The relational schema below, which will be referred to as the WORKING SCHEMA, keeps information of a literary competition (note that data types are not included):

```
Country(country_code, name)
 PK:{country_code}
 NNV:{name}

Judge(number, name, degree, country_code)
 PK:{number}
 FK:{country_code}→ Country(country_code)
   Delete in CASCADE and Update in CASCADE
 NNV:{name, degree}

Book(book_code, title, pages, resp_person)
 PK:{book_code}
 NNV:{title, pages, resp_person}
 FK:{resp_person} → Judge(number)
   Delete in CASCADE and Update in CASCADE
 Uni:{resp_person}

Read(book_code, number, date)
 PK:{book_code, number}
 NNV:{date}
 FK:{book_code} → Book
   Delete in CASCADE and Update in CASCADE
 FK:{number} → Judge
   Delete in CASCADE and Update in CASCADE

Review(code, date, comment, book_code, number)
 PK:{code}
 NNV:{book_code}
 FK:{book_code, number} → Read
   Referential Integrity TO BE DEFINED
   Delete TO BE DEFINED and Update in CASCADE
```

Where the relations have the following meaning:

Country:

- *country_code*: Country identifier
- *name*: Name of the country

Judge:

- *number*: Judge identifier
- *name*: Name of the judge
- *degree*: University degree of the judge
- *country_code*: Identifier of the country where the judge was born

Book:

- *book_code*: Book identifier
- *pages*: Number of pages of the book
- *title*: of the book
- *resp_person*: Identifier of the judge responsible for the book

Read: The judge *number* has been assigned to read the book *book_code* on day *date*. The judge will review the book.

Review: The judge *number* reviewed the book *book_code* on day *date.* The review is identified by its *code* and can include a *comment.*

Consider the following extension of the previous schema. The *id* columns are not part of the database, they are used to refer to the tuples.

BOOK

id	book_code	title	pages	resp_person
t1	O14	El mar	123	1
t2	O26	La paz	222	5
t3	O17	Hoy, no	650	2
t4	O43	Mañana	55	4

READ

id	book_code	number	date
t5	O14	1	4/4/17
t6	O14	2	5/4/17
t7	O26	3	9/4/17
t8	O26	1	5/4/17
t9	O17	3	9/4/17

JUDGE

id	number	name	degree	country_code
t10	1	Alfonso Peris	Philology	jhe09
t11	2	María Llopis	Philosophy	jhe09
t12	3	Juao Portao	Philology	kjh78
t13	4	Pierre Rius	Literature	xyz45
t14	5	Ana Pardo	Literature	

REVEW

id	code	comment	date	book_code	number
t15	1111	Excellent	5/4/17	O14	1
t16	2222			O14	
t17	3333			O26	
t18	4444	Great	10/4/17	O26	3
t19	5555	Poor	11/4/17	O17	3

COUNTRY

id	country_code	name
t20	xyz45	France
t21	kjh78	Portugal
t22	jga65	Italy
t23	jhe09	Spain

NOTE: The queries in the Unit 2 part shouldn't take into account the database content shown above.

DATABASES AND INFORMATION SYSTEMS (7/10/19)

LASTNAME, NAME: ____

P2: UNIT 2 (100 points)

1) Considering the working schema, indicate if the following sentences are **true** or **false**. Justify your answer. (30 points)

a. Every book has a unique code.
True, because PK:{book_code} in Book

b. Every judge has a known number.
True, because PK:{number} in Judge

c. Every review has a known date.
False, because date can be NULL in Review

d. One judge can only read one book.
False, because the PK:{book_code, number} contains two attributes, therefore is possible to have the same number (of judge) several times (with different book_code).

e. One judge can be the responsible person of several books.
False, because Uni:{resp_person} in Book

f. Every book must have at least one judge assigned to read it.
False, it is possible to have a book_code in Book that is not included in Read

g. One book has one and only one responsible judge.
True. One book always has a judge because NNV:{resp_person} and there is only one attribute to indicate the responsible, therefore only one judge can be assigned to a book.

h. The responsible person of a book must be one of the judge that must read the book.
False. No constraint is checking it.

i. A judge can only be assigned to read a book if there is at least one review of the book.
False. No constraint is checking it.

j. A review must be always about a book included in the *Book* relation.
It depend on the referential integrity used for FK:{book_code, number} $\rightarrow$ Read. If the R.I. is Weak, it is possible to have a Review with a NULL value and then, the value of the number attribute does not have to be in the Judge relation. If the R.I. is Weak or Full, the book_code must be in the Book relation.

2) Answer the following questions: (5 points)

a. What is the *Judge* cardinality *?*
5

b. What is the *Judge* degree?
4

3) Fill the cells in the following table with YES or NO indicating whether the referential integrity would be fulfilled in each of the possible cases of Referential Integrity (IR) when the tuple with: *code=6666, date='3/4/17', book_code=X*, is inserted in the *Review* relation. *X* will take the values indicated in the table (15 points).

X	Weak R.I.	Partial R.I.	Full R.I.
O20	YES	NO	NO
O43	YES	NO	NO
O14	YES	YES	NO

4) Consider the database of the tables above, and assume the following cases for the foreign key in the *Review* relation. For each case, indicate which tuples will be removed from the database when we delete the tuple *t10*. Use the tuple *id*s (t1 to t23) to answer the questions. (20 points).
   a. Weak R.I and on DELETE CASCADE
      The tuples t10, t1, t5, t6, t8, and t15 are deleted

   b. Partial R.I. and on DELETE CASCADE
      The tuples t10, t1, t5, t6, t8, t15, and t16 are deleted

5) What is the maximum cardinality that the *Read* relation can have? Express this cardinality in terms of the cardinality of other relations. (10 points)
   Card(Book) x Card(Judge)

6) Which of the following expressions of Relational Algebra represent the query: "Numbers of the judges who are not the responsible person of any book? (10 points)
   a. Book[resp_person](resp_person, number) – Judge[number]
   b. (Book WHERE IsNull(number) [resp_person](resp_person, number)$\otimes_{\text{number}}$ Judge) [number]
   c. Judge[number] – Book[resp_person](resp_person, number)
   d. (Book $\times$ Judge) WHERE resp_person<>number [number]

7) Which query represent the following expression? (10 points)

((Read[book_code, number] $\otimes_{\text{book\_code}}$ Read[book_code, number] (number, ZZ)) WHERE ZZ<>number)[book_code]

Book_code of the Books which have been assigned to be read by more than one judge

---

P2: UNIT 2 (200 points)

1) Solve the following queries in SQL:
   a. List how many reviews are with no comment. (20 points)
   b. List the title and code of the books assigned to be read by some judge. List the books in alphabetical order by title. (30 points)
   c. List the number and name of the judges assigned to read a book containing the word "water" in the title. (30 points)
   d. List the code and title of the books with a number of pages greater than the average number of pages of the books. (30 points)
   e. List the number and name of the judges who haven't reviewed any book. (40 points)
   f. List the number and name of the judges assigned to read a book which has received more than 5 reviews. (50 points)

ANSWERS

a)

```
SELECT COUNT(code)
FROM Review
WHERE comment IS NULL;
```

b)

```
SELECT DISTINCT B.title, B.book_code
FROM Book B, Read R
WHERE B.book_code = R.book_code
ORDER BY B.title ;
```

Also:

```
SELECT B.title, B.book_code
FROM Book B
WHERE B.book_code IN (SELECT R.book_code
                      FROM Read R )
ORDER BY B.title ;
```

Also:

```
SELECT B.title, B.book_code
FROM Book B
WHERE EXISTS (SELECT *
              FROM Read R
              WHERE R.book_code = B.book_code )
ORDER BY B.title ;
```

c)

```
SELECT DISTINCT J.number, J.name
FROM Judge J, Read R, Book B
WHERE J.number = R.number
  AND R.book_code = B.book_code
  AND B.title LIKE ‘%water%’;
```

Also:

```
SELECT J.number, J.name
FROM Judge J,
WHERE J.number IN (SELECT R.number
                   FROM Read R, Book B
                   WHERE R.book_code = B.book_code
                     AND B.title LIKE ‘%water%’ );
```

Also:

```
SELECT J.number, J.name
FROM Judge J,
WHERE EXISTS (SELECT *
              FROM Read R, Book B
              WHERE R.book_code = B.book_code
                AND R.number = J.number
                AND B.title LIKE ‘%water%’ );
```

d)

```
SELECT B.book_code, B.title
FROM Book B
WHERE pages > (SELECT AVG(pages)
               FROM Book);
```

e)

```
SELECT J.number, J.name
FROM Judge J
WHERE J.number NOT IN ( SELECT number
                        FROM Review
                        WHERE number IS NOT NULL );
```

Also:

```
SELECT J.number, J.name
FROM Judge J
WHERE NOT EXISTS ( SELECT *
                   FROM Review R
                   WHERE J.number = R.number );
```

f)

```
SELECT DISTINCT J.number, J.name
FROM Judge J, Read R
WHERE J.number = R.number
  AND 5 < (SELECT COUNT(*)
           FROM Review REV
           WHERE REV.book_code = R.book_code);
```

Also:

```
SELECT J.number, J.name
FROM Judge J
WHERE J.number IN (SELECT R.number
                   FROM Read R
                   WHERE 5 < ( SELECT COUNT(*)
                               FROM Review REV
                               WHERE REV.book_code = R.book_code);
```