Chapter 10

How to design a database

Objectives

Applied

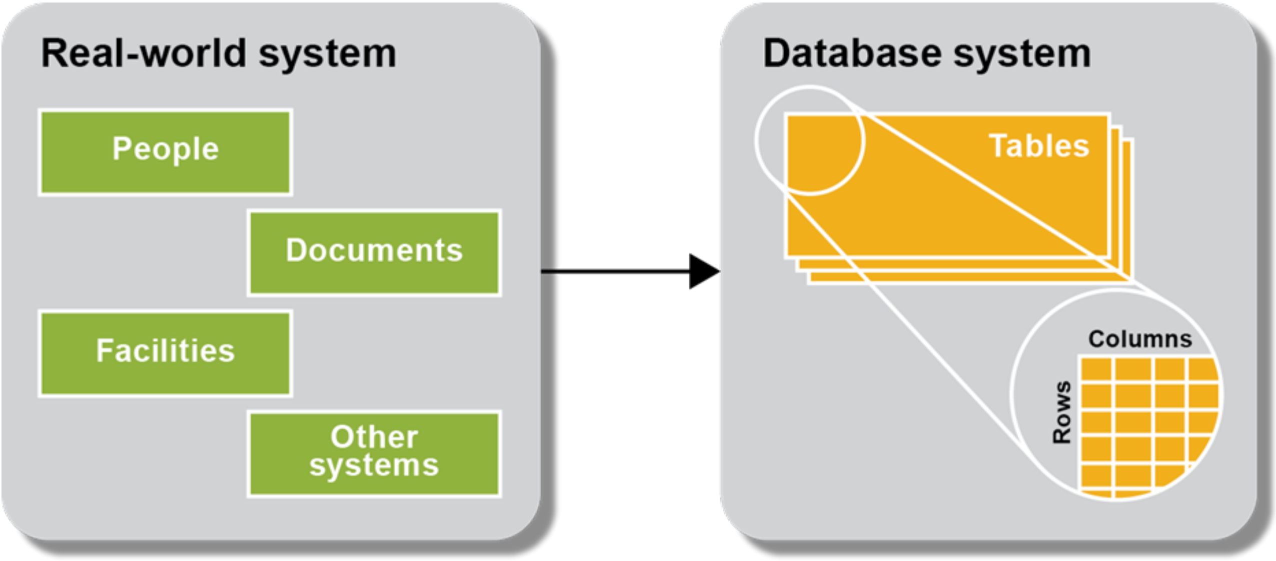
1. Given the specifications for a database, identify the tables, columns, keys, relationships, and indexes for the database.
2. Given the tables for an unnormalized database, normalize the structure to the third normal form.
3. Use MySQL Workbench to create or work with an EER model for a database and any EER diagrams that are associated with that model.

Objectives (continued)

Knowledge

1. Give three criteria for when a column should be indexed.
2. Describe referential integrity.
3. Explain how MySQL uses declarative referential integrity to prevent deletion, insertion, and update problems.
4. Explain how normalizing a database to the third normal form affects database performance.

A database system is modeled after a real-world system



The six basic steps for designing a data structure

Step 1: Identify the data elements

Step 2: Subdivide each element into its smallest useful components

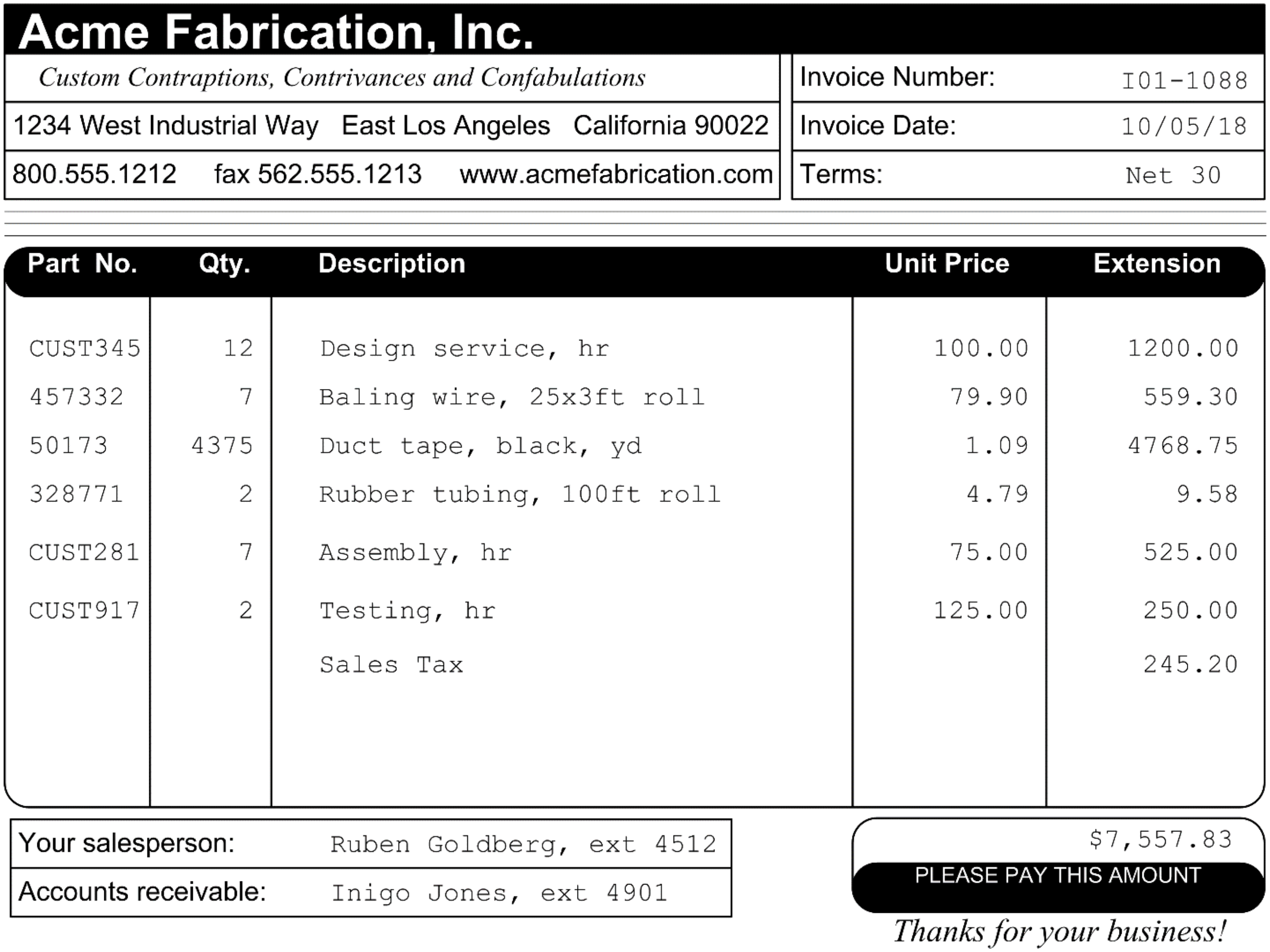
Step 3: Identify the tables and assign columns

Step 4: Identify the primary and foreign keys

Step 5: Review whether the data structure is normalized

Step 6: Identify the indexes

An invoice that's used to identify data elements



Acme Fabrication, Inc.

Custom Contraptions, Contrivances and Confabulations

1234 West Industrial Way East Los Angeles California 90022
800.555.1212 fax 562.555.1213 www.acmefabrication.com

Invoice Number:	I01-1088
Invoice Date:	10/05/18
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
Accounts receivable:	Inigo Jones, ext 4901

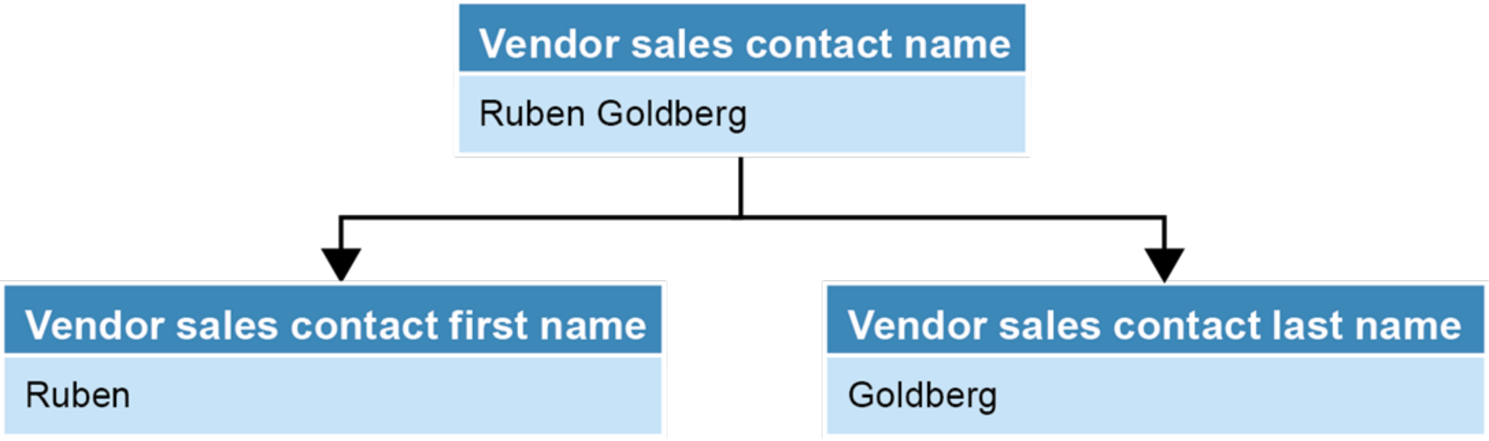
$7,557.83
PLEASE PAY THIS AMOUNT

Thanks for your business!

The data elements on the invoice document

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

A name that’s divided into first and last names

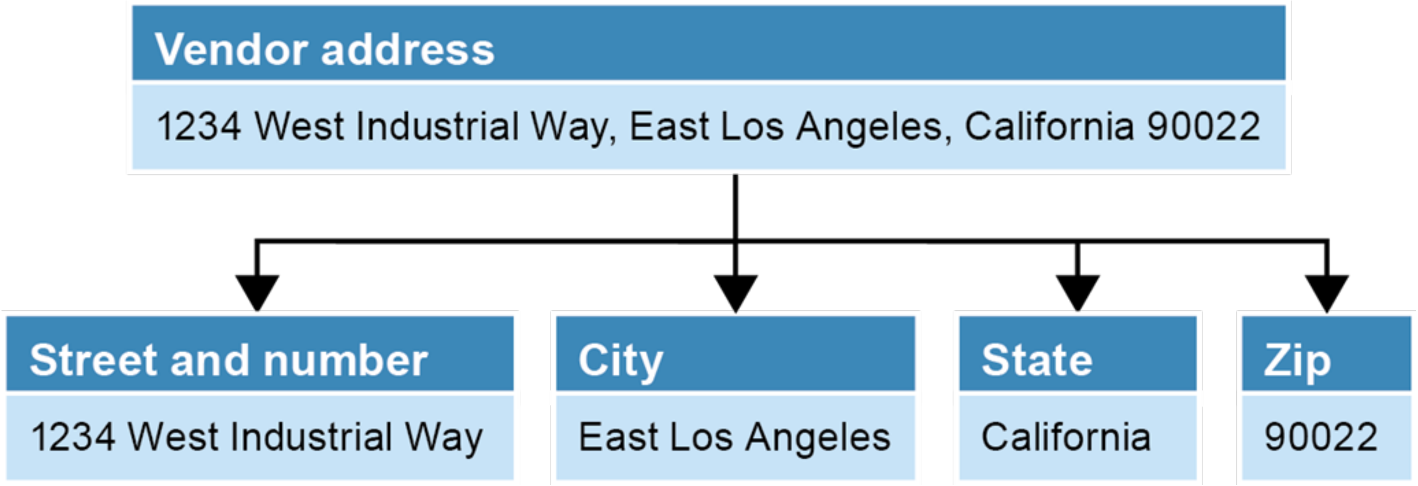


Vendor sales contact name
Ruben Goldberg

Vendor sales contact first name
Ruben

Vendor sales contact last name
Goldberg

An address that's divided into its components



Vendor address

1234 West Industrial Way, East Los Angeles, California 90022

Street and number	City	State	Zip
1234 West Industrial Way	East Los Angeles	California	90022

Possible tables and columns for an A/P system

Vendors

Vendor name

Vendor address

Vendor city

Vendor state

Vendor zip code

Vendor phone number

~~Vendor fax number~~

~~Vendor web address~~

Vendor contact first name

Vendor contact last name

~~Vendor contact phone~~

~~Vendor AR first name~~

Invoices

Invoice number*

Invoice date

Terms*

Invoice total

Payment date

Payment total

Invoice due date

Credit total

*Account number**

Possible tables and columns for an A/P system (continued)

Vendors

~~Vendor AR last name~~

~~Vendor AR phone~~

*Terms**

*Account number**

Invoice line items

Invoice number*

~~Item part number~~

Item quantity

Item description

Item unit price

Item extension

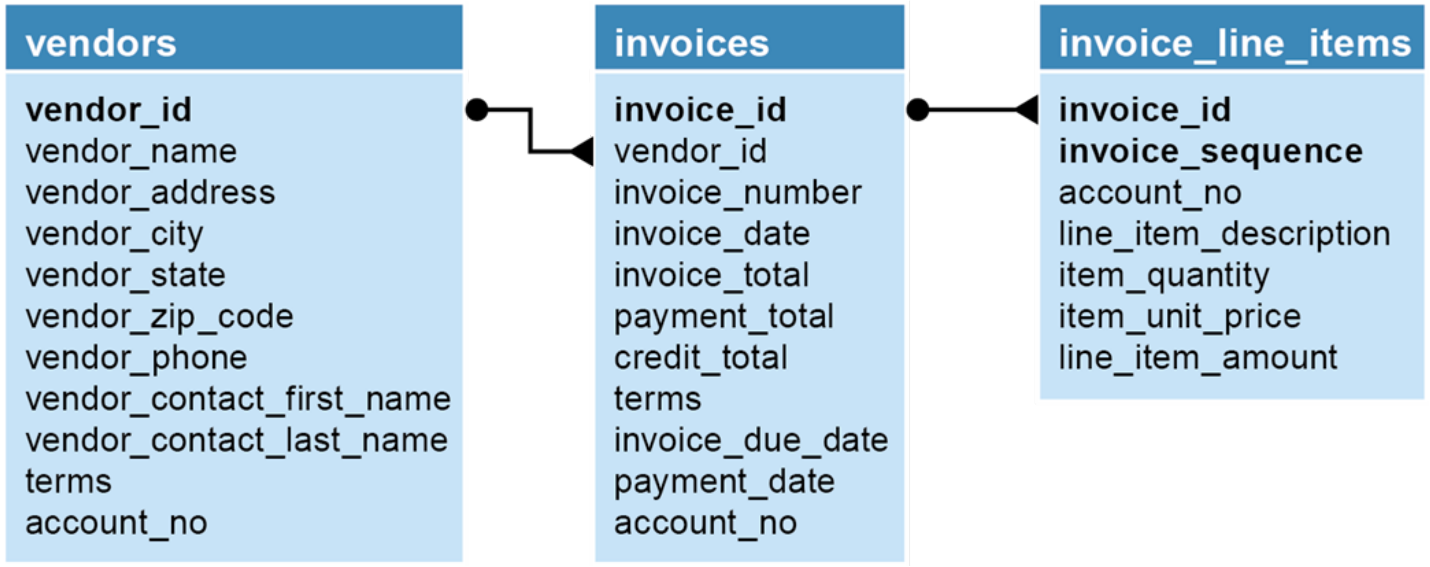
*Account number**

Sequence number

The notation for identifying data elements

- Data elements that were previously identified but aren’t needed are crossed out.
- Data elements that were added are displayed in italics.
- Data elements that are related to two or more entities are followed by an asterisk.

The relationships between the tables



vendors

vendor_id
vendor_name
vendor_address
vendor_city
vendor_state
vendor_zip_code
vendor_phone
vendor_contact_first_name
vendor_contact_last_name
terms
account_no

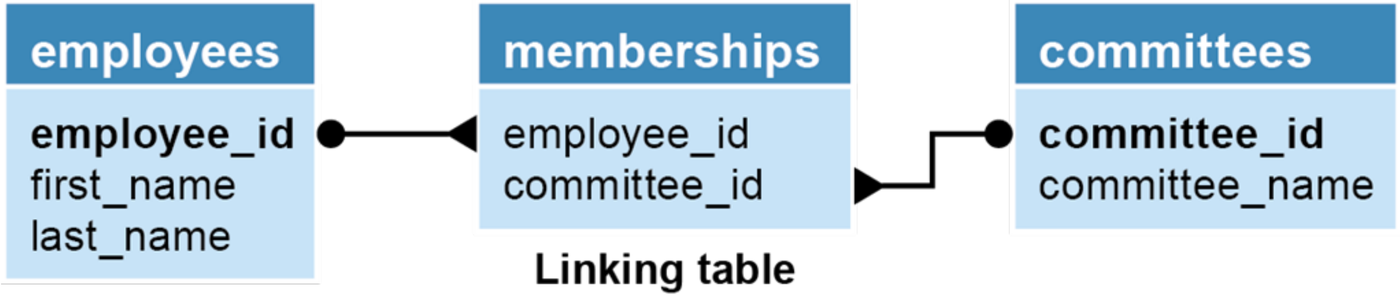
invoices

invoice_id
vendor_id
invoice_number
invoice_date
invoice_total
payment_total
credit_total
terms
invoice_due_date
payment_date
account_no

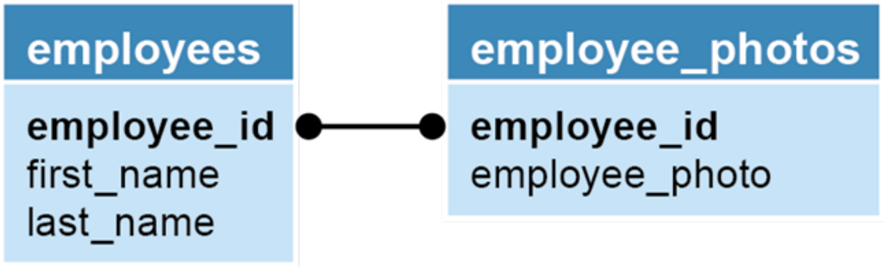
invoice_line_items

invoice_id
invoice_sequence
account_no
line_item_description
item_quantity
item_unit_price
line_item_amount

Two tables with a many-to-many relationship



Two tables with a one-to-one relationship



employees

employee_id
first_name
last_name

employee_photos

employee_id
employee_photo

Operations that can violate referential integrity

Deleting a row from the primary key table

If the foreign key table contains one or more rows related to the deleted row

Inserting a row in the foreign key table

If the foreign key value doesn’t have a matching primary key value in the related table

Updating the value of a foreign key

If the new foreign key value doesn’t have a matching primary key value in the related table

Updating the value of a primary key

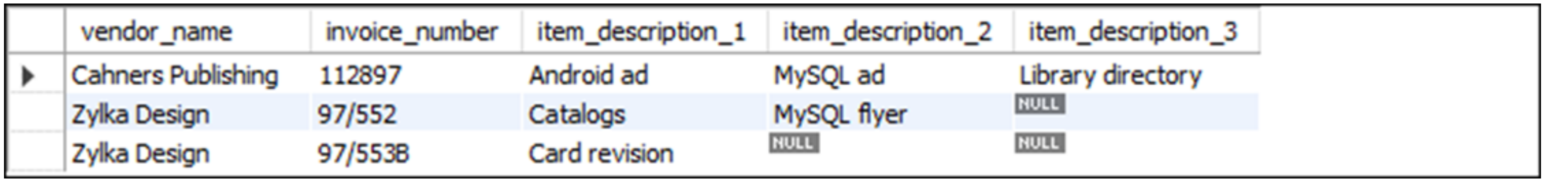
If the foreign key table contains one or more rows related to the row that’s changed

Terms to know about data structures

- Entity
- Attribute
- Instance
- Entity-relationship (ER) modeling
- Referential integrity
- Declarative referential integrity (DRI)
- Foreign key constraint
- Orphaned row

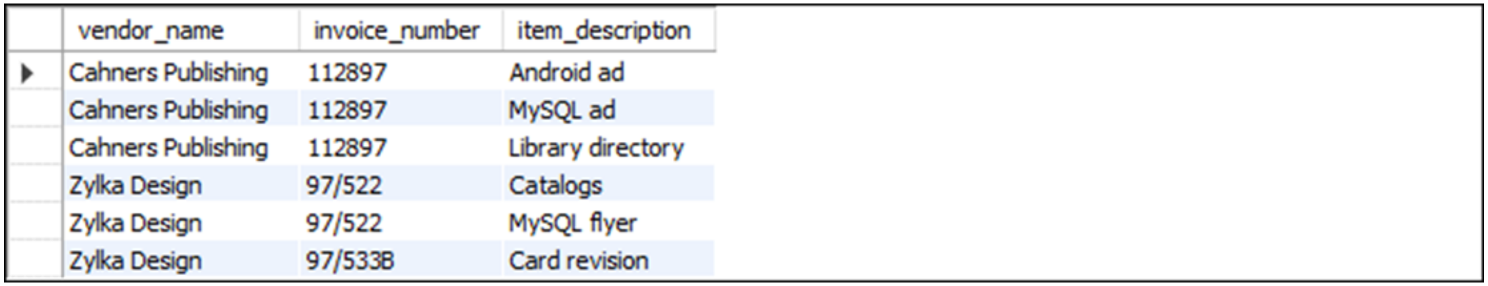
Two tables that need to be normalized

A table that contains repeating columns



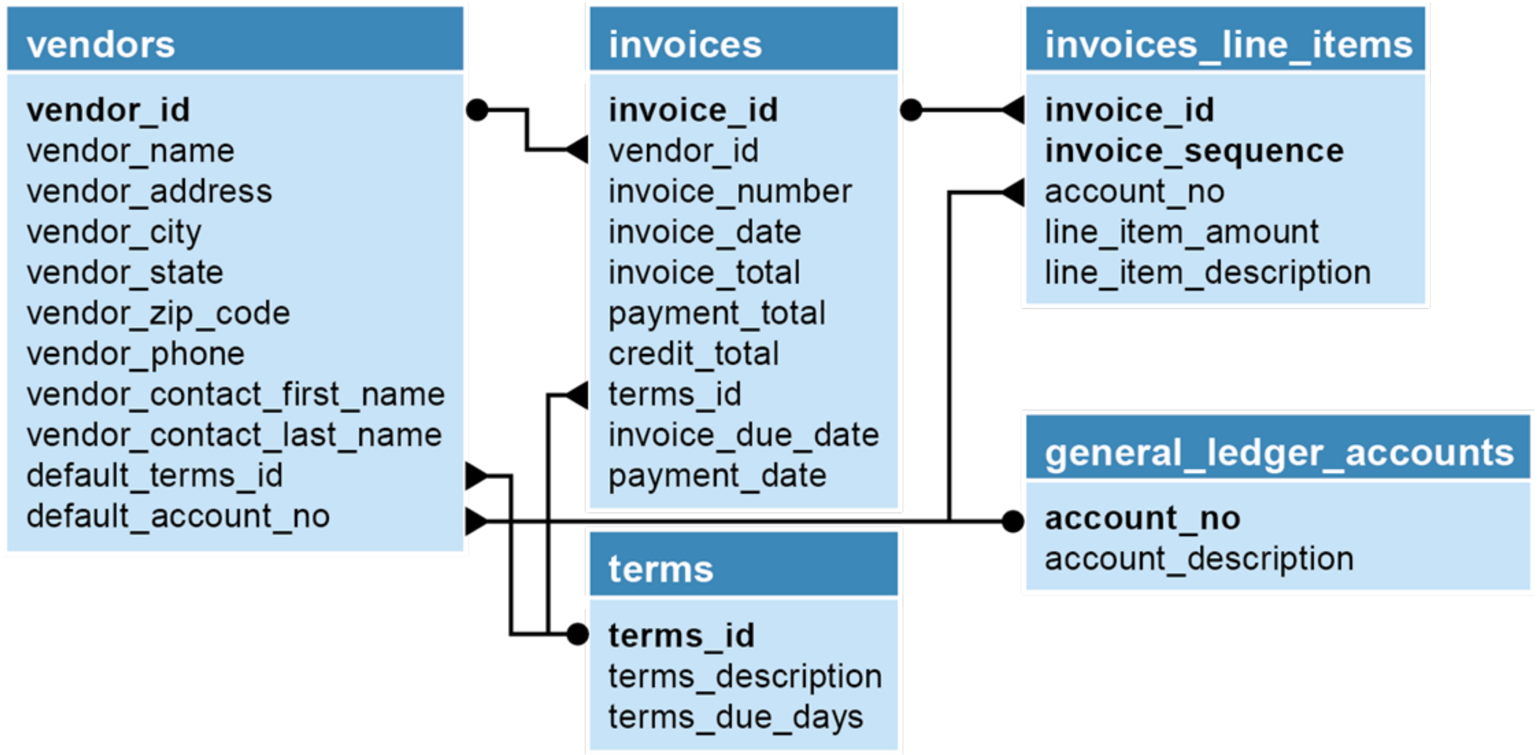
	vendor_name	invoice_number	item_description_1	item_description_2	item_description_3
▶	Cahners Publishing	112897	Android ad	MySQL ad	Library directory
	Zylka Design	97/552	Catalogs	MySQL flyer	NULL
	Zylka Design	97/553B	Card revision	NULL	NULL

A table that contains redundant data



	vendor_name	invoice_number	item_description
▶	Cahners Publishing	112897	Android ad
	Cahners Publishing	112897	MySQL ad
	Cahners Publishing	112897	Library directory
	Zylka Design	97/522	Catalogs
	Zylka Design	97/522	MySQL flyer
	Zylka Design	97/533B	Card revision

The accounts payable system in third normal form



vendors

vendor_id
vendor_name
vendor_address
vendor_city
vendor_state
vendor_zip_code
vendor_phone
vendor_contact_first_name
vendor_contact_last_name
default_terms_id
default_account_no

invoices

invoice_id
vendor_id
invoice_number
invoice_date
invoice_total
payment_total
credit_total
terms_id
invoice_due_date
payment_date

invoices_line_items

invoice_id
invoice_sequence
account_no
line_item_amount
line_item_description

general_ledger_accounts

account_no
account_description

terms

terms_id
terms_description
terms_due_days

Terms to know about normalizing a data structure

- Normalization
- Data redundancy
- Unnormalized data structure
- Normalized data structure
- Normal form

When to create an index

- When the column is used frequently in search conditions or joins
- When the column contains a large number of distinct values
- When the column is updated infrequently

The first three 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.

Note

- Most designers stop at the third normal form.

The next four normal forms

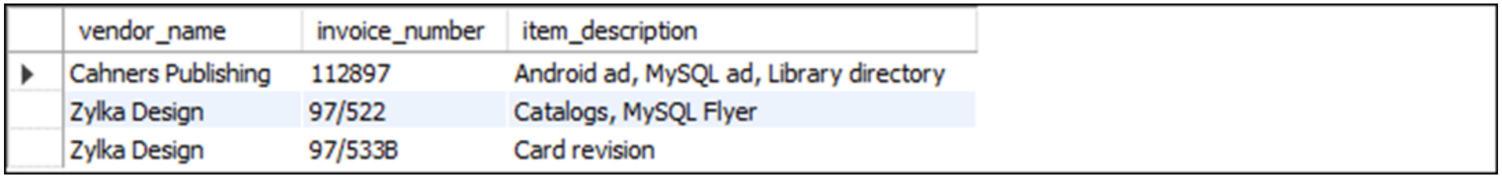
Normal form	Description
Boyce-Codd (BCNF)	A non-key column can't be dependent on another non-key column.
Fourth (4NF)	A table must not have more than one *multivalued dependency*, where the primary key has a one-to-many relationship to non-key columns.
Fifth (5NF)	The data structure is split into smaller and smaller tables until all redundancy has been eliminated.
Domain-key (DKNF) or Sixth (6NF)	Every constraint on the relationship is dependent only on key constraints and domain constraints, where a *domain* is the set of allowable values for a column.

The benefits of normalization

- More tables, and each table has an index on its primary key. That makes data retrieval more efficient.
- Each table contains information about a single entity. 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.

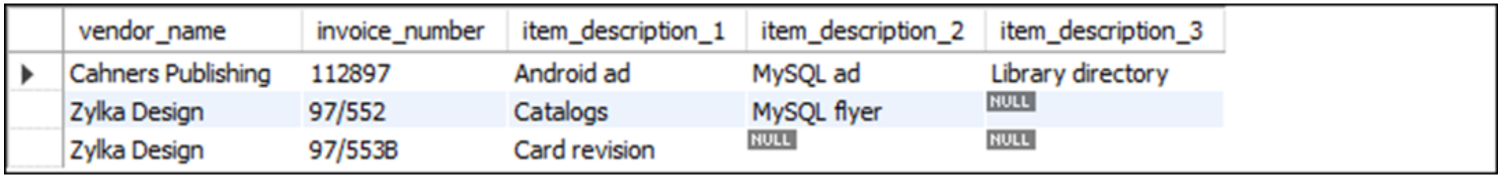
Unnormalized invoice data

The invoice data with a column that contains repeating values



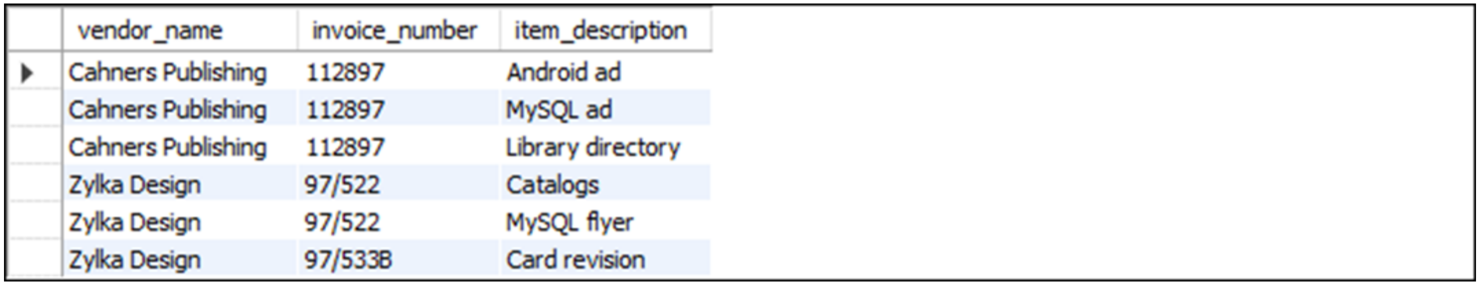
	vendor_name	invoice_number	item_description
▶	Cahners Publishing	112897	Android ad, MySQL ad, Library directory
	Zylka Design	97/522	Catalogs, MySQL Flyer
	Zylka Design	97/533B	Card revision

The invoice data with repeating columns



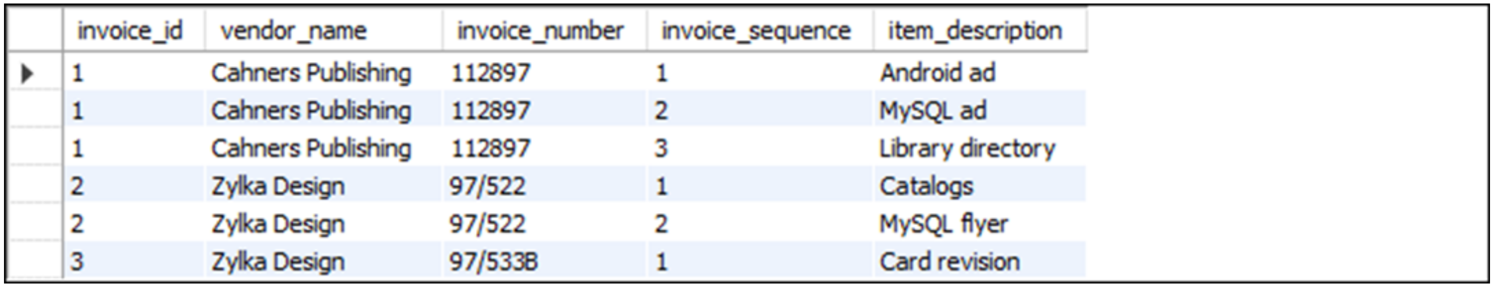
	vendor_name	invoice_number	item_description_1	item_description_2	item_description_3
▶	Cahners Publishing	112897	Android ad	MySQL ad	Library directory
	Zylka Design	97/552	Catalogs	MySQL flyer	NULL
	Zylka Design	97/553B	Card revision	NULL	NULL

The invoice data in first normal form



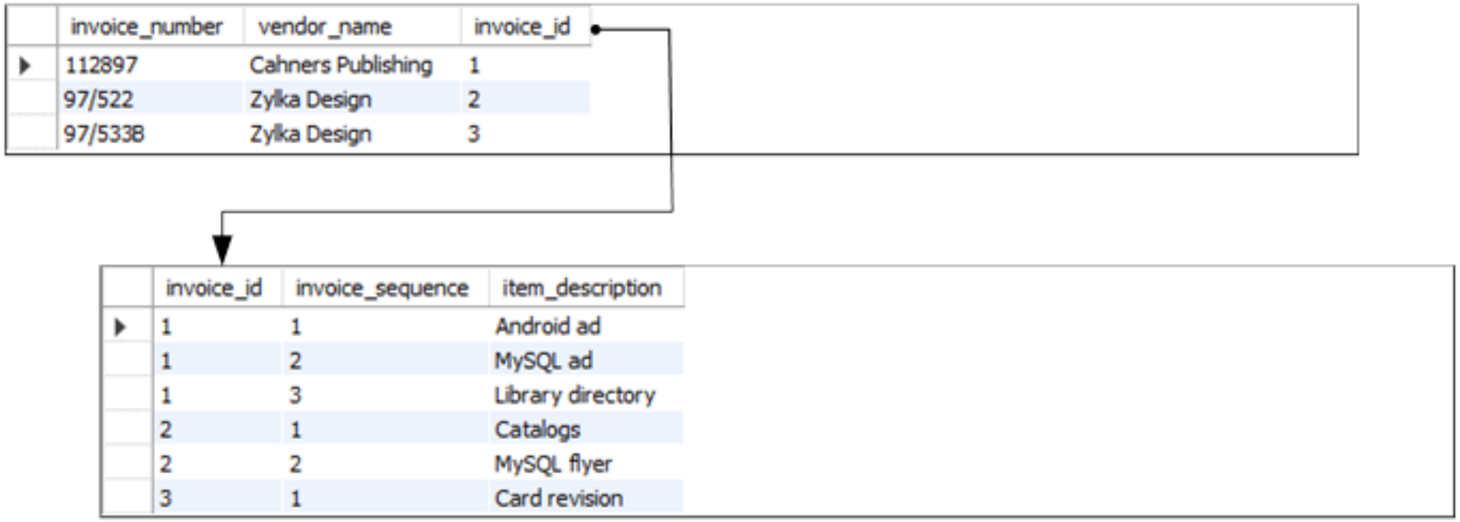
	vendor_name	invoice_number	item_description
▶	Cahners Publishing	112897	Android ad
	Cahners Publishing	112897	MySQL ad
	Cahners Publishing	112897	Library directory
	Zylka Design	97/522	Catalogs
	Zylka Design	97/522	MySQL flyer
	Zylka Design	97/533B	Card revision

The invoice data in first normal form with keys added



	invoice_id	vendor_name	invoice_number	invoice_sequence	item_description
▶	1	Cahners Publishing	112897	1	Android ad
	1	Cahners Publishing	112897	2	MySQL ad
	1	Cahners Publishing	112897	3	Library directory
	2	Zylka Design	97/522	1	Catalogs
	2	Zylka Design	97/522	2	MySQL flyer
	3	Zylka Design	97/533B	1	Card revision

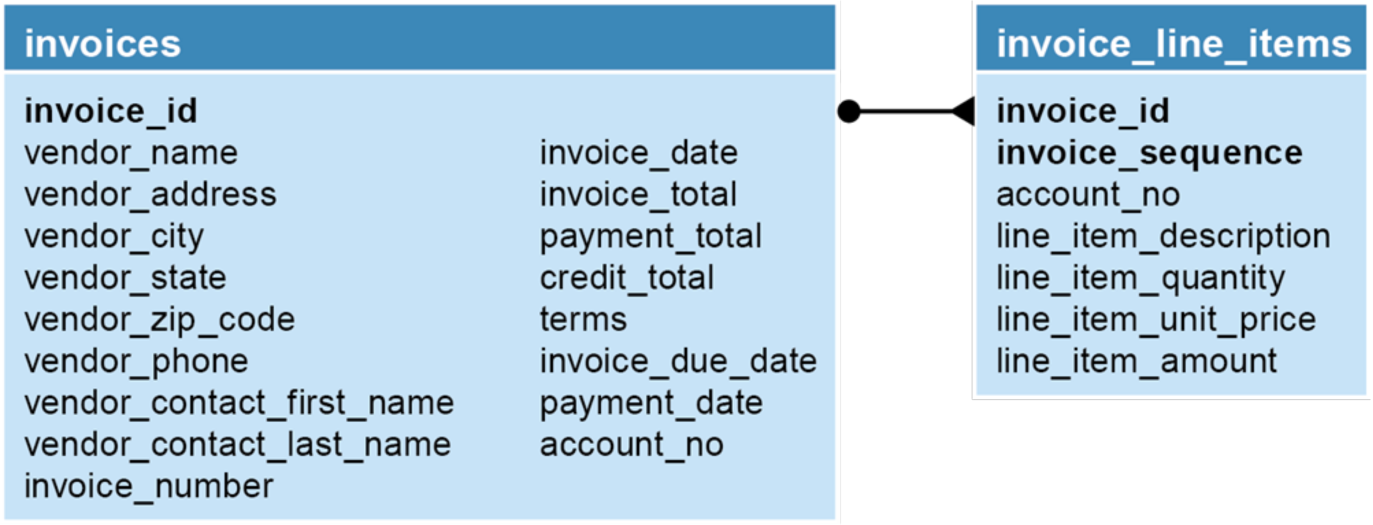
The invoice data in second normal form



invoice_number	vendor_name	invoice_id
112897	Cahners Publishing	1
97/522	Zylka Design	2
97/533B	Zylka Design	3

invoice_id	invoice_sequence	item_description
1	1	Android ad
1	2	MySQL ad
1	3	Library directory
2	1	Catalogs
2	2	MySQL flyer
3	1	Card revision

The A/P system in second normal form



invoices

invoice_id
vendor_name
vendor_address
vendor_city
vendor_state
vendor_zip_code
vendor_phone
vendor_contact_first_name
vendor_contact_last_name
invoice_number
invoice_date
invoice_total
payment_total
credit_total
terms
invoice_due_date
payment_date
account_no

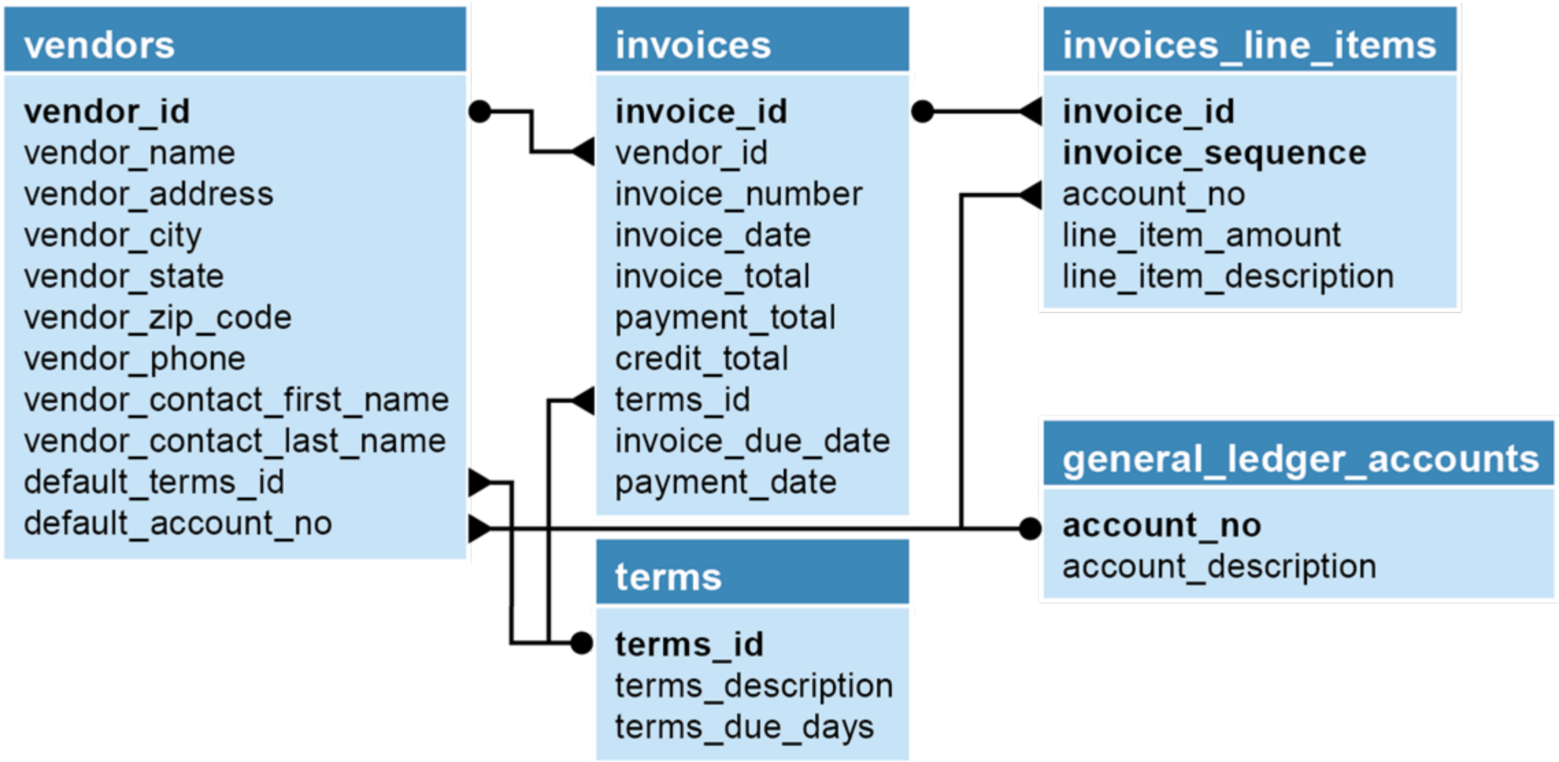
invoice_line_items

invoice_id
invoice_sequence
account_no
line_item_description
line_item_quantity
line_item_unit_price
line_item_amount

Questions 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?

The A/P system in third normal form



vendors

vendor_id
vendor_name
vendor_address
vendor_city
vendor_state
vendor_zip_code
vendor_phone
vendor_contact_first_name
vendor_contact_last_name
default_terms_id
default_account_no

invoices

invoice_id
vendor_id
invoice_number
invoice_date
invoice_total
payment_total
credit_total
terms_id
invoice_due_date
payment_date

invoices_line_items

invoice_id
invoice_sequence
account_no
line_item_amount
line_item_description

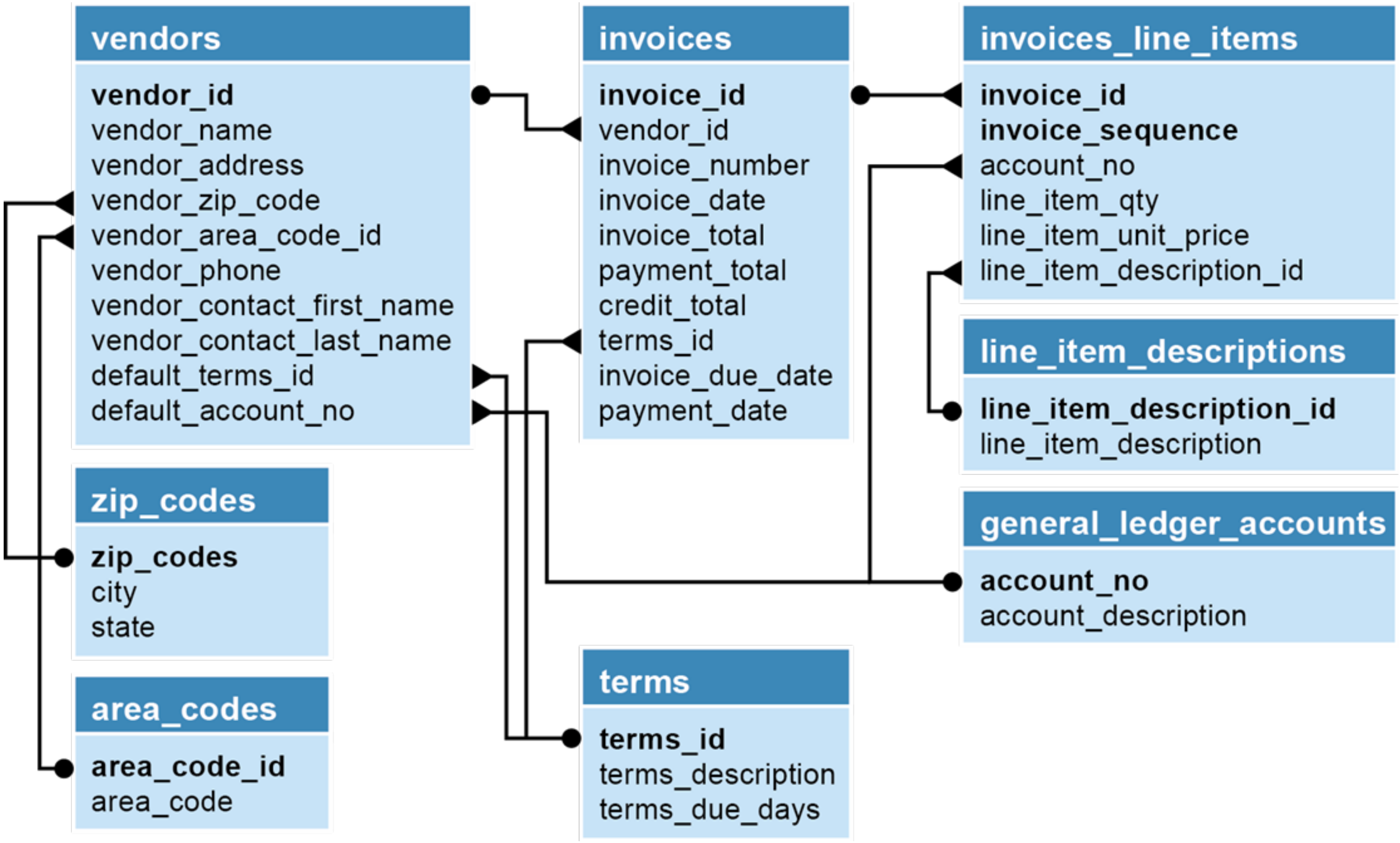
general_ledger_accounts

account_no
account_description

terms

terms_id
terms_description
terms_due_days

The A/P system in fifth normal form



vendors
vendor_id
vendor_name
vendor_address
vendor_zip_code
vendor_area_code_id
vendor_phone
vendor_contact_first_name
vendor_contact_last_name
default_terms_id
default_account_no

invoices
invoice_id
vendor_id
invoice_number
invoice_date
invoice_total
payment_total
credit_total
terms_id
invoice_due_date
payment_date

invoices_line_items
invoice_id
invoice_sequence
account_no
line_item_qty
line_item_unit_price
line_item_description_id

line_item_descriptions
line_item_description_id
line_item_description

general_ledger_accounts
account_no
account_description

zip_codes
zip_codes
city
state

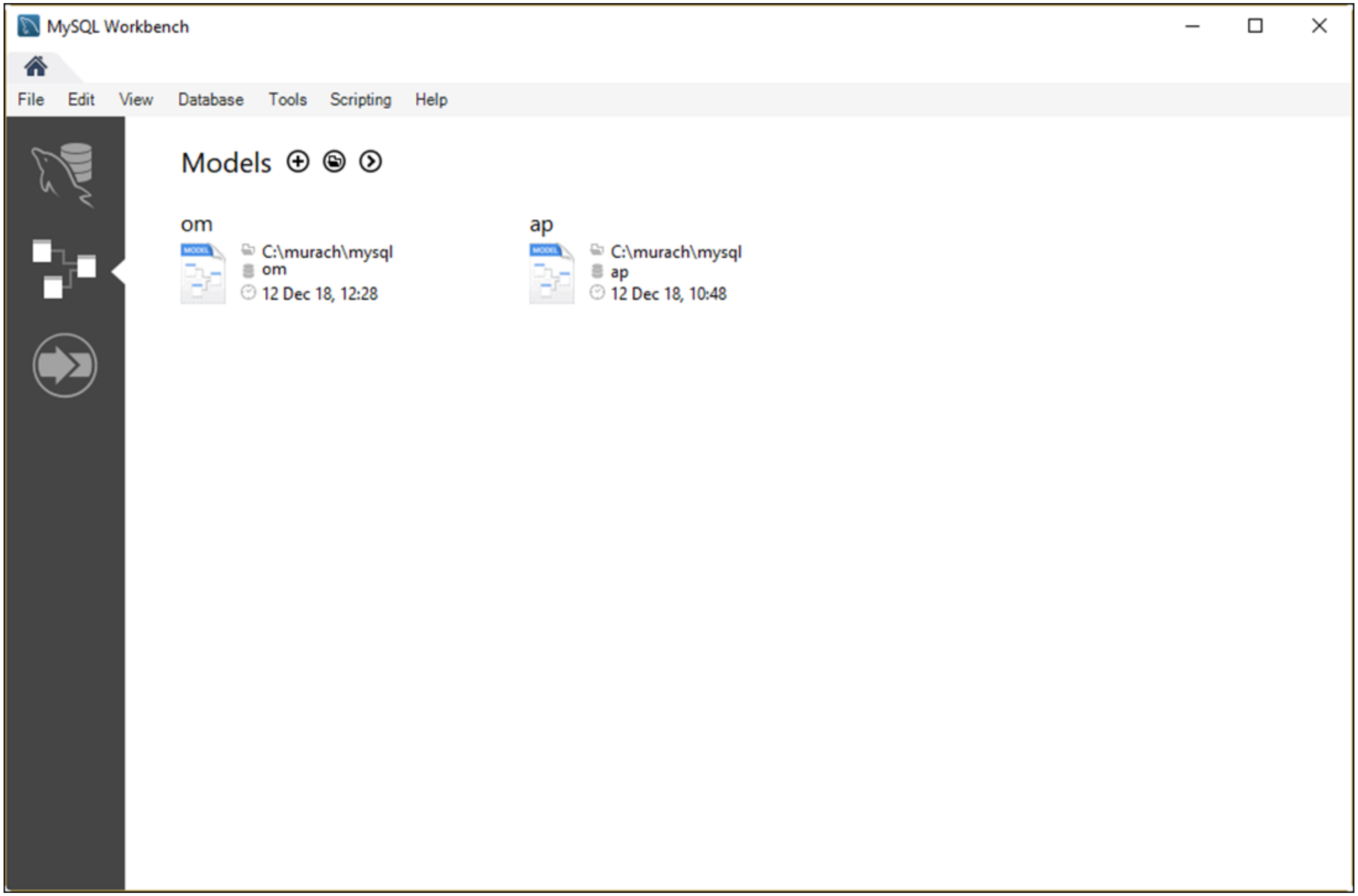
area_codes
area_code_id
area_code

terms
terms_id
terms_description
terms_due_days

When to denormalize

- When a column from a joined table is used repeatedly in search criteria.
- If a table is updated infrequently.
- Include columns with derived values when those values are used frequently in search conditions.

The Models tab of MySQL Workbench Home page



om
C:\murach\mysql
om
12 Dec 18, 12:28

ap
C:\murach\mysql
ap
12 Dec 18, 10:48

Operations you can perform from the Models tab

- Open an existing EER model
- Create a blank EER model
- Create an EER model from an existing database
- Create an EER model from a SQL creation script
- Remove an EER model from the list

The EER model for the AP database

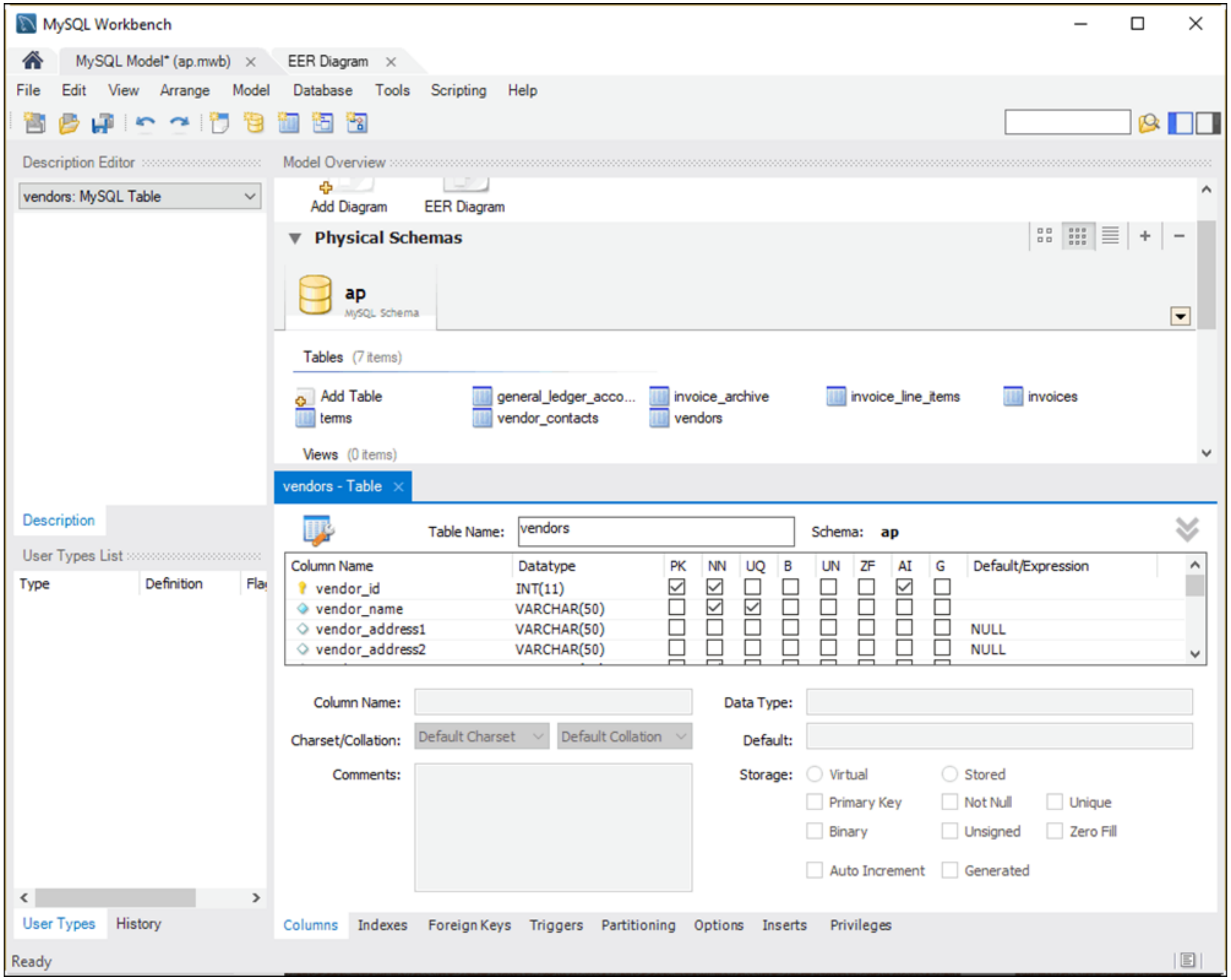


Table Name: vendors

Schema: ap

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
vendor_id	INT(11)	☑	☑	□	□	□	□	☑	□	
vendor_name	VARCHAR(50)	□	☑	☑	□	□	□	□	□	
vendor_address1	VARCHAR(50)	□	□	□	□	□	□	□	□	NULL
vendor_address2	VARCHAR(50)	□	□	□	□	□	□	□	□	NULL

Column Name:

Data Type:

Charset/Collation: Default Charset Default Collation

Default:

Comments:

Storage: ○ Virtual ○ Stored

□ Primary Key □ Not Null □ Unique

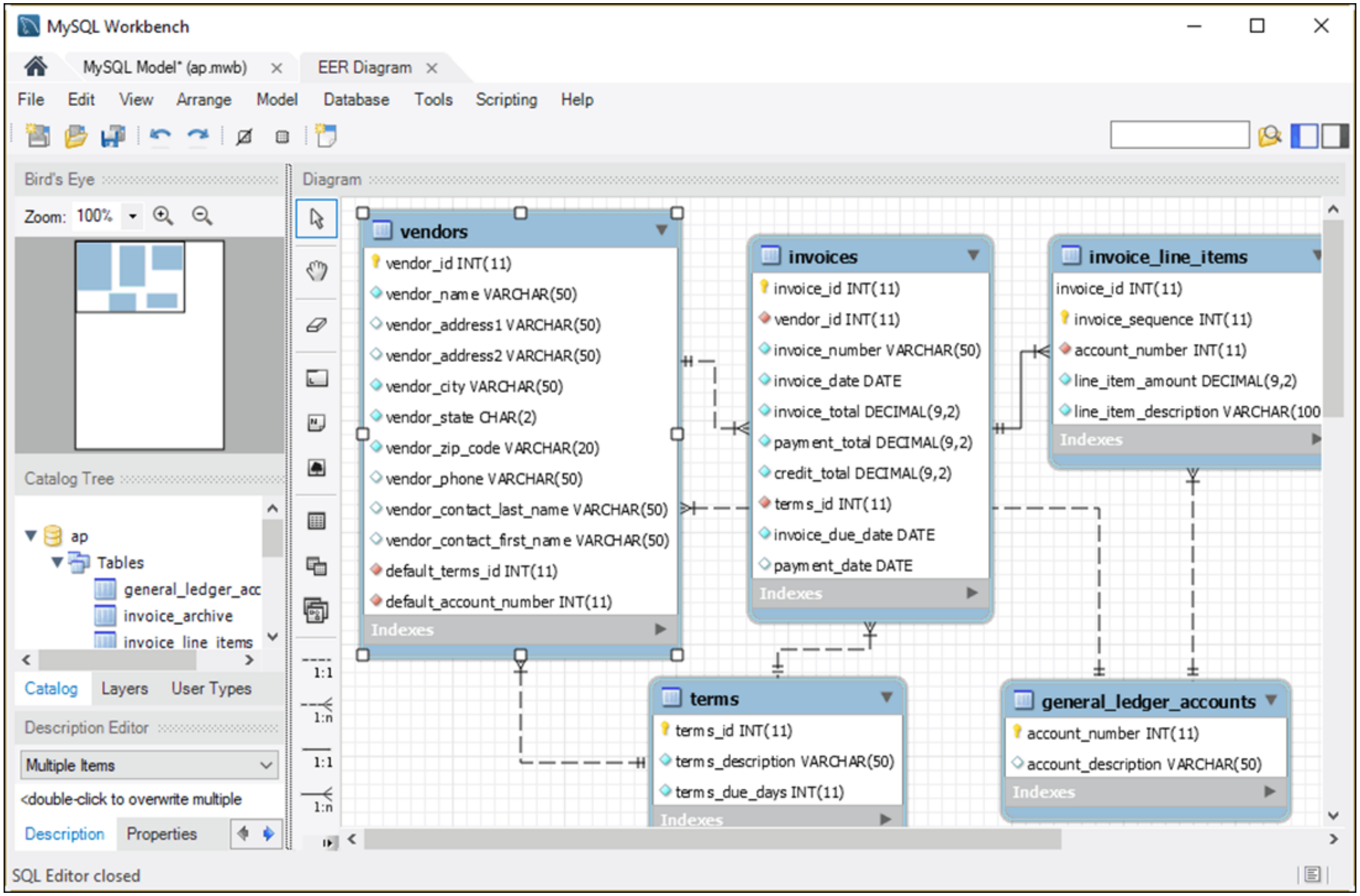
□ Binary □ Unsigned □ Zero Fill

□ Auto Increment □ Generated

Operations for working with an EER model

- Edit a table
- Add a new table
- Delete a table
- Display a diagram
- Create a new diagram
- Create a database creation script from the model

The EER diagram for the AP database



vendors

- vendor_id INT(11)
- vendor_name VARCHAR(50)
- vendor_address1 VARCHAR(50)
- vendor_address2 VARCHAR(50)
- vendor_city VARCHAR(50)
- vendor_state CHAR(2)
- vendor_zip_code VARCHAR(20)
- vendor_phone VARCHAR(50)
- vendor_contact_last_name VARCHAR(50)
- vendor_contact_first_name VARCHAR(50)
- default_terms_id INT(11)
- default_account_number INT(11)

Indexes

invoices

- invoice_id INT(11)
- vendor_id INT(11)
- invoice_number VARCHAR(50)
- invoice_date DATE
- invoice_total DECIMAL(9,2)
- payment_total DECIMAL(9,2)
- credit_total DECIMAL(9,2)
- terms_id INT(11)
- invoice_due_date DATE
- payment_date DATE

Indexes

invoice_line_items

- invoice_id INT(11)
- invoice_sequence INT(11)
- account_number INT(11)
- line_item_amount DECIMAL(9,2)
- line_item_description VARCHAR(100

Indexes

terms

- terms_id INT(11)
- terms_description VARCHAR(50)
- terms_due_days INT(11)

Indexes

general_ledger_accounts

- account_number INT(11)
- account_description VARCHAR(50)

Indexes

Operations for working with an EER Diagram

- Add an existing table to the diagram
- Add a new table to the diagram
- Display the model for a table
- Define the relationship between two tables
- Edit and delete relationships
- Remove a table from the diagram (and, optionally, the model)