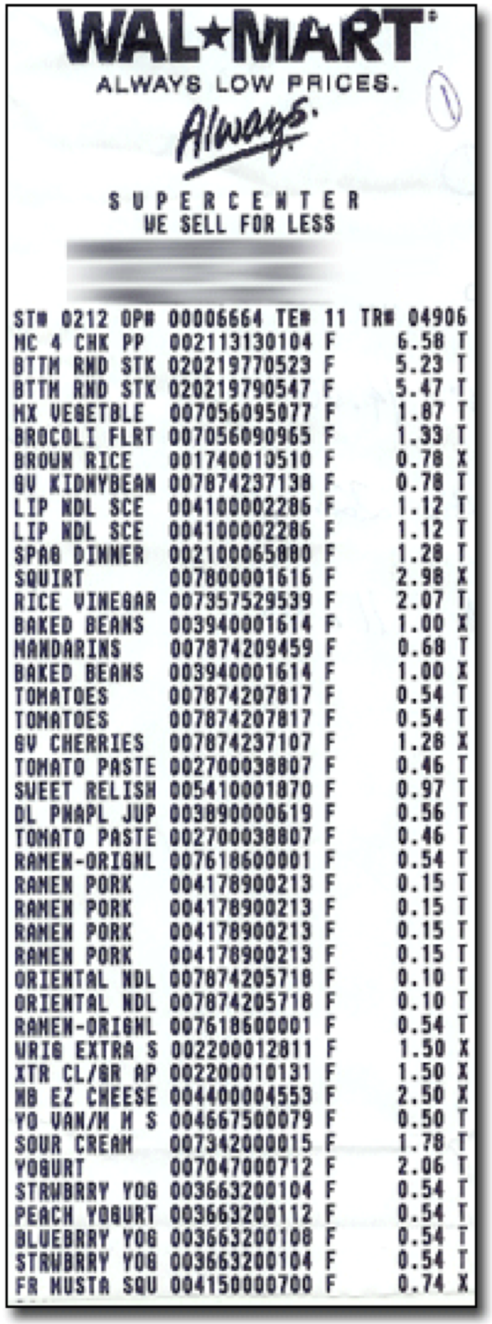
The Many-to-Many Relationship

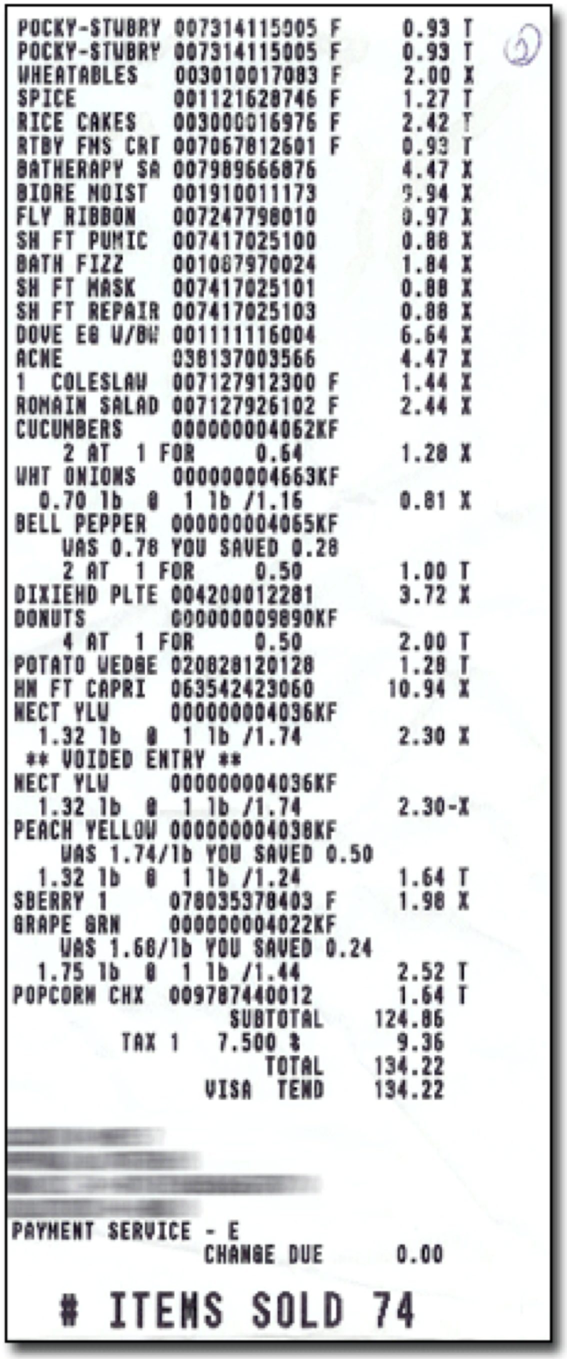
Fearful concatenation of circumstances

Daniel Webster



```
                WAL★MART
           ALWAYS LOW PRICES.
                 Always.                  1

             S U P E R C E N T E R
               WE SELL FOR LESS
[redacted]
ST# 0212 OP# 00006664 TE# 11 TR# 04906
MC 4 CHK PP  002113130104 F      6.58 T
BTTM RND STK 020219770523 F      5.23 T
BTTM RND STK 020219790547 F      5.47 T
MX VEGETBLE  007056095077 F      1.87 T
BROCOLI FLRT 007056090965 F      1.33 T
BROWN RICE   001740010510 F      0.78 X
GV KIDNYBEAN 007874237138 F      0.78 T
LIP NDL SCE  004100002286 F      1.12 T
LIP NDL SCE  004100002286 F      1.12 T
SPAG DINNER  002100065880 F      1.28 T
SQUIRT       007800001616 F      2.98 X
RICE VINEGAR 007357529539 F      2.07 T
BAKED BEANS  003940001614 F      1.00 X
MANDARINS    007874209459 F      0.68 T
BAKED BEANS  003940001614 F      1.00 X
TOMATOES     007874207817 F      0.54 T
TOMATOES     007874207817 F      0.54 T
GV CHERRIES  007874237107 F      1.28 X
TOMATO PASTE 002700038807 F      0.46 T
SWEET RELISH 005410001870 F      0.97 T
DL PNAPL JUP 003890000619 F      0.56 T
TOMATO PASTE 002700038807 F      0.46 T
RAMEN-ORIGNL 007618600001 F      0.54 T
RAMEN PORK   004178900213 F      0.15 T
RAMEN PORK   004178900213 F      0.15 T
RAMEN PORK   004178900213 F      0.15 T
RAMEN PORK   004178900213 F      0.15 T
ORIENTAL NDL 007874205718 F      0.10 T
ORIENTAL NDL 007874205718 F      0.10 T
RAMEN-ORIGNL 007618600001 F      0.54 T
WRIG EXTRA S 002200012811 F      1.50 X
XTR CL/GR AP 002200010131 F      1.50 X
KB EZ CHEESE 004400004553 F      2.50 X
YO VAN/M M S 004667500079 F      0.50 T
SOUR CREAM   007342000015 F      1.78 T
YOGURT       007047000712 F      2.06 T
STRWBRRY YOG 003663200104 F      0.54 T
PEACH YOGURT 003663200112 F      0.54 T
BLUEBRRY YOG 003663200108 F      0.54 T
STRWBRRY YOG 003663200104 F      0.54 T
FR MUSTA SQU 004150000700 F      0.74 X
```



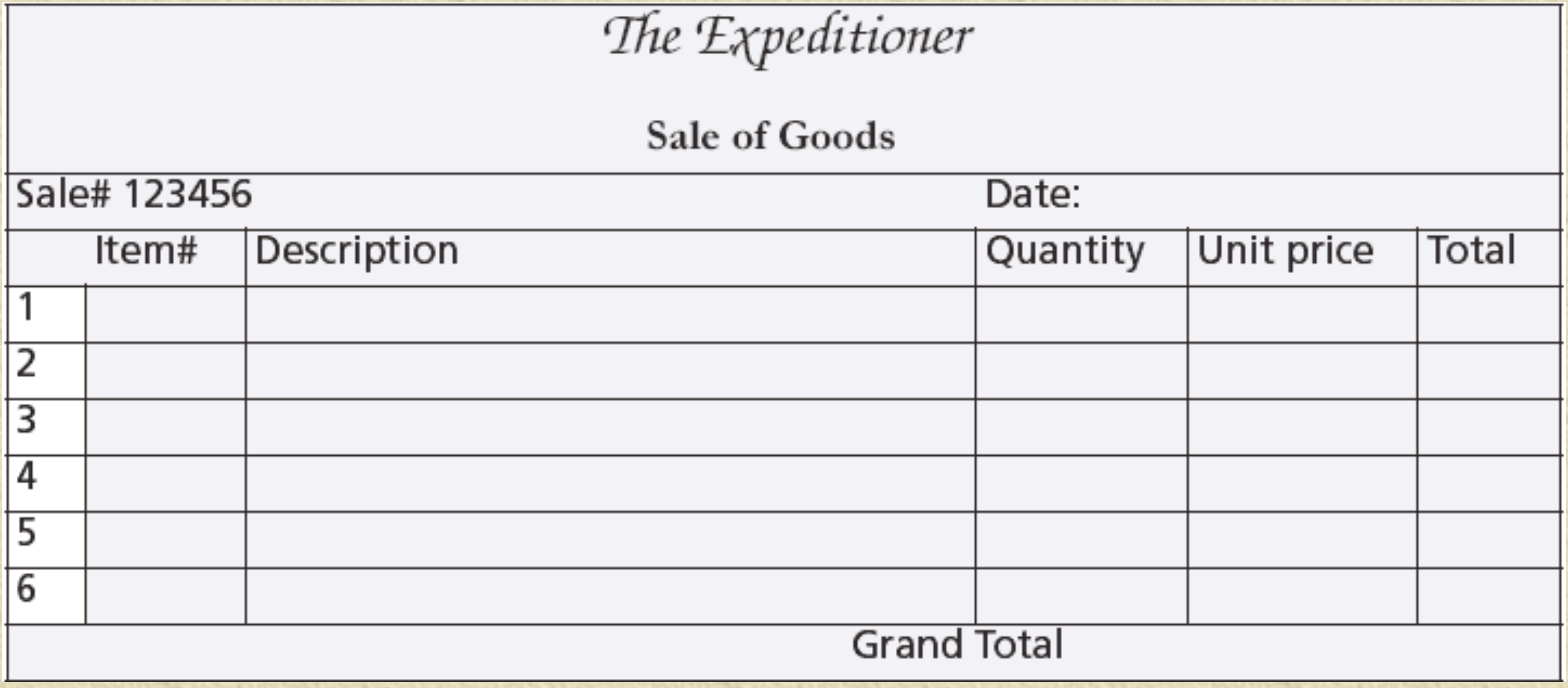
```
POCKY-STWBRY 007314115005 F      0.93 T     2
POCKY-STWBRY 007314115005 F      0.93 T
WHEATABLES   003010017083 F      2.00 X
SPICE        001121628746 F      1.27 T
RICE CAKES   003000016976 F      2.42 T
RTBY FMS CRT 007067812601 F      0.93 T
BATHERAPY SA 007989666876        4.47 X
BIORE MOIST  001910011173        9.94 X
FLY RIBBON   007247798010        0.97 X
SH FT PUMIC  007417025100        0.88 X
BATH FIZZ    001087970024        1.84 X
SH FT MASK   007417025101        0.88 X
SH FT REPAIR 007417025103        0.88 X
DOVE EG W/BW 001111116004        6.64 X
ACNE         038137003566        4.47 X
1  COLESLAW  007127912300 F      1.44 X
ROMAIN SALAD 007127926102 F      2.44 X
CUCUMBERS    000000004062KF
    2 AT  1 FOR     0.64         1.28 X
WHT ONIONS   000000004663KF
  0.70 lb  @  1 lb /1.16         0.81 X
BELL PEPPER  000000004065KF
    WAS 0.78 YOU SAVED 0.28
    2 AT  1 FOR     0.50         1.00 T
DIXIEHD PLTE 004200012281        3.72 X
DONUTS       000000009890KF
    4 AT  1 FOR     0.50         2.00 T
POTATO WEDGE 020828120128        1.28 T
HN FT CAPRI  063542423060       10.94 X
NECT YLW     000000004036KF
  1.32 lb  @  1 lb /1.74         2.30 X
 ** VOIDED ENTRY **
NECT YLW     000000004036KF
  1.32 lb  @  1 lb /1.74         2.30-X
PEACH YELLOW 000000004038KF
    WAS 1.74/lb YOU SAVED 0.50
  1.32 lb  @  1 lb /1.24         1.64 T
SBERRY 1     078035378403 F      1.98 X
GRAPE GRN    000000004022KF
    WAS 1.68/lb YOU SAVED 0.24
  1.75 lb  @  1 lb /1.44         2.52 T
POPCORN CHX  009787440012        1.64 T
                   SUBTOTAL    124.86
         TAX 1   7.500 %         9.36
                      TOTAL    134.22
                 VISA  TEND    134.22

[redacted]

PAYMENT SERVICE - E
                 CHANGE DUE       0.00

   # ITEMS SOLD 74
```

A sales form



The Expeditioner

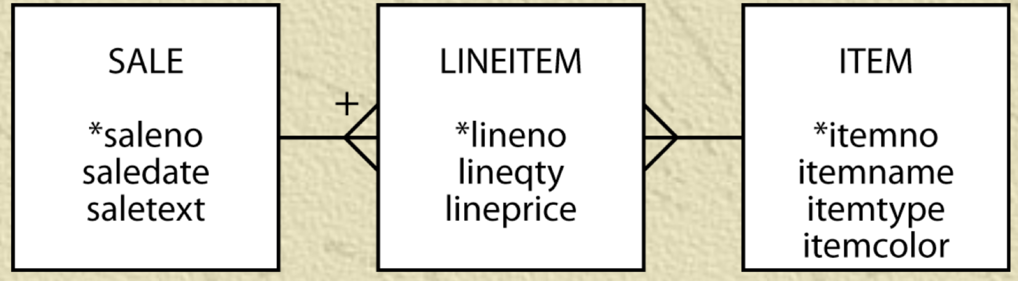
Sale of Goods

Sale# 123456 Date:

	Item#	Description	Quantity	Unit price	Total
1					
2					
3					
4					
5					
6					
			Grand Total		

The many-to-many relationship

- Create a third entity to map an m:m relationship
  - An associative entity
- The + on the crow's foot indicates that LINEITEM is identified by concatenating *saleno* and *lineno*

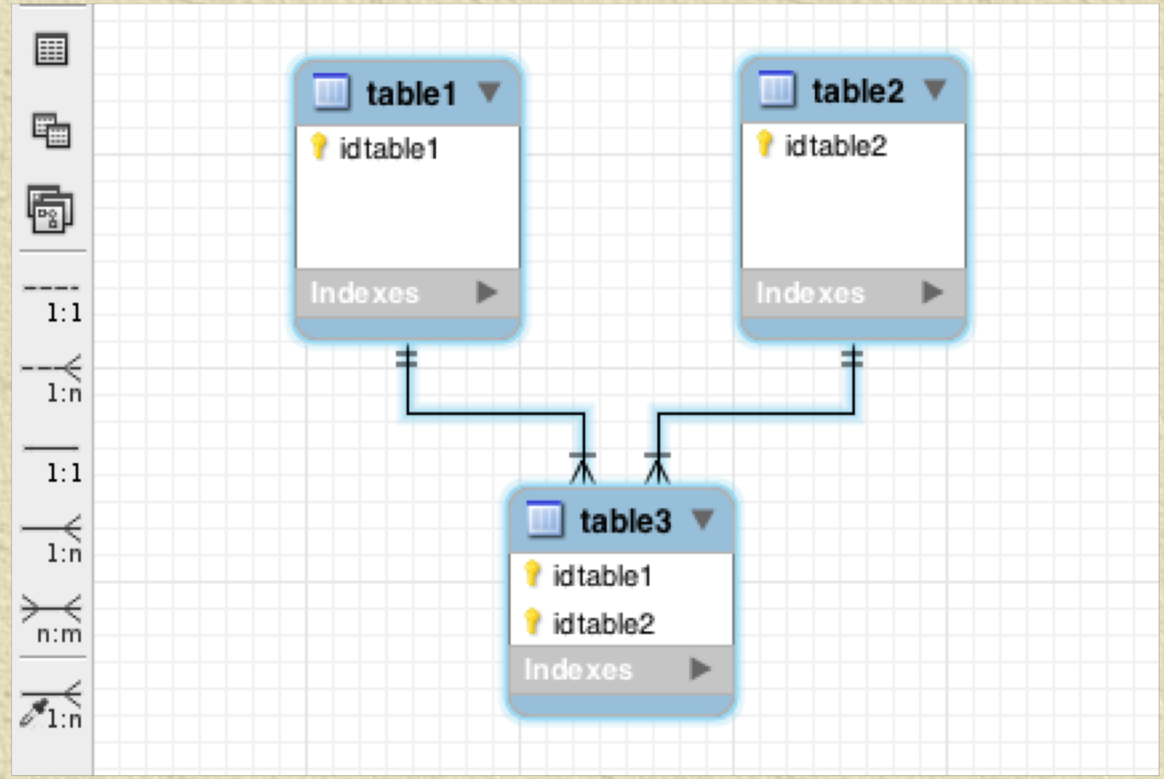


LINEITEM is known as a weak entity, and it has an ***identifying*** *relationship with SALE*

The many-to-many relationship

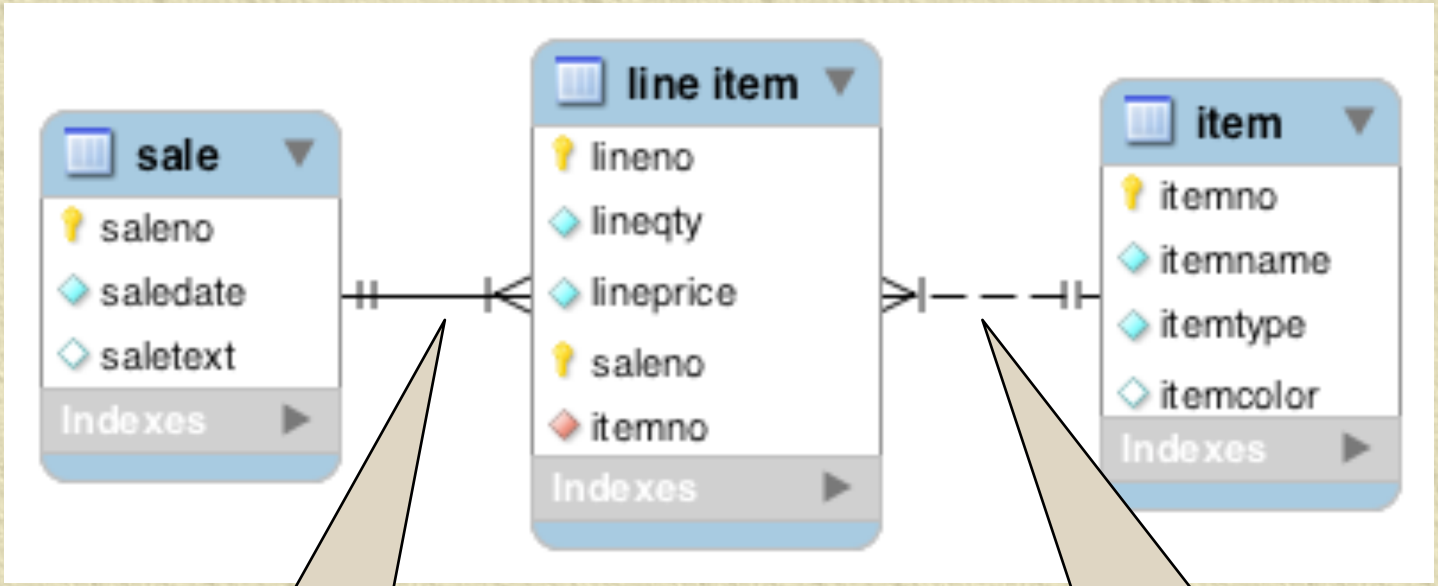
- MySQL Workbench

m:m symbol



The many-to-many relationship

- MySQL Workbench



Identifying relationship

Non-identifying relationship

Why a third entity?

- Store data about the relationship
- Think of an m:m as two 1:m relationships

Creating a relational database

- Same rules apply
- The associative table has two foreign keys
  - One for each of the entities in the m:m relationship
- A foreign key can also be part of the primary key of an associative entity

lineitem				
lineno	lineqty	lineprice	*saleno*	*itemno*
1	1	4.50	1	2
1	1	25.00	2	6
2	1	20.00	2	16
3	1	25.00	2	19

Creating a relational database

```
CREATE TABLE sale (
    saleno     INTEGER,
    saledate   DATE NOT NULL,
    saletext   VARCHAR(50),
      PRIMARY KEY(saleno));

CREATE TABLE item (
    itemno     INTEGER,
    itemname   VARCHAR(30) NOT NULL,
    itemtype   CHAR(1) NOT NULL,
    itemcolor  VARCHAR(10),
      PRIMARY KEY(itemno));

CREATE TABLE lineitem (
    lineno     INTEGER,
    lineqty    INTEGER NOT NULL,
    lineprice  DECIMAL(7,2) NOT NULL,
    saleno     INTEGER,
    itemno     INTEGER,
      PRIMARY KEY(lineno,saleno),
      CONSTRAINT fk_has_sale FOREIGN KEY(saleno) REFERENCES sale(saleno),
      CONSTRAINT fk_has_item FOREIGN KEY(itemno) REFERENCES item(itemno));
```

Exercise

- A keen field hockey fan wants to keep track of which countries won which medals in the various summer Olympics for both the men's and women's events
  - Design a data model
  - Create the database
  - Populate with data for the last two Olympics
    - http://en.wikipedia.org/wiki/Field_hockey_at_the_Summer_Olympics

A three table join

- Specify two matching conditions with the associative table in both join conditions

```
SELECT * FROM sale JOIN lineitem
   ON sale.saleno = lineitem.saleno
   JOIN item ON item.itemno = lineitem.itemno;
```

A three table join

- *List the names of items, quantity, and value of items sold on January 16, 2011*

```
SELECT itemname, lineqty, lineprice, lineqty*lineprice
  AS total FROM sale JOIN lineitem
     ON lineitem.saleno = sale.saleno
     JOIN item ON item.itemno = lineitem.itemno
     WHERE saledate = '2011-01-16';
```

itemname	lineqty	lineprice	total
Pocket knife—Avon	1	0.00	0.00
Safari chair	50	36.00	1800.00
Hammock	50	40.50	2025.00
Tent—8 person	8	153.00	1224.00
Tent—2 person	1	60.00	60.00

EXISTS

- Existential qualifier
- Returns *true* or *false*
- Returns *true* if the table contains at least one row satisfying the specified condition

Report all clothing items (type "C") for which a sale is recorded

```
SELECT itemname, itemcolor FROM item
   WHERE itemtype = 'C'
      AND EXISTS (SELECT * FROM lineitem
        WHERE lineitem.itemno = item.itemno);
```

itemname	itemcolor
Hat—Polar Explorer	Red
Boots—snake proof	Black
Pith helmet	White
Stetson	Black

```
SELECT itemname,
itemcolor FROM item
   WHERE itemtype = 'C'
      AND EXISTS (SELECT *
FROM lineitem
   WHERE lineitem.itemno
= item.itemno);
```

itemname	itemcolor
Hat—Polar Explorer	Red
Boots—snake proof	Black
Pith helmet	White
Stetson	Black

itemno	itemname	itemtype	itemcolor
1	Pocket knife—Nile	E	Brown
2	Pocket knife—Avon	E	Brown
3	Compass	N	—
4	Geopositioning system	N	—
5	Map measure	N	—
6	Hat—Polar Explorer	C	Red
7	Hat—Polar Explorer	C	White
8	Boots—snake proof	C	Green
9	Boots—snake proof	C	Black
10	Safari chair	F	Khaki
11	Hammock	F	Khaki
12	Tent—8 person	F	Khaki
13	Tent—2 person	F	Khaki
14	Safari cooking kit	E	—
15	Pith helmet	C	Khaki
16	Pith helmet	C	White
17	Map case	N	Brown
18	Sextant	N	—
19	Stetson	C	Black
20	Stetson	C	Brown

lineno	lineqty	lineprice	saleno	itemno
1	1	4.5	1	2
1	1	25	2	6
2	1	20	2	16
3	1	25	2	19
4	1	2.25	2	2
1	1	500	3	4
2	1	2.25	3	2
1	1	500	4	4
2	1	65	4	9
3	1	60	4	13
4	1	75	4	14
5	1	10	4	3
6	1	2.25	4	2
1	50	36	5	10
2	50	40.5	5	11
3	8	153	5	12
4	1	60	5	13
5	1	0	5	2

NOT EXISTS

- Returns *true* if the table contains no rows satisfying the specified condition

Report all clothing items (type "C") that have not been sold

```
SELECT itemname, itemcolor FROM item
  WHERE itemtype = 'C'
    AND NOT EXISTS
     (SELECT * FROM lineitem
       WHERE item.itemno = lineitem.itemno);
```

itemname	itemcolor
Hat—Polar Explorer	White
Boots—snake proof	Green
Pith helmet	Khaki
Stetson	Brown

```
SELECT itemname, itemcolor
FROM item
  WHERE itemtype = 'C'
   AND NOT EXISTS
 (SELECT * FROM lineitem
  WHERE item.itemno =
lineitem.itemno);
```

itemname	itemcolor
Hat—Polar Explorer	White
Boots—snake proof	Green
Pith helmet	Khaki
Stetson	Brown

itemno	itemname	itemtype	itemcolor
1	Pocket knife—Nile	E	Brown
2	Pocket knife—Avon	E	Brown
3	Compass	N	—
4	Geopositioning system	N	—
5	Map measure	N	—
6	Hat—Polar Explorer	C	Red
7	Hat—Polar Explorer	C	White
8	Boots—snake proof	C	Green
9	Boots—snake proof	C	Black
10	Safari chair	F	Khaki
11	Hammock	F	Khaki
12	Tent—8 person	F	Khaki
13	Tent—2 person	F	Khaki
14	Safari cooking kit	E	—
15	Pith helmet	C	Khaki
16	Pith helmet	C	White
17	Map case	N	Brown
18	Sextant	N	—
19	Stetson	C	Black
20	Stetson	C	Brown

lineno	lineqty	lineprice	saleno	itemno
1	1	4.5	1	2
1	1	25	2	6
2	1	20	2	16
3	1	25	2	19
4	1	2.25	2	2
1	1	500	3	4
2	1	2.25	3	2
1	1	500	4	4
2	1	65	4	9
3	1	60	4	13
4	1	75	4	14
5	1	10	4	3
6	1	2.25	4	2
1	50	36	5	10
2	50	40.5	5	11
3	8	153	5	12
4	1	60	5	13
5	1	0	5	2

Exercise

- Report all brown items that have been sold
- Report all brown items that have not been sold

Divide

- The universal qualifier
  - *forall*
- Not directly mapped into SQL
- Implement using `NOT EXISTS`

Find all items that have appeared in all sales

becomes

Find items such that there does not exist a sale in which this item does not appear

Divide

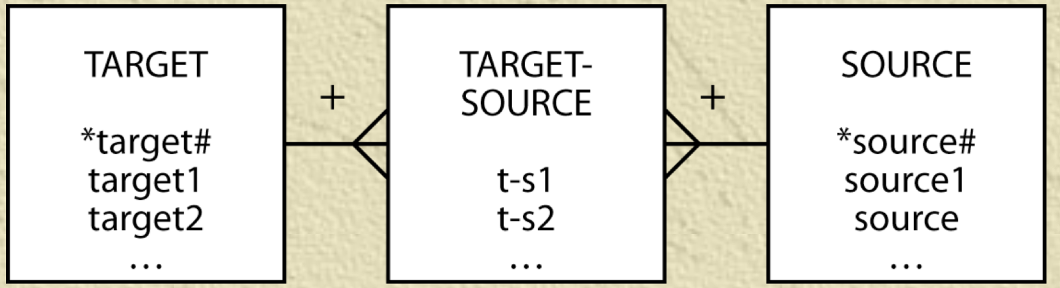
Find the items that have appeared in all sales

```
SELECT itemname FROM item
   WHERE NOT EXISTS
      (SELECT * FROM sale
         WHERE NOT EXISTS
            (SELECT * FROM lineitem
               WHERE lineitem.itemno = item.itemno
               AND lineitem.saleno = sale.saleno));
```

itemname
Pocket knife—Thames

See the book's web site for a detailed explanation of how divide works (Support/SQL Divide)

A template for divide



Find the target1 that have appeared in all sources

```
SELECT target1 FROM target
   WHERE NOT EXISTS
       (SELECT * FROM source
          WHERE NOT EXISTS
             (SELECT * FROM target-source
                WHERE target-source.target# = target.target#
                AND target-source.source# = source.source#));
```

Beyond the great divide

Find the items that have appeared in all sales

can be rephrased as

Find all the items for which the number of sales that include this item is equal to the total number of sales.

First determine the number of sales in which an item has appeared

```
SELECT item.itemno, item.itemname
   FROM item JOIN lineitem
      ON item.itemno = lineitem.itemno
          GROUP BY item. itemno, item.itemname
             HAVING COUNT(DISTINCT saleno)
             = (SELECT COUNT(DISTINCT saleno) FROM sale);
```

Second compare the number of sales to the total number of sales

Set operations

- `UNION`
  - Equivalent to OR
- `INTERSECT`
  - Equivalent to AND

UNION

List all items that were sold on January 16, 2011, or are brown.

```
SELECT itemname FROM item JOIN lineitem
   ON item.itemno = lineitem.itemno
   JOIN sale ON lineitem.saleno = sale.saleno
   WHERE saledate = '2011-01-16'
UNION
   SELECT itemname FROM item WHERE itemcolor = 'Brown';
```

itemname
Hammock
Map case
Pocket knife—Avon
Pocket knife—Nile
Safari chair
Stetson
Tent—2 person
Tent—8 person

INTERSECT

List all items that were sold on January 16, 2011, and are brown.

```
SELECT itemname FROM item JOIN lineitem
   ON item.itemno = lineitem.itemno
   JOIN sale ON lineitem.saleno = sale.saleno
   WHERE saledate = '2011-01-16'
INTERSECT
   SELECT itemname FROM item WHERE itemcolor = 'Brown';
```

itemname
Pocket knife—Avon

INTERSECT not supported by MySQL

Conclusion

- Introduced
  - m:m relationship
  - Associative entity
  - Weak entity
  - `EXISTS`
  - Divide
  - Set operations