

Rapport TP BDD

BLANC Loriane

GUIRAUD Vincent

HELLOUIN DE CENIVAL Zacharie

ZENNARO Thomas

BOURLLOT Xavier

3IMACS-AE-C, Binôme 8

3 juin 2019

Table des matières

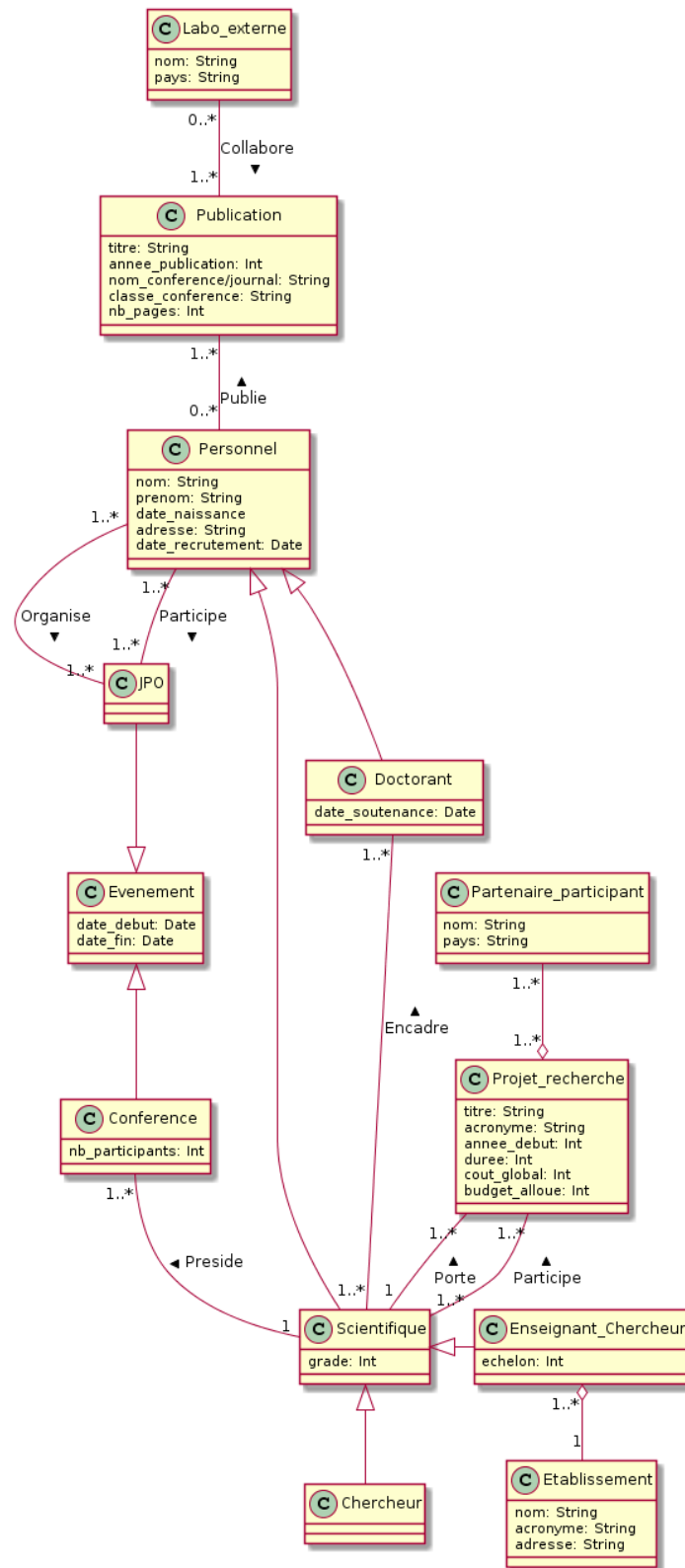
1	Introduction	2
2	Conception UML	2
3	Modèle Relationnel	3
4	Normalisation	4
5	Création des tables	4
6	Remplissage des tables	4
7	Écriture des Requêtes	4
7.1	Algèbre relationnelle	4
7.2	SQL	5
8	Conclusion	7
A	Annexes	8
A.1	Requêtes SQL	8

1 Introduction

Nous souhaitons mettre en place une base de données afin de gérer les évaluations du laboratoire du LAAS-CNRS. Pour cela nous commencerons par concevoir un diagramme de classe UML, puis nous chercherons à traduire ce diagramme en modèle relationnel en utilisant SQL. L'étape suivante consiste à remplir la base de données avec des valeurs inventées afin de pouvoir tester la viabilité de notre modèle par le biais de différentes requêtes.

2 Conception UML

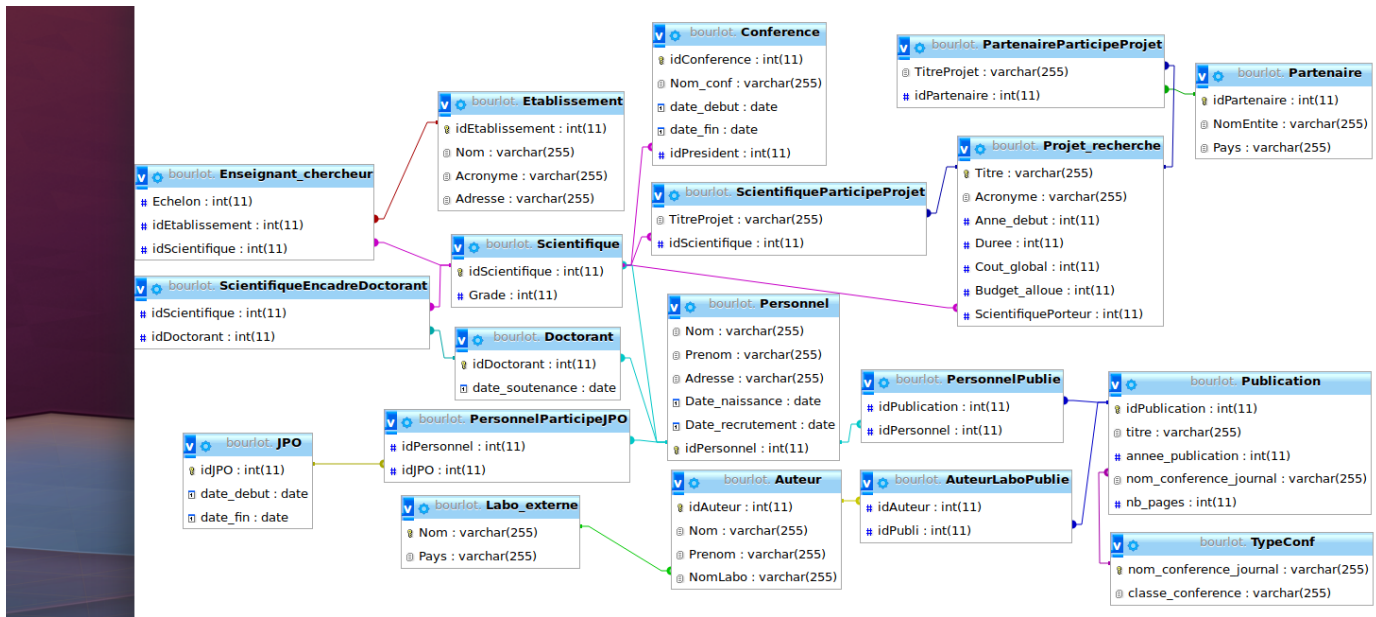
A partir du cahier des charges qui nous a été fourni, nous avons conçu un diagramme de classe. Nous avons bien pris en compte le fait qu'il y a deux types de personnel : les scientifiques et les doctorants, et qu'un scientifique peut encadrer un doctorant. Nous avons procédé de même pour les autres classes, en suivant à la lettre le cahier des charges qui nous a permis de construire les classes, définir leurs arguments, et les liens entre eux ainsi que leur cardinalité. Nous avons ensuite traduit ce diagramme de classe en UML. Voici le diagramme ainsi obtenu :



3 Modèle Relationnel

Afin de pouvoir implémenter notre base de données, nous avons utilisé le langage SQL pour créer nos tables que nous avons pu visualiser sur phpMyAdmin. Nous avons ajouté une classe Auteur et une classe AuteurPublie pour pouvoir ajouter des collaborateurs externes dans les publications. De même, nous avons ajouté une classe d'association entre Scientifiques et Doctorants, JPO et Personnel, Scientifique et Projet de recherche, et Partenaire avec Projet de recherche.

Ces classes d'association permettent de faire le lien entre deux classes pour lesquelles une action est menée (participation, encadrement de personnel, publication etc.). Nous avons donc créé le modèle relationnel suivant :



4 Normalisation

Nous n'avons pas rencontré de problèmes particuliers concernant la normalisation, toutes les formes normales semblaient être respectées. Nous pouvons donc passer à l'étape suivante : la création des tables.

5 Création des tables

Dans cette étape, nous n'avons eu qu'à créer les tables à partir des modèles précédemment établis. La syntaxe étant la même pour la création de toutes les tables, la difficulté consistait à ne pas oublier d'arguments et leur affecter le bon type. Il fallait également veiller à la présence des clés étrangères qui lie certaines classes, notamment les classes d'association. Nous avons donc créé les tables dans le langage SQL.

6 Remplissage des tables

Le remplissage des tables s'est fait de manière à pouvoir tester les différentes requêtes ; il était en effet nécessaire d'avoir des données correspondant aux interrogations SQL de la base de donnée. Nous avons essayé de faire en sorte que la BDD soit la plus complète possible, mais nous avons dû modifier plusieurs fois l'ensemble des tables afin de correspondre à une requête précise que nous n'avions pas encore traitée, ce qui a été un travail fastidieux et peu intéressant. Le code que nous avons écrit respecte donc ces contraintes et permet de tester les requêtes de la prochaine partie.

7 Écriture des Requêtes

7.1 Algèbre relationnelle

Cette partie fut la plus chronophage. En effet, la traduction des différentes requêtes en algèbre relationnelle, plus formelle, s'est révélée délicate à la fois en raison de la complexité de la base de données, et parfois à cause des différentes interprétations possibles de la requête. Nous nous sommes donc répartis les requêtes, en s'entraidant en cas de difficulté, car certaines requêtes étaient bien plus fastidieuses que d'autres. La cohérence entre les requêtes et l'architecture en évolution continue de notre base de données était également un obstacle que nous avons dû surmonter.

7.2 SQL

Pour plus de lisibilité nous n'avons pas mis l'ensemble des requêtes, le but étant d'en présenter quelques unes pour visualiser le fonctionnement. L'intégralité des requêtes est fournie en annexe.

REQUETE 1 : L'idée ici était de d'abord récupérer le nom et l'id des scientifiques encadrant le doctorant (on a choisi le doctorand dont l'id était 1). Pour obtenir le grade et l'id, nous avons dû passer par le biais de la table "Scientifique" et procéder de manière identique qu'avec le nom, qui lui était situé dans "Personnel". Il a ensuite suffi de joindre ces deux tables afin de récupérer juste le nom, associé au grade, les id récupérés permettant d'assurer une bonne correspondance.

```
1 —1.      Le nom et les grades des encadrants d un doctorant donne
2
3 SELECT p.Nom,s.grade FROM
4 (SELECT Nom, idPersonnel FROM Personnel WHERE idPersonnel IN (SELECT idScientifique FROM
5      ScientifiqueEncadreDoctorant WHERE idDoctorant=1))as p
6 JOIN
7 (SELECT s.grade,s.idScientifique FROM Scientifique s WHERE s.idScientifique IN (SELECT idScientifique
8      FROM ScientifiqueEncadreDoctorant WHERE idDoctorant=1))as s
9 ON s.idScientifique=p.idPersonnel
```

Nous obtenons le résultat suivant :

Nom	grade
Nom	8
Bond	3
Encadrator	1
Orange	2

REQUETE 6 : Nous utilisons la fonction "COUNT (*)" puis GROUP BY à partir de la table Publication afin de pouvoir les compter.

```
1 —6.      Le nombre de publications par annee de tout le laboratoire
2
3 (SELECT annee_publication ,COUNT(*) as cntPubli FROM Publication GROUP BY annee_publication);
```

Nous obtenons le résultat suivant :

annee_publication	cntPubli
2018	5
2019	3

REQUETE 16 : L'élaboration de cette requête est assez évidente, car il a suffi d'exclure de la liste des scientifiques tous ceux qui étaient présents dans "ScientifiqueEncadreDoctorant" ou "ScientifiqueParticipeProjet" ou "PersonnelPublie". Pour récupérer le nom et prénom, il a ensuite fallu passer par le biais de la table "Personnel".

```
1 —16.     Le nom et le prenom du scientifique qui n a jamais publie , encadre , ni participe a des projets .
2
3 SELECT Nom,Prenom from Personnel p WHERE p.idPersonnel=
4 (((((SELECT idScientifique from Scientifique)
5 EXCEPT
6 (SELECT idScientifique from ScientifiqueEncadreDoctorant))
7 EXCEPT
8 (SELECT idScientifique from ScientifiqueParticipeProjet))
9 EXCEPT
10 SELECT idPersonnel from PersonnelPublie)
```

Nous obtenons le résultat suivant :

nom	prenom
Kine	Miss

REQUETE 21 : Pour la requête 21 (trouver le nombre de collaborateurs par pays), nous avons joint les tables Auteur et Labo externe qui ont leur nom de laboratoire en commun (ce qui veut dire qu'ils ont collaboré), ensuite nous avons groupé ces laboratoires par pays, et enfin, nous avons compté le nombre de laboratoire qui collabore par pays. Voici son code en SQL :

```

1  —21.    Le nombre de collaborateurs par pays
2
3  SELECT pn.Pays, COUNT(idAuteur)
4  FROM(SELECT Pays,Nom FROM Labo_externe) as pn
5  LEFT JOIN(SELECT idAuteur, NomLabo FROM Auteur) as a
6  ON pn.Nom=a.NomLabo
7  GROUP BY pn.Pays

```

Nous obtenons le résultat suivant :

pays	count
France	3
URSS	1
Estonie	1

REQUETE 26 : Nous comparons le nombre de doctorants recrutés chaque année par un scientifique avec le nombre d'années écoulées entre le recrutement du scientifique et la date actuelle.

```

1  —26.    Les scientifiques qui recrutent au moins un doctorant par annee
2
3
4  SELECT da.idScientifique FROM
5  (SELECT sd.idScientifique, COUNT(sd.DateDoc) as NbDocs, AVG(EXTRACT( YEAR FROM p.Date_recrutement)) as
   DateSci
6  FROM (SELECT idPersonnel, Date_recrutement FROM Personnel) as p
7
8  JOIN (SELECT DISTINCT s.idScientifique, EXTRACT(YEAR FROM d.Date_recrutement) as DateDoc
9  FROM
10 (SELECT idScientifique, idDoctorant FROM ScientifiqueEncadreDoctorant) as s
11 JOIN
12 (SELECT idPersonnel,Date_recrutement FROM Personnel WHERE idPersonnel IN (SELECT idDoctorant FROM
   Doctorant)) as d
13 ON s.idDoctorant=d.idPersonnel )as sd
14 ON sd.idScientifique=p.idPersonnel
15 GROUP BY sd.idScientifique) as da
16 WHERE (da.nbdocs>= (EXTRACT(YEAR FROM CURRENT_DATE) - da.DateSci))

```

Nous obtenons le résultat suivant :

idscientifique
6

REQUETE 30 : Pour trouver le nombre de conférences de classe A organisées par le Laas par année, nous avons joint les tables Conférence et TypeConf pour lesquelles le nom de conférence était le même, avec uniquement des classes A (ce qui permet d'obtenir la liste des conférences de classe A données par le Laas), puis nous les avons regroupées par année et nous avons compté le nombre de conférence correspondant à l'aide d'un "COUNT(nom conférence)", tout en gardant uniquement les attributs année et nombre de conférences à l'aide d'un Select, puisque ce sont les informations qui nous intéressent. Voici le code en SQL :

```

1  —30.    Le nombre de conferences de classe A organisees par le laboratoire par annee
2
3  SELECT cd.annee, COUNT(cd.Nom_conf) as NbConf FROM
4  (SELECT Nom_conf,EXTRACT(YEAR FROM date_debut) as annee FROM Conference) as cd
5  LEFT JOIN
6  (SELECT nom_conference_journal FROM TypeConf WHERE classe_conference='A') as na
7  ON na.nom_conference_journal=cd.Nom_conf
8  GROUP BY cd.anne

```

Nous obtenons le résultat suivant :

annee	nbconf
2019	1
2018	2
2001	1
2000	1

8 Conclusion

Ce projet nous a permis d'aborder l'ensemble des notions vues en cours, en TD et de les mettre en pratique. Mise à part la requête numéro 24, nous les avons toutes testées avec notre base de données et nous avons pu les valider. Le cheminement nécessaire pour aboutir à une base de données fiable et fonctionnelle nous a permis de prendre conscience des écueils à éviter au cours d'un projet de cette ampleur. Il est par exemple très important de ne pas hésiter à modifier la structure du projet tout au long de la conception, même au moment de l'élaboration des requêtes.

A Annexes

A.1 Requêtes SQL

Ensemble de nos requêtes traduites en SQL ainsi que le résultat obtenu pour chacune :

A.1.1 R1

```
1 —1.      Le nom et les grades des encadrants d un doctorant donne
2
3 SELECT p.Nom,s.grade FROM
4 (SELECT Nom, idPersonnel FROM Personnel WHERE idPersonnel IN (SELECT idScientifique FROM
5   ScientifiqueEncadreDoctorant WHERE idDoctorant=1))as p
6 JOIN
7 (SELECT s.grade,s.idScientifique FROM Scientifique s WHERE s.idScientifique IN (SELECT idScientifique
8   FROM ScientifiqueEncadreDoctorant WHERE idDoctorant=1))as s
9 ON s.idScientifique=p.idPersonnel
```

Nous obtenons le résultat suivant :

Nom	grade
Nom	8
Bond	3
Encadrator	1
Orange	2

A.1.2 R2

```
1 —2.      Les pays avec qui un scientifique donne collabore
2
3 SELECT Pays FROM Labo_externe WHERE Nom IN(SELECT NomLabo FROM Auteur WHERE idAuteur IN(SELECT idAuteur
4   FROM AuteurLaboPublie WHERE idPubli IN(SELECT idPublication FROM PersonnelPublie WHERE idPersonnel
5   =8)));
```

Nous obtenons le résultat suivant :

Pays
Estonie

A.1.3 R3

```
1 —3.      Les noms et les pays des auteurs collaborateurs d un scientifique donne en 2018
2
3 SELECT nnl.Nom, l.Pays FROM
4 (SELECT Nom, NomLabo FROM Auteur WHERE idAuteur IN(SELECT idAuteur FROM AuteurLaboPublie WHERE idPubli
5   IN(SELECT idPublication FROM PersonnelPublie WHERE idPersonnel=8)
6 INTERSECT
7 (SELECT idPublication FROM Publication WHERE annee_publication=2018)))as nnl
8 JOIN
9 (SELECT Nom,Pays FROM Labo_externe)as l
10 ON nnl.NomLabo=l.Nom
```

Nous obtenons le résultat suivant :

Nom	Pays
Sade	Estonie
(1 row)	

A.1.4 R4


```

1  —4.      Le nombre de collaborateurs d un scientifique donne en 2018
2
3
4  SELECT COUNT(idPersonnel) as NbCollab FROM PersonnelPublie
5  WHERE (idPublication IN
6         (SELECT idPublication FROM Publication
7          WHERE annee_publication=2018 AND idPublication IN
8          (SELECT idPublication FROM PersonnelPublie
9           WHERE idPersonnel=10)
10         ) AND idPersonnel!=10)

```

Nous obtenons le résultat suivant :

NbCollab
2

A.1.5 R5

```

1  —5.      Pour chaque doctorant, on souhaiterait recuperer le nombre de ses publications
2
3  SELECT d.idDoctorant ,p.cntPubli
4  FROM (SELECT idPersonnel ,COUNT(idPublication) as cntPubli FROM PersonnelPublie GROUP BY idPersonnel) as
5  p
6  RIGHT JOIN (SELECT idDoctorant FROM Doctorant) as d
ON d.idDoctorant=p.idPersonnel

```

Nous obtenons le résultat suivant :

iddoctorant	cntpubli
121314	1
151617	NULL
1	NULL
8	2
88	NULL
66	NULL

A.1.6 R6

```

1  —6.      Le nombre de publications par annee de tout le laboratoire
2
3  (SELECT annee_publication ,COUNT(*) as cntPubli FROM Publication GROUP BY annee_publication);

```

Nous obtenons le résultat suivant :

annee_publication	cntPubli
2018	5
2019	3

A.1.7 R7

```

1  —7.      Le nombre de doctorants du laboratoire
2
3  SELECT COUNT(*) as nbDoctorants FROM Doctorant;

```

Nous obtenons le résultat suivant :

nbDoctorants
6

A.1.8 R8

```
1 —8.      Le nombre de scientifiques du laboratoire
2
3 SELECT COUNT(*) as NbScientifiques FROM Scientifique;
```

Nous obtenons le résultat suivant :

NbScientifiques
6

A.1.9 R9

```
1 —9.      Le nombre d enseignants chercheurs par etablissement d enseignement
2
3 (SELECT idEtablissement ,COUNT(*) as cntEnseignants FROM Enseignant_chercheur GROUP BY idEtablissement)
```

Nous obtenons le résultat suivant :

idEtablissement	cntEnseignants
51	2
52	1

A.1.10 R10

```
1 —10.     Le nombre de publications par scientifique/doctorant
2
3 (SELECT idPersonnel ,COUNT(idPublication) as cntPubli FROM PersonnelPublie GROUP BY idPersonnel)
```

Nous obtenons le résultat suivant :

idPersonnel	cntPubli
1	2
7	1
8	2
10	3
999	1
121314	1

A.1.11 R11

```
1 —11.     Les personnes ayant participe a toutes les journees portes ouvertes
2
3 SELECT DISTINCT idPersonnel FROM PersonnelParticipeJPO j1
4 WHERE NOT EXISTS(SELECT * FROM JPO j2
5 WHERE NOT EXISTS(SELECT * FROM PersonnelParticipeJPO j3
6 WHERE (j1.idPersonnel=j3.idPersonnel AND j2.idJPO=j3.idJPO)))
```

Nous obtenons le résultat suivant :

idPersonnel
118518

A.1.12 R12

```
1 —12.     Les personnes qui n ont jamais participe aux journees portes ouvertes
2
3 (SELECT idPersonnel from Personnel)
4 EXCEPT
5 (SELECT idPersonnel from PersonnelParticipeJPO)
```

Nous obtenons le résultat suivant :

idpersonnel
8
10
121314
151617
13
88
66
6
999
7

A.1.13 R13

```

1 —13. Le nom et l annee de toutes les conferences organisees par un scientifique donne.
2
3 SELECT Nom_conf,EXTRACT(YEAR FROM date_debut) as annee FROM Conference WHERE idPresident=7

```

Nous obtenons le résultat suivant :

Nom_conf	annee
L'inspiration dans l'infiltration : Récit	2000
Demain ne meurt pas aujourd'hui : TED talk	0
Cours de torture pour débutants	0

A.1.14 R14

```

1 —14. Le nom et le prenom du scientifique qui n a jamais encadre
2
3 SELECT s.Nom,s.Prenom
4 FROM(SELECT idPersonnel,Nom,Prenom from Personnel) as s
5 JOIN ((SELECT idScientifique from Scientifique)
6 EXCEPT
7 (SELECT idScientifique from ScientifiqueEncadreDoctorant))as e
8 ON e.idScientifique=s.idPersonnel

```

Nous obtenons le résultat suivant :

nom	prenom
Einstein	Albert
Kine	Miss

A.1.15 R15

```

1 —15. Pour une annee donnee, on veut recuperer le nombre de publications , de conferences , et de
2 doctorants de chaque scientifique
3 SELECT s.idScientifique , NbPubli,NbConf,NbDocts
4 FROM(SELECT idScientifique FROM Scientifique) as s
5 LEFT JOIN (SELECT idPresident ,COUNT(*) as NbConf FROM Conference WHERE EXTRACT(YEAR FROM date_debut)
6 =2018 GROUP BY idPresident) as c
7 ON s.idScientifique=c.idPresident
8 LEFT JOIN (SELECT idPersonnel ,COUNT(*) as NbPubli FROM PersonnelPublie WHERE idPublication IN(SELECT
9 idPublication FROM Publication WHERE annee_publication=2018) GROUP BY idPersonnel) as p
10 ON s.idScientifique=p.idPersonnel
11 LEFT JOIN (SELECT idScientifique ,COUNT(*) as NbDocts FROM ScientifiqueEncadreDoctorant WHERE
12 idDoctorant IN(SELECT idDoctorant FROM Doctorant WHERE EXTRACT(YEAR FROM date_soutenance)=2018)
13 GROUP BY idScientifique) as d
14 ON s.idScientifique=d.idScientifique

```

Nous obtenons le résultat suivant :

idScientifique	NbPubli	NbConf	NbDocts
6	NULL	NULL	1
7	NULL	NULL	NULL
10	2	NULL	NULL
13	NULL	NULL	NULL
999	1	1	NULL
118518	NULL	NULL	NULL

A.1.16 R16

```

1  —16.    Le nom et le prenom du scientifique qui n a jamais publie , encadre , ni participe a des projets .
2
3  SELECT Nom,Prenom from Personnel p WHERE p.idPersonnel=
4  (((SELECT idScientifique from Scientifique)
5  EXCEPT
6  (SELECT idScientifique from ScientifiqueEncadreDoctorant))
7  EXCEPT
8  (SELECT idScientifique from ScientifiqueParticipeProjet))
9  EXCEPT
10 SELECT idPersonnel from PersonnelPublie)

```

Nous obtenons le résultat suivant :

nom prenom
Kine Miss

A.1.17 R17

```

1  —17.    Afficher pour chaque scientifique , le nombre de ses publications , le nombre de ses projets et
2           de ses doctorants .
3
4  SELECT s.idScientifique , NbPubli,NbProjet,NbDocts
5  FROM(SELECT idScientifique FROM Scientifique) as s
6  LEFT JOIN (SELECT idPersonnel , COUNT(*) as NbPubli FROM PersonnelPublie GROUP BY idPersonnel) as p
7  ON s.idScientifique=p.idPersonnel
8  LEFT JOIN (SELECT idScientifique , COUNT(*) as NbProjet FROM ScientifiqueParticipeProjet GROUP BY
9  idScientifique) as pr
10 ON s.idScientifique=pr.idScientifique
11 LEFT JOIN (SELECT idScientifique , COUNT(*) as NbDocts FROM ScientifiqueEncadreDoctorant GROUP BY
12 idScientifique) as d
13 ON s.idScientifique=d.idScientifique

```

Nous obtenons le résultat suivant :

idScientifique	NbPubli	NbProjet	NbDocts
6	NULL	NULL	3
7	1	NULL	1
10	3	1	2
13	NULL	NULL	NULL
999	1	NULL	NULL
118518	NULL	2	2

A.1.18 R18

```

1  —18.    Les scientifiques qui ont que des doctorants ayant soutenus et pas de doctorant en cours
2
3
4  SELECT idScientifique FROM ScientifiqueEncadreDoctorant
5  EXCEPT
6  SELECT idScientifique FROM ScientifiqueEncadreDoctorant
7  WHERE (idDoctorant IN (SELECT idDoctorant FROM Doctorant WHERE (date_soutenance >= CURRENT_DATE)))

```

Nous obtenons le résultat suivant :

```
idscientifique
-----
(0 rows)
```

A.1.19 R19

```
1  —19.    Pour chaque scientifique , le nombre de ses collaborateur externes
2
3
4  SELECT s.idScientifique ,COUNT(p.idAuteur) as NbCollab
5  FROM Scientifique s
6  JOIN (SELECT idPersonnel , idAuteur
7  FROM PersonnelPublie
8  JOIN AuteurLaboPublie
9  ON idPublication=idPubli) as p
10 ON p.idPersonnel=s.idScientifique
11 GROUP BY idScientifique
```

Nous obtenons le résultat suivant :

idScientifique	NbCollab
10	1

A.1.20 R20

```
1  —20.    Les scientifiques qui encadrent mais n ont pas de doctorants ayant deja soutenu
2
3  SELECT idScientifique FROM ScientifiqueEncadreDoctorant WHERE idDoctorant IN(SELECT idDoctorant FROM
   Doctorant WHERE (date_soutenance > CURRENT_DATE))
```

Nous obtenons le résultat suivant :

idScientifique
10
118518
7
6

A.1.21 R21

```
1  —21.    Le nombre de collaborateurs par pays
2
3  SELECT pn.Pays , COUNT(idAuteur)
4  FROM(SELECT Pays,Nom FROM Labo_externe) as pn
5  LEFT JOIN(SELECT idAuteur , NomLabo FROM Auteur) as a
6  ON pn.Nom=a.NomLabo
7  GROUP BY pn.Pays
```

Nous obtenons le résultat suivant :

pays	count
France	3
URSS	1
Estonie	1

A.1.22 R22

```

1  —22.  Les doctorants qui ont un seul encadrant et qui ont toujours des publications qu avec leur
      encadrant
2
3  SELECT idDoctorant FROM Doctorant
4  WHERE idDoctorant IN (
5  SELECT idPersonnel FROM PersonnelPublie
6  WHERE idPublication IN (
7
8  (SELECT idPublication FROM PersonnelPublie
9  WHERE idPersonnel in
10 (SELECT idScientifique FROM ScientifiqueEncadreDoctorant
11 GROUP BY idScientifique)
12
13 INTERSECT
14
15 SELECT idPublication FROM PersonnelPublie
16 WHERE idPersonnel in
17 (SELECT idDoctorant FROM ScientifiqueEncadreDoctorant
18 GROUP BY idDoctorant
19 HAVING COUNT(*)=1 ))))

```

Nous obtenons le résultat suivant :

```

iddoctorant
-----
      121314
(1 row)

```

A.1.23 R23

```

1  —23.  Les doctorants qui ont plus de 3 encadrants
2
3  SELECT idDoctorant FROM ScientifiqueEncadreDoctorant GROUP BY idDoctorant HAVING COUNT(idScientifique)
      >3

```

Nous obtenons le résultat suivant :

```

idDoctorant
-----
1

```

A.1.24 R24

```

1  —24.  Les scientifiques qui ont plus de 3 doctorants qui ont d but leur thse il y a moins de 2 ans.
2
3  — NON TESTABLE ACTUELLEMENT !!!!!!!
4
5  SELECT s.idScientifique FROM (
6  SELECT idScientifique , COUNT(idDoctorant) as NbDocts FROM ScientifiqueEncadreDoctorant
7  WHERE idDoctorant IN (
8  SELECT idDoctorant FROM Doctorant
9  WHERE idDoctorant IN (
10 SELECT idPersonnel FROM Personnel
11 WHERE EXTRACT(YEAR FROM Date_recrutement)>=(EXTRACT(YEAR FROM CURRENT_DATE)-2)))
12 GROUP BY idScientifique) as s
13 WHERE s.NbDocts>3

```

Nous n'avons malheureusement pas pu tester cette requête, car le jeu de données n'était pas suffisant pour celle-ci.

A.1.25 R25

```

1  —25.  Les doctorants qui ont au moins une publication chaque annee depuis leur recrutement
2
3  SELECT np.idPersonnel
4  FROM
5  (SELECT DISTINCT dp.idPersonnel , COUNT(pa.annee_publication) as ap

```

```

6 FROM(SELECT idPersonnel , idPublication FROM PersonnelPublie
7 WHERE idPersonnel IN (SELECT idDoctorant FROM Doctorant))as dp
8 JOIN
9 (SELECT idPublication , annee_publication FROM Publication) as pa
10 ON dp.idPublication=pa.idPublication
11 GROUP BY dp.idPersonnel)as np
12 JOIN
13 (SELECT idPersonnel , EXTRACT(YEAR FROM CURRENT_DATE)-EXTRACT(YEAR FROM Date_recrutement) as Nbannees
14 FROM Personnel) as tp
15 ON tp.idPersonnel=np.idPersonnel
16 WHERE np.ap = tp.Nbannees

```

Nous obtenons le résultat suivant :

idPersonnel
1

A.1.26 R26

```

1 —26. Les scientifiques qui recrutent au moins un doctorant par annee
2
3
4 SELECT da.idScientifique FROM
5 (SELECT sd.idScientifique , COUNT(sd.DateDoc) as NbDocs, AVG(EXTRACT( YEAR FROM p.Date_recrutement)) as
6 DateSci
7 FROM (SELECT idPersonnel , Date_recrutement FROM Personnel) as p
8 JOIN (SELECT DISTINCT s.idScientifique , EXTRACT(YEAR FROM d.Date_recrutement) as DateDoc
9 FROM
10 (SELECT idScientifique , idDoctorant FROM ScientifiqueEncadreDoctorant) as s
11 JOIN
12 (SELECT idPersonnel ,Date_recrutement FROM Personnel WHERE idPersonnel IN (SELECT idDoctorant FROM
13 Doctorant)) as d
14 ON s.idDoctorant=d.idPersonnel )as sd
15 ON sd.idScientifique=p.idPersonnel
16 GROUP BY sd.idScientifique) as da
17 WHERE (da.nbdocs>= (EXTRACT(YEAR FROM CURRENT_DATE) - da.DateSci))

```

Nous obtenons le résultat suivant :

idscientifique
6

A.1.27 R27

```

1 —27. Les pays qui sont p r sents tous les projets
2
3 SELECT pc.Pays
4 FROM (SELECT DISTINCT COUNT(ppp.TitreProjet) as nbProj , pa.Pays
5 FROM
6 (SELECT DISTINCT Pays , idPartenaire FROM Partenaire)as pa
7 JOIN
8 (SELECT * FROM PartenaireParticipeProjet)as ppp
9 ON ppp.idPartenaire=pa.idPartenaire
10 GROUP BY pa.Pays) as pc
11 WHERE pc.nbProj=(SELECT COUNT(*) FROM Projet_recherche)

```

Nous obtenons le résultat suivant :



A.1.28 R28

```
1  —28.    Les scientifiques qui publient que dans des conférences de classe A
2
3  SELECT idScientifique FROM Scientifique WHERE idScientifique IN(
4  (SELECT idPersonnel FROM PersonnelPublie WHERE idPublication IN (SELECT idPublication FROM Publication
5  WHERE nom_conference_journal IN (SELECT nom_conference_journal FROM TypeConf WHERE
6  classe_conference='A'))))
7  EXCEPT
8  (SELECT idPersonnel FROM PersonnelPublie WHERE idPublication IN (SELECT idPublication FROM Publication
9  WHERE nom_conference_journal IN (SELECT nom_conference_journal FROM TypeConf WHERE
10  classe_conference!='A'))))
```

Nous obtenons le résultat suivant :

idscientifique
999

A.1.29 R29

```
1  —29 Les scientifiques qui n'ont jamais publié dans des conférences de classe A
2
3  (SELECT idScientifique FROM Scientifique)
4  INTERSECT
5  (SELECT idPersonnel FROM PersonnelPublie)
6  EXCEPT
7  (SELECT idPersonnel FROM PersonnelPublie WHERE idPublication IN (SELECT idPublication FROM Publication
8  WHERE nom_conference_journal IN (SELECT nom_conference_journal FROM TypeConf WHERE
9  classe_conference='A'))))
```

Nous obtenons le résultat suivant :

idscientifique
7

(1 row)

A.1.30 R30

```
1  —30.    Le nombre de conférences de classe A organisées par le laboratoire par année
2
3  SELECT cd.annee, COUNT(cd.Nom_conf) as NbConf FROM
4  (SELECT Nom_conf, EXTRACT(YEAR FROM date_debut) as annee FROM Conference) as cd
5  LEFT JOIN
6  (SELECT nom_conference_journal FROM TypeConf WHERE classe_conference='A') as na
7  ON na.nom_conference_journal=cd.Nom_conf
8  GROUP BY cd.annee
```

Nous obtenons le résultat suivant :

annee	nbconf
2019	1
2018	2
2001	1
2000	1

A.1.31 R31

```
1  —31.  L  tablissement  d'enseignement ayant le plus grand nombre d'enseignant chercheur
2
3  SELECT idcnt.idEtablissement FROM
4  (SELECT idEtablissement ,COUNT(*) as cntEnseignants
5   FROM Enseignant_chercheur
6   GROUP BY idEtablissement) as idcnt
7
8  WHERE idcnt.cntEnseignants=(SELECT max(nc.cntEnseignants)
9  FROM (
10   SELECT idEtablissement ,COUNT(*) as cntEnseignants
11   FROM Enseignant_chercheur
12   GROUP BY idEtablissement) as nc)
```

Nous obtenons le résultat suivant :

```
idtablissement
-----
51
(1 row)
```