

CONCEPTION DE L'ARCHITECTURE DE BASE DE DONNEES

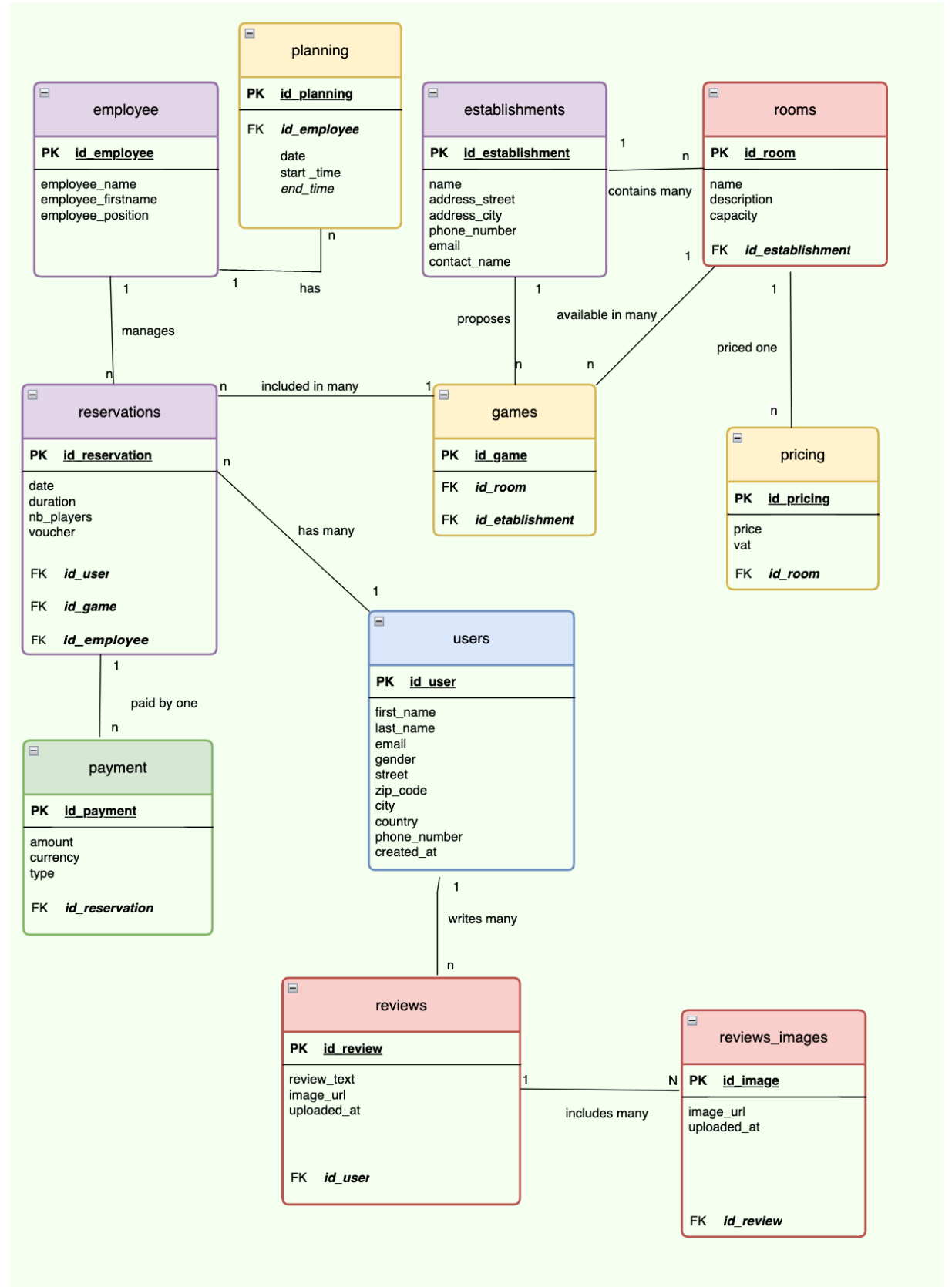
Rédigé par :

- Rami AGGOUN
- Victorien CROTTE-LAMETRIE
- Providence Hélène MABIALA POATY
- Marie Claudelle YIMGA TCHANGO

TABLE DES MATIERES

ARCHITECTURE DE BASE DE DONNEES PROPOSEES	3
Schéma entité relation (ER)	4
Normalisation (3FN)	5
Justifications des Choix d'Architecture	6
Justifications des Choix d'Index	7
ARCHITECTURE DE PRODUCTION.....	8
Proposition.....	8
Evolutivité :	8
Sécurité de l'infrastructure :	9
Sauvegarde et haute disponibilité de la donnée :	9

ARCHITECTURE DE BASE DE DONNEES PROPOSEES



SCHEMA ENTITE RELATION (ER)

Notre Modèle Conceptuel de Données (MCD) ci-joint est conçu pour structurer de manière optimale le système de réservation et de gestion de VDM Escape Game. Il articule les entités clés et leurs interactions pour refléter fidèlement les opérations et les besoins de l'entreprise, tout en posant les bases pour une évolutivité future.

Explication du MCD (Modèle Conceptuelle de données) :

- La table Employee (Employé) : Détient les informations sur les employés, qui peuvent avoir plusieurs plannings (planning) attribués. Un employé gère également des réservations (reservations).
- La table Planning : Représente les plannings attribués à chaque employé. Un employé peut avoir plusieurs plannings.
- La table Establishments (Établissements) : Contient les détails des établissements, qui peuvent contenir plusieurs salles (rooms).
- La table Rooms (Salles) : Détient les informations sur les salles disponibles dans un établissement. Chaque salle a une tarification (pricing) associée.
- La table Games (Jeux) : Représente les jeux ou activités proposés. Chaque jeu est lié à une salle et à un établissement.
- La table Reservations (Réservations) : Enregistre les réservations faites par les utilisateurs (users). Chaque réservation est liée à un utilisateur, un jeu et un employé, et elle est payée via un paiement (payment).
- La table Users (Utilisateurs) : Stocke les informations sur les utilisateurs, qui peuvent écrire des avis (reviews) et effectuer des réservations.
- La table Reviews (Avis) : Contient les avis écrits par les utilisateurs, qui peuvent inclure des images (reviews_images).
- La table Reviews_Images (Images des avis) : Stocke les images associées aux avis.
- La table Pricing (Tarification) : Représente les informations de tarification pour les salles.
- La table Payment (Paiement) : Enregistre les détails des paiements effectués pour les réservations.

Dans le cadre du développement d'un système de réservation et de gestion pour VDM Escape Game, une attention particulière a été accordée à la normalisation de la base de données. La normalisation est un processus essentiel pour minimiser la redondance des données et optimiser la cohérence des informations. Elle se décompose en plusieurs formes normales, chacune visant à résoudre des problèmes spécifiques et à améliorer la structure des données.

- **1^{ère} Forme Normale (1NF) :**
Élimination des valeurs répétitives : En assurant que chaque table a une clé primaire (comme « id_employee », « id_establishment », « id_room », etc.) et que les valeurs dans chaque colonne sont atomiques, on respecte la 1NF. Cela aide à éliminer les groupes répétitifs et permet à chaque enregistrement d'être unique.
- **2^{ème} Forme Normale (2NF) :**
Suppression des dépendances partielles : On atteint la 2NF en s'assurant que chaque attribut non-clé est pleinement fonctionnellement dépendant de la clé primaire. Cela évite les redondances, comme on le voit avec la séparation des entités « rooms » et « pricing ». La tarification est séparée car elle dépend uniquement de la salle « id_room », pas de l'ensemble de l'entité « establishments ».
- **3^e Forme Normale (3NF) :**
Élimination des dépendances transitives : En séparant les données avec des dépendances transitives dans des tables distinctes, on garantit que les modifications dans une table ne nécessitent pas de changements en cascade dans d'autres. Par exemple, les informations utilisateur « users » sont séparées des avis « reviews », car les avis dépendent de l'utilisateur mais ne sont pas une caractéristique intrinsèque de celui-ci.
- **Forme de Boyce-Codd (BCNF)**
Résolution des anomalies de mise à jour : Lorsque chaque déterminant est une clé candidate, la table est en BCNF. Notre schéma semble respecter la BCNF, car toutes les relations sont structurées de manière à ce que les clés étrangères pointent vers des clés primaires complètes.

Grâce à la normalisation, notre modèle de données offre maintenant une structure solide qui facilite la maintenance, améliore les performances des requêtes et garantit que les données reflètent fidèlement la réalité opérationnelle de VDM Escape Game. Elle permet également d'assurer la scalabilité du système pour accompagner la croissance future de l'entreprise.

JUSTIFICATIONS DES CHOIX D'ARCHITECTURE

○ Clés Primaires et Étrangères

Chaque table a une clé primaire unique, telle que « id_employee », « id_establishment », « id_room », qui identifie de manière unique chaque enregistrement. Cela garantit l'intégrité des données en empêchant les doublons et facilite la recherche rapide d'enregistrements spécifiques.

Les clés étrangères établissent des relations entre les tables. Par exemple, « FK_id_establishment » dans la table « rooms » crée un lien explicite entre une salle et son établissement. Cela permet de s'assurer que les données sont cohérentes à travers la base de données et soutient l'intégrité référentielle, empêchant les orphelins et les données incohérentes.

○ Relations entre les entités

La relation entre « establishments » et « rooms » illustre une structure hiérarchique où un établissement peut contenir plusieurs salles, mais chaque salle appartient à un seul établissement. Cela reflète une structure organisationnelle réelle et simplifie la gestion des salles.

La séparation des informations de tarification dans la table « pricing » réduit la redondance des données. Si le prix d'une salle devait être stocké dans la table « rooms », chaque changement de prix nécessiterait de multiples mises à jour. Avec une table de tarification distincte, on change simplement le prix à un endroit, préservant l'intégrité des données et facilitant la maintenance.

○ Justification des tables associatives

La table « reservations » agit comme une table associative entre « users » et « games », permettant aux utilisateurs de réserver des jeux spécifiques. Cette conception normalisée permet une grande flexibilité et évite la redondance des informations utilisateur dans la table de réservation.

De même, la séparation des avis (reviews) et des images associées (reviews_images) en deux tables distinctes permet aux utilisateurs de soumettre plusieurs images pour un seul avis, sans dupliquer l'avis pour chaque image.

○ Indexation pour les performances

L'indexation des colonnes qui sont souvent utilisées dans les requêtes, comme la date dans les réservations, permet d'accélérer considérablement les opérations de recherche et de tri.

L'indexation des clés étrangères améliore les performances des jointures entre les tables, ce qui est essentiel pour les opérations qui traversent plusieurs tables.

JUSTIFICATIONS DES CHOIX D'INDEX

- Index sur les clés Primaires et Étrangères

Clés Primaires : L'indexation automatique des clés primaires permet des recherches rapides et une identification unique des enregistrements.

Clés Étrangères : Indexer les clés étrangères accélère les jointures entre les tables et aide à maintenir l'intégrité référentielle.

- Indexation basée sur les requêtes

Colonnes Fréquemment Interrogées : Indexer les colonnes souvent utilisées dans les requêtes (WHERE, ORDER BY) comme les dates dans les réservations ou les noms dans les établissements permet des filtrages et des tris plus rapides.

Index Composites : Pour les requêtes impliquant plusieurs colonnes, un index composite peut être créé pour améliorer les performances de ces requêtes multicritères.

- Gestion de l'Overhead d'indexation :

Équilibre Lecture-Écriture : Trop d'index peuvent ralentir les opérations d'écriture, il est donc important de les utiliser judicieusement, en les limitant aux colonnes essentielles.

Maintenance des Index : Les index doivent être entretenus pour prévenir la dégradation des performances, surtout quand les données évoluent fréquemment.

ARCHITECTURE DE PRODUCTION

PROPOSITION

Nous proposons d'héberger la base de données chez **CloudSQL**, une solution haute performance servit par Google, permettant de gérer des bases de données relationnelles tel que MySQL et PostgreSQL ou SQL Server.

Chaque local VDM Escape Game possédera sa propre instance CloudSQL dédiée pour convenir aux demandes dans plusieurs pays.

Les backups seront pris en charge automatiquement par les services CloudSQL.

En ce qui concerne les ressources matérielles, il existe un choix de configuration machine en fonction de la tarification.

EVOLUTIVITE :

La tarification varie et s'applique selon :

- Le nombre de processeurs et le volume mémoire demandé
- Le volume de stockage et la mise en réseau
- Le nombre d'instances

Nombre de processeurs max : 96

Mémoire RAM max : 624 Go

Volume de stockage Max : 10 To

Nombre d'instances : illimité

Le montant sera fixé en fonction du choix de la configuration de la machine faite sur mesure.

Possibilité de faire évoluer la configuration machine à n'importe quel moment selon les besoins du client.

SECURITE DE L'INFRASTRUCTURE :

Les services CloudSQL sont hautement protégés nativement.

Ils proposent toutefois d'appliquer des règles de sécurités supplémentaires tel que :

- Adressage des ips autorisées à communiquer avec les bases de données
- Authentification aux instances ou bases de données par rôle

SAUVEGARDE ET HAUTE DISPONIBILITE DE LA DONNEE :

Possibilité de mettre en place une duplication en temps réel des instances hébergées. Les données seront alors répliquées de manière quasi instantanée, ce qui permettra de décharger le trafic de requêtes de lecture ou de trafic d'analyse.

Cette fonctionnalité peut aider également pour la reprise de service après sinistre.

La réplication des instances peut être déplacée dans différentes zones géographiques.