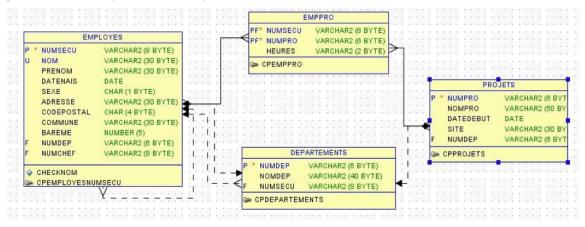
# Algèbre relationnelle

Les exercices de ce chapitre seront réalisés à partir du schéma INFOSOFT



#### Table Départements

NUMDEP	NOMDEP	NUMSECU
d00001	Applications bureautiques	935132
d00002	Applications Main Frame	123456
d00003	Applications telecom	864231
d00004	Didacticiels	121212
d00005	Images Numeriques	999999

#### Table Employés

NUMSECU	NOM	PRENOM	DATENAIS	SEXE	ADRESSE	CODEPOSTAL	COMMUNE	BAREME	NUMDEP	NUMCHEF
123456	CURTIS	Tony	1960-02-17	М	rue Egalite, 18	4430	ANS	70000	d00002	192357
451278	CELARIE	Clementine	1990-10-10	F	rue Tige, 7	4040	HERSTAL	80000	d00001	935132
654321	BEART	Emmanuelle	1995-04-04	F	quai Boverie, 102	4000	LIEGE	90000	d00002	123456
123457	HOFFMAN	Dustin	1979-03-19	М	rue Lantin, 163	4430	ANS	60000	d00004	121212
789999	CLAVIER	Christian	1987-11-10	М	La Batte, 9	4000	LIEGE	65000	d00002	654321
864231	LAFONT	Bernadette	1975-01-14	F	rue Vaudr?e, 162	4000	LIEGE	90000	d00003	192357
439549	DE NIRO	Robert	1980-11-02	М	rue Bonne Foi, 5	4040	HERSTAL	88000	d00002	123456
121212	STALLONE	Sylvester	1982-01-20	М	chaussee Romaine, 174	4300	WAREMME	95000	d00004	192357
334410	BERTHIER	Marie-Sophie L.	1990-02-02	F	allee Dubois, 5	4050	CHAUDFONTAINE	67000	d00003	864231
935132	REDFORD	Robert	1970-02-15	М	avenue Alouettes, 9	4120	NEUPRE	99000	d00001	192357
935133	MOORE	Demi	2000-12-15	F	rue Noisetiers, 9	4120	NEUPRE	54000	d00004	123457
192356	BOHRINGER	Romane	2000-01-01	F	rue Bolland, 38	4260	BRAIVES	66000	d00003	864231
192357	LHERMITTE	Thierry	1985-12-12	М	rue Vieux Pre, 6	4050	CHAUDFONTAINE	56700	NULL	NULL
999999	MONROE	Marilyn	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### Table Projets

NUMPRO	NOMPRO	DATEDEBUT	SITE	NUMDEP
p10345	Intranet	2020-01-05	SERAING	d00003
p10346	Gihep	2020-09-12	LIEGE	d00004
p10347	Eole	2020-02-01	LONDON	d00004
p10348	E-commerce	2020-01-15	SERAING	d00001
p10349	Web	2020-06-13	LIEGE	d00002
p10350	Gps - Cartographie	2020-05-19	LIEGE	d00002
p10351	Email	2020-02-13	LIEGE	d00001
p10352	Jeux - Divertissements	2020-03-19	LIEGE	d00002

Table EmpPro

NUMSECU	NUMPRO	HEURES
451278	p10345	10
451278	p10346	12
451278	p10349	10
451278	p10351	8
935132	p10348	18
935132	p10351	18
123456	p10345	15
123456	p10349	10
123456	p10350	15
654321	p10349	15
654321	p10350	10
789999	p10350	15
789999	p10352	15
439549	p10349	40
864231	p10345	20
864231	p10348	10
864231	p10349	10
334410	p10352	30
192356	p10347	10
192356	p10349	8
123457	p10346	10
123457	p10347	15
123457	p10350	15
121212	p10346	19
121212	p10347	19

# Algèbre relationnelle 1

Donner les tuples de la relation X

 $X = PROJECTION(Departements, NumSecu) \cup PROJECTION(Employes, NumChef)$ 

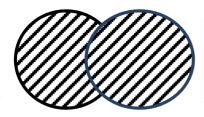
Attention à ne pas oublier le NULL qui apparaît dans la table employés

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
Union (U)	PROJECTION	

Représentez les ensembles

# NumSecu des chefs de départements



NumChef de la table Employés

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le numéro de sécurité sociale de tous les employés qui sont chefs ou responsables d'un département

Ecrivez la requête SQL pour obtenir ce résultat

SELECT NumSecu FROM Departements UNION SELECT NumChef FROM Employes;

# Algèbre relationnelle 2

Donnez les tuples de la relation X

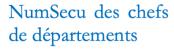
X = PROJECTION(Departements, NumSecu) - PROJECTION(Employes, NumChef)

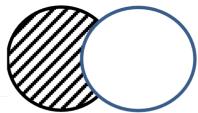
NUMSECU 999999

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
Différence (-)	PROJECTION	

Représentez les ensembles





NumChef de la table Employés

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le numéro de sécurité sociale de tous les employés responsables d'un département mais qui ne sont chefs d'aucun employé.

Ecrivez la requête SQL pour obtenir ce résultat

SELECT NumSecu
FROM Departements
MINUS
SELECT NumChef
FROM Employes;

### Algèbre relationnelle 3

Donnez les tuples de la relation Y

 $X = PROJECTION(Departements, NumDep) \times PROJECTION(SELECTION(Employes/NumSecu > 900000), Nom)$ 

NumDep	Nom
d00001	REDFORD
d00001	MOORE
d00001	MONROE
d00002	REDFORD
d00002	MOORE
d00002	MONROE
d00003	REDFORD
d00003	MOORE

```
        d00003
        MONROE

        d00004
        REDFORD

        d00004
        MOORE

        d00004
        MONROE

        d00005
        REDFORD

        d00005
        MOORE

        d00005
        MONROE
```

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
Produit cartésien (X)	PROJECTION, SELECTION	

Quelle est la question à poser pour obtenir ce résultat ?

Combinez tous les numéros de département avec le nom de chacun des employés ayant un numéro de sécurité sociale supérieur à 900000

Ecrivez la requête SQL pour obtenir ce résultat

```
SELECT departements.NumDep, Nom
FROM Departements, Employes
WHERE Employes.numsecu > 900000;
-- pas de jointure entre les deux tables => cela traduit bien le produit cartésien, mais dans la pratique, cela
n'arrivera pas dans des requêtes "logiques"
```

# Algèbre relationnelle 4

Donnez les tuples de la relation Y

 $X = PROJECTION(Employes. NumChef) \cap PROJECTION(Departements. NumSecu)$ 

935132

123456

864231

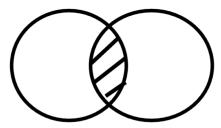
121212

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
	PROJECTION	Intersection (∩)

Représentez les ensembles





NumSecu des chefs de départements

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le numéro de sécurité sociale des employés qui sont à la fois chefs et responsables d'un département

Ecrivez deux requêtes SQL pour obtenir ce résultat

Une requête utilisera un opérateur additionnel tandis que l'autre un opérateur ensembliste.

Opérateur additionnel : intersection

SELECT NumSecu FROM Departements INTERSECT SELECT NumChef FROM Employes;

Opérateur ensembliste : MINUS

```
SELECT NumChef FROM Employés
MINUS
(SELECT NumChef FROM Employés
MINUS
SELECT NumSecu FROM Départements);
```

## Algèbre relationnelle 5

Donnez les tuples de la relation X

 $X = PROJECTION(SELECTION(JOINTURE(Departements, Employes/Departements. \ NumDep = Employes. \ NumDep)/Nom = 2000 \ (Selection (Sel$ 

NomDep Applications Main Frame

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
	PROJECTION, SELECTION	JOINTURE

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le nom du département dans lequel travaille l'employé dont le nom est 'De Niro' Donnez le nom du département dont le nom du responsable est 'De Niro'

Ecrivez la requête SQL pour obtenir ce résultat

```
SELECT NomDep
FROM Departements, Employes
WHERE Departements.NumDep = Employes.NumDep
AND UPPER(Nom) = 'DE NIRO';
```

## Algèbre relationnelle 6

Donnez les tuples de la relation X

 $X = PROJECTION(SELECTION(JOINT\_EXTLEFT(Employes, EmpPro/Employes. NumSecu = EmpPro. NumSecu)/EmpPro. NumSecu)/EmpPro. NumSecu = NumSec$ 

935133 192357 999999

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
	PROJECTION, SELECTION	JOINT_EXT LEFT (jointure externe à gauche)

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le numéro de sécurité sociale des employés qui n'ont pas de projet en cours

Ecrivez la requête SQL pour obtenir ce résultat

```
SELECT Employes.NumSecu
FROM Employes LEFT JOIN EmpPro
ON Employes.NumSecu = EmpPro.NumSecu
WHERE EmpPro.NumSecu IS NULL;
```

### Algèbre relationnelle 7

 $X = PROJECTION(JOINT\_EXTRIGHT(Employes, EmpPro/Employes.numsecu = Empro.numsecu),$ 

STALLONE
CURTIS
HOFFMAN
BOHRINGER
BERTHIER
DE NIRO
CELARIE
BEART
CLAVIER

LAFONT

Complétez le tableau ci-dessous avec le nom des opérateurs utilisés dans la question précédente

Opérateur ensembliste	Opérateur relationnel	Opérateurs additionnels
	PROJECTION	JOIN-EXT RIGHT (Jointure externe à droite)

Quelle est la question à poser pour obtenir ce résultat ?

Donnez le nom de tous les employés qui travaillent sur un projet

Ecrivez la requête SQL pour obtenir ce résultat

```
SELECT DISTINCT Nom

FROM Employes RIGHT JOIN EmpPro

ON Employes.NumSecu = EmpPro.NumSecu;
```

Quelle est l'utilité d'utiliser un RIGHT JOIN dans ce cas-ci?

Aucune utilité, au contraire, cela ajoute des tuples inutiles au résultat intermédiaire.

Donnez la requête équivalente sans jointure externe :

```
SELECT Nom
FROM Employes INNER JOIN EmpPro
ON Employes.NumSecu = EmpPro.NumSecu;
```

Demander un exemple de requête avec ces mêmes tables dans laquelle le RIGHT JOIN aurait été nécessaire

(exemple : donner les numéros de projets sur lesquels ne travaille aucun employé, ou l'exercice précédent en inversant l'ordre des tables => LEFT JOIN devient RIGHT JOIN)

# **Exercices sur l'intégrité**

Exercices sur les contraintes structurelles du modèle relationnel

#### **Integrite 1**

```
INSERT INTO PROJETS VALUES ('p10345', 'Test intégrité 1', To_DATE ('19/10/2020', 'DD/MM/YYYY'), 'LIEGE', 'd00001');

Résultat:

Erreur commençant à la ligne: 1 de la commande -
INSERT INTO PROJETS VALUES ('p10345', 'Test intégrité 1', TO_DATE ('19/10/2020', 'DD/MM/YYYY'), 'LIEGE', 'd00001')

Rapport d'erreur -
ORA-00001: violation de contrainte unique (INFOSOFT.CPPROJETS)-
```

Attirer l'attention sur la propriété de l'intégrité qui est violée : ici c'est l'unicité, cela pourrait être la propriété qui dit que la clé primaire ne peut être nulle.

#### **Integrite 2**

Exécuter une requête d'insertion pour violer l'intégrité référentielle.

```
INSERT INTO Projets VALUES ('pll111', 'Test intégrité 2',

TO_DATE ('19/10/2016', 'DD/MM/YYYY'), 'LIEGE', 'dll1111');
```

Résultat :

Erreur commençant à la ligne: 1 de la commande -

INSERT INTO Projets VALUES ('p11111', 'Test intégrité 2',

TO\_DATE ('19/10/2016', 'DD/MM/YYYY'), 'LIEGE', 'd11111')

Rapport d'erreur -

ORA-02291: violation de contrainte d'intégrité (INFOSOFT.REFPROJETSNUMDEPINDEPARTEMENTS) - clé parent introuvable

### **Integrite 3**

Exécuter une requête de suppression pour violer l'intégrité référentielle.

```
DELETE FROM Projets WHERE NumPro='p10345';
```

Résultat :

Erreur commençant à la ligne: 2 de la commande -

DELETE FROM Projets WHERE NumPro='p10345'

Rapport d'erreur -

ORA-02292: violation de contrainte (INFOSOFT.REFEMPPRONUMPROINPROJETS) d'intégrité - enregistrement fils existant

Attirer l'attention sur les numéros d'erreur Oracle :

ORA-02291 : on veut insérer une valeur dans une clé étrangère qui n'existe pas dans la table où cette clé est clé primaire ORA-02292 : on veut supprimer une valeur de clé primaire qui est référencée par une clé étrangère dans une autre table

#### **Integrite 4**

Exécuter une requête d'insertion pour insérer une valeur référençante inconnue (NULL).

```
INSERT INTO PROJETS VALUES ('p11111', 'Test intégrité 1',
TO_DATE ('19/10/2016', 'DD/MM/YYYY'), 'LIEGE', NULL);
```

Résultat :

1 ligne inséré.

#### **Integrite 5**

Exécuter la procédure AjouterProjet du package GestionProjets pour violer une intégrité de domaine

Résultat

Erreur commençant à la ligne: 1 de la commande -INSERT INTO PROJETS VALUES ('ptoto', 'Test intégrité 1', 'erreur', 'LIEGE', NULL)

Rapport d'erreur -

ORA-01858: Caractère non numérique trouvé à la place d'un caractère numérique