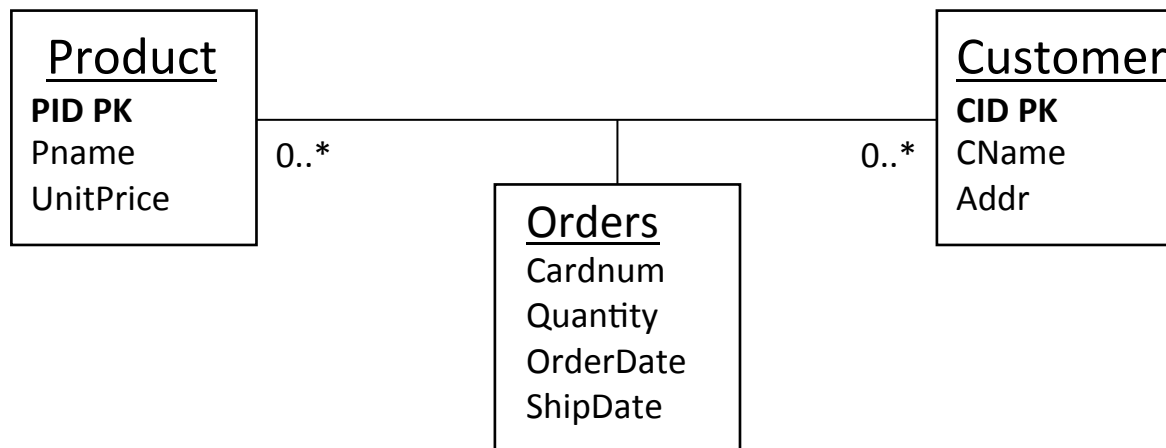
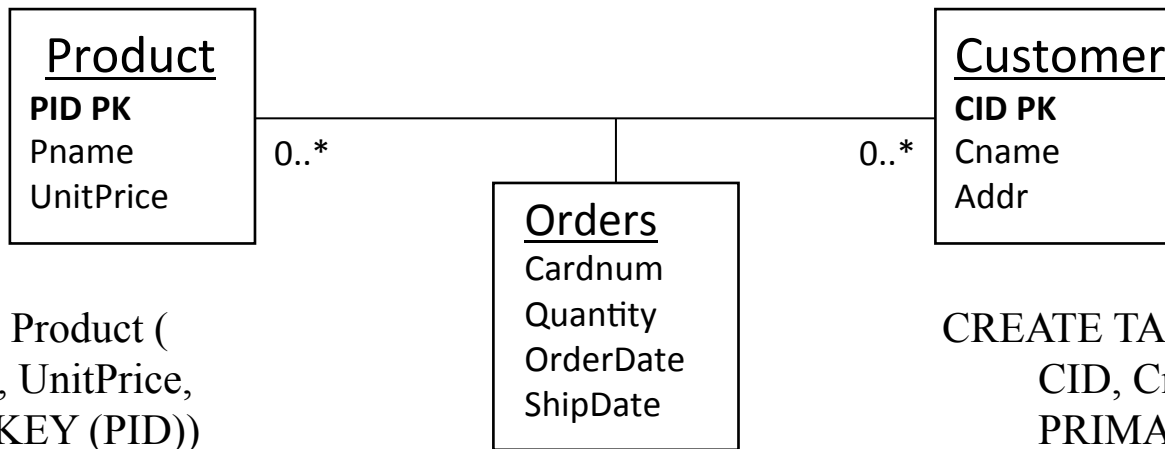


Interrogating a database design for supported functionality

Lets consider database designs intended to support purchases and shipments of products by and to customers



Translate this UML fragment into a set of CREATE statements (don't worry about attribute typing, but include all other important constraints)



```

CREATE TABLE Product (
  PID, Pname, UnitPrice,
  PRIMARY KEY (PID))
  
```

```

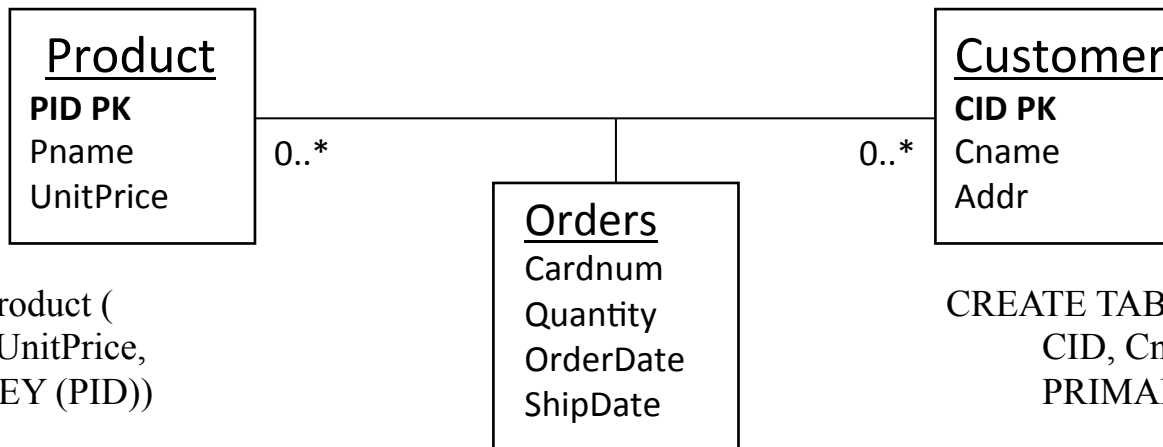
CREATE TABLE Customer (
  CID, Cname, Addr,
  PRIMARY KEY (CID))
  
```

```

CREATE TABLE Orders (
  PID, CID,
  CardNum, Quantity, OrderDate, ShipDate
  PRIMARY KEY (PID, CID),
  FOREIGN KEY (PID) REFERENCES Product,
  FOREIGN KEY (CID) REFERENCES Customer)
  
```

What if a customer orders two different products (on the same day or different days)?

What if a customer orders the same product more than once (on the same day or different days)?



```
CREATE TABLE Product (
  PID, Pname, UnitPrice,
  PRIMARY KEY (PID))
```

```
CREATE TABLE Customer (
  CID, Cname, Addr,
  PRIMARY KEY (CID))
```

```
CREATE TABLE Orders (
  PID, CID,
  CardNum, Quantity, OrderDate, ShipDate
  PRIMARY KEY (PID, CID),
  FOREIGN KEY (PID) REFERENCES Product,
  FOREIGN KEY (CID) REFERENCES Customer)
```

What if a customer orders two different products (on the same day or different days)?

Create two different Orders records, each with

- OrderDate equal CurrentDate and
- ShipDate initially set to NULL

What if a customer orders the same product more than once (on the same day or different days)?

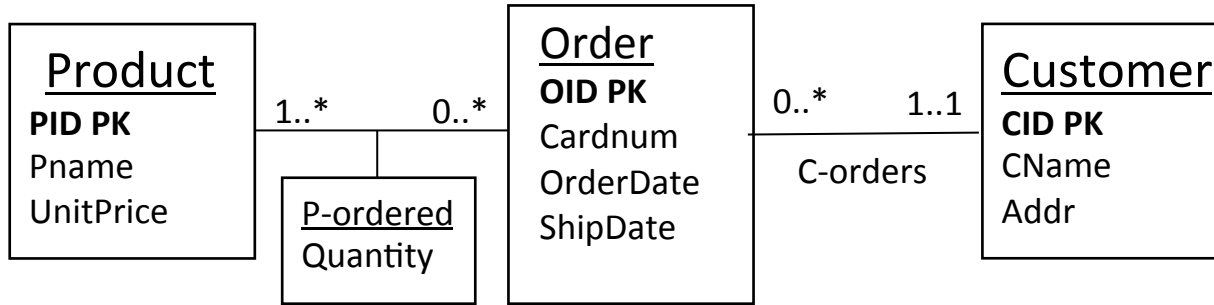
Basic problem: the DB only allows a record of no more than one instance of a PID/CID pair, but hacks are possible

If on the same day, then one could update the first created PID/CID record to reflect the subsequent order Quantity, but will lose track of separate orders

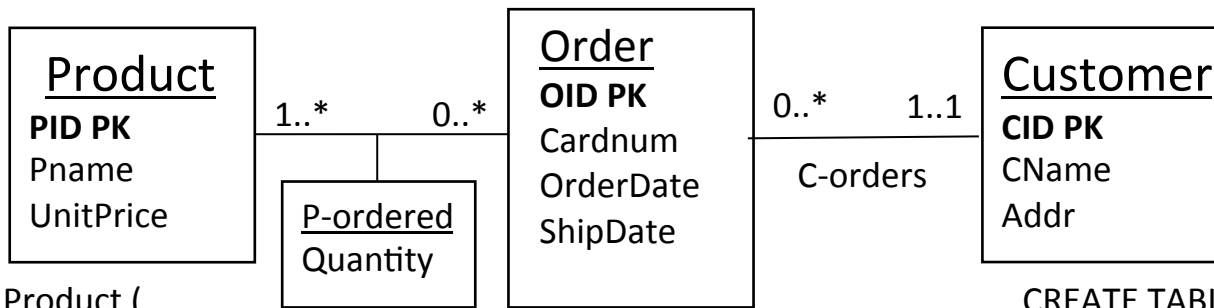
If on the different days, then one would additionally have to “lie” about the OrderDate for one of the PID/CID records

If first instance already shipped (ShipDate isn't NULL) then one could overwrite the earlier order, but loss of info

An alternative design



Translate this UML fragment into a set of CREATE statements (don't worry about attribute typing, but include all other important constraints)



```
CREATE TABLE Product (
    PID, Pname, UnitPrice,
    PRIMARY KEY (PID))
```

```
CREATE TABLE P-ordered (
    PID, OID, Quantity
    PRIMARY KEY (PID, OID),
    FOREIGN KEY (PID) REFERENCES Product,
    FOREIGN KEY (OID) REFERENCES Order)
```

```
CREATE TABLE Order&C-orders (
    OID, CID NOT NULL, CardNum, OrderDate, ShipDate
    PRIMARY KEY (OID),
    FOREIGN KEY (CID) REFERENCES Customer,
    CHECK (OID IN (SELECT P.OID /* partial implementation */
    FROM P-ordered P))) /* of 1..* constraint */
```

```
CREATE TABLE Customer (
    CID, Cname, Addr,
    PRIMARY KEY (CID))
```

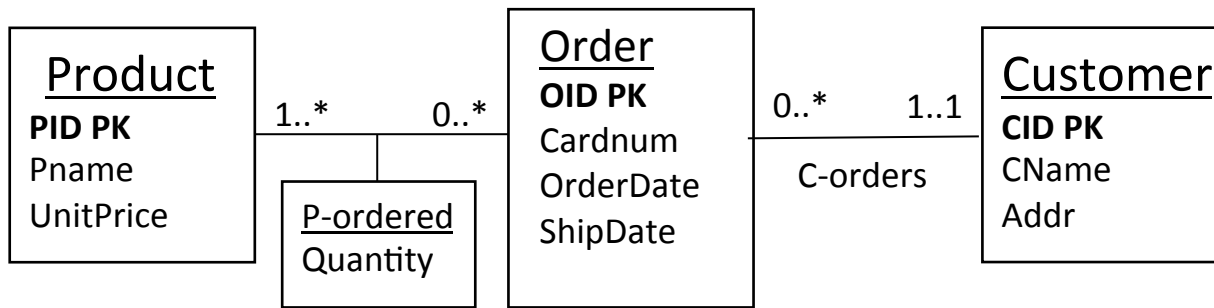
What if a customer orders two different products on the same day?

What if a customer orders the same product twice (or more) on the same day?

What if a customer orders the same product twice on different days?

What if SUPPLIER (which owns this DB) SHIPS the entirety of an order at same time?

What if SUPPLIER SHIPS only part of an order at a given time?



What if a customer orders two different products on the same day?

On same Shopping Cart?: Generate one new OID

Create/Insert one Order&C-orders record with CID and new OID

Create/Insert two P-ordered records with same (new) OID and the corresponding PID

On different Shopping Carts?: Generate two new OIDs

Create/Insert two Order&C-orders record with same CID and different new OIDs

Create/Insert two P-ordered records with different new OIDs and the corresponding PIDs

In both cases set OrderDate to CurrentDate and initially set ShipDate to NULL

What if a customer orders the same product twice (or more) on the same day? See above

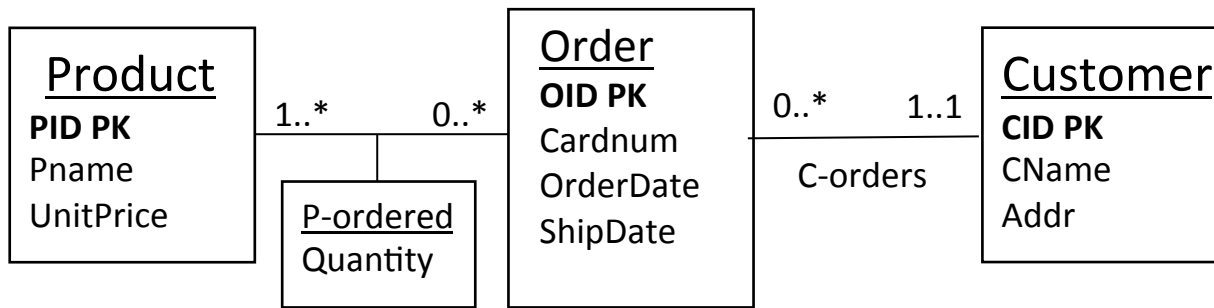
What if a customer orders the same product twice on different days? See above

What if SUPPLIER (which owns this DB) SHIPS the entirety of an order at same time?

Set ShipDate of Order to CurrentDate, which may have changed since date of order

What if SUPPLIER SHIPS only part of an order at a given time?

Hmmm, ...



What if SUPPLIER SHIPS only part of an order at a given time?

Basic idea: create new records that represent the parts of the original order that are NOT shipped, and revise the original record to reflect the parts of the order that are shipped (or vice versa)

Generate one new OID (distinct from old OID that represents original order)

Create/Insert one new Order&C-orders record with CID and new OID and copy other fields from old OID record in Order&C-orders

Set the ShipDate in old OID record to CurrentDate

UPDATE the P-ordered records of any product that is not shipped from the order with the new OID

If some product from an order has only some of the Quantity shipped,

then Create/Insert a new P-ordered record with the number of original Quantity that isn't shipped, and update the original record's Quantity by the number that isn't shipped

PID	Quantity	OID
A	5	M
B	2	M

OID	SDate
M	NULL

OID	CID
M	U



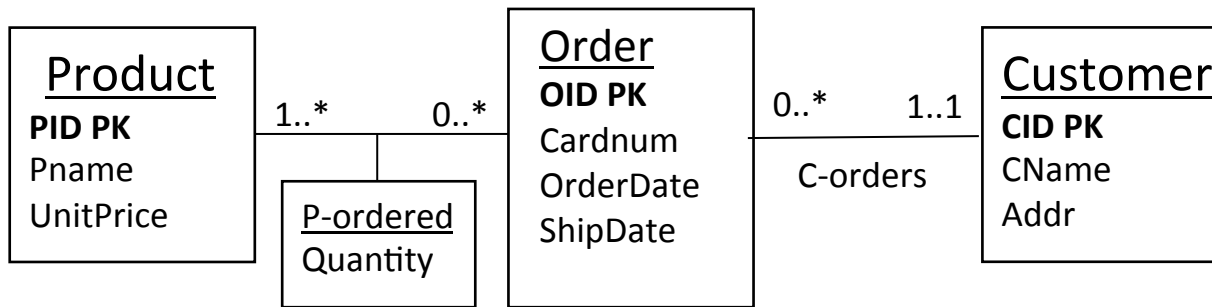
PID	Quantity	OID
A	5	M
B	2	N

OID	SDate
M	CurrDate
N	NULL

OID	CID
M	U
N	U

ready to ship

This will work, but it will lose information about items that were purchased on same transaction (shopping cart)



This will work, but it will lose information about items that were purchased on same transaction (shopping cart)

The basic problem is that the Order class is playing double duty, serving as both a record of purchase (transaction) and a record of shipment.

Solution – break Order up into two classes (see the Books database design, which contains both Transaction and Shipment classes).