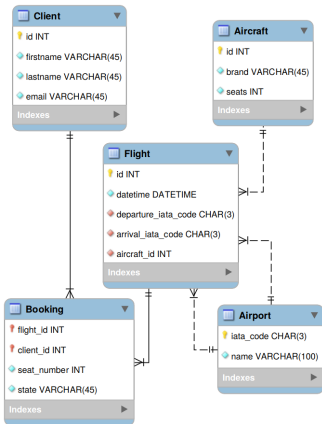
Debezium

- Plateforme de CDC
- Écoute sur MySQL (PostgreSQL / MongoDB) et publie dans Kafka
- Opérations **INSERT**, **UPDATE** ou **DELETE** surveillées
- *Matching* un-à-un entre une table MySQL et un *topic* Kafka

100

100

Bases communes

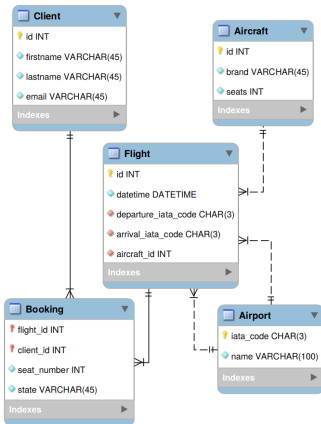


Hes-so
Master

Diagramme relationnel initial

Bases communes

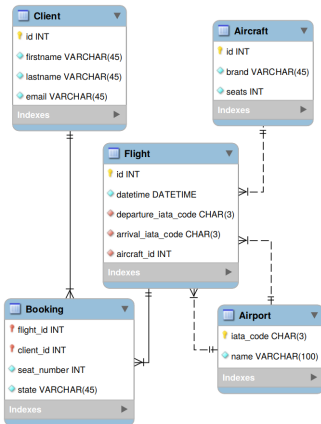
- Génération de données avec Mockaroo (+ Scala)



Hes-so
Master

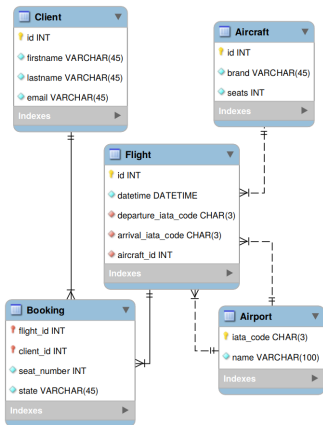
Diagramme relationnel initial

Bases communes



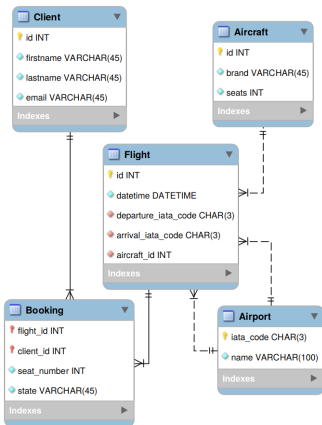
- Génération de données avec Mockaroo (+ Scala)
- Contraintes :
 - Un siège ne peut être attribué qu'à un seul client sur un vol
 - Un client ne peut avoir qu'un seul siège sur un vol
 - $aeroport_destination \neq aeroport_depart$
 - $places_vol = f(avion)$

Bases communes



- Génération de données avec Mockaroo (+ Scala)
- Contraintes :
 - Un siège ne peut être attribué qu'à un seul client sur un vol
 - Un client ne peut avoir qu'un seul siège sur un vol
 - $aeroport_destination \neq aeroport_depart$
 - $places_vol = f(avion)$
- Contrat (**trait**) **FlyManager**

Bases communes

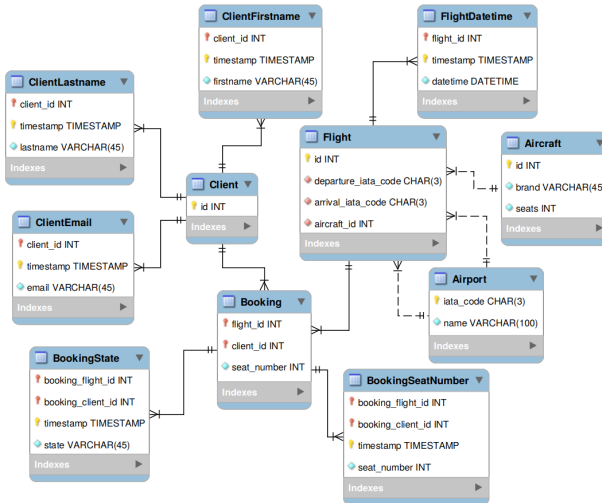


- Génération de données avec Mockaroo (+ Scala)
- Contraintes :
 - Un siège ne peut être attribué qu'à un seul client sur un vol
 - Un client ne peut avoir qu'un seul siège sur un vol
 - $aeroport_destination \neq aeroport_depart$
 - $places_vol = f(avion)$
- Contrat (**trait** FlyManager
- *History et Status

Hes-so
Master

Diagramme relationnel initial

MySQL *full*(1)



MySQL *full*(2)

MySQL *full*(2)

- Trois fonctions de base (`command`, `queryToList` et `query`)
- *Wrapper* autour de déclarations SQL

MySQL *full*(2)

- Trois fonctions de base (`command`, `queryToList` et `query`)
- *Wrapper* autour de déclarations SQL
- Transactions SQL pour certaines fonctions (`update` et `delete`)
- Problème avec requêtes retournant le dernier état d'un enregistrement : **ORDER BY** avant **GROUP BY** interdit

MySQL *full*(2)

- Trois fonctions de base (`command`, `queryToList` et `query`)
- *Wrapper* autour de déclarations SQL
- Transactions SQL pour certaines fonctions (`update` et `delete`)
- Problème avec requêtes retournant le dernier état d'un enregistrement : **ORDER BY** avant **GROUP BY** interdit
- Solution : sous-requête + `ROW_NUMBER()` (fonction *window*)

MySQL *full*(2)

- Trois fonctions de base (`command`, `queryToList` et `query`)
- *Wrapper* autour de déclarations SQL
- Transactions SQL pour certaines fonctions (`update` et `delete`)
- Problème avec requêtes retournant le dernier état d'un enregistrement : **ORDER BY** avant **GROUP BY** interdit
- Solution : sous-requête + `ROW_NUMBER()` (fonction *window*)
- **Défauts critiques (en l'état) : perte (définitive) de l'information en cas de suppression d'enregistrements**

Kafka *full*

Kafka *full*

- À l'opposé de MySQL *full*, message comme unité de base, doit indiquer son opération, un *topic* par entité

Kafka *full*

- À l'opposé de MySQL *full*, message comme unité de base, doit indiquer son opération, un *topic* par entité
- *Event Sourcing* par défaut, rejouer l'historique

Kafka *full*

- À l'opposé de MySQL *full*, message comme unité de base, doit indiquer son opération, un *topic* par entité
- *Event Sourcing* par défaut, rejouer l'historique
- "Il faut tout faire soi-même"

Kafka *full*

- À l'opposé de MySQL *full*, message comme unité de base, doit indiquer son opération, un *topic* par entité
- *Event Sourcing* par défaut, rejouer l'historique
- "Il faut tout faire soi-même"
- *Thread* consommateur à l'écoute des *topics*, désérialisation + insertion dans tables associatives (**Map**[K, V]) (exist + perms)
- Commandes : check **Map**[K, V] + sérialisation + envoi de message
- Requêtes "simples" : *lookup* dans la **Map**[K, V] correspondante
- Requêtes "complexes" : pseudo **JOIN** ou **GROUP BY** => usage des fonctions Scala standard sur les collections (*map*, *filter*, etc.)
- **Défauts critiques (en l'état) : concurrence lors des commandes, pas de transactions**

MySQL + Debezium

MySQL + Debezium

- "Mix" des deux implémentations précédentes
- Modèle relationnel redevient "simple", dernier état dans MySQL et historique dans Kafka

MySQL + Debezium

- "Mix" des deux implémentations précédentes
- Modèle relationnel redevient "simple", dernier état dans MySQL et historique dans Kafka
- Mécanisme d'écoute Kafka comme Kafka *full*, format JSON

MySQL + Debezium

- "Mix" des deux implémentations précédentes
- Modèle relationnel redevient "simple", dernier état dans MySQL et historique dans Kafka
- Mécanisme d'écoute Kafka comme Kafka *full*, format JSON
- Fonctions retournant un **Status** ou un historique fonctionnelles : *lookup* MySQL + *lookup* dans le *topic*
- **Meilleur des deux mondes : contraintes + historique**

Démo

Récapitulatif et synthèse

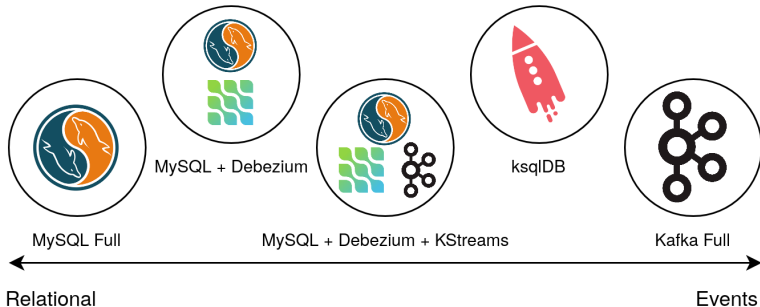


FIGURE – Distribution des implémentations, selon les technologies et *patterns*

100

Bilan

- Étude de différentes solutions de bases de données orientées événements

Bilan

- Étude de différentes solutions de bases de données orientées événements
- Choix et implémentations d'un *use case*

Bilan

- Étude de différentes solutions de bases de données orientées événements
- Choix et implémentations d'un *use case*
- Comparaisons avec le modèle relationnel

Améliorations possibles

Améliorations possibles

- Kafka Streams
- "Vrai" simulateur / API => comparaisons sur les performances

Améliorations possibles

- Kafka Streams
- "Vrai" simulateur / API => comparaisons sur les performances
- Approfondir les fonctionnalités avancées de Kafka (transactions) et appliquer des *patterns* de systèmes distribués

Merci pour votre attention ! Questions ?