

Bases de données SQL

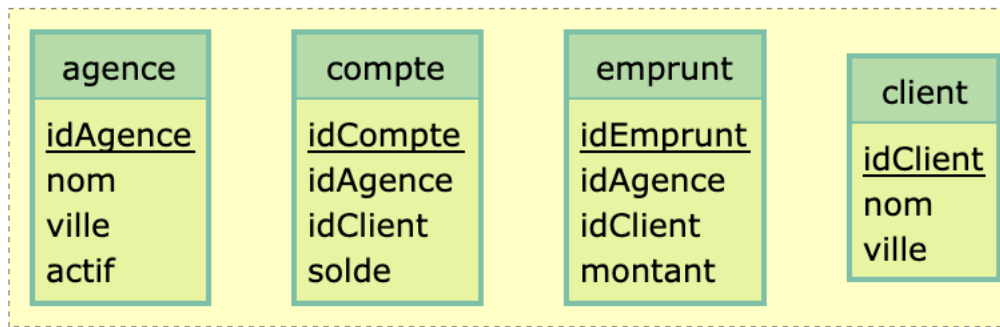
Exercices de TD

max.silberztein@univ-fcomte.fr

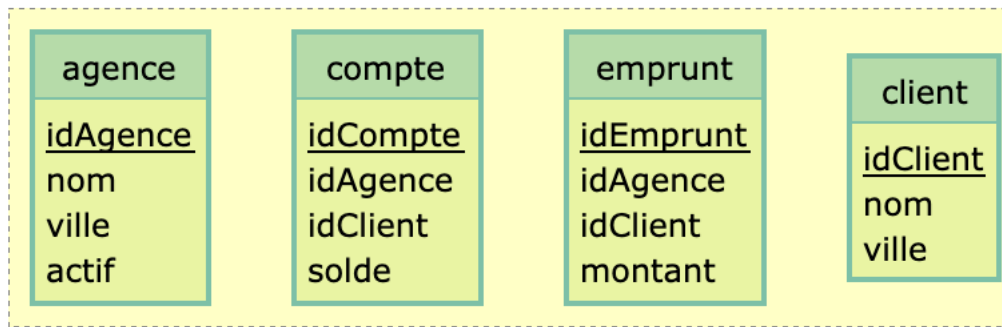
TD4

SQL :

- Entrer les commandes sans aucune faute d'orthographe ou de ponctuation ; éviter les double espaces
- Respecter la notation "Camel", pas d'accents dans les noms de table ni d'attributs
- Respecter la casse



1. Entrer la commande SQL pour obtenir la liste des agences ayant des comptes-clients
2. Clients ayant un compte à l'agence de la ville “La Rochelle”
3. Clients ayant un compte ou un emprunt dans la ville “La Rochelle”



1. Entrer la commande SQL pour obtenir la liste des agences ayant des comptes-clients

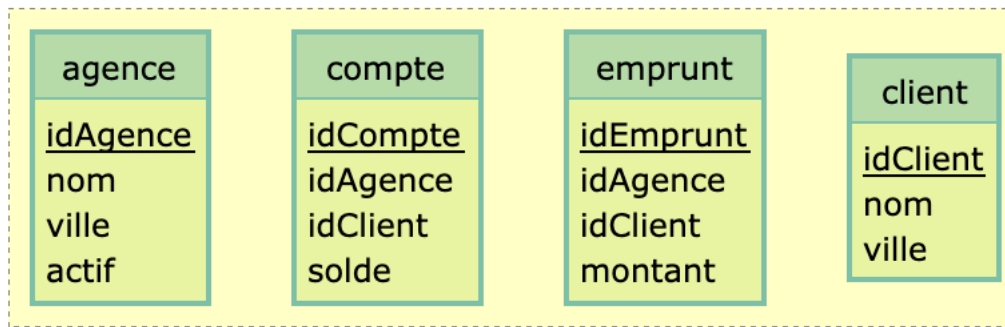
```
SELECT DISTINCT agence.nom FROM agence, compte  
WHERE agence.idAgence = compte.idAgence ;
```

2. Clients ayant un compte à l'agence de la ville “La Rochelle”

```
SELECT DISTINCT client.nom FROM client, agence, compte  
WHERE client.idClient = compte.idClient AND compte.idAgence = agence.idAgence  
AND agence.ville = 'La Rochelle';
```

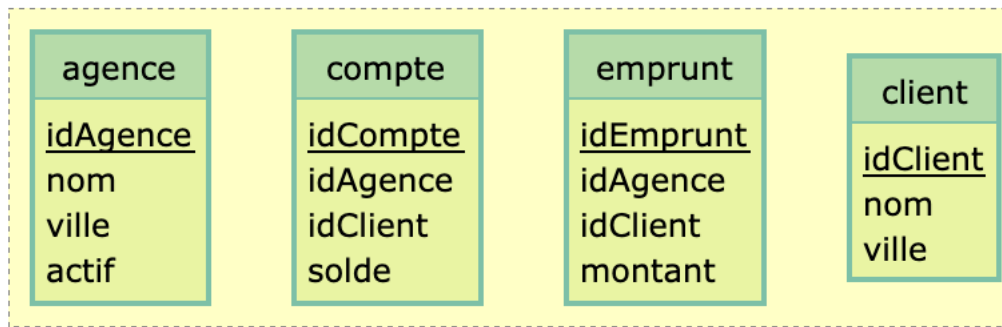
3. Clients ayant un compte ou un emprunt dans la ville “La Rochelle”

```
SELECT DISTINCT client.nom FROM client, agence, compte WHERE  
(client.idClient = compte.idClient AND compte.idAgence = agence.idAgence AND agence.ville = 'La Rochelle') OR  
(client.idClient = emprunt.idClient AND emprunt.idAgence = agence.idAgence AND agence.ville = 'La Rochelle');
```



4. Noms des clients ayant un compte et un emprunt dans une agence de la ville “La Rochelle”

5. Liste de clients ayant un compte dans une agence de la ville “La Rochelle” mais pas d’emprunt

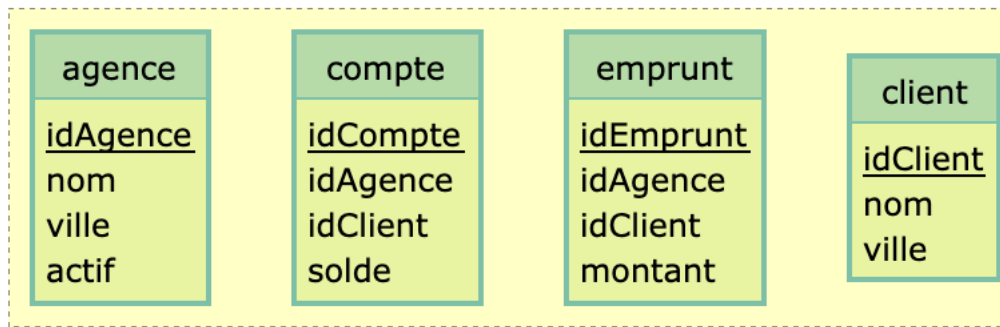


4. Noms des clients ayant un compte et un emprunt dans une agence de la ville “La Rochelle”

```
SELECT DISTINCT client.nom FROM client, agence, compte, emprunt  
WHERE client.idClient = compte.idClient AND compte.idAgence = agence.idAgence  
AND agence.ville = 'La Rochelle' AND client.idClient = emprunt.idClient  
AND emprunt.idAgence = agence.idAgence ;
```

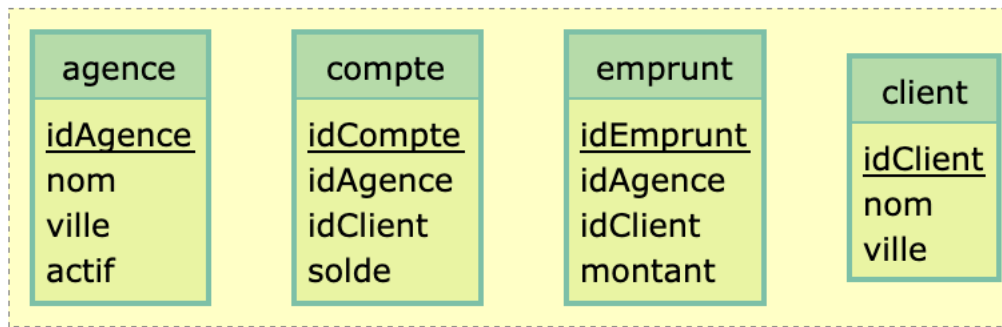
5. Liste de clients ayant un compte dans une agence de la ville “La Rochelle”, mais pas d’emprunt

```
SELECT DISTINCT client.nom FROM client, agence, compte  
WHERE client.idClient = compte.idClient AND compte.idAgence = agence.idAgence AND agence.ville = 'La  
Rochelle' AND client.idClient NOT IN  
(SELECT DISTINCT idClient FROM emprunt, agence WHERE agence.ville = 'La Rochelle' AND  
agence.idAgence = emprunt.idAgence);
```



6. Clients ayant un compte dans la ville où ils habitent

7. Clients et nom de la ville où ils habitent ayant un compte à l'agence “Paris-Etoile”

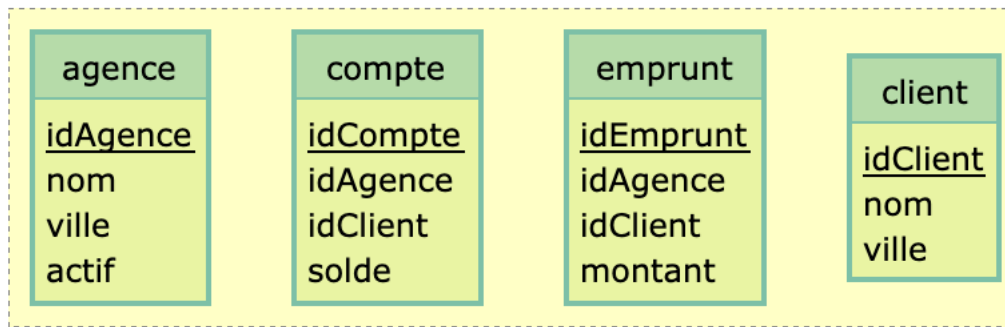


6. Clients ayant un compte dans la ville où ils habitent

```
SELECT DISTINCT client.nom FROM client, agence, compte WHERE  
client.idClient = compte.idClient AND compte.idAgence = agence.idAgence AND agence.ville =  
client.ville ;
```

7. Clients et nom de la ville où ils habitent ayant un compte à l'agence "Paris-Etoile"

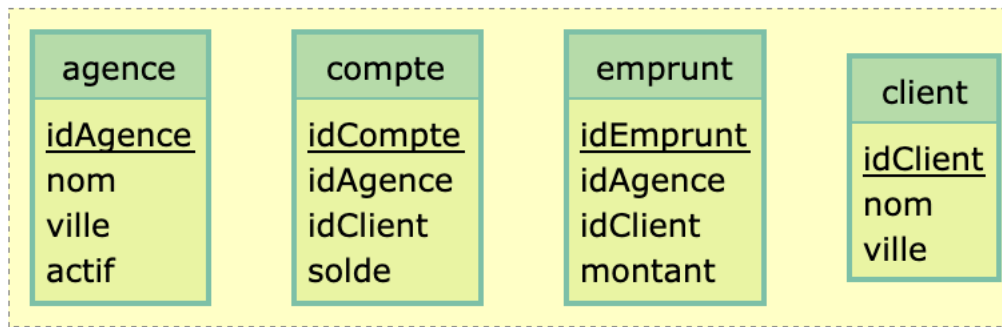
```
SELECT DISTINCT client.nom, client.ville FROM client, agence, compte WHERE  
agence.idAgence = compte.idAgence AND compte.idClient = client.idClient AND agence.nom =  
'Paris-Etoile'
```

8. Clients ayant un compte dans une agence où “Dupuy” a un compte

9. Agences ayant l'actif le plus élevé de Dijon

10. Clients ayant un compte dans l'une des agences de Besançon



8. Clients ayant un compte dans une agence où “Dupuy” a un compte

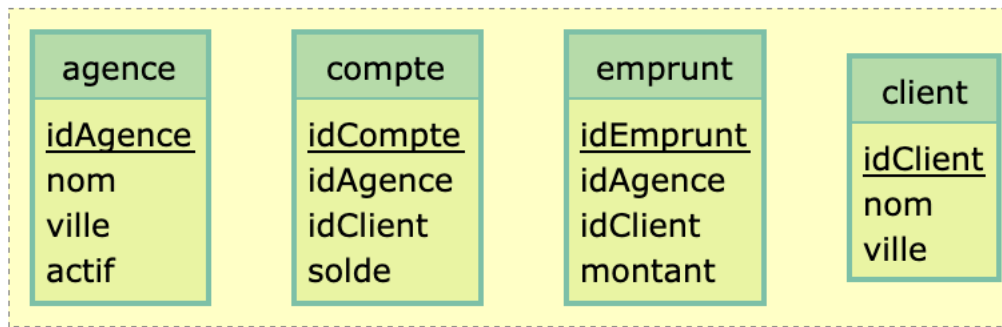
```
SELECT DISTINCT client.nom FROM client, agence, compte WHERE  
client.nom != 'Dupuy' AND client.idClient=compte.idClient AND compte.idAgence =  
agence.idAgence AND agence.idAgence IN (SELECT compte.idAgence FROM compte, client  
WHERE compte.idClient = client.idClient AND client.nom = 'Dupuy');
```

9. Agences ayant l'actif le plus élevé de Dijon

```
SELECT agence.nom FROM agence WHERE  
agence.actif = (SELECT MAX(actif) FROM agence WHERE ville='Dijon');
```

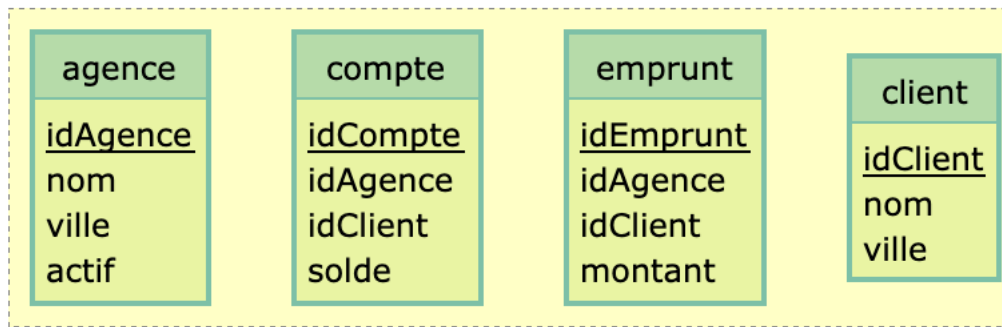
10. Clients ayant un compte dans l’une des agences de Besançon

```
SELECT DISTINCT client.nom FROM agence, compte, client WHERE  
client.idClient = compte.idClient AND compte.idAgence = agence.idAgence AND agence.ville  
= 'Besançon';
```



11. Emprunteurs de l'agence de Dole classés par ordre alphabétique

12. Solde moyen des comptes clients de chaque agence

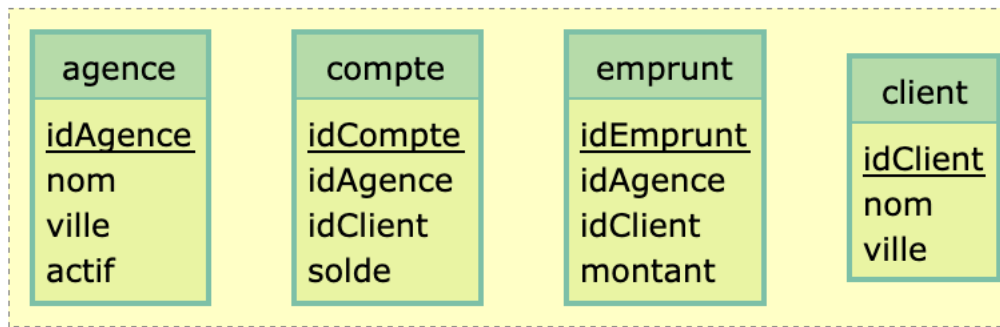


11. Emprunteurs de l'agence de la ville de Dole classés par ordre alphabétique

```
SELECT DISTINCT client.nom FROM agence, emprunt, client  
WHERE agence.idAgence = emprunt.idAgence AND  
emprunt.idClient = client.idClient AND agence.ville = 'Dole'  
ORDER BY client.nom ;
```

12. Solde moyen des comptes clients de chaque agence

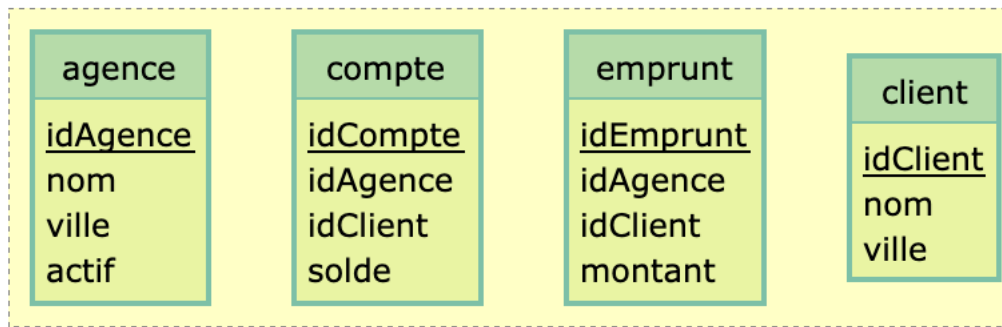
```
SELECT agence.nom, AVG(compte.solde) FROM agence, compte  
WHERE agence.idAgence = compte.idAgence  
GROUP BY agence.idAgence ;
```



13. Solde moyen des comptes-clients des agences dont le solde moyen est > 10.000

14. Nombre de clients habitant à Paris

15. Nom des clients de l'agence "République" n'ayant pas d'adresse dans la table client



13. Solde moyen des comptes-clients des agences dont le solde moyen est > 10.000

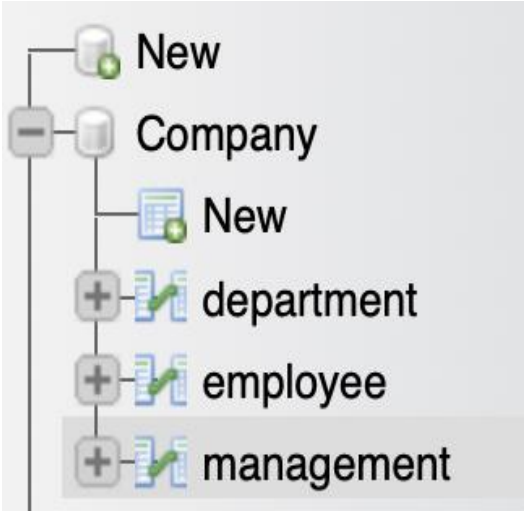
```
SELECT agence.nom, AVG(compte.solde) FROM agence, compte  
WHERE agence.idAgence = compte.idAgence  
GROUP BY agence.idAgence HAVING AVG(compte.solde) > 10000;
```

























14. Nombre de clients habitant à Paris












```
SELECT COUNT(*) FROM client WHERE client.ville = 'Paris';
```












15. Nom des clients de l'agence "République" n'ayant pas d'adresse dans la table client

```
SELECT DISTINCT client.nom FROM agence, compte, client  
WHERE agence.idAgence = compte.idAgence AND  
compte.idClient = client.idClient AND  
agence.nom = 'République' AND client.ville IS NULL;
```



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	idEmployee  	int		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	familyName	varchar(50)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	firstName	varchar(50)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4	sex	char(1)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	5	salary	decimal(10,2)			No	None			 Change  Drop  More
<input type="checkbox"/>	6	contractDate	date			No	None			 Change  Drop  More
<input type="checkbox"/>	7	idDepartment 	int		UNSIGNED	Yes	NULL			 Change  Drop  More

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	idManager  	int		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	familyName	varchar(50)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	firstName	varchar(50)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	idDepartment  	int		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/>	2	idManager	int		UNSIGNED	No	None			 Change  Drop  More
<input type="checkbox"/>	3	name	varchar(50)	utf8mb4_0900_ai_ci		No	None			 Change  Drop  More

Exercice

1. Les prénoms des employés qui gagnent plus que la moyenne ; les trier par leur prénom.

1a. Utiliser une sous-requête

1b. Utiliser une variable

2. le prénom des employés qui gagnent plus de 20% plus que le salaire moyen

2a. Utiliser une sous-requête

2b. Utiliser une variable

Exercice

1. Les prénoms des employés qui gagnent plus que la moyenne ; les trier par leur prénom.

1a. Utiliser une sous-requête

```
SELECT firstName FROM employee WHERE salary > (SELECT AVG(salary) FROM employee) ORDER BY  
firstName;
```

1b. Utiliser une variable

```
SET @a = (SELECT AVG(salary) FROM employee) ;
```

```
SELECT firstName FROM employee WHERE salary > @a ORDER BY firstName ;
```

2. le prénom des employés qui gagnent plus de 20% plus que le salaire moyen

2a. Utiliser une sous-requête

```
SELECT firstName FROM employee WHERE salary > 1.2 * (SELECT AVG(salary) FROM employee) ;
```

2b. Utiliser une variable

```
SET @a = (SELECT AVG(salary) FROM employee) ;
```

```
SELECT firstName FROM employee WHERE salary > 1.2 * @a ;
```

Exercice

3. Les noms de famille et les départements des employés qui sont attachés à un département

3.a Sans utiliser l'opérateur JOIN

3.b Effectuer une jointure avec l'opérateur JOIN

4. les noms de famille et les départements de tous les employés, y compris ceux qui ne sont attachés à aucun département

4.a Effectuer une jointure avec l'opérateur JOIN

4.b Sans utiliser l'opérateur JOIN

Exercice

3. Les noms de famille et les départements des employés qui sont attachés à un département

3.a Sans utiliser l'opérateur JOIN

```
SELECT employee.familyName, department.name FROM employee, department WHERE  
employee.idDepartment = department.idDepartment
```

3.b Effectuer une jointure avec l'opérateur JOIN

```
SELECT employee.familyName, department.name FROM employee JOIN department USING(idDepartment)
```

Exercice

4. Les noms de famille et de départements de tous les employés, y compris ceux qui ne sont attachés à aucun département

4.a Effectuer une jointure avec l'opérateur JOIN

4.b Sans utiliser l'opérateur JOIN

Exercice

4. Les noms de famille et de départements de tous les employés, y compris ceux qui ne sont attachés à aucun département

4.a Effectuer une jointure avec l'opérateur JOIN

```
SELECT employee.familyName, department.name FROM employee LEFT JOIN department USING  
(idDepartment)
```

4.b Sans utiliser l'opérateur JOIN

```
SELECT employee.familyName, department.name FROM employee, department WHERE  
employee.idDepartment = department.idDepartment
```

```
UNION
```

```
SELECT employee.familyName, 'NULL' FROM employee WHERE  
employee.idDepartment IS NULL
```

Exercice

5. Les noms de famille, prénoms et les départements de tous les employés, y compris les départements qui n'ont aucun employé

5.a Effectuer une jointure avec l'opérateur JOIN

5.b Effectuer une jointure sans l'opérateur JOIN

Exercice

5. Les noms de famille et les départements de tous les employés, y compris les départements qui n'ont aucun employé

5.a Effectuer une jointure avec l'opérateur JOIN

```
SELECT employee.familyName, department.name  
FROM employee RIGHT JOIN department USING (idDepartment)
```

5.b Effectuer une jointure sans l'opérateur JOIN

```
SELECT employee.familyName, department.name FROM employee, department  
WHERE employee.idDepartment = department.idDepartment  
  
UNION  
  
SELECT 'NULL', department.name FROM department  
WHERE department.idDepartment NOT IN (SELECT department.idDepartment FROM department, employee  
WHERE employee.idDepartment = department.idDepartment)
```