

Session 16

How to design a database

Objectives

- How to design a data structure
- How to normalize a data structure
- A database design tool

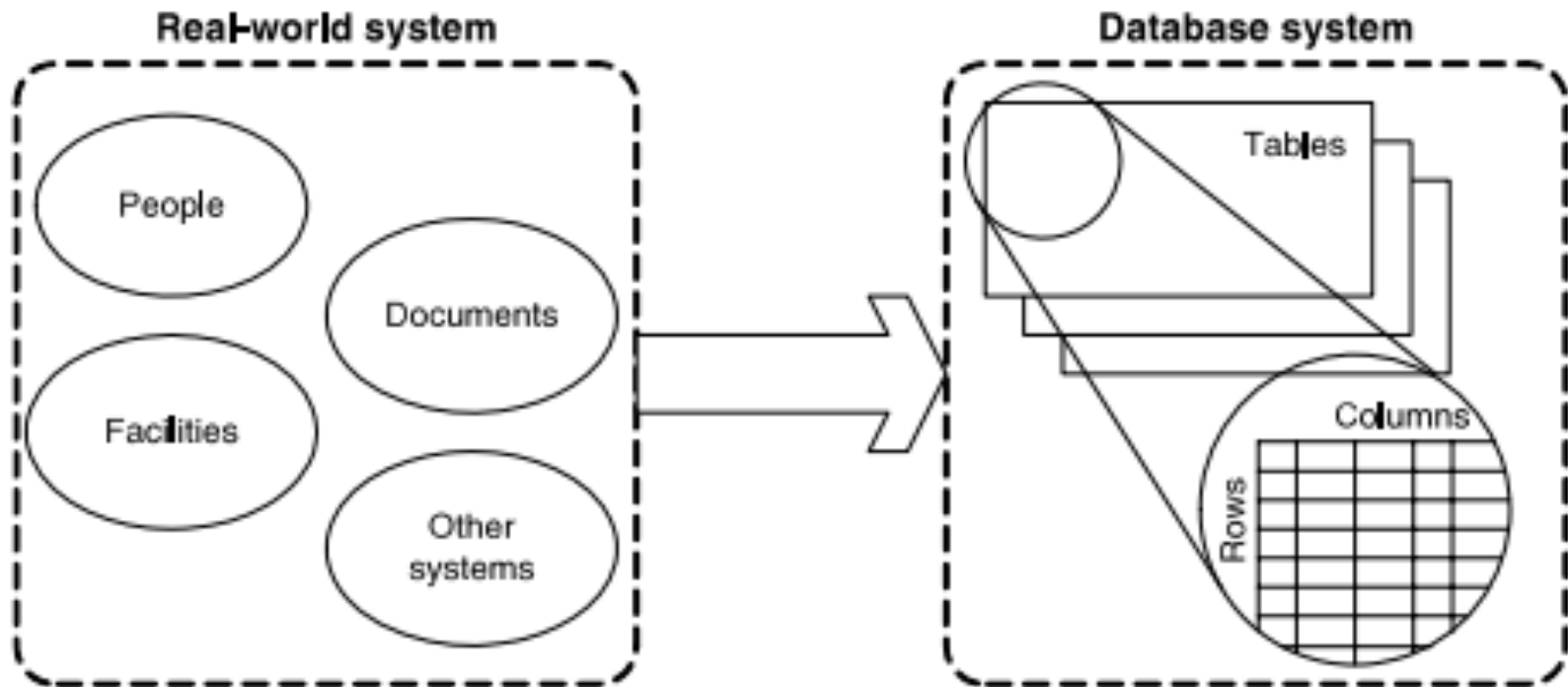
How to design a data structure

The basic steps for designing a data structure

- A relational database system should model the real-world environment where it's used.
- A table in a relational database typically represents an object, or entity, in the real world.
- To model a database and the relationships between its tables after a real-world system, you can use a technique called entity-relationship (ER) modeling.

The basic steps for designing a data structure (cont.)

- A database system is modeled after



The basic steps for designing a data structure (cont.)

- The six basic steps for designing a data structure
 - Step 1: Identify the data elements
 - Step 2: Subdivide each element into its smallest
 - Step 3: Identify the tables and assign columns
 - Step 4: Identify the primary and foreign keys
 - Step 5: Review whether the data structure is normalized
 - Step 6: Identify the indexes

How to identify the data elements

- Depending on the nature of the system, you can identify data elements in a variety of ways, including interviewing users, analyzing existing systems, and evaluating comparable systems.
- The data elements identified on the invoice document

Vendor name	Invoice date	Item extension
Vendor address	Invoice terms	Vendor sales contact name
Vendor phone number	Item part number	Vendor sales contact extension
Vendor fax number	Item quantity	Vendor AR contact name
Vendor web address	Item description	Vendor AR contact extension
Invoice number	Item unit price	Invoice total

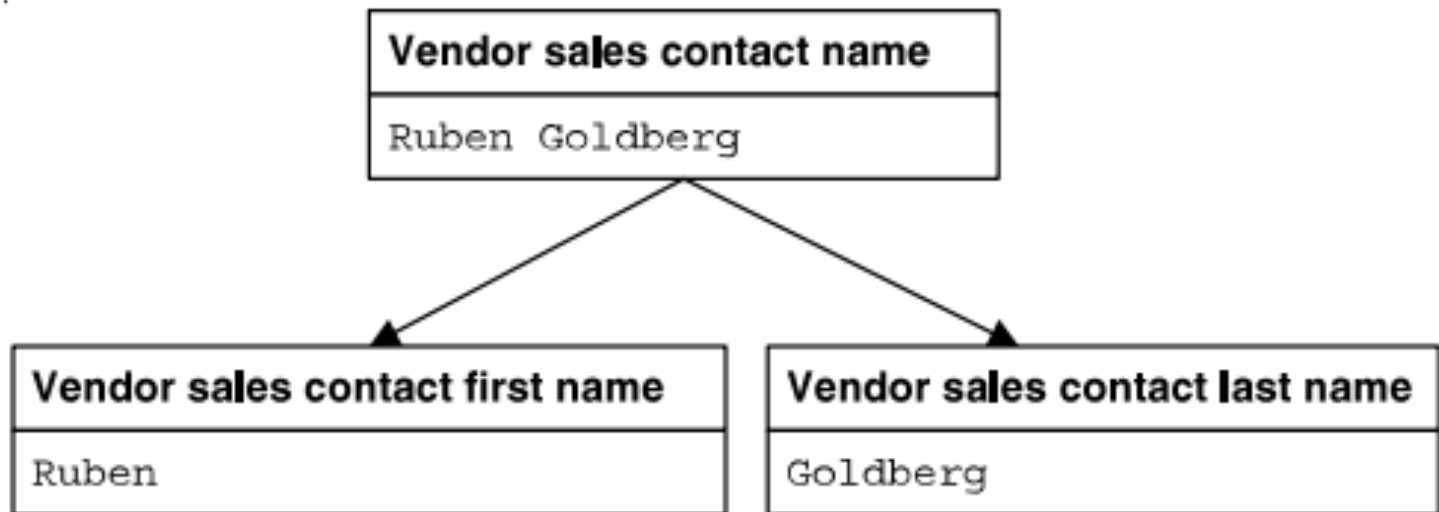
How to identify the data elements (cont.)

- An invoice that can be used to identify data elements

Acme Fabrication, Inc.				
<i>Custom Contraptions, Contrivances and Confabulations</i>			Invoice Number: I01-1088	
1234 West Industrial Way East Los Angeles California 90022			Invoice Date: 10/05/10	
800.555.1212 fax 562.555.1213 www.acmefabrication.com			Terms: Net 30	
Part No.	Qty.	Description	Unit Price	Extension
CUST345	12	Design service, hr	100.00	1200.00
457332	7	Baling wire, 25x3ft roll	79.90	559.30
50173	4375	Duct tape, black, yd	1.09	4768.75
328771	2	Rubber tubing, 100ft roll	4.79	9.58
CUST281	7	Assembly, hr	75.00	525.00
CUST917	2	Testing, hr	125.00	250.00
		Sales Tax		245.20
Your salesperson: Ruben Goldberg, ext 4512			\$7,557.83	
Accounts receivable: Inigo Jones, ext 4901			PLEASE PAY THIS AMOUNT	
Thanks for your business!				

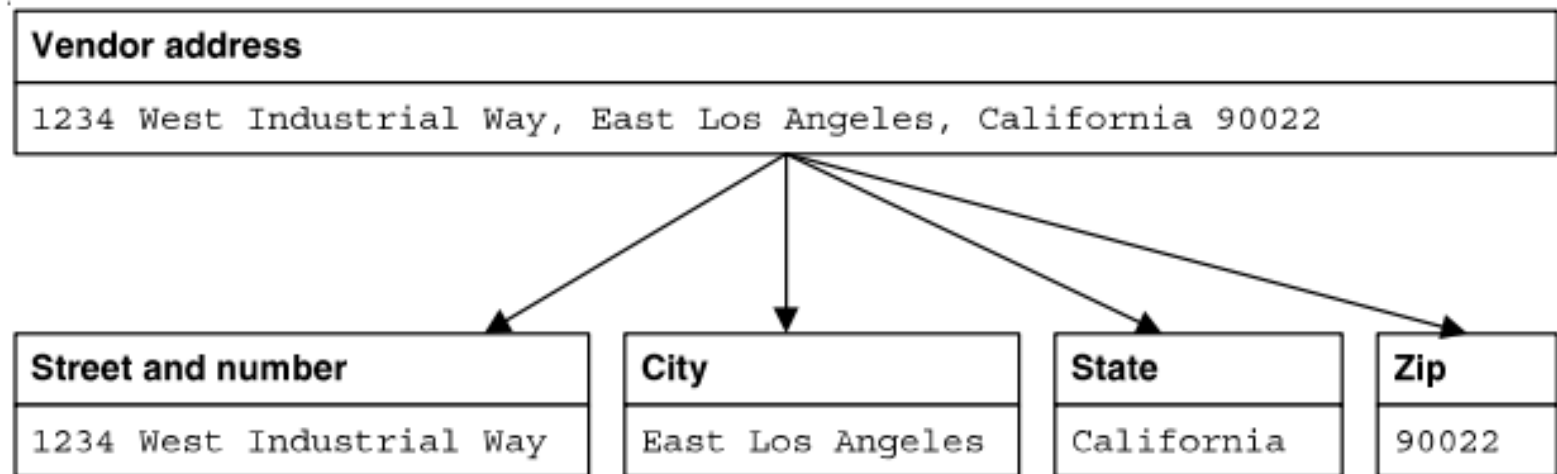
How to subdivide the data elements

- If a data element contains two or more components, you should consider subdividing the element into those components.
- A name that's divided into first and last names



How to subdivide the data elements (cont.)

- An address that's divided into street address, city, state, and zip code



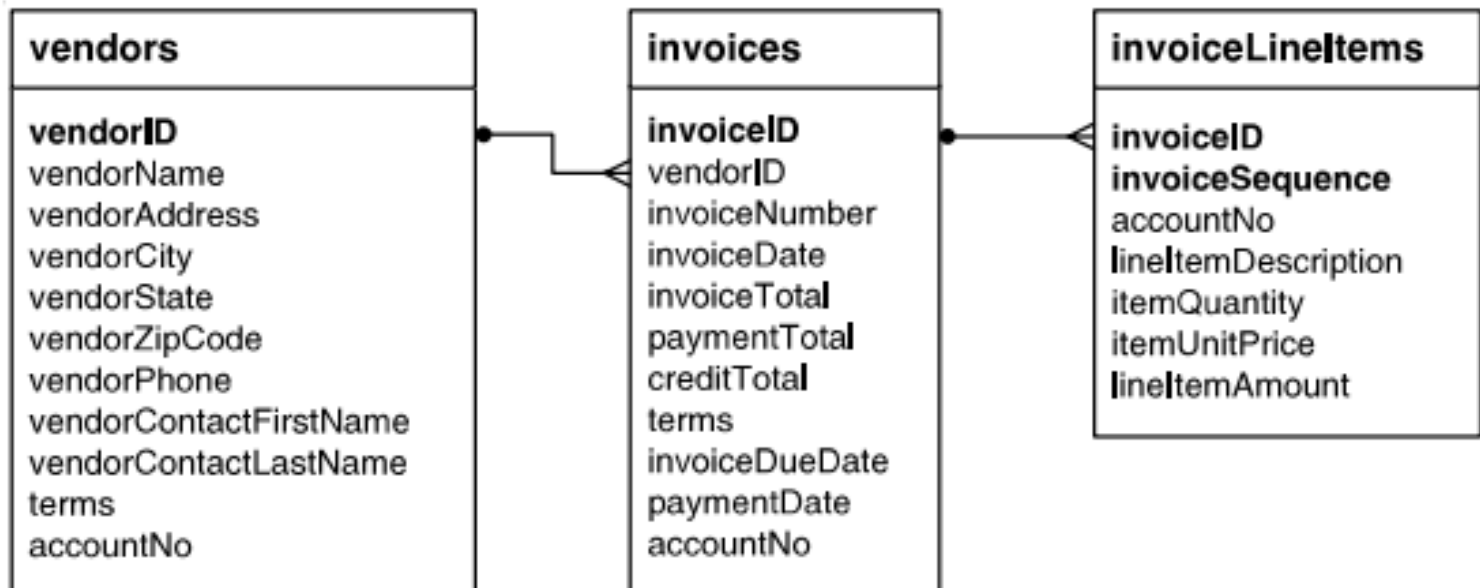
How to identify the tables and assign columns

- Possible tables and columns for an accounts payable system

Vendors	Invoices	Invoice line items
Vendor name	Invoice number*	Invoice number*
Vendor address	Invoice date	Item part number
Vendor city	Terms*	Item quantity
Vendor state	Invoice total	Item description
Vendor zip code	<i>Payment date</i>	Item unit price
Vendor phone number	<i>Payment total</i>	Item extension
Vendor fax number	<i>Invoice due date</i>	<i>Account number*</i>
Vendor web address	<i>Credit total</i>	<i>Sequence number</i>
Vendor contact first name	<i>Account number*</i>	
Vendor contact last name		
Vendor contact phone		
Vendor AR first name		
Vendor AR last name		
Vendor AR phone		
<i>Terms*</i>		
<i>Account number*</i>		

How to identify the primary and foreign keys

- Each table should have a primary key that uniquely identifies each row.
- The relationships between the tables in the accounts payable system

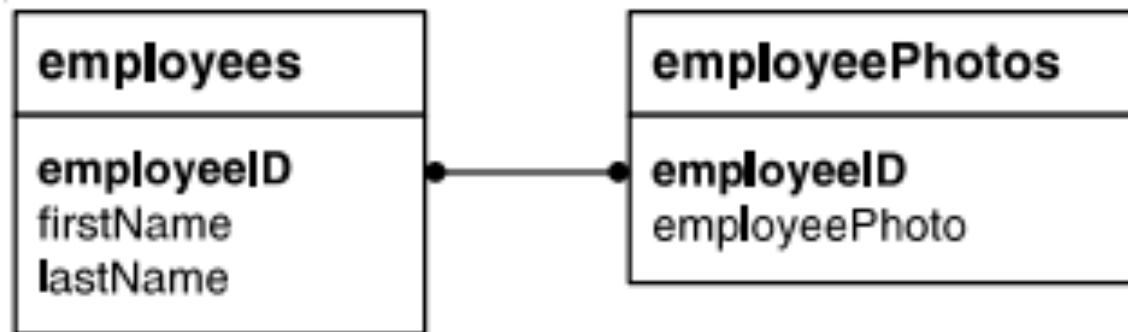


How to identify the primary and foreign keys (cont.)

- Two tables with a many-to-many relationship



- Two tables with a one-to-one relationship



How to enforce the relationships between tables

- Referential integrity means that relationship tables are maintained correctly.
- A table with a foreign key doesn't have rows with foreign key values that don't have matching primary key values in the related table.

How to enforce the relationships between tables (cont.)

- Operations that can violate referential integrity

This operation...	Violates referential integrity if...
Delete a row from the primary key table	The foreign key table contains one or more rows related to the deleted row
Insert a row in the foreign key table	The foreign key value doesn't have a matching primary key value in the related table
Update the value of a foreign key	The new foreign key value doesn't have a matching primary key value in the related table
Update the value of a primary key	The foreign key table contains one or more rows related to the row that's changed

How normalization works

- Normalization is a formal process that you can use to separate the data in a data structure into related tables
- Normalization reduces data redundancy, which can cause storage and maintenance problems.
- To normalize a data structure, you apply the normal forms in sequence.

How normalization works (cont.)

- A table that contains repeating columns

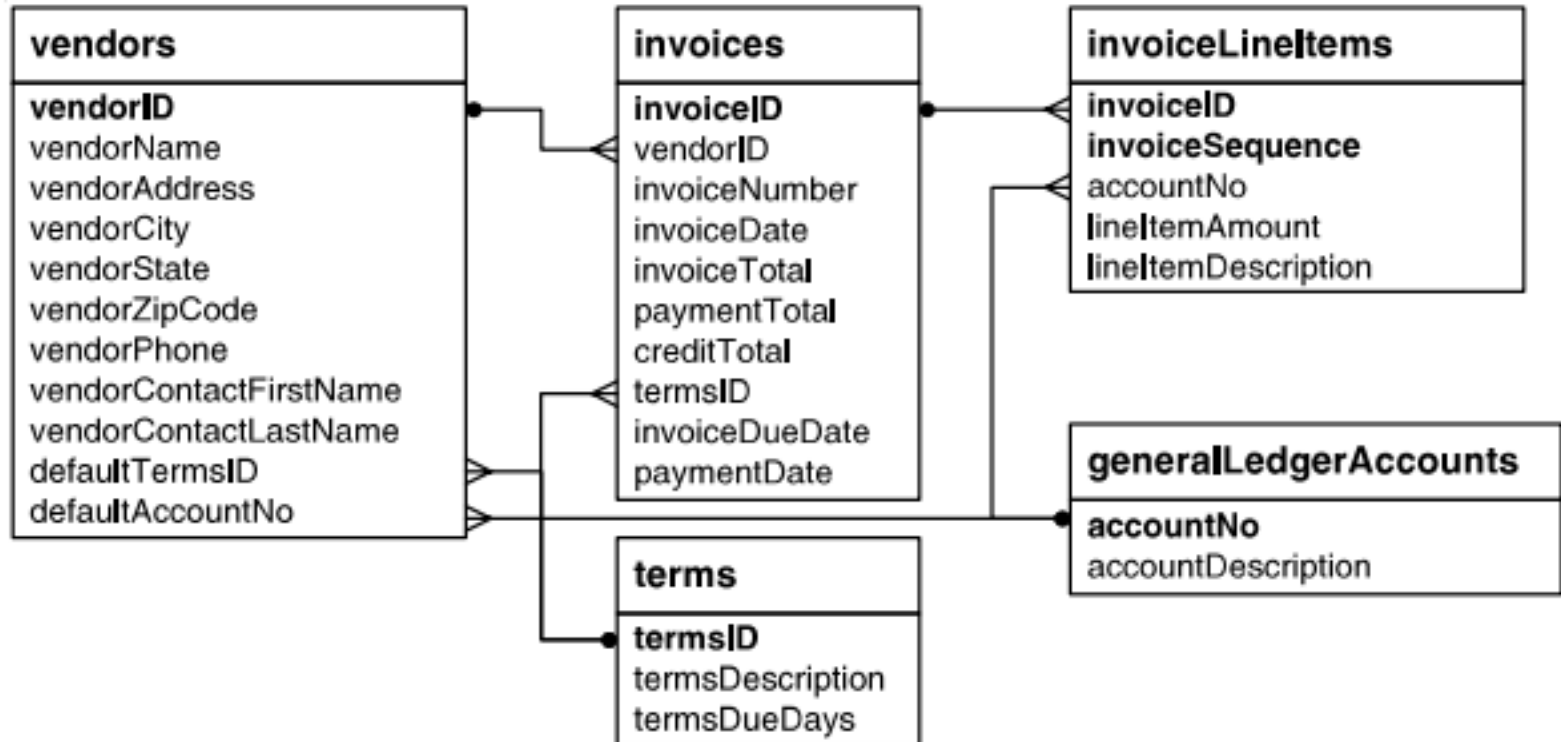
vendorName	invoiceNumber	itemDescription_1	itemDescription_2	itemDescription_3
Cahners Publishing	112897	VB ad	SQL ad	Library directory
Zylka design	97/552	Catalogs	SQL flyer	NULL
Zylka design	97/553B	Card revision	NULL	NULL

- A table that contains redundant data

vendorName	invoiceNumber	itemDescription
Cahners Publishing	112897	VB ad
Cahners Publishing	112897	SQL ad
Cahners Publishing	97/533B	Card revisions
Zylka design	112897	Library directory
Zylka design	97/522	Catalogs
Zylka design	97/522	SQL flyer

How normalization works (cont.)

- The accounts payable system in third normal form



How to identify the columns to be indexed

- MySQL automatically creates an index for a primary key
- AN index provides a way for a database management system to locate information more quickly.
- Indexes speed performance when searching and joining tables.

How to identify the columns to be indexed (cont.)

- When to create an index
 - When the column is a foreign key
 - When the column is used frequently in search conditions or joins
 - When the columns contains a large number of distinct values
 - When the column is updated infrequently.

How to normalize a data structure

The seven normal forms

- The benefits of normalization
 - Since a normalized database has more tables than an unnormalized database, and since each table has an index on its primary key, the database has more indexes. That makes data retrieval more efficient
 - Since each table contains information about a single entity, each index has fewer columns (usually one) and fewer rows. That makes data retrieval and insert, update, and delete operations more efficient.
 - Each table has fewer indexes, which makes insert, update, and delete operations more efficient
 - Data redundancy is minimized, which simplifies maintenance and reduces storage.

The seven normal forms (cont.)

- The seven normal forms

Normal form	Description
First (1NF)	The value stored at the intersection of each row and column must be a scalar value, and a table must not contain any repeating columns.
Second (2NF)	Every non-key column must depend on the entire primary key.
Third (3NF)	Every non-key column must depend only on the primary key.
Boyce-Codd (BCNF)	A non-key column can't be dependent on another non-key column. This prevents <i>transitive dependencies</i> , where column A depends on column C and column B depends on column C. Since both A and B depend on C, A and B should be moved into another table with C as the key.
Fourth (4NF)	A table must not have more than one <i>multivalued dependency</i> , where the primary key has a one-to-many relationship to non-key columns. This form gets rid of misleading many-to-many relationships.
Fifth (5NF)	The data structure is split into smaller and smaller tables until all redundancy has been eliminated. If further splitting would result in tables that couldn't be joined to recreate the original table, the structure is in fifth normal form.
Domain-key (DKNF) or (6NF)	Every constraint on the relationship is dependent only on key constraints and domain constraints, where a <i>domain</i> is the set of allowable values Sixth for a column. This form prevents the insertion of any unacceptable data by enforcing constraints at the level of a relationship, rather than at the table or column level. DKNF is less a design model than an abstract "ultimate" normal form.

How to apply the first normal form

- For a table to be in first normal form , its columns must not contain repeating values.
- A table in first normal form often has repeating values in its rows. This can be resolved by applying the second normal form.
- The invoice data with a column that contains repeating values

vendorName	invoiceNumber	itemDescription
Cahners Publishing	112897	VB ad, SQL ad, Library directory
Zylka design	97/522	Catalogs, SQL Flyer
Zylka design	97/533B	Card revision

How to apply the first normal form (cont.)

- The invoice data with repeating columns

vendorName	invoiceNumber	itemDescription_1	itemDescription_2	itemDescription_3
Cahners Publishing	112897	VB ad	SQL ad	Library directory
Zylka design	97/552	Catalogs	SQL flyer	NULL
Zylka design	97/553B	Card revision	NULL	NULL

- The invoice data in first normal form

vendorName	invoiceNumber	itemDescription
Cahners Publishing	112897	VB ad
Cahners Publishing	112897	SQL ad
Cahners Publishing	97/533B	Card revisions
Zylka design	112897	Library directory
Zylka design	97/522	Catalogs
Zylka design	97/522	SQL flyer

How to apply the second normal form

- For a table to be in second normal form, every non-key must depend on the entire primary key.
- To apply second normal form, you move columns that don't depend on the entire primary key to another table and then establish a relationship between the two tables.
- Second normal form helps remove redundant row data. Which can save storage space, make maintenance easier, and reduce the chance of storing inconsistent data.

How to apply the second normal form (cont.)

- The invoice data in first normal form with keys added

invoiceID	vendorName	invoiceNumber	invoiceSequence	itemDescription
1	Cahners Publishing	112897	1	VB ad
1	Cahners Publishing	112897	2	SQL ad
1	Cahners Publishing	112897	3	Library directory
2	Zylka design	97/522	1	Catalogs
2	Zylka design	97/522	2	SQL flyer
3	Zylka design	97/533B	1	Card revision

How to apply the second normal form (cont.)

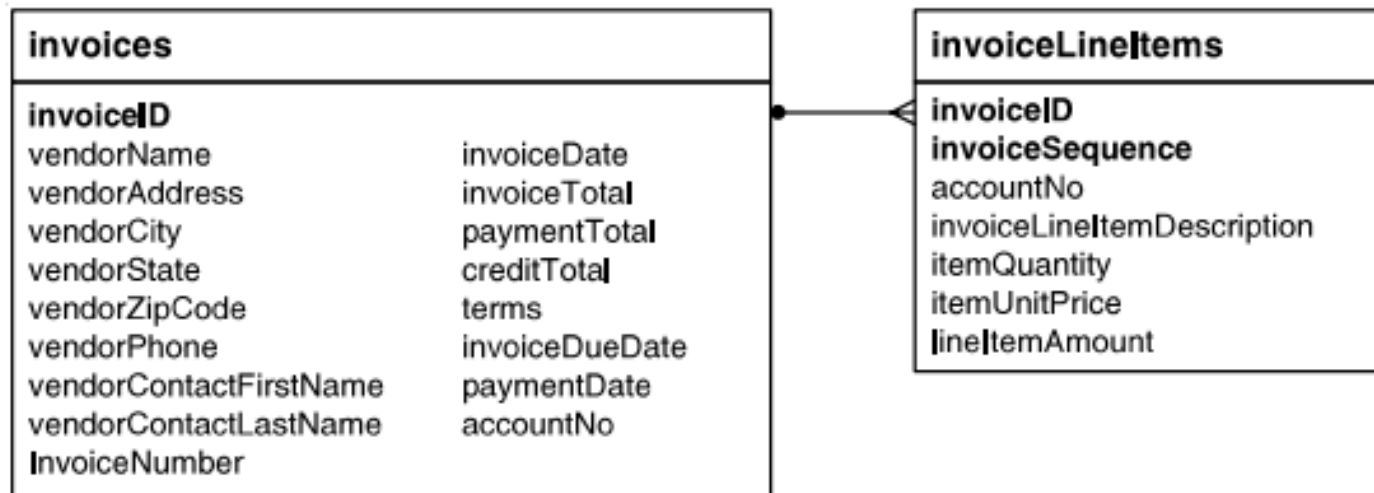
- The invoice data in second normal form

invoiceNumber	vendorName	invoiceID
11287	Cahners Publishing	1
97/522	Zylka design	2
97/533B	Zylka design	3

invoiceID	invoiceSequence	itemDescription
1	1	VB ad
1	2	SQL ad
1	3	Library directory
2	1	Catalogs
2	2	SQL flyer
3	1	Card revision

How to apply the third normal form

- For a table to be in third normal form, every non-key columns must depend only on the primary key.
- The accounts payable system in second normal form

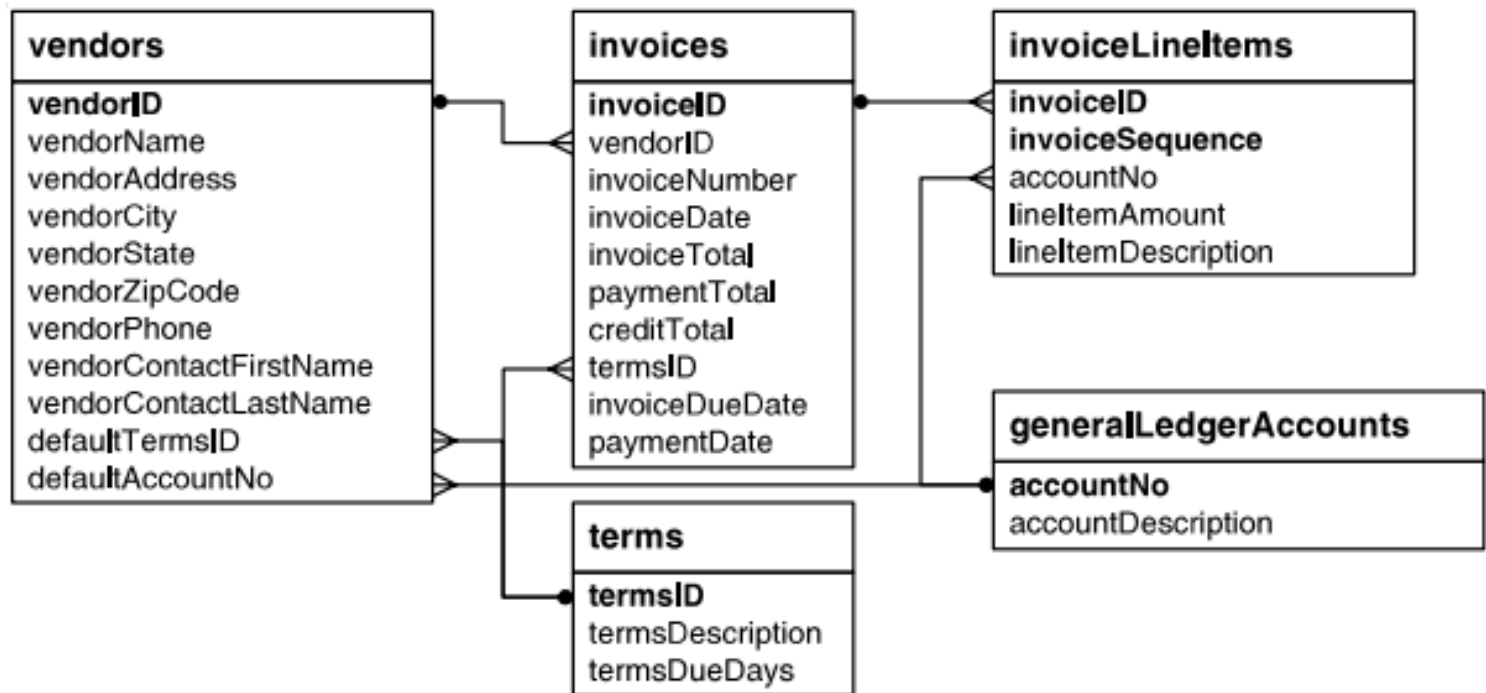


How to apply the third normal form (cont.)

- Question about the structure
 1. Does the vendor information (vendor_name, vendor_address, etc) depend only on the invoice_id column?
 2. Does the terms column depend only on the invoice_id column?
 3. Does the account_no column depend only on the invoice_id column?
 4. Can the invoice_due_date and line_item_amount columns be derived from other data?

How to apply the third normal form (cont.)

- The accounts payable system in third normal form

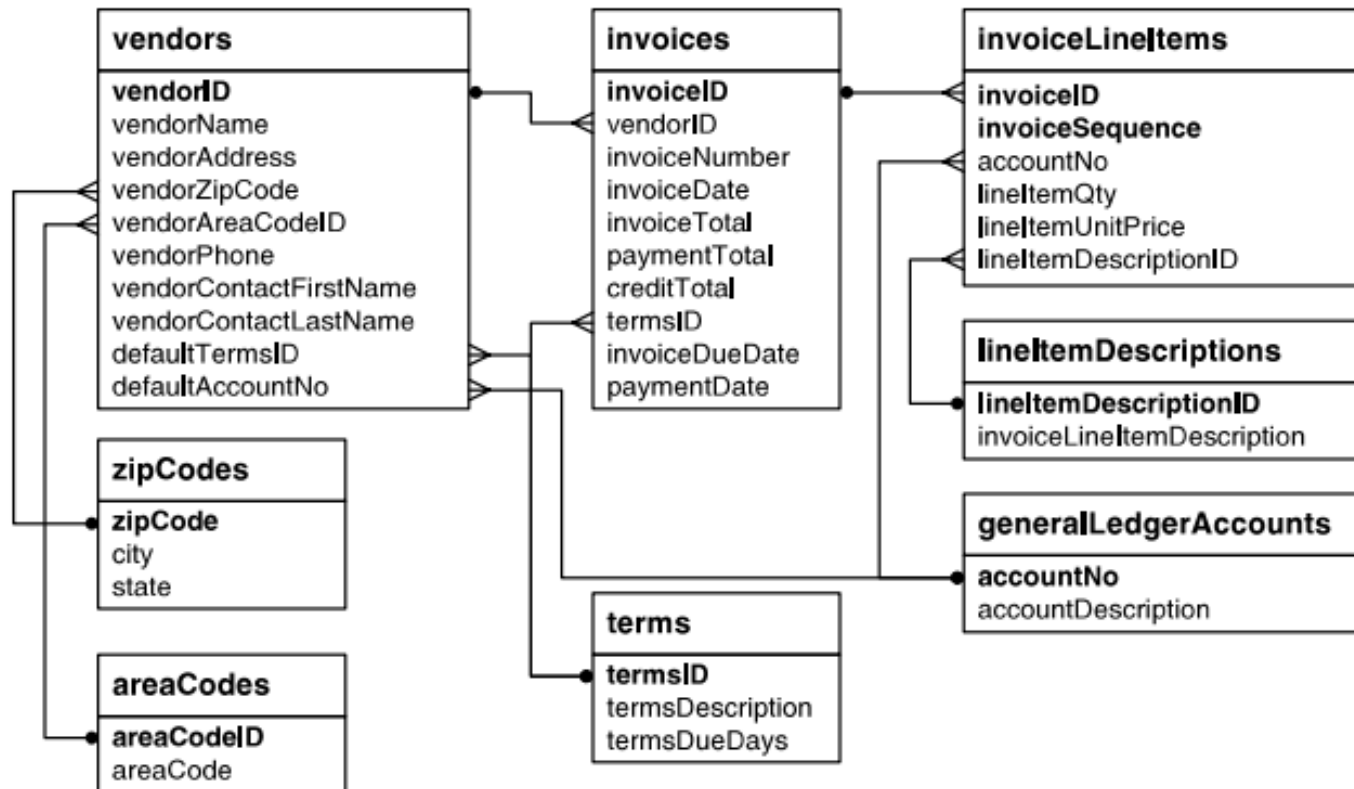


When and how to denormalize a data structure

- Data structures that are normalized to the fourth normal form and beyond typically require more joins than tables normalized to the third form and can therefore be less efficient.
- Denormalization can result in larger tables, redundant data, and reduced performance.

When and how to denormalize a data structure (cont.)

- The accounts payable system in fifth normal form



When and how to denormalize a data structure (cont.)

- When to denormalize
 - When a column from a joined table is used repeatedly in search criteria, you should consider moving that column to the primary key table if it will eliminate the need for a join
 - If a table is updated infrequently, you should consider denormalizing it to improve efficiency.
 - Include columns with derived values when those values are used frequently in search conditions.

A database design tool

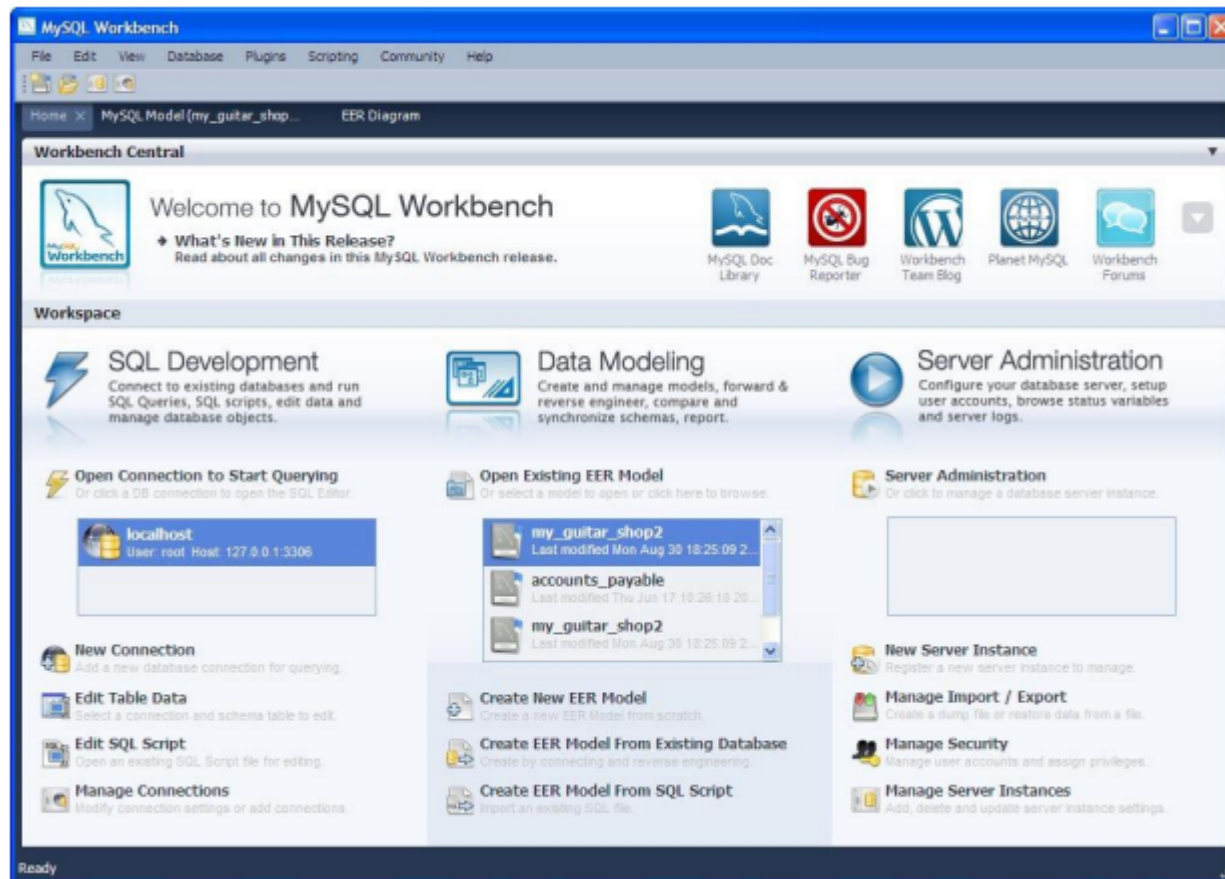
An introduction to MySQL Workbench

- MySQL Workbench is a database design tool that's available for free and runs on most modern operating systems.
- The MySQL Workbench web site

<http://wb.mysql.com/>

An introduction to MySQL Workbench (cont.)

- The Home Page



An introduction to MySQL Workbench (cont.)

- MySQL Workbench
 - Lets you create and edit diagrams
 - Let you define the table, columns, and indexes for a database
 - Lets you define the relationships between the tables in a database
 - Lets you generate a diagram from a SQL creation script
 - Lets you generate a SQL creation script from diagram

An introduction to MySQL Workbench (cont.)

- How to install MySQL Workbench
 1. Go to the MySQL Workbench web site shown above
 2. Download the version for your system
 3. Run the installer or setup file and respond to the prompts.

Summary

- Database are often designed and maintained by database administrators (DBAs)
- A table in a relational database typically represents an object, or entity, in the real world.
- To model database and the relationships between its tables after real-world system. You can use entity-relationship (ER) modeling.
- You can use a CASE (Computer-Aided Software Engineering) tool to help you design a database.

Summary (2)

- Add foreign key column : one-to-many relationship, many – to –many relationship, one-to-one relationship
- Normalization is a formal process you can use to separate the data in a data structure into related tables.
- Denormalized data structures to some extent, usually to the third normal form.

Discussion