## Database Model Levels

- A Conceptual model represents reality in an abstracted form that can be used in developing an information system in a wide variety of formats (e.g. relational, object-oriented, flat-file, etc.)
  - It is hardware and software independent
  - It is independent of any logical model type
- A Logical model represents reality in the format required by a particular database model (e.g. relational or object-oriented)
  - Is still hardware and software independent
  - Depends on the chosen logical model type
- A Physical model is created specifically for a particular database software package
  - Is dependent on hardware, software, and on the chosen logical model type

## Relational Database Model

- The relational model is a type of logical database model that was conceived by E.F. Codd in 1969
- The relational model is based on set theory and predicate logic
  - It is well formalized, so its behavior is predictable
- A relational database consists of tables (relations) that are linked together via the use of primary and foreign keys
  - A FOREIGN KEY in a table is a primary key from a different table that has been posted into the table to create a link between the two tables
- Relational database tables are made up of rows and columns
  - Rows are called the table extension or tuples
    - The ordering of rows in a table does not matter
  - Columns are called the table intension or schema
    - The ordering of columns in a table does not matter
    - All values in a column must conform to the same data format (e.g. date, text, currency, etc.)
  - Each cell in a database table (a row-column intersection) can contain only one value
    - no repeating groups are allowed

## Foreign Key Example

| SaleID | Date | Amount | Salesperson |
|---|---|---|---|
| 061401A | 6/14 | $4,218 | 123456 |
| 061401B | 6/14 | $6,437 | 654321 |
| 061501A | 6/15 | $1,112 | 654321 |

| SalespersonID | Name |
|---|---|
| 123456 | Fred |
| 654321 | Francis |

## Relational Database Model

- Some principles of the relational model
  - Entity Integrity
    - A primary key in a table must not contain a null value
      - Guarantees uniqueness of entities and enables proper referencing of primary key values by foreign key values
  - Referential Integrity
    - A value for a foreign key in a table must either
      - Be null (blank)
      - Match exactly a value for the primary key in the table from which it was posted
  - One Fact, One Place
    - Fact = a pairing of a candidate key attribute value with another attribute value
      - Facts are found in the extensional data

## Referential Integrity Example

**EXHIBIT 6–1 Foreign Key Examples**

(a) Meets referential integrity principle

**Sale**

| SaleID | Date | Amount | Salesperson |
|---|---|---|---|
| 061401A | 6/14 | $4,218 | 123456 |
| 061401B | 6/14 | $6,437 | 654321 |
| 061501A | 6/15 | $1,112 | 654321 |
| 061501B | 6/15 | $3,300 | |
| 061501C | 6/15 | $1,776 | |

**Salesperson**

| SalespersonID | Name | Telephone |
|---|---|---|
| 123456 | Fred | 555-0063 |
| 654321 | Francis | 555-0007 |

(b) Violates referential integrity principle

**Sale**

| SaleID | Date | Amount | Salesperson |
|---|---|---|---|
| 061401A | 6/14 | $4,218 | 123456 |
| 061401B | 6/14 | $6,437 | 654321 |
| 061501A | 6/15 | $1,112 | 654321 |
| 061501B | 6/15 | $3,300 | |
| 061501C | 6/15 | $1,776 | 234567 |

**Salesperson**

| SalespersonID | Name | Telephone |
|---|---|---|
| 123456 | Fred | 555-0063 |
| 654321 | Francis | 555-0007 |

## One Fact-One Place Violations One fact in multiple places

**Sale**

| SaleID | Date | Amount | | CustomerID | CustomerName | CustomerAddress |
|---|---|---|---|---|---|---|
| 8532 | Oct. 2 | $13 | 1 | C422 | Andy | 456 Pine St. |
| 9352 | Oct. 14 | $14 | | C821 | Jennifer | 987 Forest St. |
| 10215 | Oct. 27 | $20 | 3 | C363 | Arlie | 321 Beech St. |
| 14332 | Nov. 5 | $18 | 1 | C422 | Andy | 456 Pine St. |
| 17421 | Nov. 16 | $22 | 3 | C363 | Arlie | 321 Beech St. |

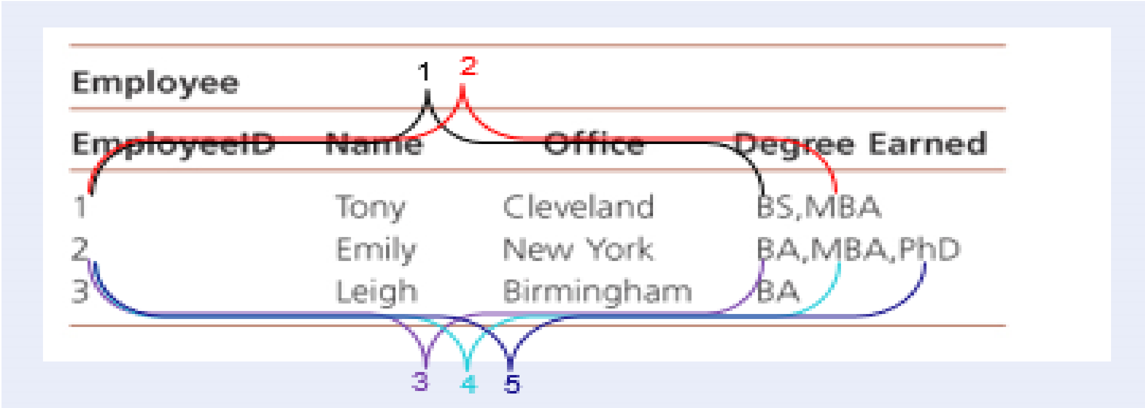
2

4

2

4

Multiple facts in one place



| Employee | | | |
|---|---|---|---|
| EmployeeID | Name | Office | Degree Earned |
| 1 | Tony | Cleveland | BS,MBA |
| 2 | Emily | New York | BA,MBA,PhD |
| 3 | Leigh | Birmingham | BA |



Each value of each attribute in a row is paired with the primary key, so if any cell has two or more attribute values, by definition there are multiple facts in one place (also known as a *repeating group*)

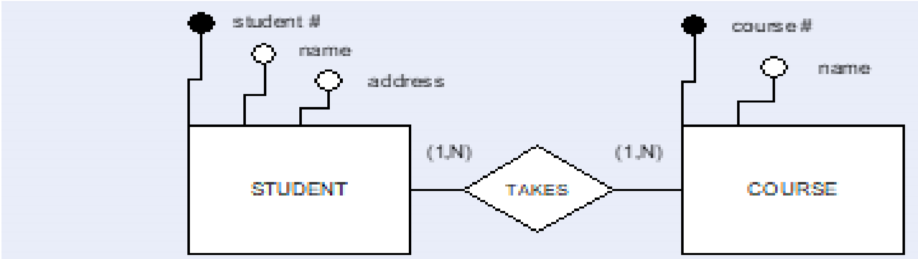
**Converting Conceptual to Relational**

- Step 1: Create a separate table to represent each entity in the conceptual model
    - 1A: Each attribute of the entity becomes a column in the relational table
    - 2A: Each instance (member) of the entity set will become a row in the relational table
- Steps 2-4 (detailed in the next few slides) involve determining whether each relationship in the conceptual model should be represented as a separate table or as a posted foreign key
    - Redundancy and Load are important determinants
        - Redundancy = one fact in multiple places or multiple facts in one place
        - Load = the percentage of non-null values in a column
    - Participation Cardinalities communicate some of the information regarding redundancy and load

**Relationship Conversion**

- Maximum Cardinalities
    - The general rule is to post into a "1" entity table
        - This avoids "repeating groups" redundancy
    - You can NEVER post into an "N" entity
        - This causes "repeating groups" redundancy
- Minimum Cardinalities
    - The general rule is to post into a "1" (mandatory) entity table
        - This avoids null values in the foreign key column
    - This rule should be violated in some circumstances (to be discussed soon)
- Step 2: Create a separate table to represent each many-to-many relationship in the conceptual model, I.e., for the following participation cardinality patterns (0,N)-(0,N) (0,N)-(1,N) (1,N)-(0,N) (1,N)-(1,N)
    - You <u>must</u> create a separate table to represent the relationship
        - The primary keys of the related entity tables are posted into the relationship table to form its primary key. This kind of primary key is called a composite or concatenated primary key
        - This avoids redundancy
        - There are no exceptions to this rule!!!
    - If you post a foreign key in either direction, redundancy <u>will</u> be a problem for many-to-many relationships

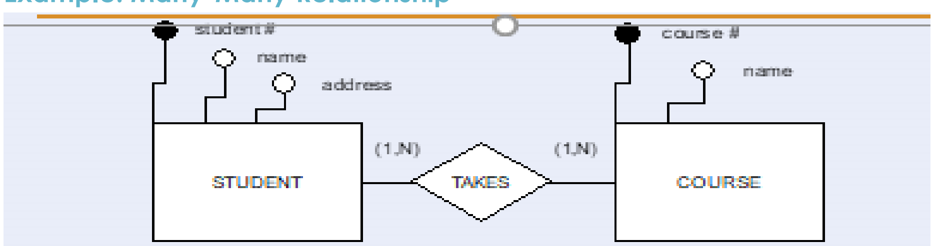
## Example: Many-Many Relationships




| Student# | Name | Address | Cours[illegible] |
|---|---|---|---|
| 1 | Tony | Cleveland | Ac[illegible]40[illegible]cg310[illegible] |
| 2 | Emily | New York | Ac[illegible]401, A[illegible]101 |
| 3 | Leigh | Birmingham | Acg4[illegible]Acg3[illegible] |

| Course# | Name | S[illegible]den[illegible] |
|---|---|---|
| Acg4401 | AIS | [illegible], 2[illegible] |
| Acg3101 | FAR 1 | [illegible]2, 3 |

## Example: Many-Many Relationship




| Student# | Name | Address |
|---|---|---|
| 1 | Tony | Cleveland |
| 2 | Emily | New York |
| 3 | Leigh | Birmingham |

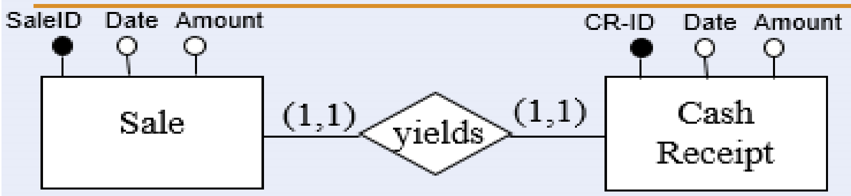
| Course# | Name |
|---|---|
| Acg4401 | AIS |
| Acg3101 | FAR 1 |

| Student# | Course# |
|---|---|
| 1 | Acg4401 |
| 1 | Acg3101 |
| 2 | Acg4401 |
| 2 | Acg3101 |
| 3 | Acg3101 |

## Relationship Conversion

- Step 3: For participation cardinality pattern (1,1)-(1,1), consider whether the two entities are conceptually separate or whether they should be combined
- If they should remain separate, then
  - 3A: Post the primary key from one entity's table into the other entity's table as a foreign key
  - 3B: It doesn't matter which entity's primary key is posted into the other entity's table, but DO NOT post both
  - DO NOT make a separate table
  - Redundancy is automatically avoided and load is not an issue when you post a foreign key into either table in a (1,1)-(1,1) relationship

## Example: (1,1)-(1,1)




| SaleID | Date | Amount | CR-ID * |
|---|---|---|---|
| S1 | 6/12 | $10 | CR1 |
| S2 | 6/12 | $15 | CR2 |
| S3 | 6/13 | $12 | CR3 |

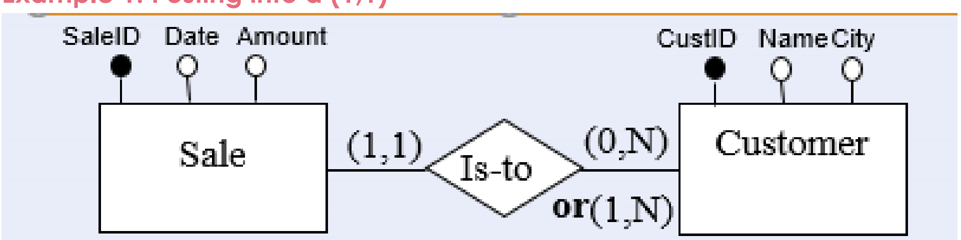
| CR-ID | Date | Amount | S-ID * |
|---|---|---|---|
| CR1 | 6/12 | $10 | S1 |
| CR2 | 6/12 | $15 | S2 |
| CR3 | 6/13 | $12 | S3 |

**Choose ONE of these; DO NOT do both!!**

## Relationship Conversion

- Step 4: For remaining relationships that have (1,1) participation by one entity set, post the related entity's primary key into the (1,1) entity's table as a foreign key
  - I.e., for the following participation cardinality patterns (0,N)-(1,1) (1,N)-(1,1) (1,1)-(0,N) (1,1)-(1,N) (0,1)-(1,1) (1,1)-(0,1)
    - Do NOT make a separate table
    - Post a foreign key INTO the (1,1) entity's table from the other entity's table
    - Redundancy is avoided and load is not an issue if you follow this instruction
    - If you post the opposite direction, either redundancy [for N maximums] OR load [for 0 minimums] will be a problem

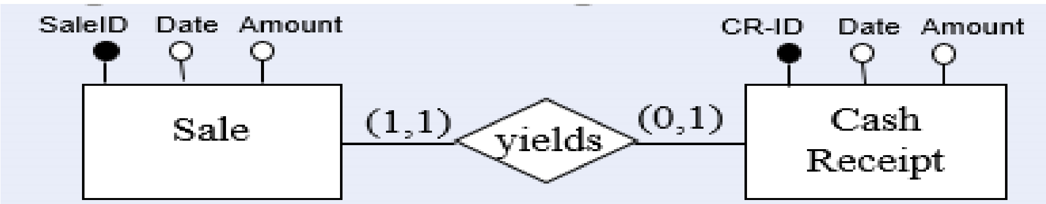
## Example 1: Posting into a (1,1)




| SaleID | Date | Amount | Cust-ID* |
|---|---|---|---|
| S1 | 6/12 | $10 | C1 |
| S2 | 6/12 | $15 | C2 |
| S3 | 6/13 | $12 | C1 |

| Cust-ID | Name | Address | S-ID* |
|---|---|---|---|
| C1 | Heather | Walnut Creek | S3 |
| C2 | Steven | Cincinnati | S2 |
| C3 | Dave | Cincinnati | |

### Example 2: Posting into a (1,1)




| SaleID | Date | Amount | CR-ID* |
|---|---|---|---|
| S1 | 6/12 | $10 | CR1 |
| S2 | 6/12 | $15 | CR2 |
| S3 | 6/13 | $12 | CR3 |

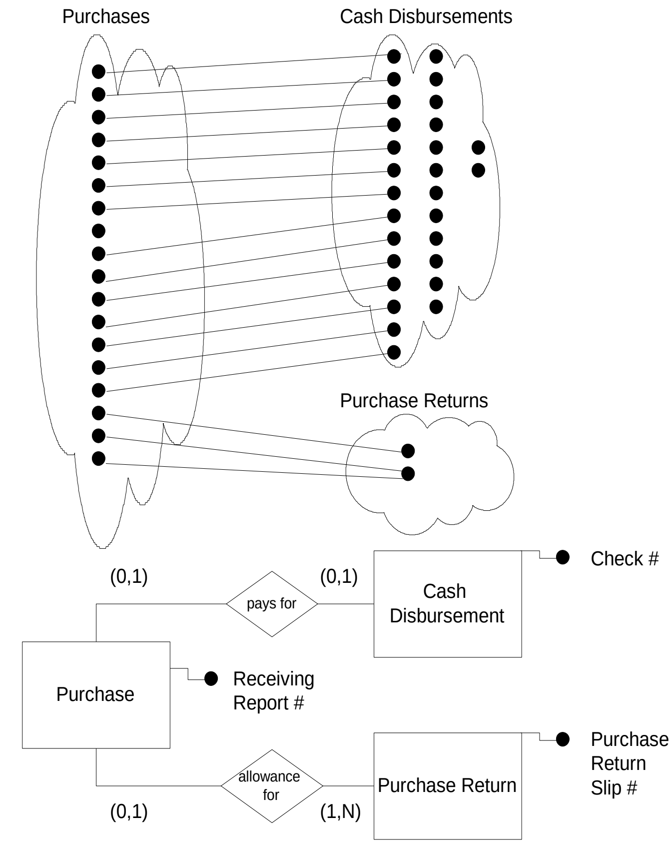
| CR-ID | Date | Amount | S-ID* |
|---|---|---|---|
| CR1 | 6/12 | $10 | [illegible] |
| CR2 | 6/12 | $15 | S2 |
| CR3 | 6/13 | $12 | [illegible] |
| CR4 | 6/13 | $1,000 | |

### Relationship Conversion

- Step 5: For remaining relationships that have (0,1) participation by one or both of the entities, consider load
  I.e., for the following participation cardinality patterns
  (0,N)-(0,1) (1,N)-(0,1) (0,1)-(0,N) (0,1)-(1,N) (0,1)-(0,1)
  - The rule for maximum cards requires posting into a (0,1) or making a separate table; you CANNOT post into the (0,N) or (1,N)
  - The rule for minimum cards says you really shouldn't post into the (0,1) because it will create null values that waste valuable space in the database
    - However, if a separate table would waste more space, then it is better to follow the maximum rule and break the minimum rule
  - 5A: Post the related entity's primary key into the (0,1) entity's table as a foreign key for any relationships for which that results in a high load
  - 5B: Create a separate table for any relationships for which posting a foreign key results in low load
    - Note: For (0,1)-(0,1), step 5A, post whichever direction results in highest load; if neither direction yields high load, then follow step 5B

### Example: Load Considerations

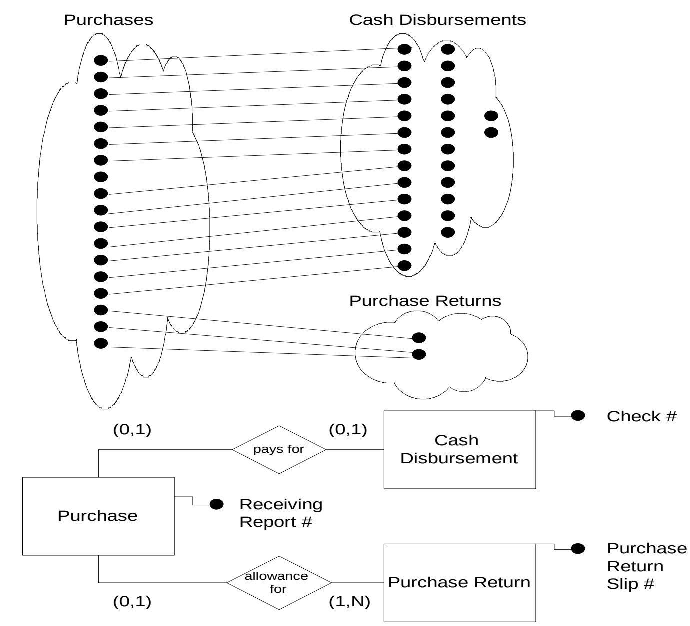
- Some cash disbursements (13/26) pay for purchases
  - If we post Receiving Report# into Cash Disbursement, 13 out of 26 will be non-null
  - This is a medium load
  - Might be worth breaking minimum rule
  - Consider other posting option
- Most purchases (14/18) result in cash disbursements
  - If we post Check# into Purchase, 14 out of 18 will be non-null
  - This is a high load
  - Worth breaking the minimum rule




Conclusion: post Check# into Purchase table to represent the "pays for" relationship

### Example: Load considerations

- Few purchases (3/18) result in purchase returns
  - If we post Purchase Return Slip# into Purchase, only 3 out of 18 will be non-null
  - This is low load
  - Must either make a separate table or consider posting the other direction
- Can't post receiving report# into purchase return because one purchase return slip # can be associated with multiple purchases

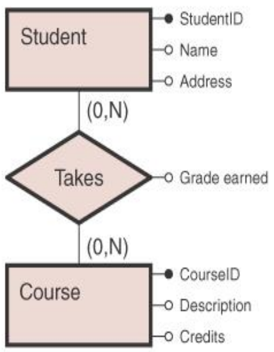


Conclusion: Make a separate table to represent the "allowance for" relationship

## Relationship Attribute Placement

- If relationship becomes a separate table, then relationship attributes are placed in that table
- If relationship can be represented by a posted foreign key, relationship attribute is posted alongside the foreign key

Diagram format




Grammar format

Entity: Student
Attributes: StudentID
Name
Address
Identifier: StudentID

Entity: Course
Attributes: CourseID
Description
Credits
Identifier: CourseID

Relationship: Takes
Connected entities: (0,N) Student
(0,N) Course
Attributes: Grade earned

**Student**

| StudentID | Name | Address |
|---|---|---|
| 999888 | Mildred | 123 Almanac St. |
| 888777 | Kent | 456 Market Dr. |
| 777666 | Candace | 789 Harriet Ave. |

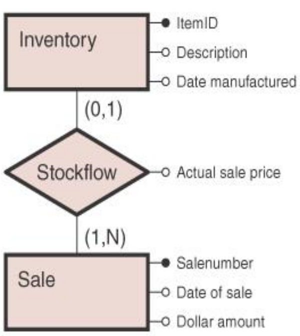
**Course**

| CourseID | Description | Credits |
|---|---|---|
| ACG611 | Advanced AIS | 3 |
| FIN642 | Financial Markets | 3 |
| MIS650 | IT Management | 3 |

**Takes**

| StudentID | CourseID | Grade Earned |
|---|---|---|
| 999888 | ACG611 | B |
| 999888 | MIS650 | A– |
| 888777 | MIS650 | B+ |

Diagram format




Grammar format

Entity: Inventory
Attributes: ItemID
Description
Date manufactured
Identifier: ItemID

Entity: Sale
Attributes: Salenumber
Date of sale
Dollar amount
Identifier: Salenumber

Relationship: Stockflow
Connected entities: (0,1) Inventory
(1,N) Sale
Attributes: Actual sale price

**Inventory**

| ItemID | Description | Date Manufactured | Salenumber | Actual Sale Price |
|---|---|---|---|---|
| I1 | Big blue item | 9/24/05 | 1 | $450 |
| I2 | Triangle green item | 9/25/05 | 1 | $375 |
| I3 | Small square item | 9/26/05 | | |
| I4 | Medium pink item | 9/27/05 | 2 | $500 |

**Sale**

| Salenumber | Date | Dollar Amount |
|---|---|---|
| S1 | 10/12/05 | $825 |
| S2 | 10/15/05 | $500 |

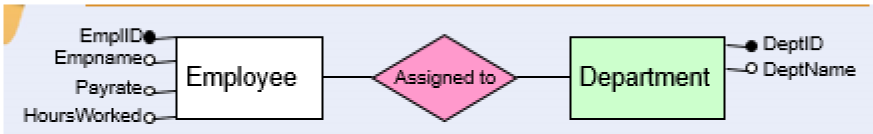
### Fixing One Fact Multiple Places

**Employee**

| EmpID | EmpName | Payrate | Hours Worked | Dept# | DeptName |
|---|---|---|---|---|---|
| 8532 | Andy | $13 | 36 | D423 | Audit |
| 7352 | Jennifer | $14 | 45 | D423 | Audit |
| 215 | Arlie | $20 | 50 | D777 | ISAAS |
| 4332 | Craig | $18 | 60 | D821 | Tax |
| 74 | Steven | $22 | 64 | D821 | Tax |

- What facts are in multiple places in this table?
- Reverse engineer to get the ER model that this table must represent
- Is the ER model that results in this table correct?
- What SHOULD the ER model have been instead?
- What is the correct relational model?

### Fixing One Fact Multiple Places




**Employee**

| EmpID | EmpName | Payrate | Hours Worked | Dept# |
|---|---|---|---|---|
| 8532 | Andy | $13 | 36 | D423 |
| 7352 | Jennifer | $14 | 45 | D423 |
| 215 | Arlie | $20 | 50 | D777 |
| 4332 | Craig | $18 | 60 | D821 |
| 74 | Steven | $22 | 64 | D821 |

**Department**

| Dept# | DeptName |
|---|---|
| D423 | Audit |
| D777 | ISAAS |
| D821 | Tax |

### Fixing Multiple Facts in One Place

**Warehouse**

| Warehouse# | Address | QOH |
|---|---|---|
| W1 | 123 Oak | 2,14,784 |
| W2 | 456 Pine | 4,23,873 |

**Inventory**

| Product# | Description | StdCost | QOH |
|---|---|---|---|
| AB12 | Granddaddy | $5,000 | 2,4 |
| BC445 | Mama | $3,000 | 14,23 |
| DD2 | Littlebabe | $100 | 784,873 |

**InventoryInWarehouse**

| Warehouse# | Product# |
|---|---|
| W1 | AB12 |
| W1 | BC445 |
| W1 | DD2 |
| W2 | AB12 |
| W2 | BC445 |
| W2 | DD2 |

- What facts are in multiple places?
- How could this be avoided?

**Warehouse**

| <u>Warehouse#</u> | Address |
|---|---|
| W1 | 123 Oak |
| W2 | 456 Pine |

**Inventory**

| <u>Product#</u> | Description | StdCost |
|---|---|---|
| AB12 | Granddaddy | $5,000 |
| BC445 | Mama | $3,000 |
| DD2 | Littlebabe | $100 |

**InventoryInWarehouse**

| <u>Warehouse#</u> | <u>Product#</u> | QOH |
|---|---|---|
| W1 | AB12 | 2 |
| W1 | BC445 | 14 |
| W1 | DD2 | 784 |
| W2 | AB12 | 4 |
| W2 | BC445 | 23 |
| W2 | DD2 | 873 |

### Relational Database Design Summary

- The relational model is based on set theory and predicate logic and the resultant relations (tables) can be manipulated for information retrieval purposes if they are properly constructed
- To create well-behaved tables, follow the rules we discussed
  - Conversion rules for cardinality patterns
  - One Fact-One Place
- Think at the data (extensional) level!!
- When creating physical databases, use the conceptual and logical models to help you realize the important issues and potential pitfalls