Question 1. [4 MARKS]

Consider the following database:

Item

IID	price
1	5
4	6
2	8
3	4
7	12
8	3
9	1

OrderItem

OID	IID
2	1
2	2
8	7
1	9
9	3
1	3
8	3
1	1

Rating

CID	IID	number
12	4	3
15	2	5
3	4	1
6	1	3
3	8	5
12	8	6
8	3	3
6	7	8

Give the result of following queries, using the same tabular format as above; do **not** describe the result in English. Assume the set semantics (not bag semantics) for Relational Algebra.

Part (a) [2 MARKS]

$((\Pi_{IID} Item) - (\Pi_{IID} OrderItem)) \bowtie (\sigma_{number>3} Rating)$

Solution:

CID	IID	number
3	8	5
12	8	6

Part (b) [2 MARKS]

$One := OrderItem \bowtie Item$

$Two(OID, IID) := \Pi_{A.OID,A.IID}(\sigma_{A.OID=B.OID \wedge A.price<B.price}(\rho_A One \times \rho_B One)$

$Answer := (\Pi_{OID,IID} One) - Two$

Solution:

OID	IID
2	2
8	7
9	3
1	1

Question 2. [12 marks]

Consider the following schema for an online system that runs tests. It is capable of representing only a single test.

Relations

- Question(QID, questionText, correctAnswer)
- Response(QID, SID, answer, timestamp)
- Student(SID, lastname, firstname)

Integrity constraints

- Response[QID] $\subseteq$ Question[QID]
- Response[SID] $\subseteq$ Student[SID]

Write the following queries using only the basic Relational Algebra operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$. Assume the set semantics (not bag semantics) for Relational Algebra.

1. Report the first name, last name, and SID of the first student(s) to get question 19711 correct.
   Solution:

   $Correct := \Pi_{SID,timestamp}\sigma_{answer=correctAnswer \wedge QID=19711}(Response \bowtie Question)$
   $Beaten(SID) := \Pi_{C1.SID}\sigma_{C1.timestamp>C2.timestamp}(\rho_{C1}Correct \times \rho_{C2}Correct)$
   $Answer := (\Pi_{SID}Correct - Beaten) \bowtie Student$

2. Find the SID of students who answered every question, but didn't have any correct answers.
   Solution:

   $ShouldHave := \Pi_{QID,SID}(Question \times Student)$
   $DidAnswer := \Pi_{QID,SID}Response$
   $MissedSome := \Pi_{SID}Student(ShouldHave - DidAnswer)$
   $AnsweredAll := (\Pi_{SID}Student) - MissedSome$
   $SomeCorrect := \Pi_{SID}\sigma_{answer=correctAnswer}(Response \bowtie Question)$
   $NoneCorrect := (\Pi_{SID}Student) - SomeCorrect$
   $Answer := AnsweredAll \cap NoneCorrect$

Question 3. [8 marks]

This question uses the same schema as the previous question. It is repeated here for convenient reference:

Relations

- Question(QID, questionText, correctAnswer)
- Response(QID, SID, answer, timestamp)
- Student(SID, lastname, firstname)

Integrity constraints

- Response[QID] $\subseteq$ Question[QID]
- Response[SID] $\subseteq$ Student[SID]

Which of the following statements are enforced by the schema? Circle one answer for each. If the statement is enforced, say what part of the schema enforces it. If it is not enforced, write an integrity constraint that would enforce it (using one of the two forms defined in the texbook).

1. For each student, at most one answer can be recorded for each question.

   [✓ Enforced] This part of the schema enforces it:
   QID, SID is a key for Response, so you can't repeat the same QID and SID with two different answers.

   Not enforced This new integrity constraint would enforce it:

2. The same question text can't be associated with two different questions (QIDs).

   Enforced This part of the schema enforces it:

   [✓ Not enforced] This new integrity constraint would enforce it:

   $\sigma_{Q1.QID \neq Q2.QID \wedge Q1.questionText=Q2.questionText}(\rho_{Q1}Question \times \rho_{Q2}Question) = \emptyset$

3. One question (one QID) can't have two different correct answers.

[✓ Enforced] This part of the schema enforces it:
QID is a key for Question, so you can't repeat the same QID with two different values for correctAnswer.

Not enforced This new integrity constraint would enforce it:

4. Every question has been answered by at least one student.

Enforced This part of the schema enforces it:

[✓ Not enforced] This new integrity constraint would enforce it:

$(\Pi_{QID} Question) \subseteq (\Pi_{QID} Response)$

Question 4. [8 marks]

Suppose relation Guesses contains the values shown below.

```
 number |   name    | guess | age
--------+-----------+-------+-----
      1 | Cole      |   365 |   5
      2 | Avery     |   585 |   5
      3 | Sam       |   502 |  12
      4 | Madeleine |   511 |  18
      5 | Cole      |   450 |   5
      6 | Michael   |  1000 |  12
      7 | Mackenzie |   700 |   5
      8 | Mackenzie |   701 |   5
      9 | Micah     |   498 |   4
     10 | Jiaqi     |   509 |   4
     11 | Jamieson  |   502 |   6
```

Part (a) [2 marks]

Show the result of the following SQL query:

```
select name as "guesser"
from Guesses
where age < 10 and number < 10;
```

Solution:

```
  guesser
-----------
 Cole
 Avery
 Cole
 Mackenzie
 Mackenzie
 Micah
```

Part (b) [2 marks]

Show the result of the following SQL query:

```
select g1.name, g2.name
from Guesses g1, Guesses g2
where g1.name <> g2.name and g1.guess = g2.guess;
```

Solution:

```
 name    |   name
----------+----------
 Jamieson | Sam
 Sam      | Jamieson
```

Part (c) [2 marks]

Show the result of the following SQL query. Note that `@` is a function that returns the absolute value.

```
select *
from Guesses g1
where not exists (
   select * from Guesses g2
   where @(g2.guess - 500.0) < @(g1.guess - 500.0)
);
```

Solution:

```
 number |   name   | guess | age
--------+----------+-------+-----
      3 | Sam      |   502 |  12
      9 | Micah    |   498 |   4
     11 | Jamieson |   502 |   6
```

Part (d) [2 MARKS]

Consider the following SQL query:

```
select _______________
from Guesses
group by age;
```

Which of the following could go in the `select` list for this query? Circle "Okay" or "Error" for each. 0.5 marks each.

Solution:

`number`	Okay	[✓ Error]
`count(name)`	[✓ Okay]	Error
`max(guess)`	[✓ Okay]	Error
`age`	[✓ Okay]	Error