

Database Management Systems

- Design and Creation

Doug Shook

Database Creation

- Design is very important
 - Long lasting implications
- How is our data being stored again?
 - How do we manipulate data?

Example

- We wish to create a database for the following:
 - We have a company that is organized into departments
 - Name, number, employee
 - May have several locations
 - Each department controls a number of projects
 - Name, number, single location
 - Each department as a number of employees
 - Each employee can work for one department, but may have many projects
 - Number, name, address, salary
 - Each employee may have a number of dependents
 - Name, birthdate, relationship

Creating a Database

- First step:
 - Define your entities and attributes
 - Simple vs. Composite attributes

Key Attributes

- Key attributes are values that must be unique for an entity
 - What would qualify in this example?

Design

- Let's start drawing a diagram to represent our design

Relationships

- Identifying relationships is important
 - Why?
- What types of relationships exist?
- What relationships exist in our example?
- Recursive Relationships
- Let's update our design

Normalization

- Goal: reduce data redundancy
 - Data stored in exactly one place
- Accomplished by applying forms
 - Seven forms total
 - Three is sufficient

Normalization

- First normal form
 - The value stored at the intersection of each row and column must be scalar
 - A table must not contain any repeating column

- Will still likely have repeating values in rows

Normalization

| | VendorName | InvoiceNumber | ItemDescription |
|---|--------------------|---------------|----------------------------------|
| 1 | Cahners Publishing | 112897 | VB ad, SQL ad, Library directory |
| 2 | Zylka Design | 97/522 | Catalogs, SQL flyer |
| 3 | Zylka Design | 97/533B | Card revision |

| | VendorName | InvoiceNumber | ItemDescription 1 | ItemDescription2 | ItemDescription3 |
|---|--------------------|---------------|-------------------|------------------|-------------------|
| 1 | Cahners Publishing | 112897 | VB ad | SQL ad | Library directory |
| 2 | Zylka Design | 97/522 | Catalogs | SQL flyer | NULL |
| 3 | Zylka Design | 97/533B | Card revision | NULL | NULL |

Normalization

| | VendorName | InvoiceNumber | ItemDescription |
|---|--------------------|---------------|-------------------|
| 1 | Cahners Publishing | 112897 | VB ad |
| 2 | Cahners Publishing | 112897 | SQL ad |
| 3 | Cahners Publishing | 112897 | Library directory |
| 4 | Zylka Design | 97/522 | Catalogs |
| 5 | Zylka Design | 97/522 | SQL flyer |
| 6 | Zylka Design | 97/533B | Card revision |

Normalization

- Second Normal Form
 - Every non-key column must depend on the entire primary key
 - If the above is false it indicates there are multiple entities within the table
- Application
 - Move columns that don't depend on the entire key to a different table
 - Establish a relationship between the tables
- The step removes redundant row data

Normalization

| | InvoiceID | VendorName | InvoiceNumber | InvoiceSequence | ItemDescription |
|---|-----------|--------------------|---------------|-----------------|-------------------|
| 1 | 1 | Cahners Publishing | 112897 | 1 | VB ad |
| 2 | 2 | Cahners Publishing | 112897 | 2 | SQL ad |
| 3 | 3 | Cahners Publishing | 112897 | 3 | Library directory |
| 4 | 4 | Zylka Design | 97/522 | 1 | Catalogs |
| 5 | 5 | Zylka Design | 97/522 | 2 | SQL flyer |
| 6 | 6 | Zylka Design | 97/533B | 1 | Card revision |

Normalization

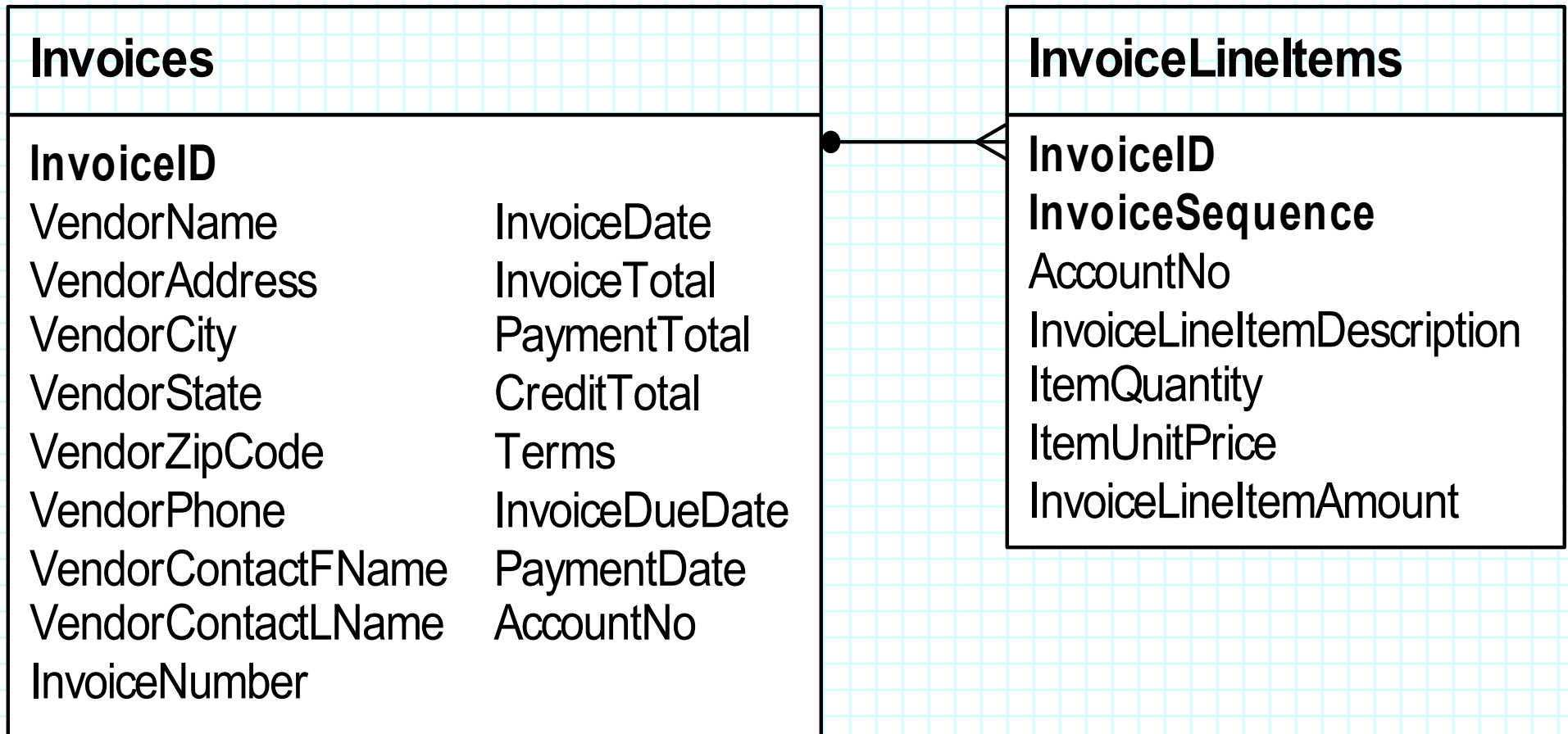
| | InvoiceNumber | VendorName | InvoiceID |
|---|---------------|--------------------|-----------|
| 1 | 112897 | Cahners Publishing | 1 |
| 2 | 97/522 | Zylka Design | 2 |
| 3 | 97/533B | Zylka Design | 3 |

| | InvoiceID | InvoiceSequence | ItemDescription |
|---|-----------|-----------------|-------------------|
| 1 | 1 | 1 | VB ad |
| 2 | 1 | 2 | SQL ad |
| 3 | 1 | 3 | Library directory |
| 4 | 2 | 1 | Catalogs |
| 5 | 2 | 2 | SQL flyer |
| 6 | 3 | 1 | Card revision |

Normalization

- Third Normal Form
 - Each non-key column must depend only on the primary key
- If a column does not depend only on the primary key
 - Assigned to the wrong table
 - Can be computed from other columns
 - Derived data

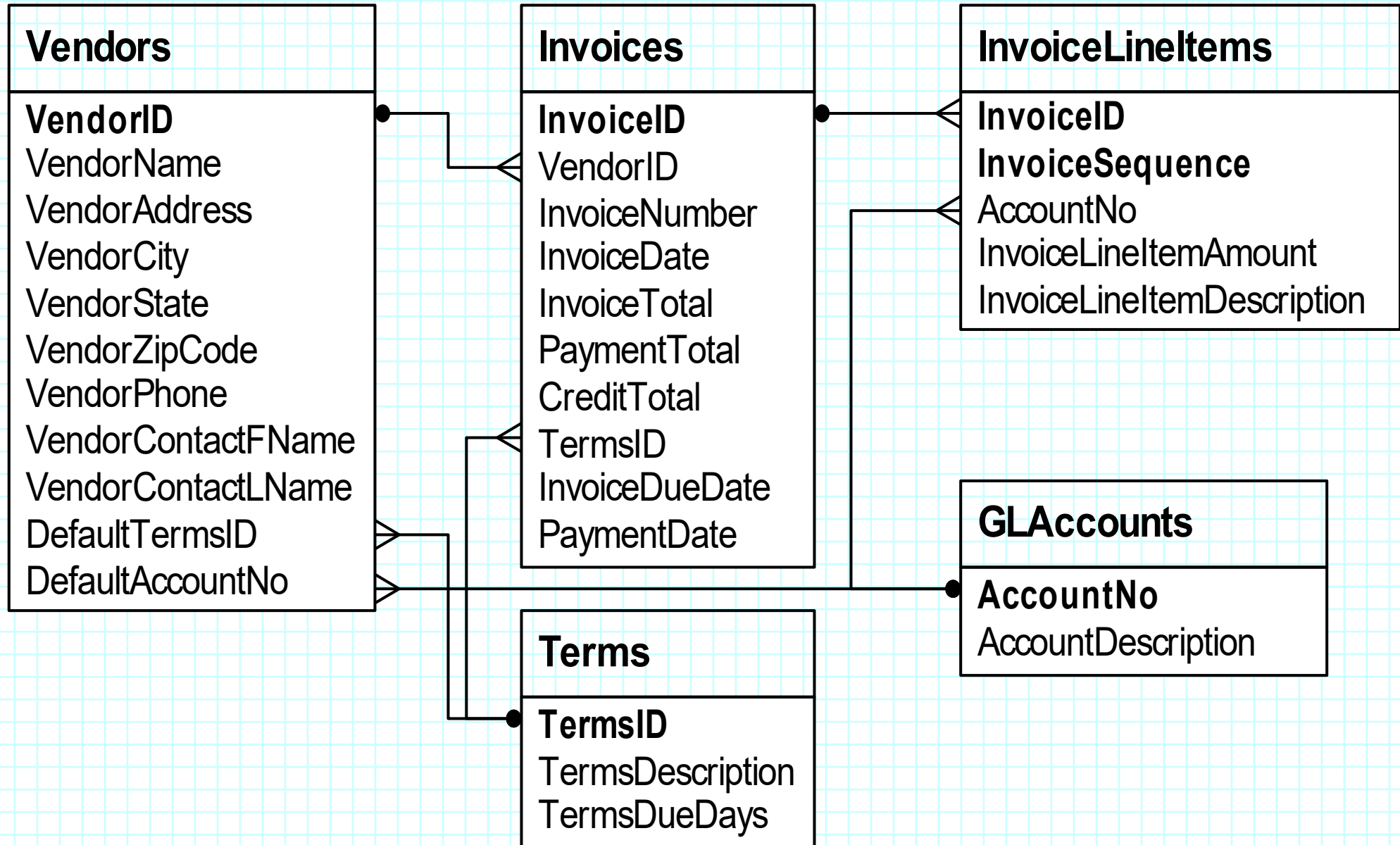
Normalization



Normalization

- Does the vendor information depend only on the InvoiceID column?
- Does the Terms column depend only on the InvoiceID column?
- Does the AccountNo column depend only on the InvoiceID column?
- Can the InvoiceDueDate and InvoiceLineItemAmount columns be derived from other data?

Normalization



Implementation

- How can we turn this into an actual database?
 - What are we missing?

Data Types

- Each column has a data type associated with it
 - Not exactly the same as the data types from other languages....
- What data types have you seen?

Creating Databases

- Each table must be contained in a separate database
- `CREATE DATABASE company;`
- What happens when we run this query?

Create Tables

- When we create tables, we must specify column names and types
 - Primary key?
 - Foreign key?
- `CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE);`
- `NULL, AUTO_INCREMENT`
- What happens when we run this query?

INSERT

- Enters a new row
 - Must specify values
 - NULL?
 - AUTO_INCREMENT?
 - Data types?

```
INSERT INTO pet(name, owner, species, sex,  
birthdate) VALUES ('Seth', 'Doug Shook', 'Cat',  
'M', '2007-04-03');
```

UPDATE

- Used to modify existing values

```
UPDATE pet  
SET name = 'Franklin'  
WHERE name = 'Seth';
```

- Order of operations?

DELETE

- Removes a row

```
DELETE FROM pet  
WHERE name = 'Franklin';
```

- Careful!

Exercises

- Practice creating the remaining tables from our employees example
- Practice inserting, updating, and deleting values from the tables you created