

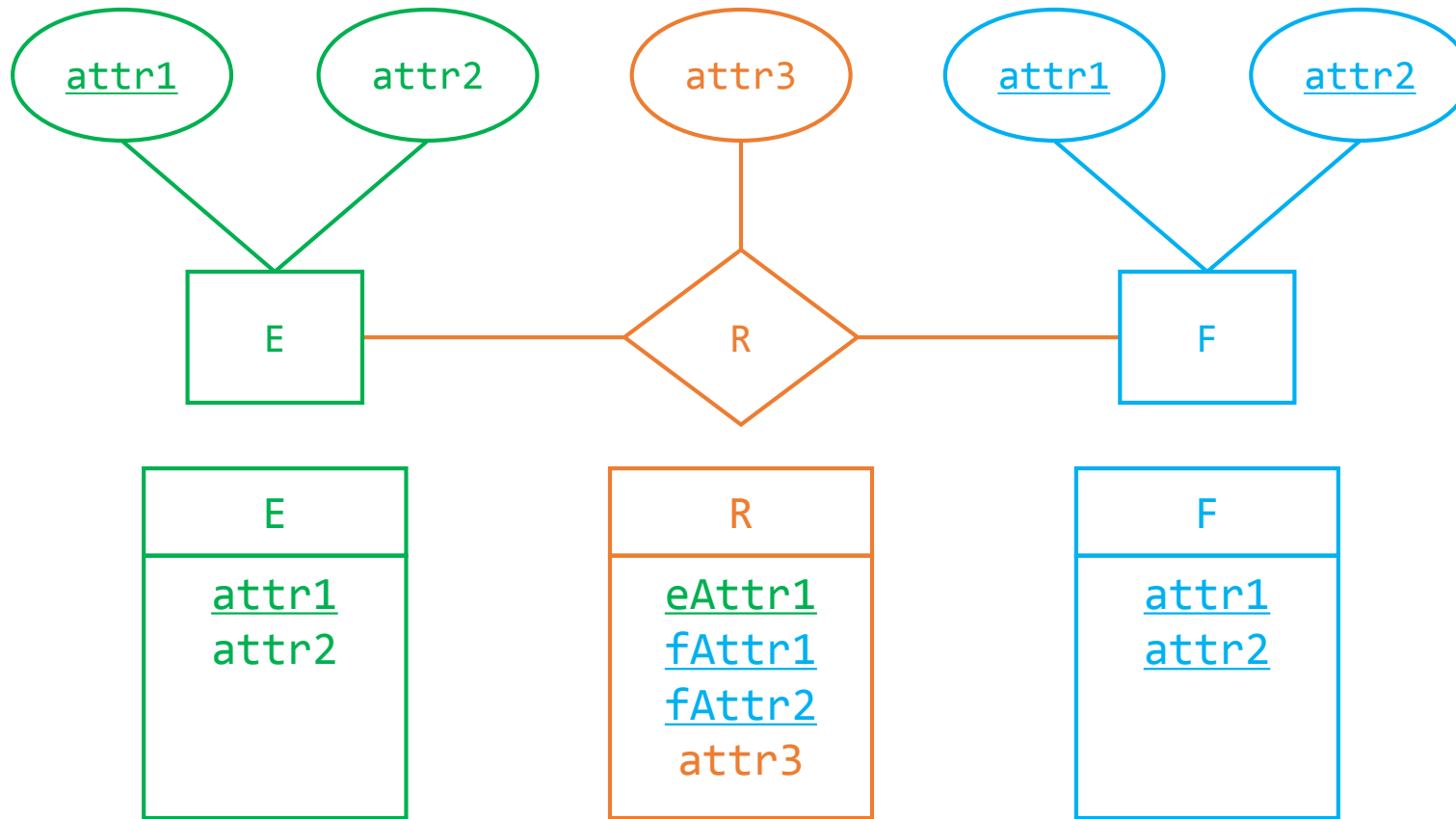
CS 336 Recitation

E/R Diagrams → Relational Tables

Fei Deng

Mon 10AM – 11AM, Hill 202

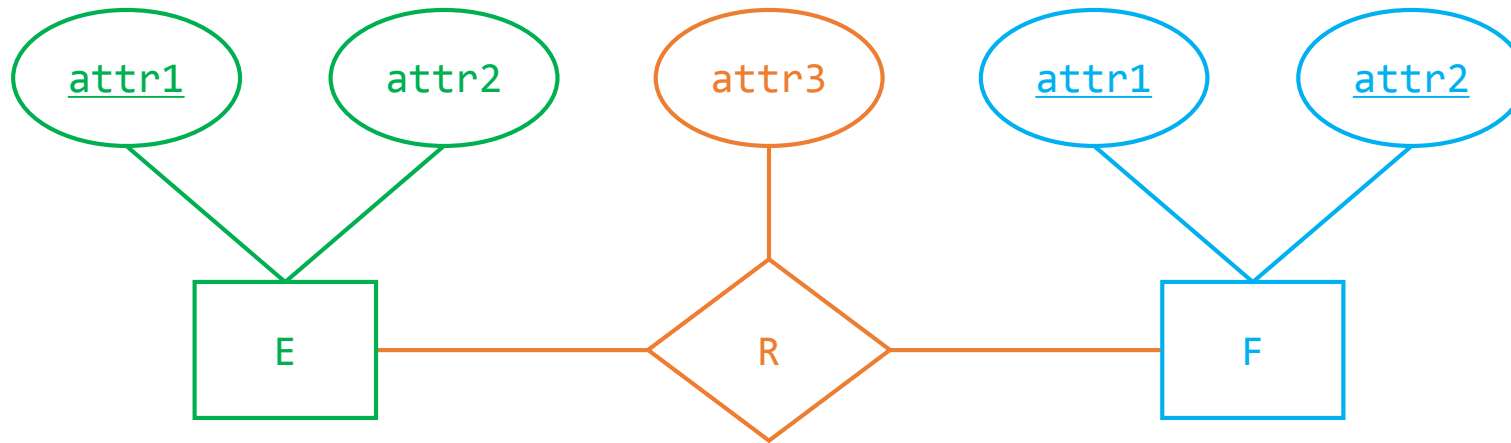
fei.deng@rutgers.edu



```
CREATE TABLE E (
  attr1 INT PRIMARY KEY,
  attr2 BOOL DEFAULT 0
);
```

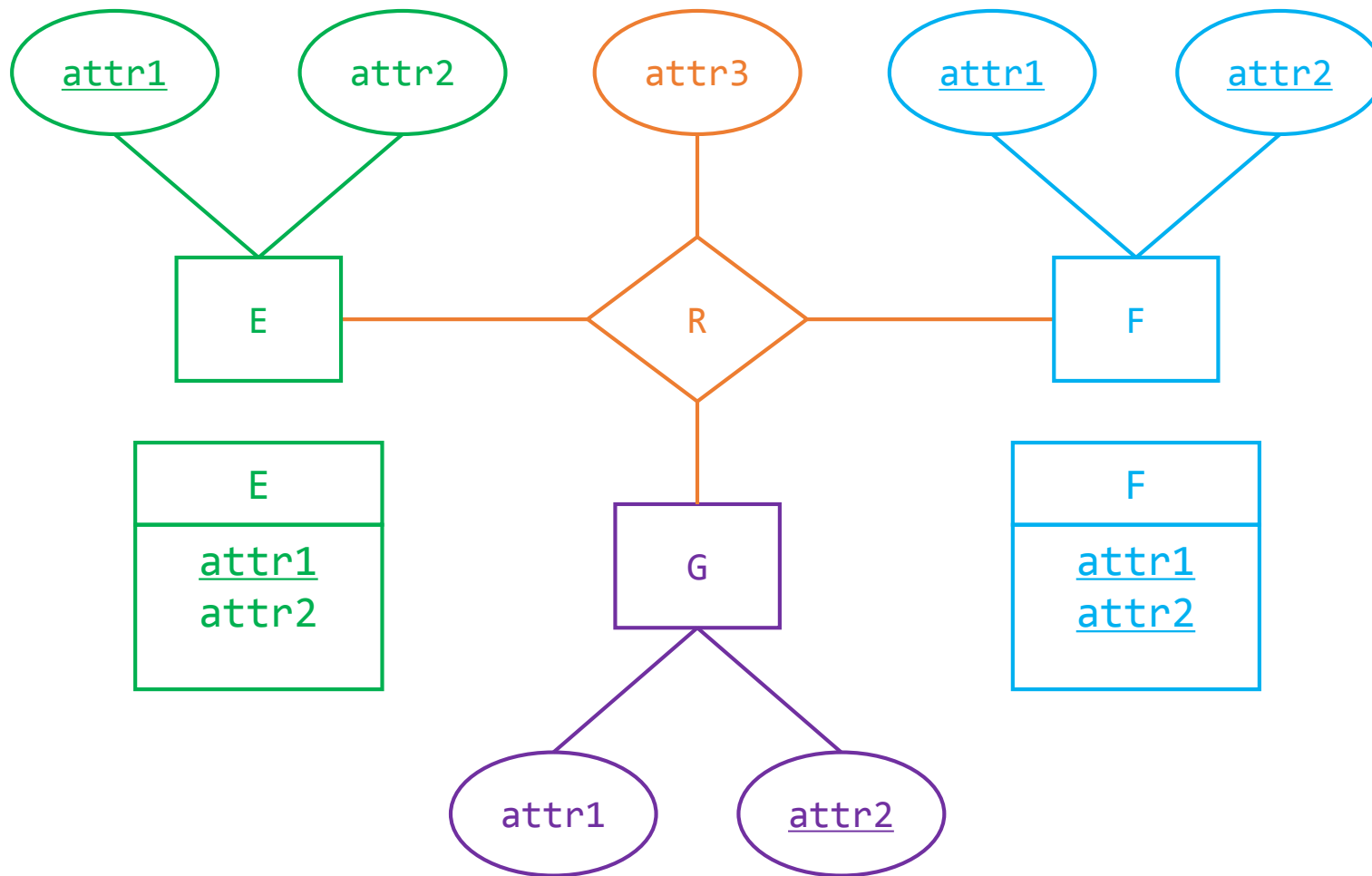
```
CREATE TABLE F (
  attr1 DATETIME,
  attr2 VARCHAR(10),
  PRIMARY KEY (attr1, attr2)
);
```

case insensitive



```
CREATE TABLE R (  
    eAttr1 INT,  
    fAttr1 DATETIME,  
    fAttr2 VARCHAR(10),  
    attr3 FLOAT,  
    PRIMARY KEY (eAttr1, fAttr1, fAttr2),  
    FOREIGN KEY (eAttr1)  
        REFERENCES E(attr1),  
    FOREIGN KEY (fAttr1, fAttr2)  
        REFERENCES F(attr1, attr2)  
);
```

```
CREATE TABLE E (  
    attr1 INT PRIMARY KEY,  
    attr2 BOOL DEFAULT 0  
);  
  
CREATE TABLE F (  
    attr1 DATETIME,  
    attr2 VARCHAR(10),  
    PRIMARY KEY (attr1, attr2)  
);
```

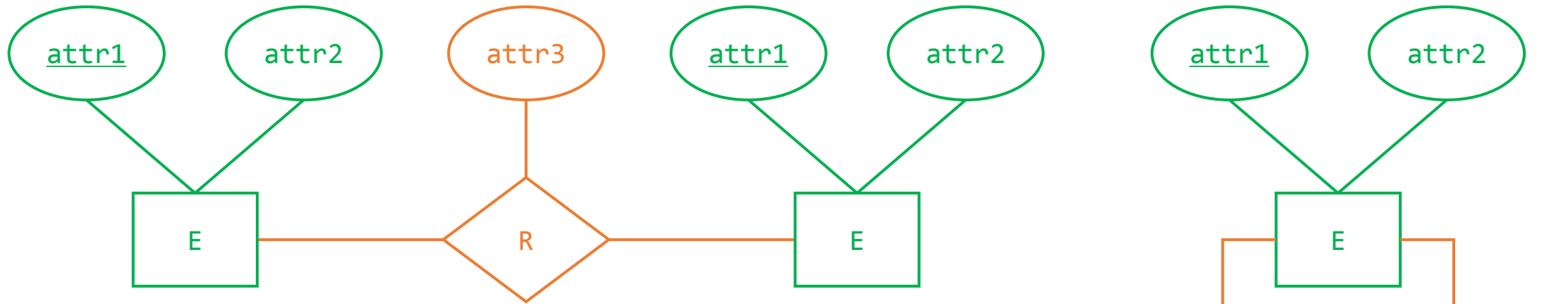


E
<u>attr1</u>
attr2

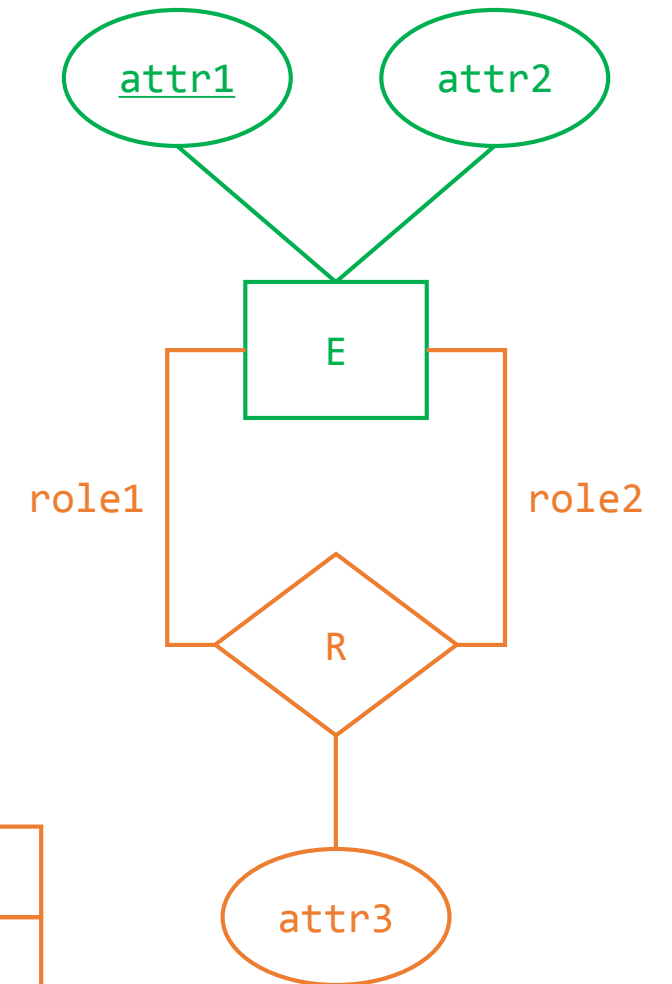
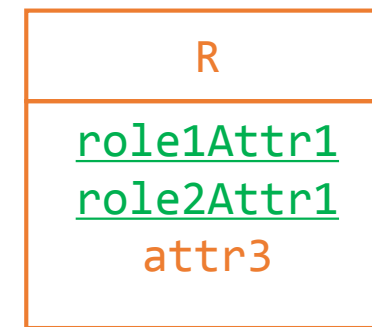
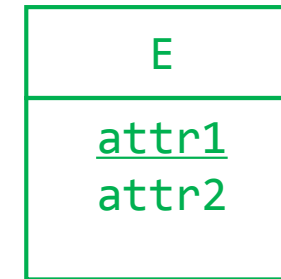
F
<u>attr1</u>
attr2

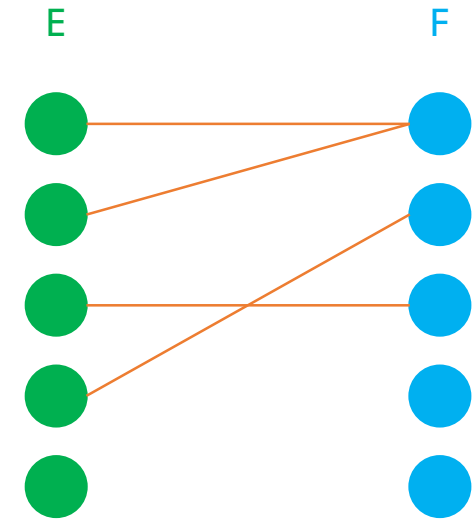
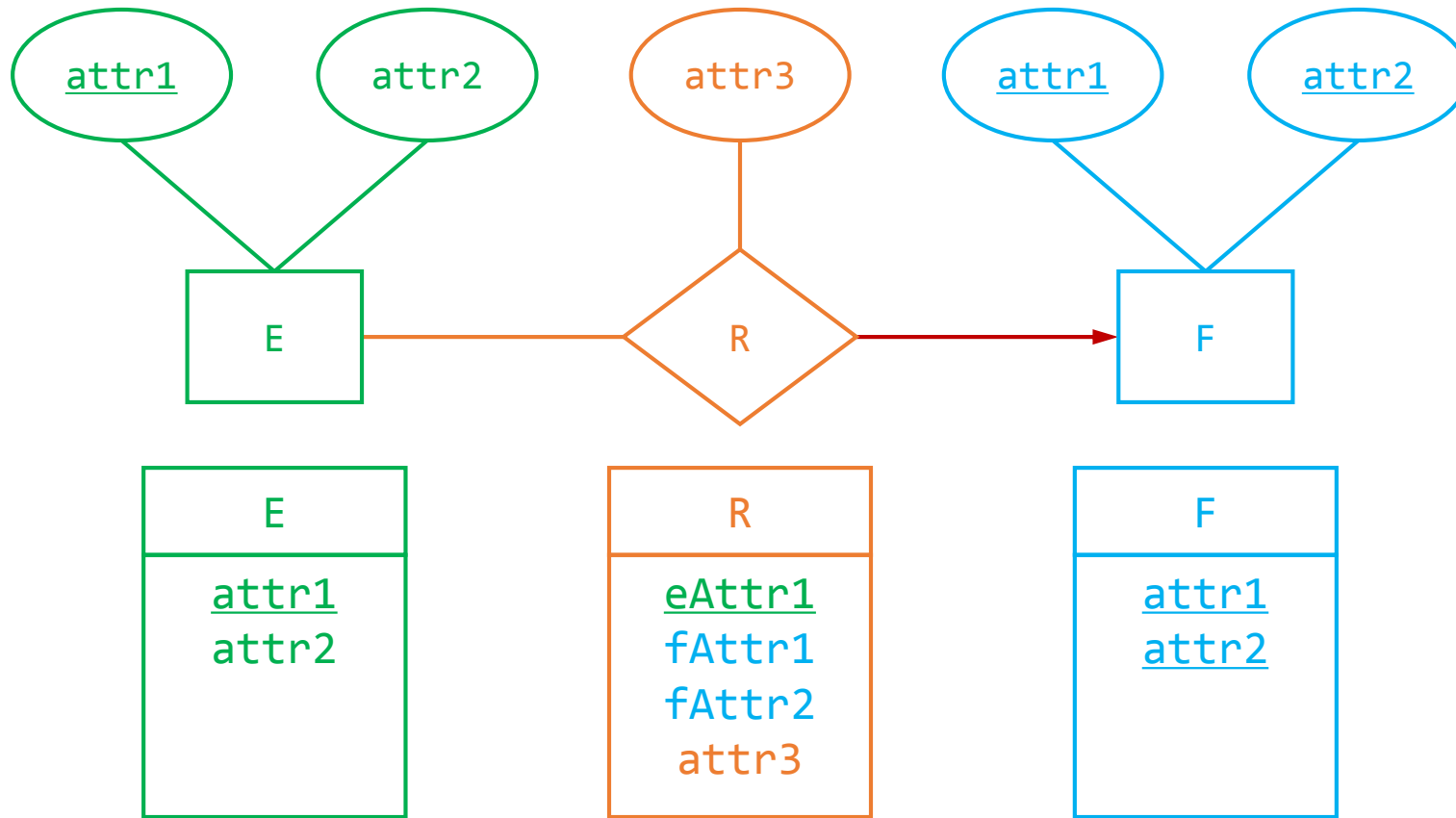
R
<u>eAttr1</u>
<u>fAttr1</u>
<u>fAttr2</u>
<u>gAttr2</u>
attr3

G
attr1
<u>attr2</u>

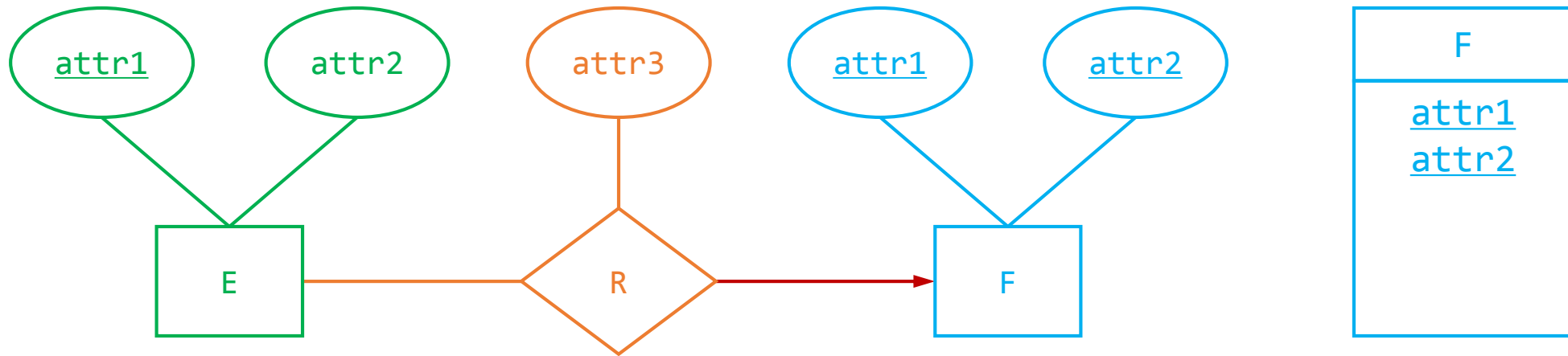


```
CREATE TABLE R (
  role1Attr1 INT,
  role2Attr1 INT,
  attr3      FLOAT,
  PRIMARY KEY (role1Attr1, role2Attr1),
  FOREIGN KEY (role1Attr1)
    REFERENCES E(attr1),
  FOREIGN KEY (role2Attr1)
    REFERENCES E(attr1)
);
```





The primary key of R is now eAttr1!
 We can merge R into E!
 Query will be more efficient!



E
<u>attr1</u>
attr2
fAttr1
fAttr2
attr3

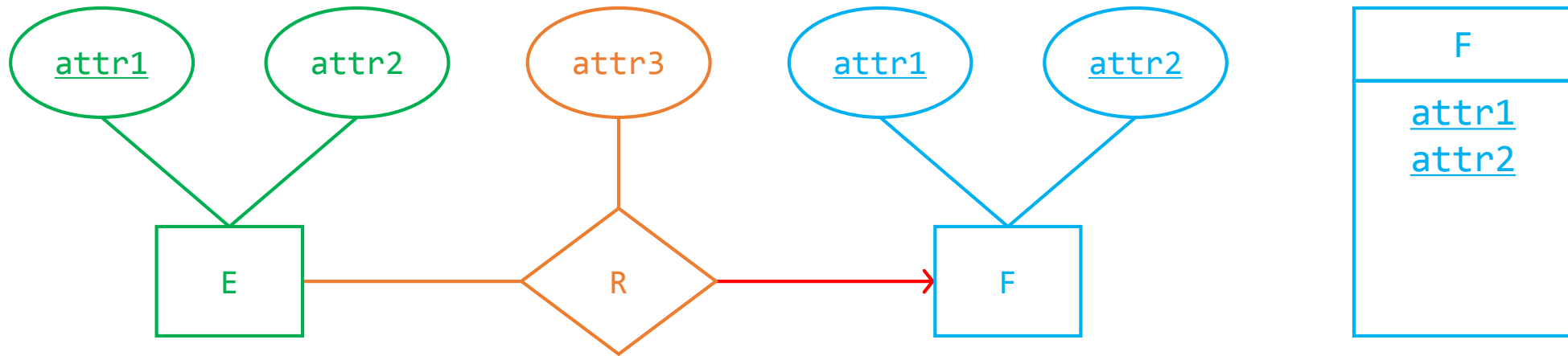
composite primary key of F
attribute of R

```

CREATE TABLE E (
    attr1 INT PRIMARY KEY,
    attr2 BOOL DEFAULT 0,
    fAttr1 DATETIME,
    fAttr2 VARCHAR(10),
    attr3 FLOAT,
    FOREIGN KEY (fAttr1, fAttr2)
        REFERENCES F(attr1, attr2)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);
  
```

“at most one”:

fAttr1, fAttr2 can be NULL in table E
FOREIGN KEY allows NULL values
(PRIMARY KEY implies NOT NULL)



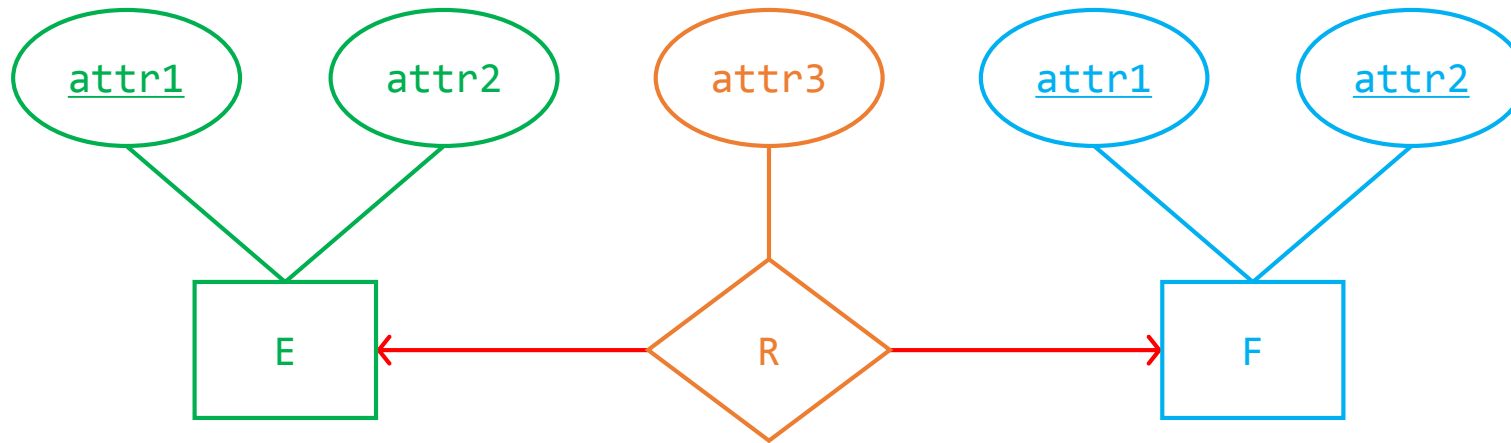
E
<u>attr1</u>
attr2
fAttr1
fAttr2
attr3

composite primary key of F
attribute of R

```

CREATE TABLE E (
    attr1 INT PRIMARY KEY,
    attr2 BOOL DEFAULT 0,
    fAttr1 DATETIME NOT NULL,
    fAttr2 VARCHAR(10) NOT NULL,
    attr3 FLOAT,
    FOREIGN KEY (fAttr1, fAttr2)
        REFERENCES F(attr1, attr2)
);
  
```

“exactly one”:
fAttr1, fAttr2 cannot be NULL in table E



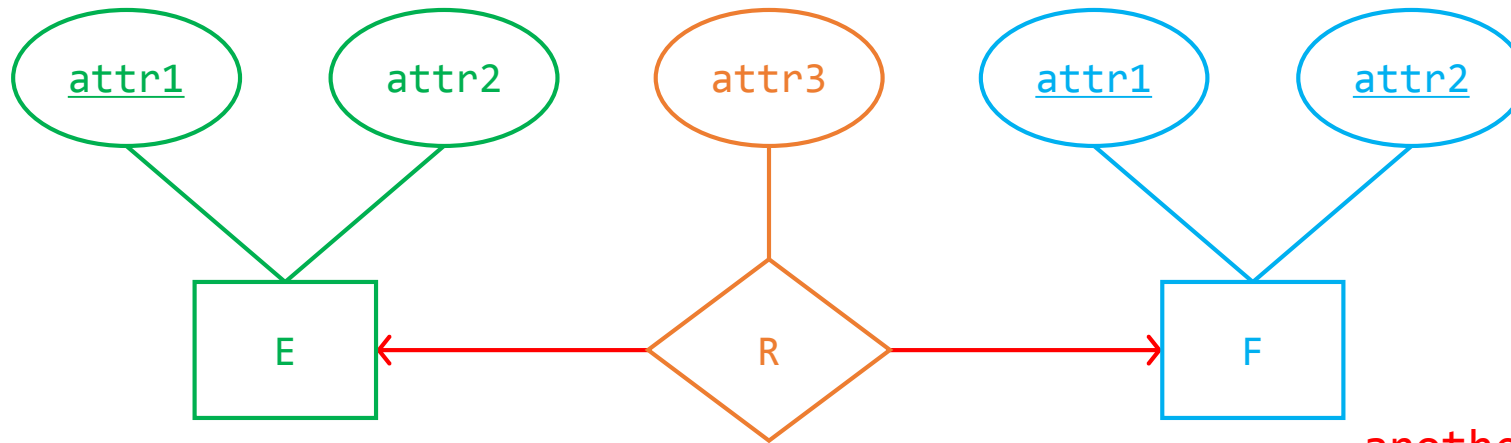
E/R/F
<u>eAttr1</u>
eAttr2
fAttr1
fAttr2
attr3

```

CREATE TABLE R (
    eAttr1 INT PRIMARY KEY,
    eAttr2 BOOL DEFAULT 0,
    fAttr1 DATETIME NOT NULL,
    fAttr2 VARCHAR(10) NOT NULL,
    attr3 FLOAT,
    UNIQUE (fAttr1, fAttr2)
);
  
```

“exactly one to exactly one”:
only need one table

UNIQUE is used for candidate keys
UNIQUE allows NULL values
UNIQUE NOT NULL attributes can be
referenced by FOREIGN KEY



another option:

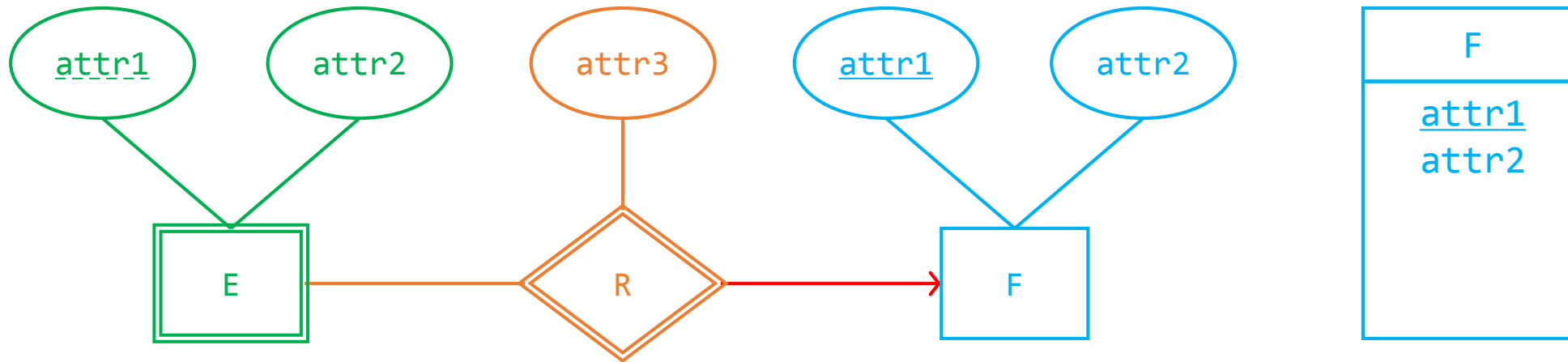
E/R/F
eAttr1
eAttr2
<u>fAttr1</u>
<u>fAttr2</u>
attr3

```

CREATE TABLE R (
    eAttr1 INT UNIQUE NOT NULL,
    eAttr2 BOOL DEFAULT 0,
    fAttr1 DATETIME,
    fAttr2 VARCHAR(10),
    attr3 FLOAT,
    PRIMARY KEY (fAttr1, fAttr2)
);
  
```

“exactly one to exactly one”:
only need one table

UNIQUE is used for candidate keys
UNIQUE allows NULL values
UNIQUE NOT NULL attributes can be
referenced by FOREIGN KEY



E
<u>attr1</u>
attr2
<u>fAttr1</u>
attr3

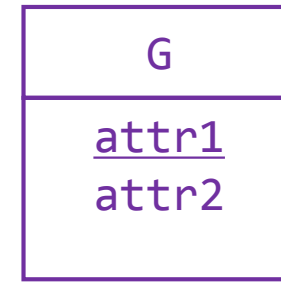
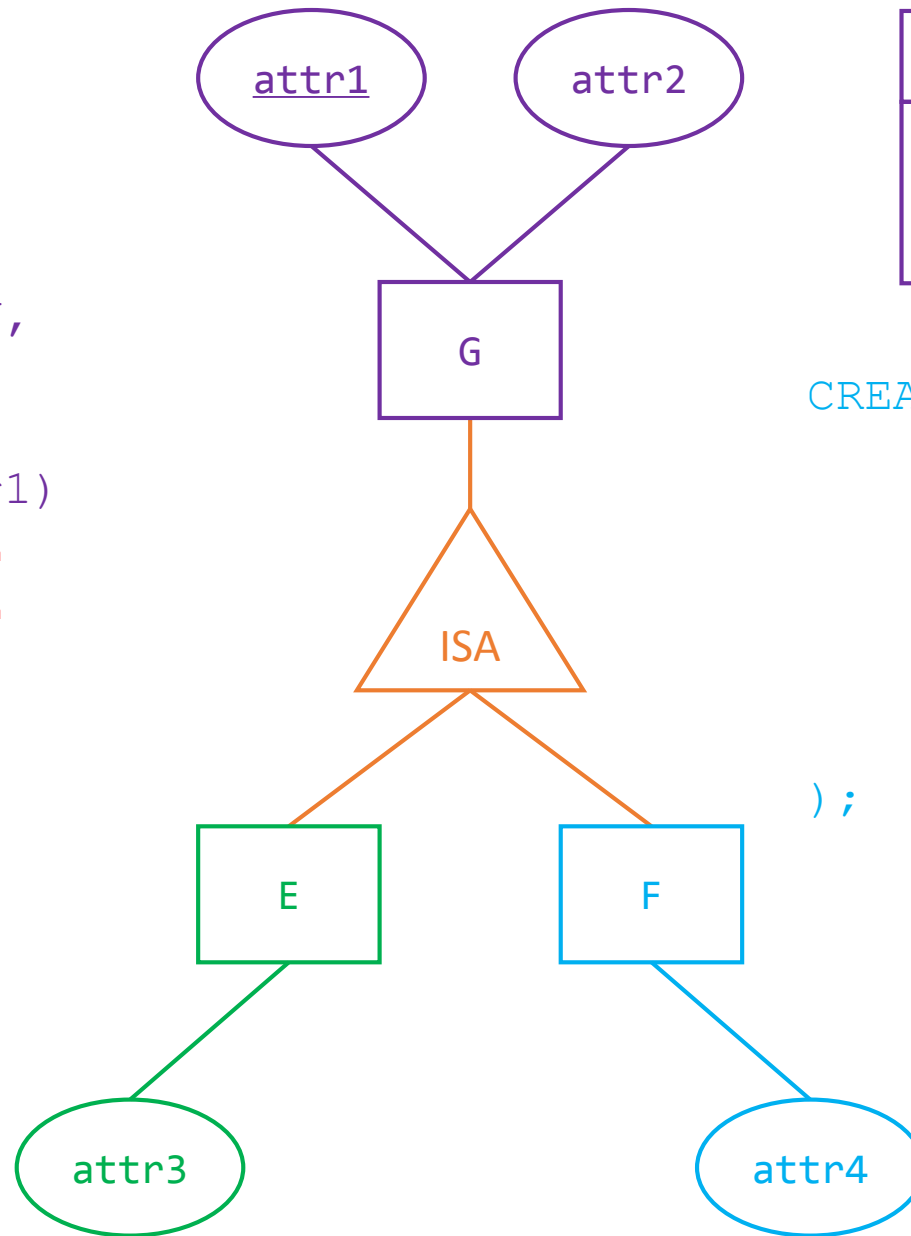
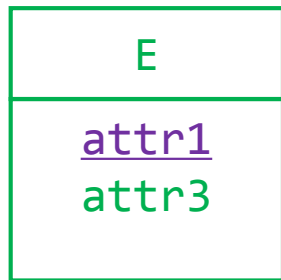
```

CREATE TABLE E (
    attr1 INT,
    attr2 BOOL DEFAULT 0,
    fAttr1 DATETIME,
    attr3 FLOAT,
    PRIMARY KEY (fAttr1, attr1),
    FOREIGN KEY (fAttr1)
        REFERENCES F(attr1)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
  
```

supporting relationship must be
“many to exactly one”: does not
need a separate table!

fAttr1 is now part of PRIMARY KEY in E:
cannot be NULL!

```
CREATE TABLE E (
  attr1 INT PRIMARY KEY,
  attr3 BOOL DEFAULT 0,
  FOREIGN KEY (attr1)
    REFERENCES G(attr1)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```



```
CREATE TABLE F (
  attr1 INT PRIMARY KEY,
  attr4 VARCHAR(10),
  FOREIGN KEY (attr1)
    REFERENCES G(attr1)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

