

UNIVERSITÉ DE MONTRÉAL

PROJET 9 : GESTION D'UN CLUB VIDÉO

PAR

JOANIE BIRTZ (20244554)

JANIC FOURNEL (20201900)

KAREN-FAE LAURIN (20019033)

CYNTHIA-ALEXANDRA VARGAS (20030603)

BACCALAURÉAT EN INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

TRAVAIL PRÉSENTÉ À JIHEN REZGUI
DANS LE CADRE DU COURS IFT-2935-B
BASES DE DONNÉES

10 AVRIL 2024

TABLE DES MATIÈRES

1. DESCRIPTION DU PROJET	3
2. MODÈLE ENTITÉ-ASSOCIATION	4
3. MODÈLE RELATIONNEL	7
4. NORMALISATION	8
1 ^{ère} forme normale (1FN)	8
Dépendances fonctionnelles	8
2 ^e forme normale (2FN)	9
3 ^e forme normale (3FN)	9
5. ENSEMBLE DES REQUÊTES SQL	11
1. Donner tous les genres des vidéos louer par un client.	11
2. Donner les genres préférées des clients.	11
3. Quel sont les sujets de tous les vidéos d'un certain acteur?	12
4. Donner le nom des réalisateurs et des acteurs du vidéo le plus louer.	12
5. Donner l'id, le titre, la date de sortie et l'état des vidéos ayant un sujet, un genre et un acteur en particulier.	13
6. Quels sont les vidéos archivés qui sont sortis avant 1995?	13
7. Donner les événements qui auront lieu entre le 15 avril 2024 et le 15 mai 2024.	14
8. Donner le nom du genre, le sujet et le titre du vidéo le moins louer (tous indépendant).	15
9. Quels sont les nouveautés vidéos qui ont été ajoutés dans le dernier mois?	15
10. Donner tous les informations d'un client selon son courriel.	15
11. Donner tous les informations d'un administrateur selon son courriel.	16
12. Donner le nombre de client qui ont un abonnement.	16
6. APPLICATION EN SQL ET CAPTURES D'ÉCRAN	17

1. DESCRIPTION DU PROJET

Notre projet est de mettre en place un centre de distribution de vidéos en ligne. Il sera donc possible de louer toutes sortes de contenus vidéos (films, séries, vidéos amateurs, etc.) pour une durée maximale de 48h pour chaque location. Le centre dispose d'un catalogue de films qui est mis à jour régulièrement par l'ajout de nouveauté et par l'archivage. Les clients pourront rechercher des films selon le titre, le réalisateur, les acteurs, le genre ou le sujet. Le contenu de location sera distribué aux clients seulement par téléchargement et ne sera plus accessible une fois le délai de location terminé. Ils pourront toujours le retélécharger s'ils n'ont pas eu le temps de terminer le visionnement, mais ils devront payer à nouveau les frais de location. Pour récompenser les clients fréquents, le centre de distribution offre un abonnement qui permettra aux abonnés de louer gratuitement certains contenus sélectionnés par le centre. Les abonnés recevront aussi des renseignements exclusifs sur les nouveautés, les projections et les expositions organisées par le centre de distribution. Tous les clients pourront participer aux événements organisés par le centre. Les clients et les administrateurs du centre de distribution de vidéos pourront se connecter à leur compte grâce à leur adresse courriel et à un mot de passe. Les comptes des administrateurs ne permettent pas de louer du contenu vidéo. Si les administrateurs du centre de distribution veulent louer du contenu vidéo, ils devront se créer un compte client en utilisant un courriel différent de celui de leur compte administrateur.

2. MODÈLE ENTITÉ-ASSOCIATION

En nous basant sur la description du projet faite dans la section précédente, nous avons fait le schéma entité-association (E-A) ci-dessous. Quelques explications sur notre modèle suivront le schéma.

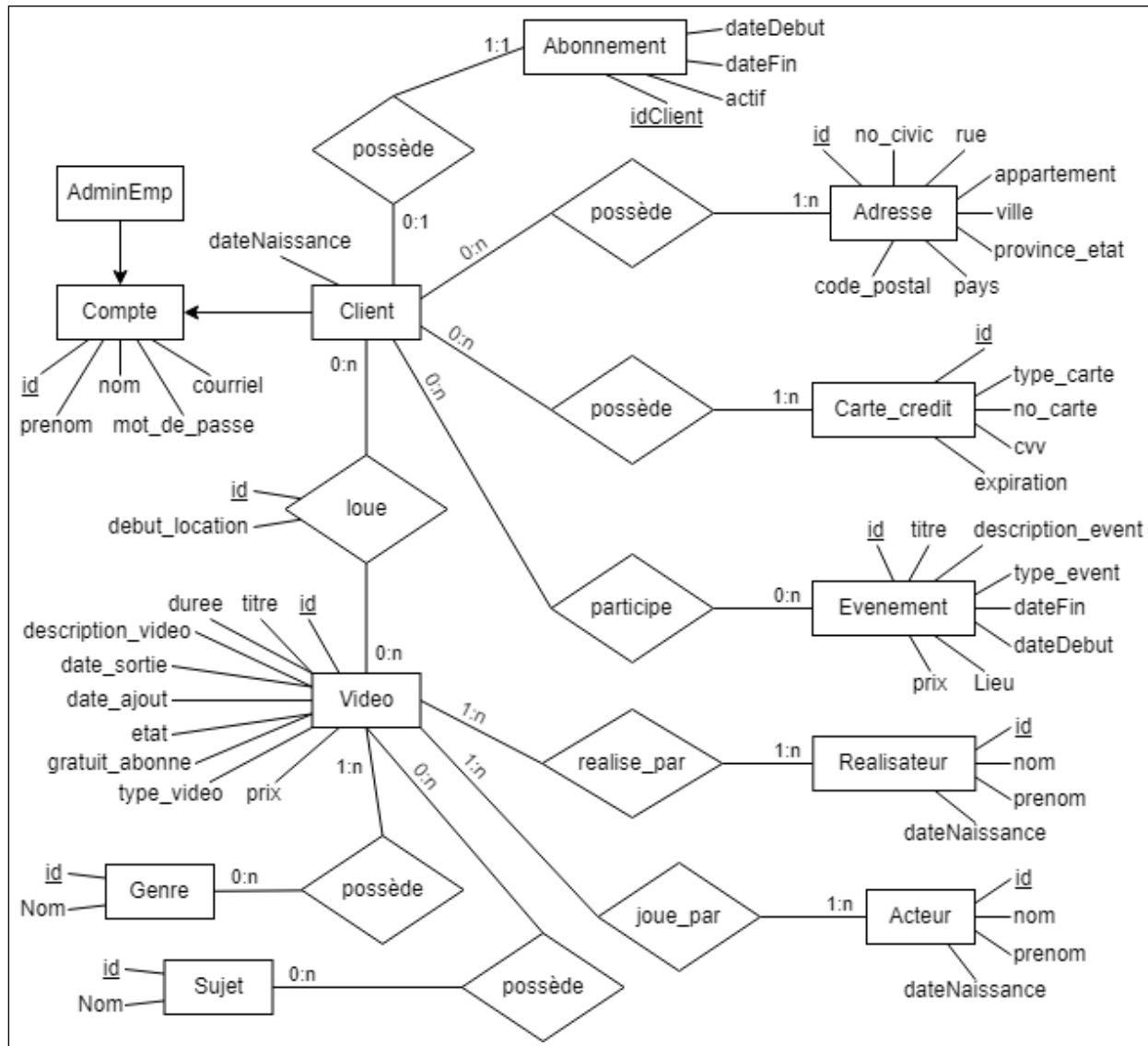


Figure 1 : Schéma Entité-Association

Dans l'entité **Compte**, nous avons mis comme attributs *nom* et *prenom* pour permettre l'identification du propriétaire du compte et *courriel* et *mot_de_passe* pour la connexion au compte.

Nous avons défini l'entité **Client** pour représenter tous les comptes des clients qu'ils aient un abonnement ou non. Nous avons ajouté un attribut date de naissance (*dateNaissance*), car nous acceptons qu'un mineur ait un compte, mais il n'aura pas accès aux contenus adultes. Pour les informations de paiement, nous avons créé les entités **Carte_crédit** et **Adresse**. Les clients

possèdent $0:n$ carte de crédit puisqu'il est possible de n'avoir aucun mode de paiement enregistré dans le compte, mais il est aussi possible d'avoir plusieurs modes de paiement enregistrés. Puisqu'il est possible d'avoir plusieurs cartes de crédit, il est aussi possible d'avoir $0:n$ adresses associées. Les clients peuvent participer à $0:n$ événements, puisqu'ils ne sont pas obligés de participer et ils peuvent participer à tous les événements qu'ils souhaitent. La même chose est vraie pour la location de vidéo, ils peuvent louer plusieurs vidéos ou aucune vidéo. Les clients possèdent $0:1$ abonnement, car ils peuvent avoir un seul abonnement ou ne pas en avoir.

Pour l'entité **Abonnement**, les attributs sont la date de début de l'abonnement (*dateDebut*), la date de fin (*dateFin*) et si l'abonnement est actif ou non (*actif*). L'attribut *actif* est un bit qui est 0 si l'abonnement n'est pas actif sinon 1. L'ajout de la date de fin de l'abonnement permet au client de résilier son abonnement à n'importe quel moment sans perdre accès au contenu jusqu'à la date de fin de son abonnement. Un abonnement peut être associé seulement à $1:1$ clients puisqu'il n'est pas possible pour le même abonnement d'être associée à plus qu'un compte client.

Pour l'entité **Carte_crédit**, les attributs sont les normes donc le numéro de carte (*no_carte*), le CVV et la date d'expiration (*expiration*). Nous avons ajouté l'attribut *type_carte* puisque nous permettons seulement l'utilisation de certaines cartes. Une carte de crédit peut être associée à $1:n$ clients, car si la carte n'est pas liée à au moins un compte, nous n'allons pas la conserver dans notre base de données, mais elle peut aussi être utilisée par plusieurs comptes, par exemple par les membres d'une même famille qui ont chacun un compte, mais utilise les mêmes informations de paiement. Les mêmes raisons sont valables pour l'adresse qui peut être associée à $1:n$ clients.

L'entité **Evenement** à plusieurs attributs pour décrire l'événement comme un titre, une description (*description_event*), une date de début et de fin (*dateDebut* et *dateFin*), le lieu et le prix. Nous avons aussi ajouté un attribut *type_event* pour que les clients puissent plus facilement savoir de quel type d'événement il s'agit. Chaque événement peut avoir de $0:n$ clients puisqu'il est possible que personne ne participe.

Pour l'association « loue », nous avons ajouté la date du début de la location (*debut_location*) pour pouvoir respecter la période autorisée de location de 48h.

Pour l'entité **Video**, nous avons utilisé des attributs assez typiques pour la description d'un matériel vidéo comme le titre, la durée, une description (*description_video*), la date de sortie (*date_sortie*) et le prix. Pour pouvoir savoir si le film est une nouveauté sur la plateforme, nous avons ajouté l'attribut date d'ajout (*date_ajout*). Nous utilisons l'attribut *etat* pour pouvoir rapidement voir si la vidéo est une nouveauté, archivé ou normal (état par défaut lorsqu'une vidéo est sur la plateforme, mais qu'elle n'est pas archivée, ni nouvelle). L'ajout de l'attribut *type_video* permet de savoir si le contenu vidéo est un film, une série, etc. L'attribut *gratuit_abonne* nous permet d'identifier les vidéos qui sont gratuites pour les clients qui sont abonnés. Les vidéos peuvent être loués par $0:n$ clients puisqu'il est possible que personne ne loue certains contenus. Les vidéos possèdent $1:n$ genres, acteurs et réalisateurs, car il n'est pas possible, selon nous, qu'il n'y ait aucun genre, acteur

ou réalisateur pour les vidéos. Cependant, il est possible qu'une vidéo n'ait pas de sujet, c'est donc pour cela que les vidéos ont $0:n$ sujets.

Pour les entités **Genre** et **Sujet**, nous avons simplement identifié ceux-ci par un nom et ils peuvent être possédés par $0:n$ vidéo puisqu'il est possible qu'il n'y ait aucune vidéo sur la plateforme de ce genre ou sujet. Le genre fait référence au type typique de vidéo comme horreur, comédie, action, etc. Tandis que les sujets sont plus précis, par exemple, apocalypse de zombie, 2^e guerre mondiale, catastrophe naturelle, etc.

Pour les entités **Réalisateur** et **Acteur**, les attributs sont le prénom, le nom et la date de naissance (*dateNaissance*). Nous avons ajouté la date de naissance comme attribut pour pouvoir différencier deux acteurs ou deux réalisateurs qui auraient le même nom et prénom. Les réalisateurs peuvent avoir réalisé et les acteurs peuvent avoir joué dans $1:n$ vidéo, car selon nous, ils doivent être associés à au moins une vidéo.

3. MODÈLE RELATIONNEL

En se basant sur notre modèle entité-association présenté dans la section précédente, nous avons fait le modèle relationnel suivant.

AdminEmp(id, prenom, nom, courriel, mot_de_passe)

Client(id, prenom, nom, courriel, mot_de_passe, dateNaissance)

Abonnement(idClient, actif, dateDebut, dateFin)

Adresse(id, no_civic, rue, appartement, ville, province_etat, pays, code_postal)

Carte_credit(id, type_carte, no_carte, cvv, expiration)

Evenement(id, type_event, titre, description_event, DateFin, DateDebut, Lieu, prix)

ClientAdresse(#idClient, #idAdresse)

ClientCarte_credit(#idClient, #idCarte)

ClientEvenement(#idClient, #idEvenement)

Video(id, titre, duree, description_video, date_sortie, date_ajout, etat, gratuit_abonne, prix, type_video)

VideoClient(id, #idClient, #idVideo, debut_location)

Realisateur(id, nom, prenom, dateNaissance)

Acteur(id, nom, prenom, dateNaissance)

Genre(id, Nom)

Sujet(id, Nom)

VideoRealisateur(#idVideo, #idRealisateur)

VideoActeur(#idVideo, #idActeur)

VideoGenre(#idVideo, #idGenre)

VideoSujet(#idVideo, #idSujet)

4. NORMALISATION

1^{ère} forme normale (1FN)

Pour qu'une relation soit en première forme normale (1FN), il faut que tous ses attributs soient atomiques. En observant notre modèle relationnel et son schéma entité-association, il est possible de constater que tous les attributs de toutes nos relations sont des attributs simples. On peut conclure que toutes les relations de notre modèle relationnel sont en 1FN.

Dépendances fonctionnelles

Avant de vérifier si notre modèle relationnel est en 2^e et 3^e formes normales, il faut définir toutes les dépendances fonctionnelles pour chaque relation de notre modèle relationnel. Voici les dépendances fonctionnelles que nous avons ressorties de notre modèle relationnel selon les différentes relations.

- AdminEmp
 - $id \rightarrow prenom, nom, courriel, mot_de_passe$
 - $courriel \rightarrow id, prenom, nom, mot_de_passe$
- Client
 - $id \rightarrow prenom, nom, courriel, mot_de_passe, dateNaissance$
 - $courriel \rightarrow id, prenom, nom, mot_de_passe, dateNaissance$
- Abonnement
 - $idClient \rightarrow actif, dateDebut, dateFin$
- Adresse
 - $id \rightarrow no_civic, rue, appartement, ville, province_etat, pays, code_postal$
- Carte_credit
 - $id \rightarrow type_carte, no_carte, cvv, expiration$
- Evenement
 - $id \rightarrow type_event, titre, description_event, DateFin, DateDebut, Lieu, prix$
- Video
 - $id \rightarrow \{ titre, duree, description_video, date_sortie, date_ajout, etat, gratuit_abonne, prix, type_video \}$
 - $\{ titre, duree, description_video, date_sortie, type_video \} \rightarrow \{ id, date_ajout, etat, gratuit_abonne, prix \}$

- VideoClient
 - $id \rightarrow idClient, idVideo, debut_location$
- Realisateur
 - $id \rightarrow nom, prenom, dateNaissance$
- Acteur
 - $id \rightarrow nom, prenom, dateNaissance$
- Genre
 - $id \rightarrow Nom$
- Sujet
 - $id \rightarrow Nom$

En ce qui concerne les relations **ClientAdresse**, **ClientCarte_credit**, **ClientEvenement**, **VideoRealisateur**, **VideoActeur**, **VideoGenre** et **VideoSujet**, nous n'avons pas trouvé de dépendance fonctionnelle, car ces relations possèdent uniquement deux attributs indépendants l'un de l'autre et qui, ensemble, forment la clé.

2^e forme normale (2FN)

Pour qu'une relation soit en deuxième forme normale (2FN), il faut qu'elle soit en 1FN et que tout attribut qui ne fait pas partie de la clé ne dépende pas d'une partie de la clé. La 2FN concerne donc les relations qui ont une clé composite, c'est-à-dire une clé composée de deux attributs ou plus. Notre modèle relationnel contient seulement 7 relations qui ont une clé composite, c'est-à-dire les relations **ClientAdresse**, **ClientCarte_credit**, **ClientEvenement**, **VideoRealisateur**, **VideoActeur**, **VideoGenre** et **VideoSujet**. En observant le modèle relationnel de ces 7 relations, il est possible de constater qu'aucune d'elles n'a d'attributs non-clé. Elles possèdent toutes uniquement 2 attributs, lesquels forment la clé. On peut donc conclure que ces 7 relations sont en 2FN. Pour ce qui est des relations ayant une clé simple, elles sont nécessairement en 2FN. Ainsi, toutes les relations de notre modèle relationnel sont en 2FN.

3^e forme normale (3FN)

Pour qu'une relation soit en troisième forme normale (3FN), il faut qu'elle soit en 2FN et que tout attribut qui ne fait pas partie de la clé ne dépende pas d'un attribut non-clé. Ici, nous allons analyser les différentes relations à l'aide des dépendances fonctionnelles ressorties plus haut. Commençons par les relations qui n'ont aucune dépendance fonctionnelle, c'est-à-dire les relations **ClientAdresse**, **ClientCarte_credit**, **ClientEvenement**, **VideoRealisateur**, **VideoActeur**, **VideoGenre** et **VideoSujet**. Comme mentionné précédemment, ces 7 relations-là n'ont pas de dépendance fonctionnelle puisqu'elles n'ont pas d'attributs non-clé. Elles sont donc nécessairement en 3FN.

Ensuite, il y a les relations qui ont une seule dépendance fonctionnelle, c'est-à-dire les relations **Abonnement**, **Adresse**, **Carte_credit**, **Evenement**, **VideoClient**, **Realisateur**, **Acteur**, **Genre** et **Sujet**. Ces relations ont une seule dépendance fonctionnelle dans laquelle tous les attributs non-clé dépendent de la clé simple. Il n'y a donc aucun attribut non-clé qui dépend d'un autre attribut non-clé dans ces relations. Elles sont donc en 3FN.

Puis, il y a les relations **AdminEmp**, **Client** et **Video** qui ont chacune deux dépendances fonctionnelles. Ces 3 relations ne sont malheureusement pas en 3FN.

Pour ce qui est de la relation **AdminEmp**, les attributs non-clé *prenom*, *nom* et *mot_de_passe* dépendent de l'attribut non-clé *courriel*. C'est la raison pour laquelle cette relation n'est pas en 3FN. En fait, l'attribut *courriel* est une clé tout comme l'attribut *id*. Pour que **AdminEmp** soit en 3FN, il aurait fallu soit remplacer la clé primaire *id* par l'attribut *courriel* et supprimer l'attribut *id* de la relation, ou soit décomposer **AdminEmp** en deux relations : **AdminEmp**(*id*, *courriel*) et **AdminInfo**(*courriel*, *prenom*, *nom*, *mot_de_passe*). Cependant, nous planifions qu'un administrateur puisse modifier son courriel, il serait donc plus simple que cet attribut ne soit pas la clé de la table. De plus, nous voulons garder toutes les informations d'un administrateur ensemble dans la même relation. Nous avons donc gardé la relation **AdminEmp** telle quelle avec la clé primaire *id*. Cette relation n'est donc pas en 3FN.

Pour ce qui est de la relation **Client**, c'est sensiblement la même chose que pour la relation **AdminEmp**, sauf qu'il y a en plus l'attribut non-clé *dateNaissance* qui dépend de l'attribut non-clé *courriel*. Pour transformer la relation **Client** en 3FN, les deux mêmes options que pour **AdminEmp** s'offraient à nous. Cependant, pour la même raison que pour **AdminEmp**, nous avons décidé de garder la relation **Client** telle quelle. **Client** n'est donc pas en 3FN.

Pour ce qui est de la relation **Video**, les attributs non-clé *date_ajout*, *etat*, *gratuit_abonne* et *prix* dépendent des attributs non-clé *titre*, *duree*, *description_video*, *date_sortie* et *type_video*. C'est la raison pour laquelle cette relation n'est pas en 3FN. En fait, la combinaison des attributs *titre*, *duree*, *description_video*, *date_sortie* et *type_video* est une clé tout comme l'attribut *id*. Pour que **Video** soit en 3FN, il aurait fallu la décomposer en deux relations : **VideoInfo**(*id*, *titre*, *duree*, *description_video*, *date_sortie*, *type_video*) et **VideoAppli**(*titre*, *duree*, *description_video*, *date_sortie*, *type_video*, *date_ajout*, *etat*, *gratuit_abonne*, *prix*). Une autre option aurait été de remplacer la clé primaire *id* par la combinaison des attributs ci-dessus et de retirer l'attribut *id*. Cependant, nous préférons garder toutes les informations d'une vidéo ensemble dans la même relation et avoir une clé composite de 5 attributs nous semblait plus complexe que nécessaire. Nous avons donc décidé de garder la relation **Video** telle quelle avec la clé primaire simple *id*. Cette relation n'est donc pas en 3FN.

Pour conclure cette partie sur la normalisation, notre base de données atteint le degré de normalisation 2FN, car c'est le plus haut niveau atteint par toutes les relations de notre base de données. Par contre, 16 des 19 relations de notre base de données sont en 3FN.

5. ENSEMBLE DES REQUÊTES SQL

1. Donner tous les genres des vidéos louer par un client.

- Avec idClient = 7

```
SELECT Nom FROM Genre
JOIN (SELECT VideoGenre.idGenre FROM VideoGenre
      JOIN (SELECT idVideo FROM VideoClient WHERE idClient = 7
            GROUP BY idVideo) AS VideoClientLouer
      ON VideoClientLouer.idVideo = VideoGenre.idVideo) AS idGenreLouer
ON idGenreLouer.idGenre = Genre.id;
```

Résultat : Drame, Science-fiction

- Avec courriel = camille.moreau@live.ca

```
SELECT Nom FROM Genre
JOIN (SELECT VideoGenre.idGenre FROM VideoGenre
      JOIN (SELECT idVideo FROM VideoClient WHERE idClient = (SELECT id FROM
                                                                Client WHERE courriel = 'camille.moreau@live.ca'))
      AS VideoClientLouer ON VideoClientLouer.idVideo = VideoGenre.idVideo)
AS idGenreLouer ON idGenreLouer.idGenre = Genre.id;
```

Résultat : Comédie, Romance

2. Donner les genres préférées des clients.

```
SELECT c.id, c.prenom, c.nom, g.nom AS genrePrefere, nombreEcoutes
FROM Client c
JOIN (SELECT vc.idClient, vg.idGenre, COUNT(*) AS nombreEcoutes
      FROM VideoClient vc JOIN VideoGenre vg ON vc.idVideo = vg.idVideo
      GROUP BY vc.idClient, vg.idGenre) AS ecoutes_client_genre
ON c.id = ecoutes_client_genre.idClient
JOIN Genre g ON ecoutes_client_genre.idGenre = g.id
JOIN (SELECT idClient, MAX(nombreEcoutes) AS maxEcoutes
      FROM (SELECT vc.idClient, vg.idGenre, COUNT(*) AS nombreEcoutes
            FROM VideoClient vc JOIN VideoGenre vg ON vc.idVideo = vg.idVideo
            GROUP BY vc.idClient, vg.idGenre ) AS subquery
      GROUP BY idClient ) AS max_ecoutes_client
ON ecoutes_client_genre.idClient = max_ecoutes_client.idClient
AND ecoutes_client_genre.nombreEcoutes = max_ecoutes_client.maxEcoutes;
```

Résultat :

<i>id</i>	<i>prenom</i>	<i>nom</i>	<i>genrePrefere</i>	<i>nombreEcoutes</i>
10	Hugo	Rousseau	Drame	1
9	Manon	Leroy	Romance	1
9	Manon	Leroy	Comédie	1
8	Antoine	Garcia	Drame	1
7	Chloé	Martin	Science-fiction	3
5	Emma	Dupont	Comédie	2
3	Camille	Moreau	Comédie	1
3	Camille	Moreau	Romance	1
2	Thomas	Dubois	Comédie	2
1	Sophie	Lefebvre	Comédie	2
1	Sophie	Lefebvre	Drame	2

3. Quel sont les sujets de tous les vidéos d'un certain acteur?

- Avec l'acteur = Leonardo DiCaprio

```
SELECT nom as Sujet FROM VideoSujet
INNER JOIN (SELECT id FROM Video
            INNER JOIN (SELECT idVideo FROM VideoActeur
                        WHERE idActeur = (SELECT id FROM Acteur
                                         WHERE prenom = 'Leonardo'
                                         AND nom = 'DiCaprio')) AS Video_LD
            ON Video.id = Video_LD.idVideo WHERE type_video = 'film')
AS Films_LD ON VideoSujet.idVideo = Films_LD.id
INNER JOIN Sujet ON VideoSujet.idSujet = Sujet.id;
```

Résultat : Espionnage

4. Donner le nom des réalisateurs et des acteurs du vidéo le plus louer.

```
SELECT DISTINCT Realisateur.nom AS Nom_Réalisateur, Acteur.nom AS Nom_Acteur
FROM (SELECT idVideo, COUNT(*) AS nbr_location FROM VideoClient
      GROUP BY idVideo
      HAVING COUNT(*) = (SELECT MAX(nbr_location)
                        FROM (SELECT COUNT(*) AS nbr_location FROM VideoClient
                              GROUP BY idVideo) AS MAXLocations))
AS VideoPlusLouer
JOIN VideoRealisateur ON VideoPlusLouer.idVideo = VideoRealisateur.idVideo
JOIN Realisateur ON VideoRealisateur.idRealisateur = Realisateur.id
JOIN VideoActeur ON VideoActeur.idVideo = VideoPlusLouer.idVideo
JOIN Acteur ON VideoActeur.idActeur = Acteur.id;
```

Résultat :

<i>Nom_Réalisateur</i>	<i>Nom_Acteur</i>
Crane	Aniston
Crane	Cox
Crane	Kudrow
Crane	LeBlanc
Crane	Perry
Crane	Schwimmer
Kauffman	Aniston
Kauffman	Cox
Kauffman	Kudrow
Kauffman	LeBlanc
Kauffman	Perry
Kauffman	Schwimmer

5. Donner l'id, le titre, la date de sortie et l'état des vidéos ayant un sujet, un genre et un acteur en particulier.

- Avec sujet = Science-Fiction, le genre = Espionnage et acteur = Leonardo Dicaprio

```
SELECT v.id, v.titre, v.date_sortie, v.etat FROM Video v
INNER JOIN VideoActeur va on v.id = va.idVideo
INNER JOIN Acteur a on a.id = va.idActeur
INNER JOIN VideoGenre vg on vg.idVideo = v.id
INNER JOIN Genre g on g.id = vg.idGenre
INNER JOIN VideoSujet vs on vs.idVideo = v.id
INNER JOIN Sujet s on s.id = vs.idSujet
WHERE g.Nom = 'Science-fiction' AND s.Nom = 'Espionnage'
AND a.nom = 'DiCaprio' AND a.prenom = 'Leonardo' AND v.etat != 'archive'
```

Résultat :

<i>id</i>	<i>titre</i>	<i>date_sortie</i>	<i>etat</i>
1	Inception	2010-07-16	normal

6. Quels sont les vidéos archivés qui sont sortis avant 1995?

```
SELECT *
FROM Video
WHERE etat = 'archive' AND date_sortie < '1995-01-01';
```

Résultat :

<i>id</i>	<i>prix</i>	<i>titre</i>	<i>type_video</i>	<i>description_video</i>	<i>duree</i>	<i>date_sortie</i>	<i>date_ajout</i>	<i>etat</i>	<i>gratuit_abonne</i>
6	9.99	Friends	serie	Une série comique emblématique sur un groupe d'amis à New York.	22	1994-09-22	2022-11-15	archive	1
10	7.99	The Shawshank Redemption	film	Un film dramatique basé sur une nouvelle de Stephen King.	142	1994-09-23	2022-10-30	archive	0

7. Donner les événements qui auront lieu entre le 15 avril 2024 et le 15 mai 2024.

```
SELECT *  
FROM Evenement  
WHERE dateDebut BETWEEN '2024-04-15' AND '2024-05-15';
```

Résultat :

<i>id</i>	<i>prix</i>	<i>titre</i>	<i>type_event</i>	<i>description_event</i>	<i>lieu</i>	<i>date_debut</i>	<i>date_fin</i>
1	15.00	Projection du film 'Inception'	projection	Une projection spéciale du célèbre film 'Inception' de Christopher Nolan.	Central Park	2024-04-15 19 : 00	2024-04-15 21 : 30
2	5.00	Exposition d'art contemporain	exposition	Une exposition mettant en vedette des œuvres d'art contemporain de divers artistes locaux et internationaux.	1380 Rue Sherbrooke O, Montreal, QC H3G 1J5	2024-05-10 10 : 00	2024-05-10 18 : 00

8. Donner le nom du genre, le sujet et le titre du vidéo le moins louer (tous indépendant).

```
SELECT GenreLeMoinsLoue.nomGenre AS GenreLeMoinsLoue,  
       SujetLeMoinsLoue.nomSujet AS SujetLeMoinsLoue,  
       VideoLaMoinsLouee.titre AS VideoLaMoinsLouee  
FROM (SELECT TOP 1 g.nom AS nomGenre, COUNT(vc.id) AS nbLocations  
      FROM Video v  
      JOIN VideoGenre vg ON v.id = vg.idVideo  
      JOIN Genre g ON vg.idGenre = g.id  
      LEFT JOIN VideoClient vc ON v.id = vc.idVideo  
      GROUP BY g.nom ORDER BY COUNT(vc.id) ASC) AS GenreLeMoinsLoue  
CROSS JOIN (SELECT TOP 1 s.nom AS nomSujet, COUNT(vc.id) AS nbLocations  
            FROM Video v  
            JOIN VideoSujet vs ON v.id = vs.idVideo  
            JOIN Sujet s ON vs.idSujet = s.id  
            LEFT JOIN VideoClient vc ON v.id = vc.idVideo  
            GROUP BY s.nom ORDER BY COUNT(vc.id) ASC) AS SujetLeMoinsLoue  
CROSS JOIN (SELECT TOP 1 v.titre, COUNT(vc.id) AS nbLocations  
            FROM Video v  
            LEFT JOIN VideoClient vc ON v.id = vc.idVideo  
            GROUP BY v.titre ORDER BY COUNT(vc.id) ASC) AS VideoLaMoinsLouee
```

Résultat : Aventure, Espionnage, Avatar

9. Quels sont les nouveautés vidéos qui ont été ajoutés dans le dernier mois?

```
SELECT * FROM Video  
WHERE etat = 'nouveaute'  
AND date_ajout > (SELECT DATEADD(month, -1, CONVERT(date, GETDATE())));
```

Résultat : Aucun résultat

10. Donner tous les informations d'un client selon son courriel.

- Avec courriel = lea.fournier@gmail.ca

```
SELECT * FROM Client  
WHERE courriel = 'lea.fournier@gmail.ca';
```

Résultat :

id	prenom	nom	dateNaissance	mot_de_passe	courriel
11	Léa	Fournier	1987-12-18	StrongPassword123	lea.fournier@gmail.ca

11. Donner tous les informations d'un administrateur selon son courriel.

- Avec courriel = bob.johnson@gmail.com

```
SELECT * FROM AdminEmp  
WHERE courriel = 'bob.johnson@gmail.com';
```

Résultat :

<i>id</i>	<i>prenom</i>	<i>nom</i>	<i>mot_de_passe</i>	<i>courriel</i>
2	Bob	Johnson	Securite456&	bob.johnson@gmail.com

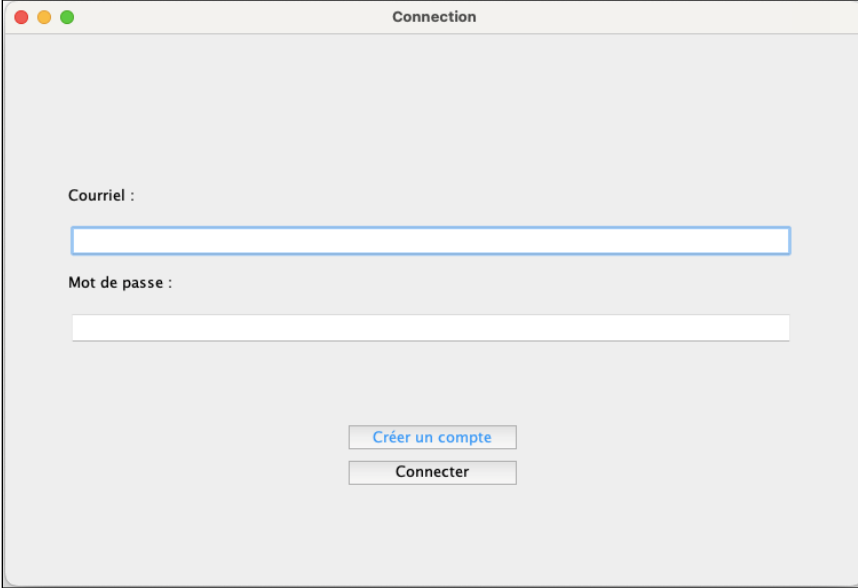
12. Donner le nombre de client qui ont un abonnement.

```
SELECT COUNT(*) AS nombreAbonnes  
FROM Abonnement a  
Where a.actif='true';
```

Résultat : 3

6. APPLICATION EN SQL ET CAPTURES D'ÉCRAN

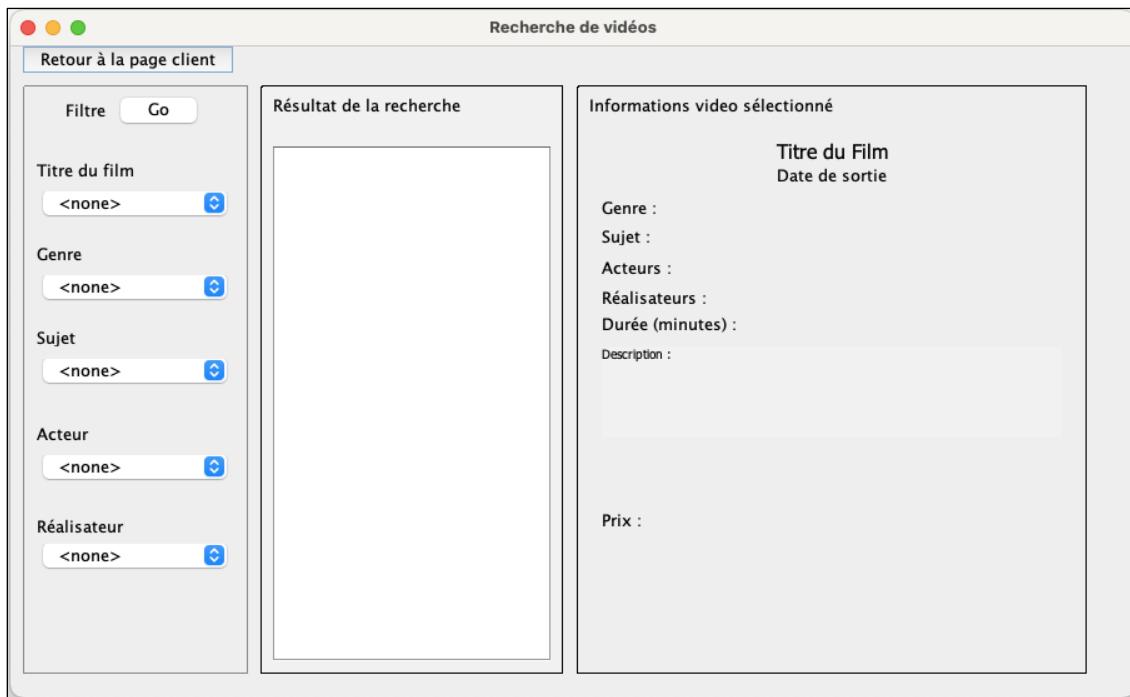
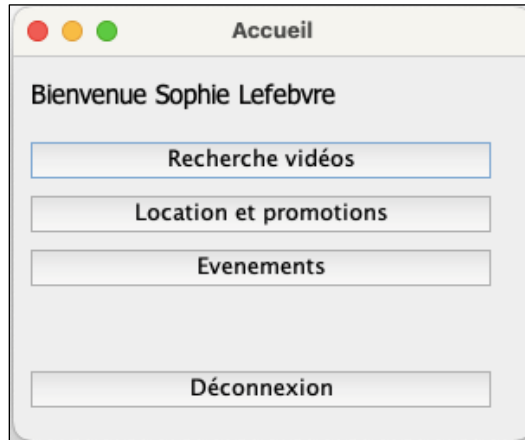
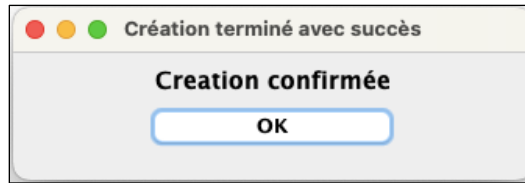
En ce qui concerne le lancement de l'application, il faut lancer la fonction « main » qui se trouve dans le fichier « Connection_Form.java ». De plus, notre application utilise l'utilisateur « E9 » avec le mot de passe « sql » pour se connecter à la base de données. Ceci peut être modifié au besoin dans le fichier « DbConnection.java ».



The screenshot shows a window titled "Connection". It contains two input fields: "Courriel :" and "Mot de passe :". Below the input fields are two buttons: "Créer un compte" and "Connecter".



The screenshot shows a window titled "Création compte". It contains five input fields: "Courriel", "Mot de passe", "Prénom", "Nom", and "Date de naissance (YYYY-MM-JJ)". Below the input fields is a button labeled "Créer compte".



Location

Retour à la page client

Mes Locations | Videos Gratuits | Nouveauté

Informations video sélectionné

Titre du Film

Date de sortie

Genre :

Sujet :

Acteurs :

Réalisateurs :

Durée (minutes) :

Description :

Prix :

Confirmation Location

Confirmez votre location du video suivant :

Film Test

En cliquant confirmer, vous acceptez que le prix suivant vous soit facturé.

Prix : 9.99\$

Confirmer

Cancel

Téléchargement

Retour à la page client

Vous êtes en train de télécharger ...

Titre du video :

Film Test

Evenement

[Retour à la page client](#)

Mes Evenements

Tous les évènements

Projection du film "Inception" (projection) --- 2024-04-15T19:00

Exposition d'art contemporain (exposition) --- 2024-05-10T10:00

Atelier de poterie (activité) --- 2024-06-20T14:00

Q&A avec les acteurs de Friends (autre) --- 2024-07-05T18:30

Visite guidée de Poudlard (activité) --- 2024-08-12T11:00

Informations évènement sélectionné

Titre de l'évènement

Date de debut :

Date de fin :

Lieu :

Type Event :

Description :

Prix :

Confirmation participation evenement

Confirmez votre participation à l'évènement suivant :

Evenement Test

En cliquant confirmer, vous acceptez que le prix suivant vous soit facturé.

Prix : 9.99\$

Confirmer

Cancel

Admin

Filtre

Rechercher

Titre du Film

Genre

<none>

Sujet

Prenom acteur

nom acteur

Prenom Réalis

nom Réalis

Filtre

Rechercher

Prenom

Nom

Courriel

Abonner

<None>

Filtre

Rechercher

Titre

Type

<None>

Contenu video

Inception (2010-07-16)
Friends (1994-09-22) --- archivé
Breaking Bad (2008-01-20)
Avatar (2009-12-18) --- nouveauté
Stranger Things (2016-07-15)
The Shawshank Redemption (1994-09-23) --- archivé
Game of Thrones (2011-04-17)
The Matrix (1999-03-31) --- archivé
Stranger Than Fiction (2006-11-10)
The Office (2005-03-24) --- archivé

Ajouter

Archiver/Déarchiver

Clients

id=1, courriel=sophie.lefebvre@hotmail.com, mot_de_passe
id=2, courriel=thomas.dubois@gmail.ca, mot_de_passe=Pa:
id=3, courriel=camille.moreau@live.ca, mot_de_passe=Secu
id=4, courriel=lucas.lefort@hotmail.com, mot_de_passe=P@
id=5, courriel=emma.dupont@hotmail.ca, mot_de_passe=St
id=6, courriel=louis.durand@live.com, mot_de_passe=Super
id=7, courriel=chloe.martin@hotmail.com, mot_de_passe=P
id=8, courriel=antoine.garcia@hotmail.com, mot_de_passe=
id=9, courriel=manon.leroy@gmail.com, mot_de_passe=P@:
id=10, courriel=hugo.rousseau@gmail.com, mot_de_passe=
id=11, courriel=lea.fournier@gmail.ca, mot_de_passe=Stror

Événement

Projection du film "Inception" (projection) --- 2024-04-15T19
Exposition d'art contemporain (exposition) --- 2024-05-10T1
Atelier de poterie (activité) --- 2024-06-20T14:00
Q&A avec les acteurs de Friends (autre) --- 2024-07-05T18:
Visite guidée de Poudlard (activité) --- 2024-08-12T11:00

Ajouter

Ajout contenu

Titre *

Prix *

format: chiffre seulement

Type vidéo *

film

Description vidéo

Durée *

format: nombre entier

État *

nouveauté

Gratuit abonné *

Oui

Non

Genre

Sujet

Liste acteur

Liste réalisateur

Date de sortie *

Format : yyyy-mm-dd

Ajouter

Supprimer

Ajouter

Supprimer

Ajouter

Supprimer

Ajouter

Supprimer

Créer

Ajouter acteur

Ajouter acteur

Existant

Creer acteur

Leonardo DiCaprio

Jennifer Aniston

Courteney Cox

Lisa Kudrow

Matt LeBlanc

Matthew Perry

David Schwimmer

Bryan Cranston

Ajouter

Ajouter acteur

☐ Existant

☒ Creer acteur

Prenom Nom Date de naissance (yyyy-mm-jj)

Ajout realisateur

☒ Existant

☐ Creer réalisateur

Christopher Nolan

David Crane

Marta Kauffman

Vince Gilligan

James Cameron

Matt Duffer

Ross Duffer

Frank Darabont

Ajout realisateur

Ajouter réalisateur

☐ Existant

☒ Créer réalisateur

Prenom

Nom

Date de naissance (yyyy-mm-jj)

Creer

Ajout evenement

Titre

Prix Chiffre seulement

Type

projection

Description

Lieu

Date de début

yyyy-mm-dd hh:mm

Date de fin

yyyy-mm-dd hh:mm

Créer