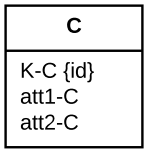
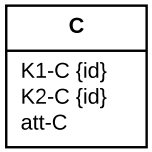
UML Class Diagram* to Relational Model Rules + Example

*Adaptation for DBs

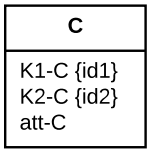
Classes



C(K-C, att1-C, att2-C);



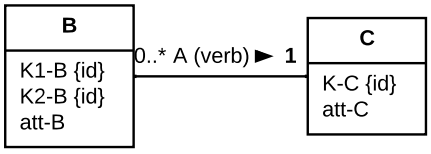
C(K1-C, K2-C, att-C);
(Composed key)



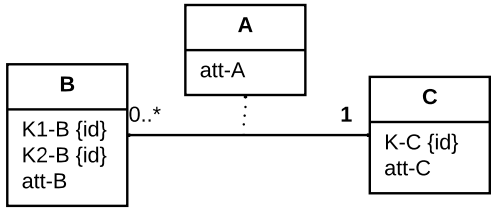
C(K1-C, K2-C, att-C);
(Two candidate keys)

Associations & Association Classes

1 to Many

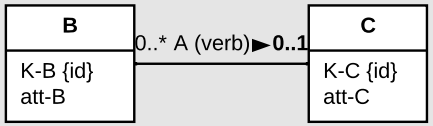


B(K1-B, K2-B, att-B, **K-C**); **C**(K-C, att-C);



B(K1-B, K2-B, att-B, att-A, **K-C**); **C**(K-C, att-C);

0..1 to Many



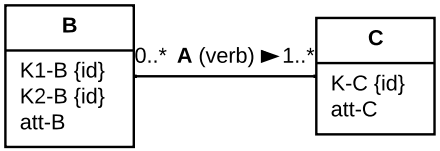
K-C could be NULL **B**(K-B, att-B, **K-C**); **C**(K-C, att-C);

OR

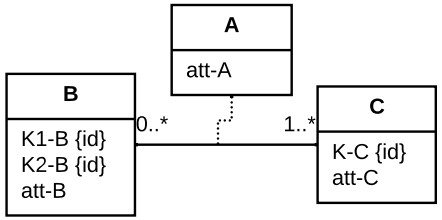
K-C NOT NULL **B**(K-B, att-B); **A**(**K-B**, **K-C**); **C**(K-C, att-C);

this option is generally better

Many to Many

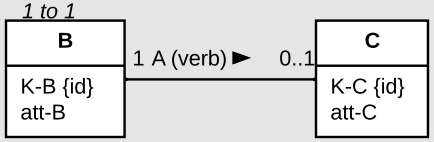


B(K1-B, K2-B, att-B); **C**(K-C, att-C);
A (K1-B, K2-B, K-C)



B(K1-B, K2-B, att-B); **C**(K-C, att-C);
A (K1-B, K2-B, K-C, att-A)

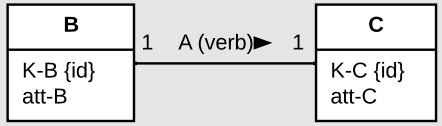
1 to 1



B(K-B, att-B); **C**(K-C, att-C, K-B)

Not composed key but two candidate keys

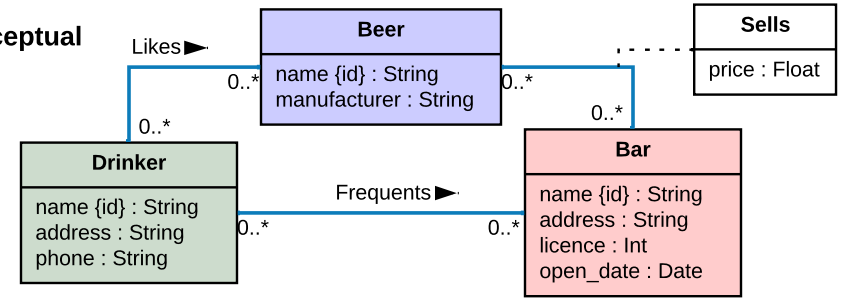
+Constraints



B(K-B, att-B); **C**(K-C, att-C, K-B)
OR
B(K-B, att-B, K-C); **C**(K-C, att-C);

+Constraints

Conceptual



Logical

Beers (beer_name, beer_manufacturer)

Drinkers (drinker_name, drinker_address, drinker_phone)

Bars (bar_name, bar_address, bar_licence, bar_open_date)

Likes (drinker_name, beer_name)

Likes[drinker_name] ⊆ Drinkers[drinker_name]

Likes[beer_name] ⊆ Beers[beer_name]

Frequents (drinker_name, bar_name)

Frequents[drinker_name] ⊆ Drinkers[drinker_name]

Frequents[bar_name] ⊆ Bars[bar_name]

Sells (bar_name, beer_name, sells_price)

sells_price > 0

Sells[beer_name] ⊆ Beers[beer_name]

Sells[bar_name] ⊆ Bars[bar_name]

SQL Implementation

```
-- SQLite Syntax
CREATE TABLE Beers (
  beer_name TEXT NOT NULL,
  -- NOT NULL not necessary for PK attributes in some systems such as ORACLE
  beer_manufacturer TEXT,
  CONSTRAINT pk_beers_c0 PRIMARY KEY (beer_name)
);

CREATE TABLE Bars (
      bar_name TEXT NOT NULL,
      bar_address TEXT,
      bar_licence DATE,
      bar_open_date DATE,
      CONSTRAINT pk_bars_c0 PRIMARY KEY (bar_name)
);

CREATE TABLE Sells(
  bar_name TEXT NOT NULL,
  beer_name TEXT NOT NULL,
  sells_price REAL NOT NULL,
  CONSTRAINT pk_sells_c0 PRIMARY KEY (bar_name,beer_name),
  CONSTRAINT fk_sells_c1 FOREIGN KEY (bar_name) REFERENCES Bars(bar_name),
  CONSTRAINT fk_sells_c2 FOREIGN KEY (beer_name) REFERENCES Beers(beer_name),
  CONSTRAINT ck_sells_c3 CHECK (sells_price > 0)
);

....
```