Q1. (SQL) For each department, list the department number, name, and the average salary for the department employees.

```
select d.dept_no, dept_name, avg(salary)
from ex1departments d join ex1dept_emp de on d.dept_no = de.dept_no
join ex1employees e on e.emp_no = de.emp_no
group by dept_no, dept_name;
```

Q2. (SQL) List office number, building, and floor for any office whose employees' average age is less than the average age for the whole company.

```
SELECT o.office_no, building, floor, avg(age) as avg_age
FROM ex1offices o join ex1employees e on o.office_no = e.office_no
Group by o.office_no, building, floor
having avg(age) < (select avg(age) from ex1employees);
```

Q3. list department number, name, and budget for departments whose total employee salaries exceeds (is greater than) one tenth (1/10) of the department budget.

```
select d.dept_no, dept_name, budget
from ex1departments d
where budget/10 <
    (select sum(salary)
     from ex1dept_emp de join ex1employees e on e.emp_no = de.emp_no
     where de.dept_no = d.dept_no)
```

Q4. Using the union operator, write a relational algebra query to find the names of employees who work in both the Accounting and Research-and-

↳ I should have included emp-id as name is not unique (two employees can have the same name)

Development departments. Write your relational algebra on paper (or use your favorite tool), take a photo (or scan) your solution, and upload the solution using the add-file button below.

$R_1 := \Pi_{name}(\sigma_{dept\text{-}name=\text{"Research-and-Development"}}(ex1employees \bowtie ex1dept\text{-}emp \bowtie ex1departments),$

$R_2 := \Pi_{name}(\sigma_{dept\text{-}name=\text{"Accounting"}}(ex1employees \bowtie ex1dept\_emp \bowtie ex1departments))$

$R_3 := R_1 \cap R_2$

Q5. Consider relations r(A B C) and s(D E F). The leftyjoin of r with s on r.C = s.D. written:

r ¥r.C=s.D s consists of every tuple in r that joins with at least one tuple in s. For example, if r and s are given as:

r

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c3
a2	b2	c2

s

D	E	F
c1	e1	f1
c3	e1	f2
c3	e2	f2
c4	e2	f1
c5	e1	f2

Then r ¥r.C=s.D s is

A	B	C
a1	b1	c1
a2	b1	c3

a. Compute s ¥r.C=s.D r
Write your answer in the text box below.

D	E	F
c_1	e_1	f_1
c_3	e_1	f_2
c_3	e_2	f_2

b. Show how leftyjoin can be computed by the relational algebra operators we already have (select, project, cross product, join, union, intersect, difference). Use the text box below to write your answer (including a description of any relational algebra you may want to use).

We can do a natural join of the two relations, then project the attributes of the left relation.

For example, we can write the expression in point a as:

$\Pi_{D,E,F}(s \bowtie r)$

Note: left outer join does not work for this question as it returns attributes of the right relation ~~and~~

True and False Questions

Question 6 (1 point) ✓ *Saved*

A table can have multiple keys.

● True
○ False

Question 7 (1 point) ✓ *Saved*

We can always update a view.

○ True
● False

Question 8 (1 point) ✓ *Saved*

Unknown AND (TRUE OR FALSE) = TRUE

○ True
● False

Question 9 (1 point) ✓ *Saved*

Having a private data center costs about the same as paying to use cloud resources.

○ True
● False

Question 10 (1 point) ✓ *Saved*

When transforming an ERD to a relational schema, a many-to-many relationship set in ERD between entity sets e1 and e2 translates to adding the primary key of e1 in the table that represents e2 and vice versa.

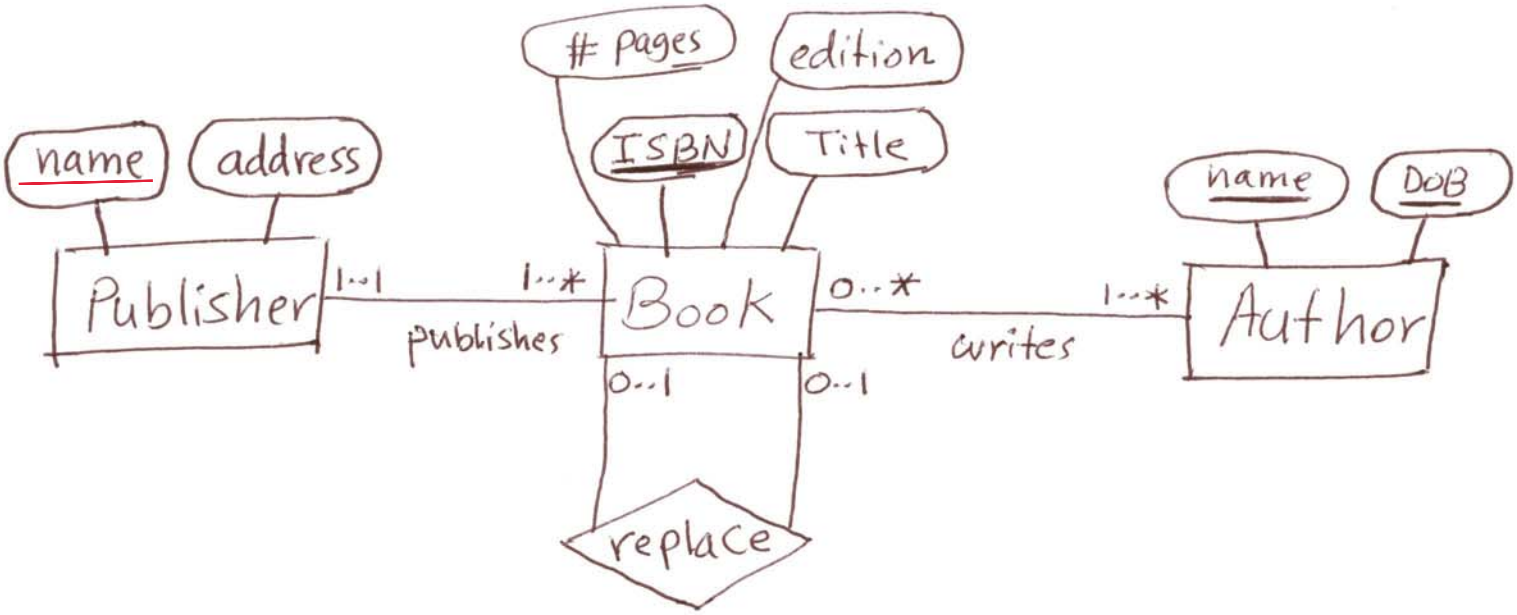
○ True
● False many-to-many relationship translates to a Separate table.

Question 11 (20 points)

Draw an ER diagram for the following scenario:

- There are books, identified by ISBN, and each book has a title, edition, and number of pages.
- There are publishers, identified by name and they have an address. A publisher can publish one or more books; a book is published by one publisher.
- Authors have names and birth date; authors may have the same name, but it is impossible for authors to have the same name and the same date of birth. Authors write books. For example, Avi Silberschatz has written the "Database System Concepts" and "Operating System Concepts" books. Note that Henry Korth and S. Sudarshan are also authors for the "Database System Concepts" book.
- A new edition of a book can replace an older edition. For example, the "Database System Concepts" 7th edition replaced the 6th edition of the book. The two editions have different ISBN numbers.

For the scenario above, use only three entities: Book, Publisher, and Author. Use only the description and the example data provided to decide what attributes, relationships, and cardinalities you include in your ERD. If the minimum cardinality is not clarified in the text, then you can use either 0 or 1. Draw your ERD on paper (or using your favorite tool) and upload a photo of your ERD using the add-file button below. Use the textbox, to write any notes you have.



# pages
edition
name
address
ISBN
Title
name
DoB
Publisher
1..1
1..*
Book
0..*
1..*
Author
publishes
writes
0..1
0..1
replace