Administrivia

HW2 solutions released: see HW2 repo

Extra credit 1 released

Thursday

- Midterm 1
- HW3 release
- Project 1 Part 3 release

Midterm Logistics

Exam: expected 75min in length

You may take full class time

Gradescope + Zoom

Zoom camera must be on the entire time.

Extra credit question.

Midterm Logistics

Open book and physical notes.

Writing tool allowed

No electronic resources.

No breaks except emergencies

Professor Wu will answer Qs in the **FIRST 60** minutes of exam.

Midterm Content

All material including SQL lectures

Columbia CS academic honesty policy

Excludes

- Details of hierarchical/network models (should know their pros and cons)
- Use of the Divide relational algebra operator

Midterm 1 Review

Topics

Data Models

- ER models
- Relational model
- Pros/Cons of hierarchical/network models
- ER → Relational Model translation

Relational Algebra

Topics

DDL statements

- CREATE TABLE statements
- Integrity Constraints

SQL

- Select, From, Where clauses
- Expressions
- Group by, Joins
- Conceptual evaluation
- WITH, Views, Tables

Relational Constraints

- Gives the DBMS a list of consistency checks
- It is run with respect to whatever data that exists in the database
  - DBMS doesn’t understand *anything* about what application wants or intends or “should” have
  - It simply goes through each constraint one by one and checks them against the data in the database

Relational Constraints

Domain constraints

Foreign key constraints

Unique constraints

Primary key constraints

NOT NULL constraints

CHECK constraints

ER → Relational translation

Translate entities and relationships into relational tables

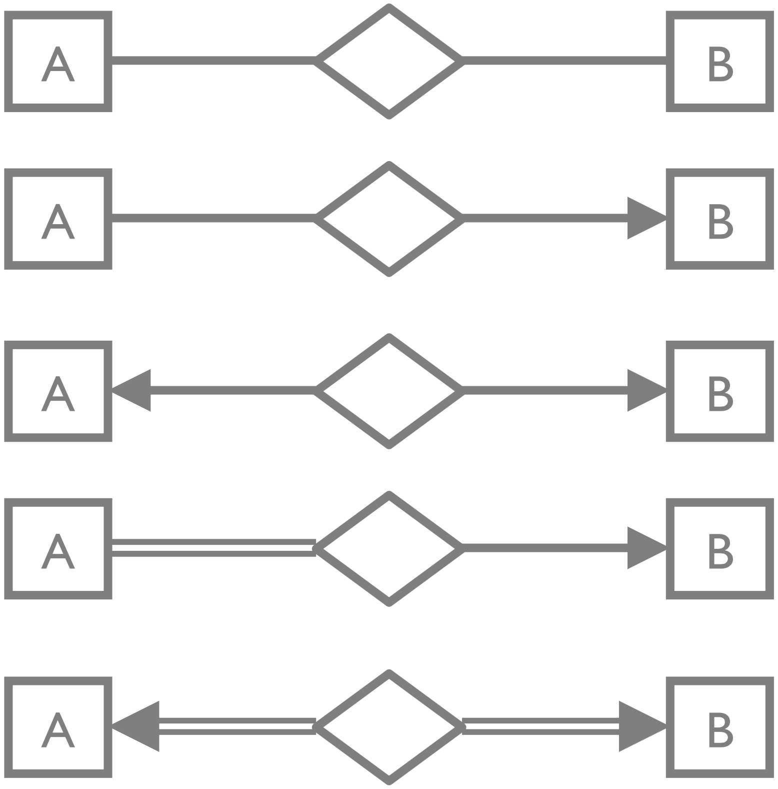
Translate ER constraints into relational constraints

The translation is correct if for *any* database instance:

- Constraints violated in ER are violated in relational
- Constraints violated in relational are violated in ER
- If ER doesn't violate, neither should relational
- If relational doesn't violate, neither should ER

Note: some translations are not possible

What constraints are translatable?



(notice there's an "at most 1" if there's an "at least 1")

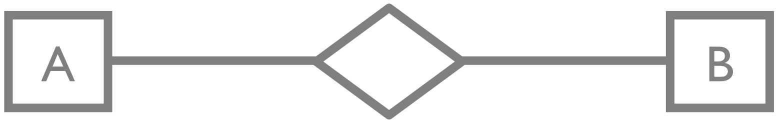
How to adhoc check a translation?

1. Come up with data that satisfies the ER constraints and check that they don’t violate relational version
2. For each ER constraint
    1. Come up with data that violated the ER constraint. Does relational version identify the violation given the same data?
3. Vice Versa

Non-exhaustive examples

(entity sets don’t include attrs for simplicity)

And how to check them



```
A(
 a int primary key,
 name text
)
```

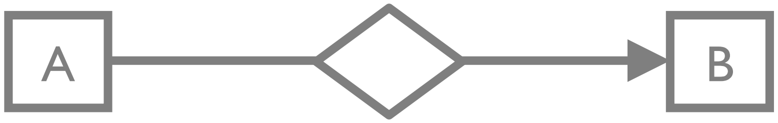
a	name
1	eve
2	bob

```
ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
)
```

a	b
1	3
1	4
2	3

```
B(
 b int primary key
)
```

b
3
4



```
A(
 a int primary key,
 name text
)
```

```
ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
 unique(a)
)
```

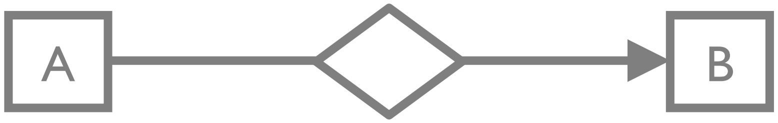
```
B(
 b int primary key
)
```

a	name
1	eve
2	bob

a	b
1	3
2	3

b
3
4

This database satisfies ER constraints.
Thus it should satisfy the relational constraints



A(
 a int primary key,
 name text
)

ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
 unique(a)
)

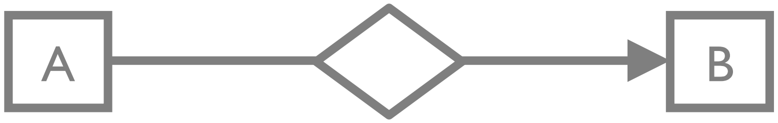
B(
 b int primary key
)

a	name
1	eve
2	bob

a	b
1	3
2	null

b
3
4

This database violates ER constraints.
Thus, it should violate the relational constraints



```
A(
 a int primary key,
 name text
)
```

a	name
1	eve
2	bob

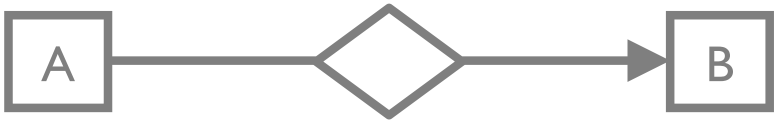
```
ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
 unique(a)
)
```

a	b
1	3
1	4
2	3

```
B(
 b int primary key
)
```

b
3
4

This database violates ER constraints.
Thus, it should violate the relational constraints



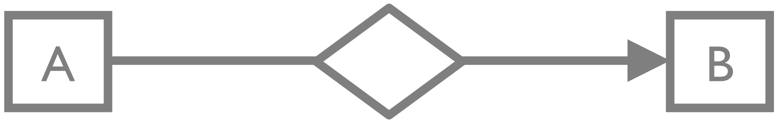
```
A(
 a int primary key,
 name text,
 b references B(b)
)
```

a	name	b
1	eve	3
2	bob	3

```
B(
 b int primary key
)
```

b
3
4

This database satisfies ER constraints.
Thus should satisfy relational constraints



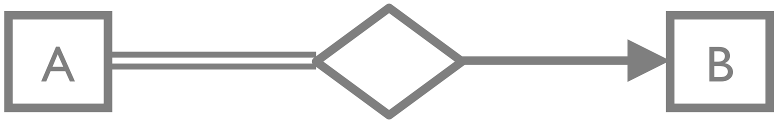
A(
 a int primary key,
 name text,
 b references B(b)
)

B(
 b int primary key
)

a	name	b
1	eve	3
2	bob	3
1	eve	4

b
3
4

This database **violates** ER constraints.
Thus should **violate** relational constraints



```
A(
 a int primary key,
 name text
)
```

a	name
1	eve
2	bob

```
ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
 unique(b)
)
```

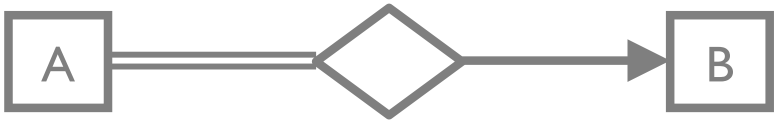
a	b
1	3
2	3

```
B(
 b int primary key
)
```

b
3
4

This database satisfies ER constraints.
BUT it doesn’t satisfy the relational constraints.

FAIL



Can this be expressed relationally?

```
A(
 a int primary key,
 name text
)
```

```
ab(
 a references A(a)
 b references B(b)
 primary key (a, b)
 unique(a)
)
```

```
B(
 b int primary key
)
```

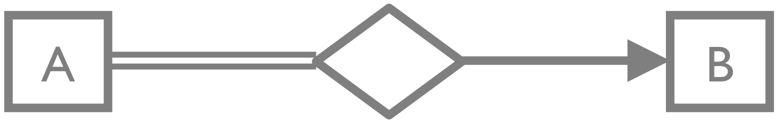
a	name
1	eve
2	bob

a	b
2	3

b
3
4

This database violates ER constraints.
BUT it does satisfy the relational constraints.

FAIL



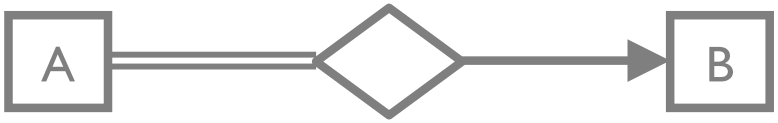
```
A(
 a int primary key,
 name text,
 b references B(b) NOT NULL
)
```

```
B(
 b int primary key
)
```

a	name	b
1	eve	3
2	bob	3

b
3
4

This database satisfies ER constraints.
Also satisfies relational constraints. OK



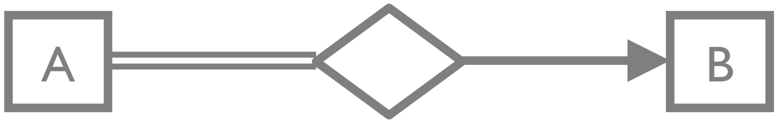
```
A(
 a int primary key,
 name text,
 b references B(b) NOT NULL
)
```

```
B(
 b int primary key
)
```

a	name	b
1	eve	null
2	bob	3

b
3
4

This database violates ER constraints.
Also violates relational constraints. OK



A(
 a int primary key,
 name text,
 b references B(b) NOT NULL
)

B(
 b int primary key
)

a	name	b
2	bob	3

b
3
4

This database satisfies ER constraints.
Also satisfies relational constraints. OK

Some tips

There are not that many ways to express a relationship between A and B for any combination of constraints

```
A(…),AB(...), B(...)  // all three
A_AB(...), B(...)     // A and AB are merged
A(...), B_AB(...)     // AB and B are merged
AB(...)               // all three merged
```

You should understand…

Why the following cannot be expressed or cannot be expressed without redundancy