3-2 public List <Float> addMark (int sid, int cid, float Mark) throws SQLException {

List <Float> studentMarks = new ArrayList <Float> ;

String query = " ~~select~~ sid from Enrolled where sid=?" +
" and cid=? and mark=?" ;

~~PreparedStatement~~ stmt = connection . prepareStatement (query);

Stmt . ~~setInt~~ (1, sid);
Stmt . setInt (2, cid);
Stmt . setFloat (3, mark);

ResultSet rs = Stmt . ~~execute~~Query ();

Update

while (rs . next ()) {

studentMarks = add . getFloat (" mark ");

}

return studentMarks;

}

.5

ECE PARIS
ÉCOLE D'INGÉNIEURS

DEVOIR SURVEILLÉ

CADRE À COMPLETER PAR L'ÉLÈVE

NOM/Prénom : GRIERE Lisa

PROMOTION : 2019 GROUPE : 5105 DATE : 12/12/17

INTITULÉ DE L'ÉPREUVE : Database

FEUILLE N° : 1 /

NOTE OBTENUE 11

REMARQUES

Marge réservée au professeur correcteur

Exercice 1 /2

(1) Select sname from Students
natural join Enrolled
where sid not in
(select sid from Enrolled
where mark < 10) ; /1

(2) Select sname from Students
~~natural join Enrolled~~ where
~~group by~~ sid
~~having~~ sid not in
(select sid from Enrolled group by sid
~~where~~ having max(mark) < 10) ; not enough 0

(4) select cname, count(~~distinct~~(sid)), max (mark)
from courses
natural join Enrolled
group by cid ; /1

(5) select ~~cyear~~, cname from courses c1
where duration =
(select max(duration) from courses c2
where c1. ~~cid~~ = c2. ~~cid~~) ; 0
cyear

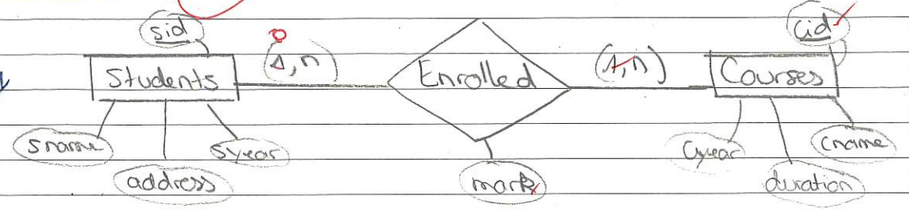
③ select sname from Students S1
natural join Enrolled e1 natural join Courses C1
where S.id not in (select sid from Enrolled
where S1.syear = S2.syear)

0

Exercice 2

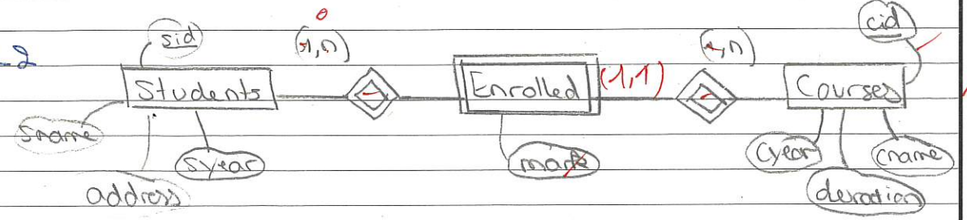
3

2.1



1

2.2



1,5

2.3 The enrolled table can be represented as a relationship between the tables Students and Courses because enrolled's primary key is the combination of these 2 other tables' primary key.
It can also be represented as a weak entity set because its primary key depends entirely of 2 foreign keys.

,5

Exercice 4:

4

① db.courses.aggregate([{$group: {_id: null, maxDuration: {$max: "$duration"}}}]);

1

② db.courses.aggregate([{$match: {year: 3}}, {$group: {_id: "$name"}}, {$count: 0}])

0

③ db.courses.aggregate([{$group: {_id: "$year", totalDuration: {$sum: "$duration"}}}]);

1

④ db.courses.aggregate([{$group: {_id: "$year", totalDuration: {$sum: "$duration"}}}, {$group: {_id: null, maxtotalDuration: {$max: "$totalDuration"}}}]);

1

⑤ db.courses.aggregate([{$match: {year: 4}}, {$match: {name: /^.*d.*d/}}]);

1

Exercice 3:

2

3.1

```
public List<Float> getMarks(int year)
    throws SQLException {

    List<Float> marks = new ArrayList<Float>();

    String query = "select mark from Enrolled natural join Students" +
                   " where syear = ?";

    CreateStatement stmt = connection.createStatement();

    stmt.setInt(1, year);

    ResultSet rs = Stmt.ExecuteQuery(query);

    while (rs.next()) {
        marks = add.getFloat("mark");
    }
    return marks;
```

1,5