

# TD4 : EXISTS et HAVING

## EXISTS

La requête suivante permet d'afficher les lignes de la table `Table` qui vérifie qu'une certaine requête possiblement dans une autre table n'est pas vide.

```
SELECT *
FROM Table
WHERE EXISTS (
    Requete
)
```

## Questions

Dans la base de données `pac.odb` et `cinema.odb`, quelles requêtes permettent de trouver les informations suivantes :

- Les noms des produits qui ont été achetés au moins une fois.

```
SELECT * FROM Produits
WHERE EXISTS (
    SELECT * FROM Achats
    WHERE Produits.NP = Achats.NP
)
```

- Les noms et identifiants des clients qui ont fait au moins un achat.

```
SELECT * FROM Clients
WHERE EXISTS (
    SELECT * FROM Achats
    WHERE Clients.NCLI = Achats.NCLI
)
```

- les titres et identifiants des films qui sont projetés au moins une fois.

```
SELECT * FROM Films
WHERE EXISTS (
    SELECT * FROM Projections
    WHERE Projections.IdFilm = Films.ID
)
```

- les titres et identifiants des films qui sont projetés au moins le lundi.

```
SELECT * FROM Films
WHERE EXISTS (
  SELECT * FROM Projections
  WHERE Projections.IdFilm = Films.ID
  AND Projections.Jour = 'Lundi'
)
```

- les identifiants des cinémas qui projettent au moins un film le jeudi et qui ont au moins 20 salles.

```
SELECT * FROM Cinemas
WHERE EXISTS (
  SELECT * FROM Projections
  WHERE Cinemas.ID = Projections.IdCinema
  AND Projections.Jour = 'Jeudi'
)
AND NbSalles >= 20
```

## HAVING

La commande HAVING est similaire à WHERE mais permet d'utiliser les fonctions d'agrégation (SUM, COUNT, MAX, MIN). La requête suivante permet d'afficher les lignes de la table [Table](#) en les regroupant selon l'attribut [attribut\\_groupe](#) mais en ne gardant les groupes que si la condition de HAVING est respectée.

```
SELECT attribut_groupe, fonction1(attribut_numerique)
FROM Table
GROUP BY attribut_groupe
HAVING fonction2(attribut_numerique) operateur valeur
```

## Questions

- afficher les identifiants des clients qui ont achetés au moins 20 en quantité total

```
SELECT NCLI, SUM(Quantite)
FROM Achats
GROUP BY NCLI
HAVING SUM(Quantite) >= 20
```

- afficher les identifiants des produits qui ont été achetés en quantité totale au moins 20

```
SELECT NP, SUM(Quantite)
FROM Achats
GROUP BY NP
HAVING SUM(Quantite) >= 20
```

- afficher les identifiants et la quantité totale achetée des produits dont toutes les commandes sont en quantité strictement inférieur à 10

```
SELECT NP, SUM(Quantite)
FROM Achats
GROUP BY NP
HAVING MAX(Quantite) < 10
```

- afficher les identifiants et le nombre de projections des films qui ont été projetés au moins 3 fois

```
SELECT IdFilm, COUNT(*)
FROM Projections
GROUP BY IdFilm
HAVING COUNT(*) >= 3
```

- même question que précédemment mais on veut le titre des films en plus : afficher l'identifiant le titre et le nombre de projections des films qui ont été projetés au moins 3 fois

```
SELECT IdFilm, Titre, COUNT(*)
FROM Projections JOIN Films ON Projections.IdFilm = Films.ID
GROUP BY IdFilm, Titre
HAVING COUNT(*) >= 3
```

```
SELECT * FROM
(
SELECT IdFilm, COUNT(*) AS NbProjections
FROM Projections
GROUP BY IdFilm
HAVING COUNT(*) >= 3
) AS Ta JOIN Films ON Ta.IdFilm = Films.ID
```

- afficher l'identifiant, la ville et le nombre de projections des cinémas qui projettent au moins 3 fois

```
SELECT IdCinema, Ville, COUNT(*)
FROM Projections JOIN Cinemas ON Projections.IdCinema = Cinemas.ID
GROUP BY IdCinema, Ville
HAVING COUNT(*) >= 3
```

