# Exercice 1

## 1. Tables Creation

## Client

```
create table clientTp5 (
numcl number,
nom varchar2(50),
prenom varchar2(50),
genere varchar2(15),
adresse varchar2(50),
tel varchar2(50),
constraint pk_clientTP5 primary key (numcl),
constraint chk_clienTP5_genere check (genere in ('homme','femme'))
);
```

### Agence

```
create table agenceTP5 (
idagence number,
adresse varchar2(50),
responsable varchar2(50),
constraint pk_agenceTP5 primary key (idagence)
);
```

### Voiture

```
create table voitureTP5 (
matricule number,
modele varchar2(50),
marque varchar2(50),
couleur varchar2(50),
description varchar2(50),
idagence number,
constraint pk_voitureTP5 primary key (matricule),
constraint fk_voitureTP5_agenceTP5 foreign key (idagence) references agencetp5(idagence)
);
```

#### Louer

```
create table louerTP5 (
  idagence number,
  numcl number,
  matricule number,
  datedebut date,
  datefin date,
  prix number,
  typelocation varchar2 (50),
  constraint pk_louerTP5 primary key (idagence,numcl,matricule,datedebut),
  constraint fk_louerTP5_agenceTP5 foreign key (idagence) references agenceTP5(idagence),
10
  constraint fk_louerTP5_clientTP5 foreign key (numcl) references clientTP5(numcl),
constraint fk_louerTP5_voitureTP5 foreign key (matricule) references voitureTP5 (matricule),
  constraint chk_louerTP5_date check (datefin>datedebut),
  constraint chk_louerTP5_type check (typelocation in ('longue duree', 'courte duree'))
  |);
```

### Insert

i can't put the insert queries in this pdf because they have a lot of lines, you can find all the insert sql code inside the SQL/EX1/insert/ folder

## 2. Client That Never Ranted For Longue Period

```
--method with nested select
select c.nom , c.prenom from clienttp5 c where c.numcl not in (
select l.numcl from louertp5 l where l.typelocation='longue duree'
);

--method with left join
select c.nom , c.prenom from clienttp5 c LEFT JOIN louertp5 l on c.numcl = l.numcl and
l.typelocation='longue duree'
where l.numcl is null;
```

# Nested Select Method

## Inner Query

```
select l.numcl from louertp5 l where l.typelocation='longue duree'
```

This inner query fetches set of numcl that at least ranted a car for a 'longue duree' once

#### Outer Query

```
select c.nom , c.prenom from clienttp5 c where c.numcl not in ( inner query );
```

This outer query select first name and last name of the client that didn't do 'longue duree' at least once  $\rightarrow$  meaning client that never ranted for 'longue duree'

# Joing Method

#### Join

```
select c.nom , c.prenom from clienttp5 c LEFT JOIN louertp5 1
```

We are doing left join between client and louer, so all client from client table will be fetched and if the is no match with louer table it will return null

#### Join Condition

```
on c.numcl = l.numcl and l.typelocation='longue duree'
```

This join condition macthes for client that exist on louer table and ranted for 'longue duree', so it would return null if client doesn't exist in louer table or didn't ranted for 'longue duree' (at least once instance)

## Where Clause

```
where l.numcl is null;
```

Null Client mean client that didn't match → don't exist in louer table or never ranted for 'longue duree'

## 3. Algeria Agency That Offer 'BMW' Or 'Volkswagen' Cars Only

```
--method nested select
select a.idagence from agencetp5 a where a.adresse='Alger'
and a.idagence not in (select v.idagence from voituretp5 v where v.marque not in ('BMW','Volkswagen'));

--method join
select distinct a.idagence from
agencetp5 a inner join voituretp5 v on a.idagence = v.idagence and a.adresse='Alger'
left join voituretp5 v1 on v1.idagence = a.idagence and v1.marque not in ('BMW','Volkswagen')
where v1.idagence is null;
```

# Nested Select Method

## Inner Query

```
select v.idagence from voituretp5 v where v.marque not in ('BMW', 'Volkswagen')
```

We are fetching idagence from voiture table that doesn't offer cars 'BMW' or 'Volkswagen' at least once

### **Outer Query**

```
select a.idagence from agencetp5 a where a.adresse='Alger'and a.idagence not in (inner query);
```

We are fetching idagence from agence table of Algeria that !(doesn't offer cars 'BMW' or 'Volkswagen' at least once)  $\rightarrow$  agency that offers 'BMW' or 'Volkswagen' Only

# Joins Method

#### Inner Join

```
select distinct a.idagence from agencetp5 a inner join voituretp5 v
on a.idagence = v.idagence and a.adresse='Alger'
```

This inner join fetches distinct idagence that exist on voiture table and are located in 'Algeria', we used distinct because this inner join can match the same agency multiple time (agency that offers >1)

#### Left Join

```
inner_Join_Output left join voituretp5 v1
on v1.idagence = a.idagence and v1.marque not in ('BMW', 'Volkswagen')
```

This left join matches the output of first inner join with voiture table, condition join: if agency exist on voiture table and if it doesn't offer 'BMW' or 'Volkswagen' cars at least once, it returns null if agency doesn't exist in voiture (but that is impossible due to first inner join) or the agency always offers 'BMW' or 'Volkswagen' cars

#### Where Clause

```
where v1.idagence is null;
```

This makes sure only agency that aren't matched from the left join that are returned  $\rightarrow$  agency that offers only 'BMW' or 'Volkswagen' cars

# 4.Most Loyal Client

```
create view v1TP5EX1(idagence,numcl,nblocation) as select l.idagence,l.numcl ,count(numcl) nblocation
from louertp5 l group by l.idagence,l.numcl
having count(numcl) = (
select max(count(numcl)) from louertp5 l2 where l2.idagence = l.idagence group by l2.numcl
);
```

# Nested Select Method

### **Inner Query**

```
select max(count(numcl)) from louertp5 12 where 12.idagence = 1.idagence group by 12.numcl
```

It fetches the max rants for each agency

#### **Outer Query**

```
create view v1TP5EX1(idagence,numcl,nblocation) as select l.idagence,l.numcl ,count(numcl) nblocation
from louertp5 l group by l.idagence,l.numcl
having count(numcl) = inner query
```

Creates external view v1 with idagence,numcl,nblocaton from tables louer and groups by idagence,numcl and we filter it with having by making sure each client from same igency must have max ranted count

# Exercice 2

# Exercice 3