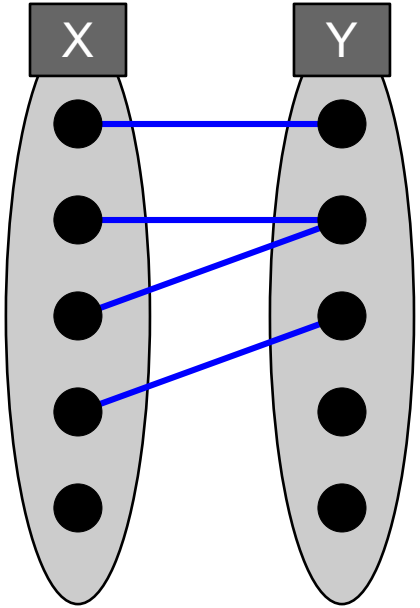
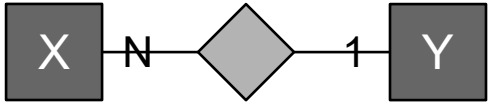
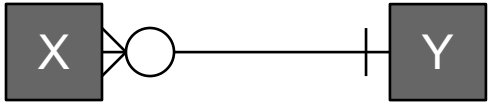
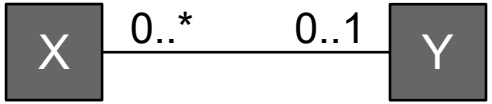


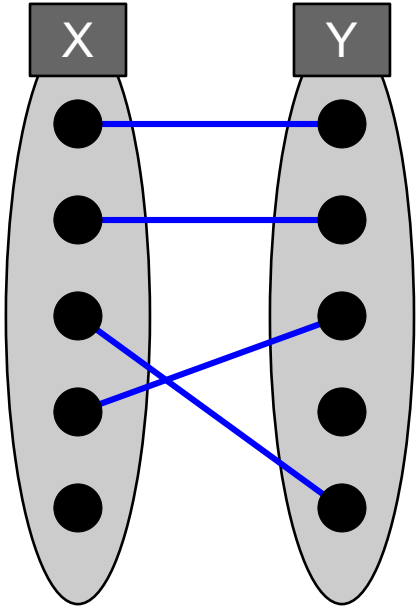

An **X** may be related to **0 or 1 Y**s.
A **Y** may be related to any number (**0 or more**) **X**s.



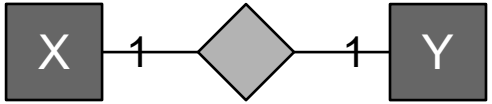



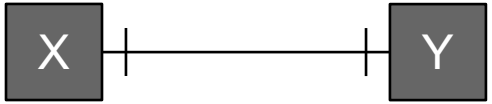



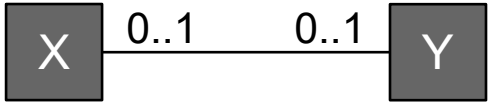

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```



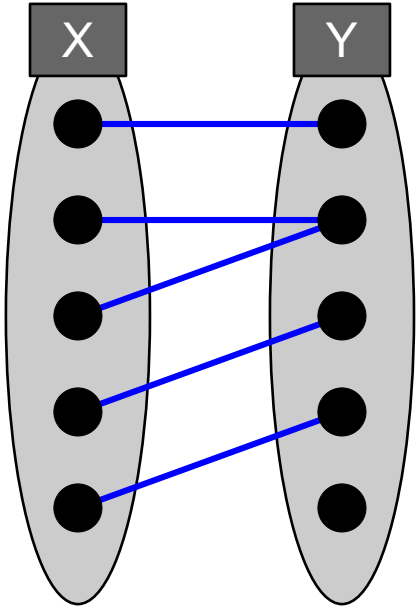

An **X** may be related to **0 or 1 Y**s.
A **Y** may be related to **0 or 1 X**s.



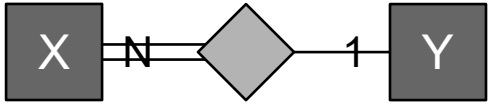



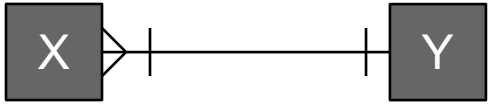



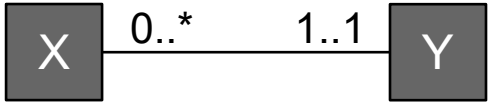

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT UNIQUE REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```



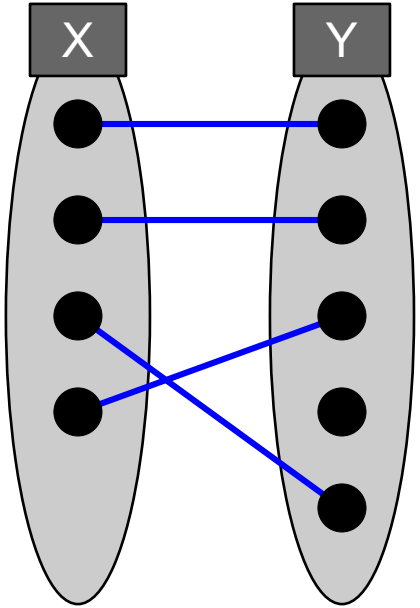

An **X** must be related to **exactly 1 Y**.
A **Y** may be related to any number (**0 or more**) **X**s.



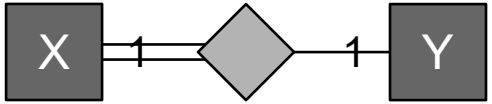



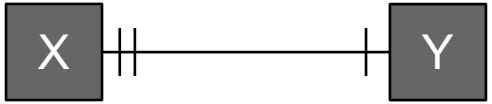



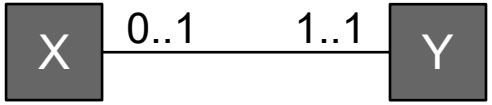

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT NOT NULL REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```



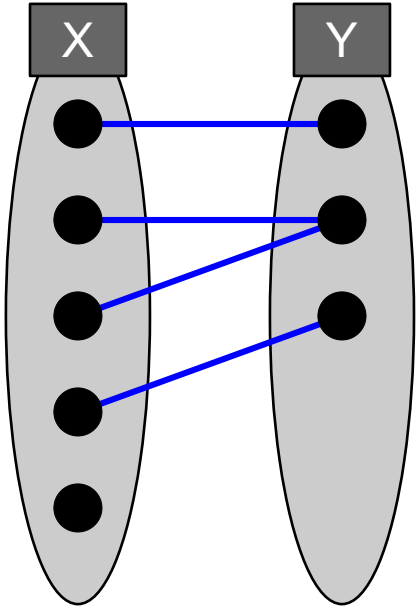

An **X** must be related to **exactly 1 Y**.
A **Y** may be related to **0 or 1 X**s.



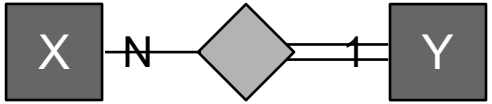



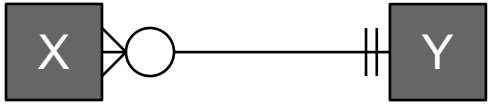



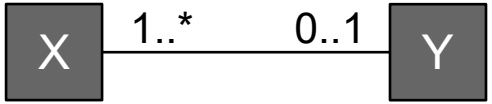

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT NOT NULL  UNIQUE REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```




An **X** may be related to **0 or 1 Ys**.
A **Y** must be related to **1 or more X**s.

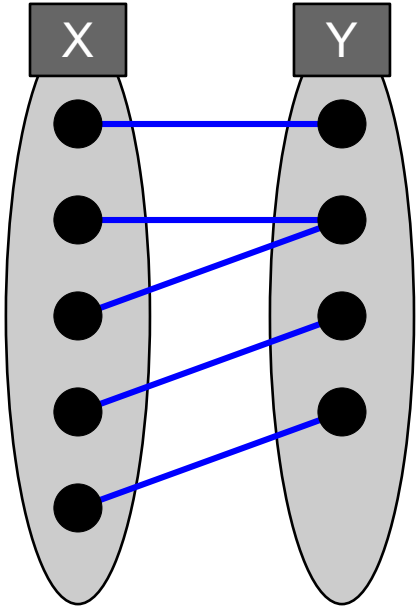


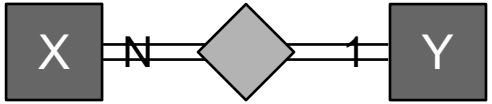



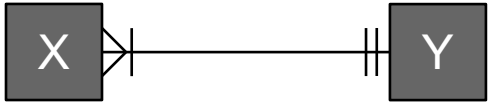

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```

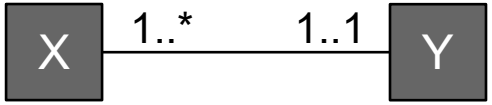
+ application logic




An **X** may be related to **exactly 1 Y**.
A **Y** must be related to **1 or more X**s.

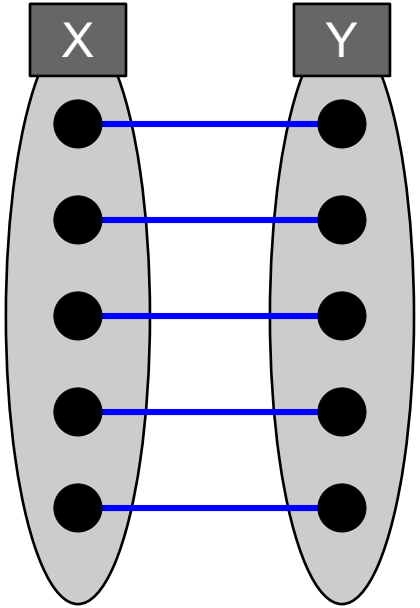


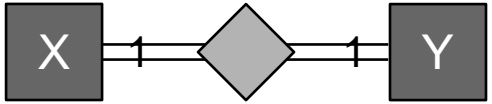



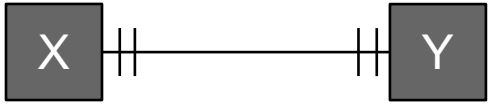

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  y_id INT NOT NULL REFERENCES Y(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);
```

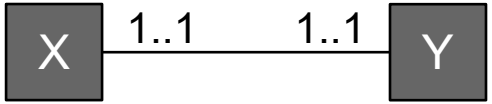
+ application logic



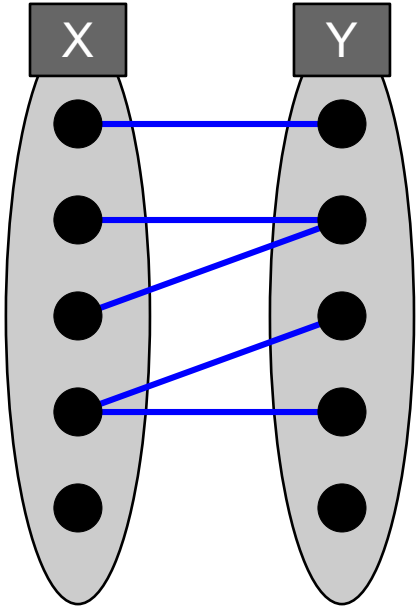

All **X**s and **Y**s must be matched **1 to 1**



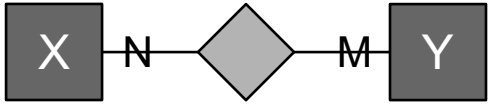



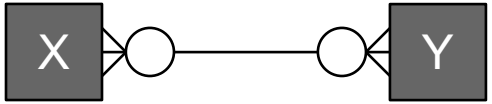



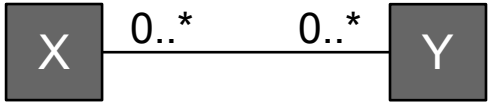

```
CREATE TABLE XY (
  x_id INT PRIMARY KEY,
  y_id INT NOT NULL  UNIQUE
);
```



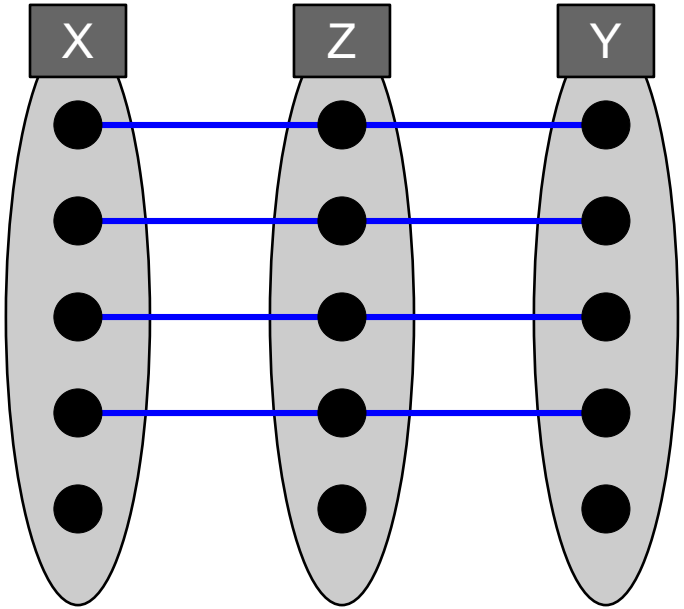

An **X** may be related to any number (**0 or more) Y**s.
A **Y** may be related to any number (**0 or more**) **X**s.

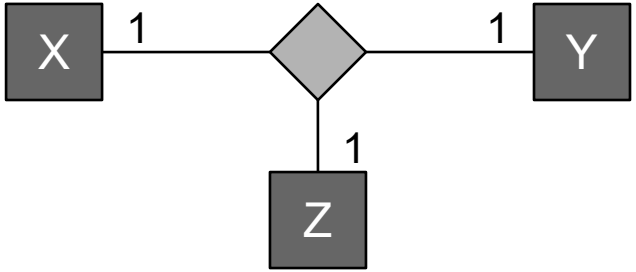


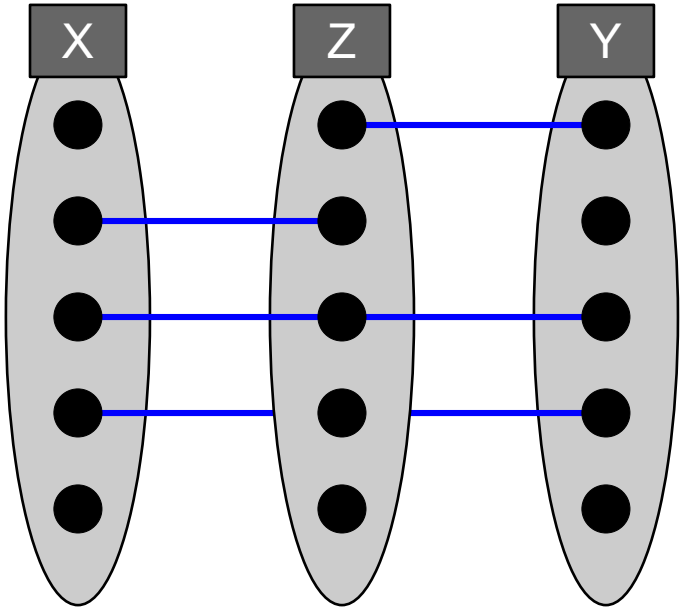

```
CREATE TABLE X (
  id INT PRIMARY KEY,
);

CREATE TABLE Y (
  id INT PRIMARY KEY
);

CREATE TABLE R (
  id INT PRIMARY KEY,
  x_id INT NOT NULL REFERENCES X(id),
  y_id INT NOT NULL REFERENCES Y(id)
);
```

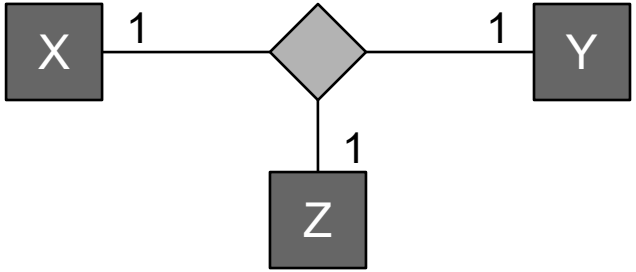



```
CREATE TABLE X (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Z (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY ,
  r_id INT REFERENCES R(id)
);
```

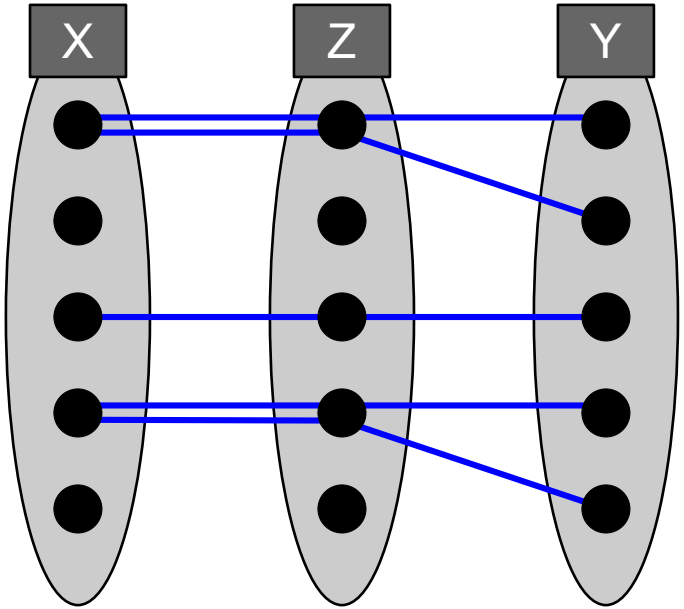
```
CREATE TABLE R (
  id INT PRIMARY KEY,
  x_id INT NOT NULL UNIQUE REFERENCES X(id),
  z_id INT NOT NULL UNIQUE REFERENCES Z(id),
  y_id INT NOT NULL UNIQUE REFERENCES Y(id)
);
```

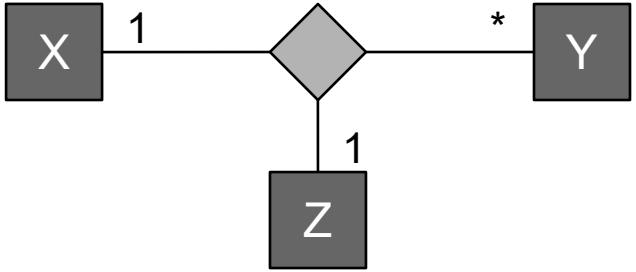


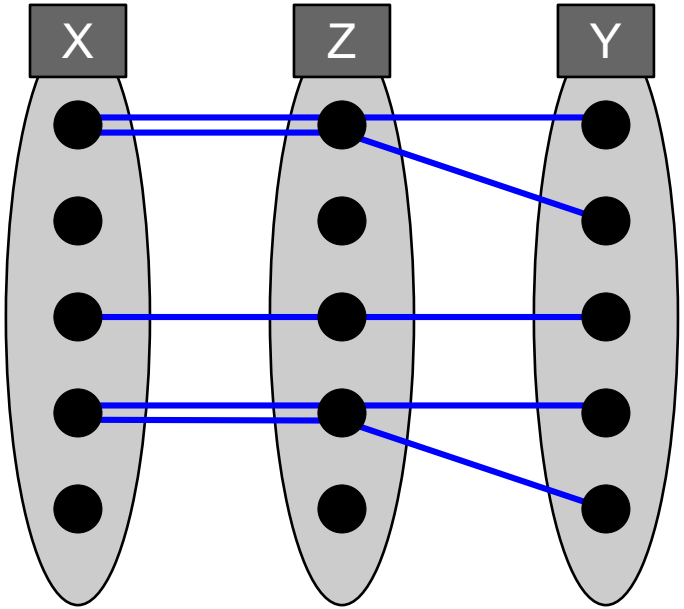

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Z (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY ,
  r_id INT REFERENCES R(id)
);
```

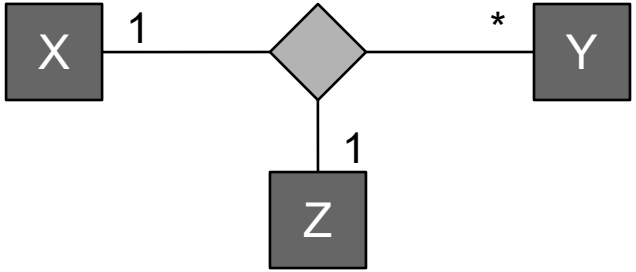
```
CREATE TABLE R (
  id INT PRIMARY KEY,
  x_id INT UNIQUE REFERENCES X(id),
  z_id INT UNIQUE REFERENCES Z(id),
  y_id INT UNIQUE REFERENCES Y(id)
);
```



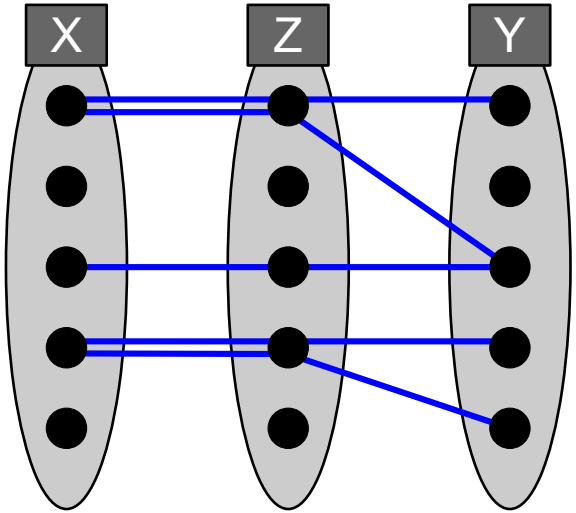



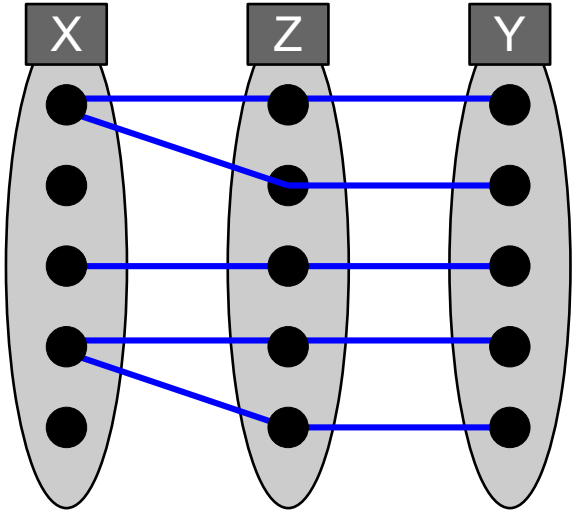

```
CREATE TABLE X (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Z (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY ,
  r_id INT REFERENCES R(id)
);




CREATE TABLE R (
  id INT PRIMARY KEY,
  x_id INT NOT NULL REFERENCES X(id),
  z_id INT NOT NULL REFERENCES Z(id),
  y_id INT NOT NULL UNIQUE REFERENCES Y(id)
);



+ application logic
```

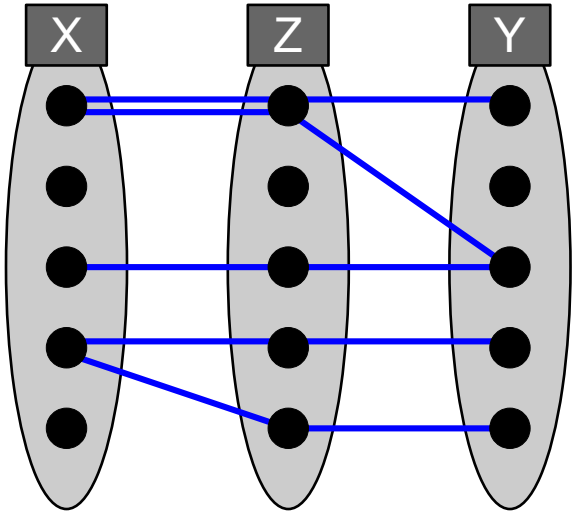



```
CREATE TABLE X (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Z (
  id INT PRIMARY KEY,
  r_id INT REFERENCES R(id)
);

CREATE TABLE Y (
  id INT PRIMARY KEY ,
  r_id INT REFERENCES R(id)
);

CREATE TABLE XZ (
  id INT PRIMARY KEY,
  x_id NOT NULL UNIQUE REFERENCES X(id),
  z_id NOT NULL UNIQUE REFERENCES Z(id)
);


CREATE TABLE R (
  id INT PRIMARY KEY,
  xz_id INT NOT NULL REFERENCES XZ(id),
  y_id INT NOT NULL UNIQUE REFERENCES Y(id)
);
```


X
Z
Y


X
Z
Y


X
Z
Y