Chapter 1

An introduction to relational databases

Objectives

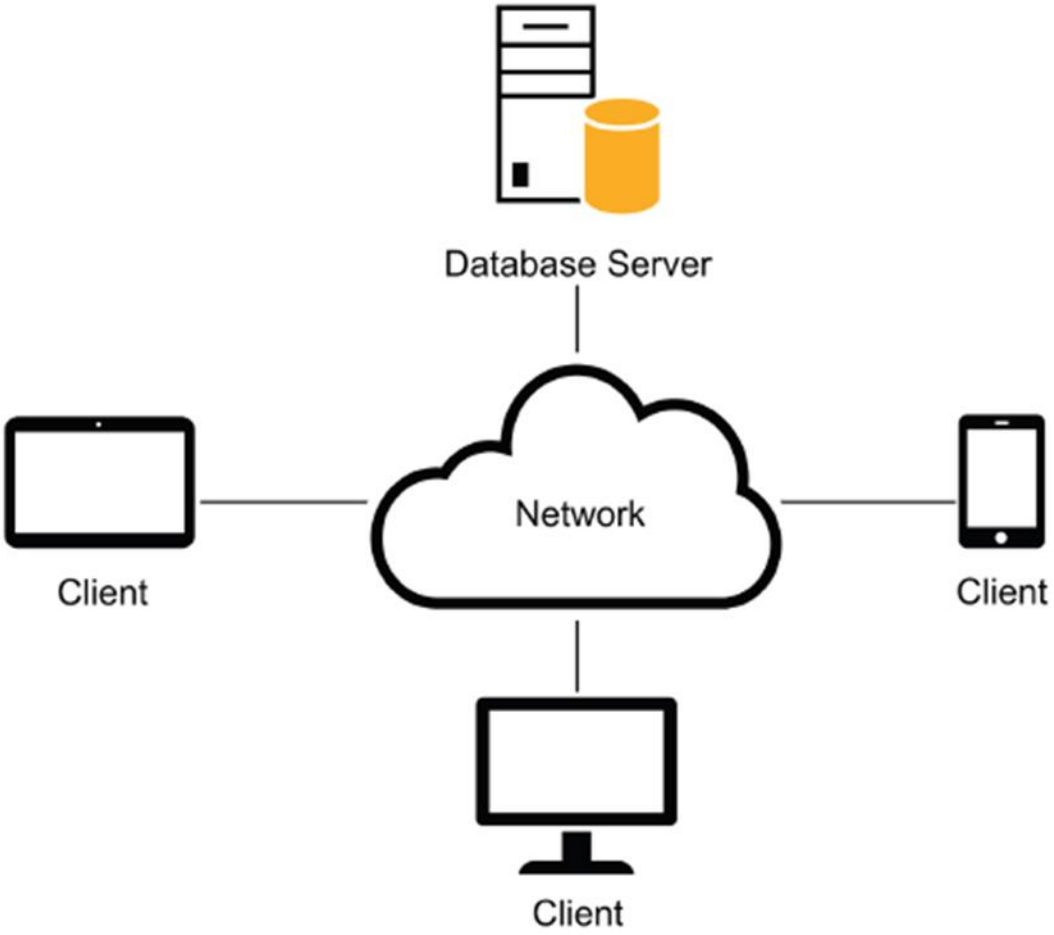
Knowledge

1. Identify the three main hardware components of a client/server system.
2. Describe the way a client accesses the database on a server using these terms: application software, data access API, database management system, SQL query, and query results.
3. Describe the way a relational database is organized using these terms: tables, columns, rows, cells, primary keys, unique keys, and foreign keys.
4. Identify the three types of relationships that can exist between two tables.
5. Describe the way the columns in a table are defined using these terms: data type, null value, and default value.

Objectives (continued)

6. Describe how an enhanced entity relationship diagram can show how the tables in a database are defined and related.
7. Describe the difference between DML statements and DDL statements.
8. List three coding techniques that can make your SQL code easier to read and maintain.

A simple client/server system



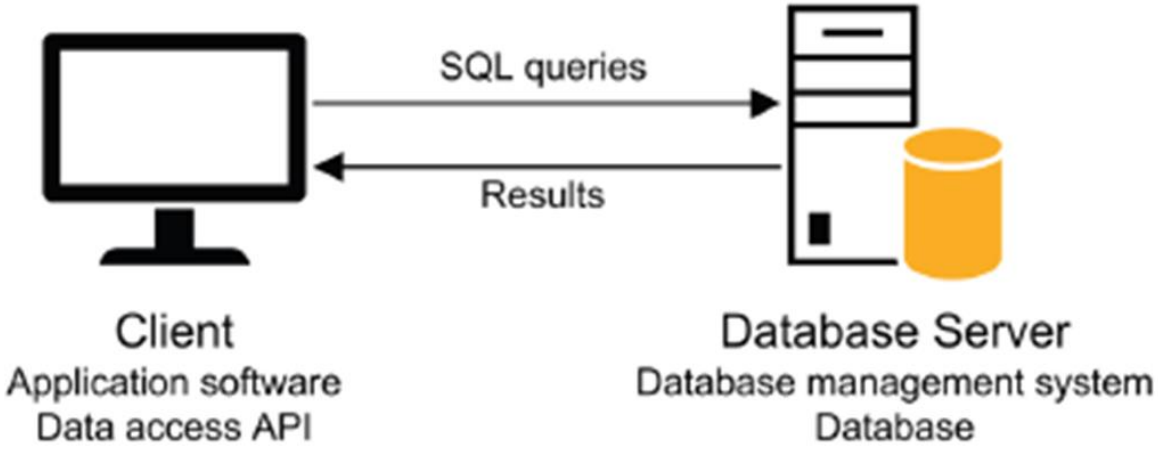
The three hardware components of a client/server system

- Clients
- Server
- Network

Terms to know about client/server systems

- Local area network (LAN)
- Wide area network (WAN)
- Cloud computing platform
- Enterprise system

Client software, server software, and the SQL interface



Server software

- Database management system (DBMS)
- The DBMS does the *back-end processing*

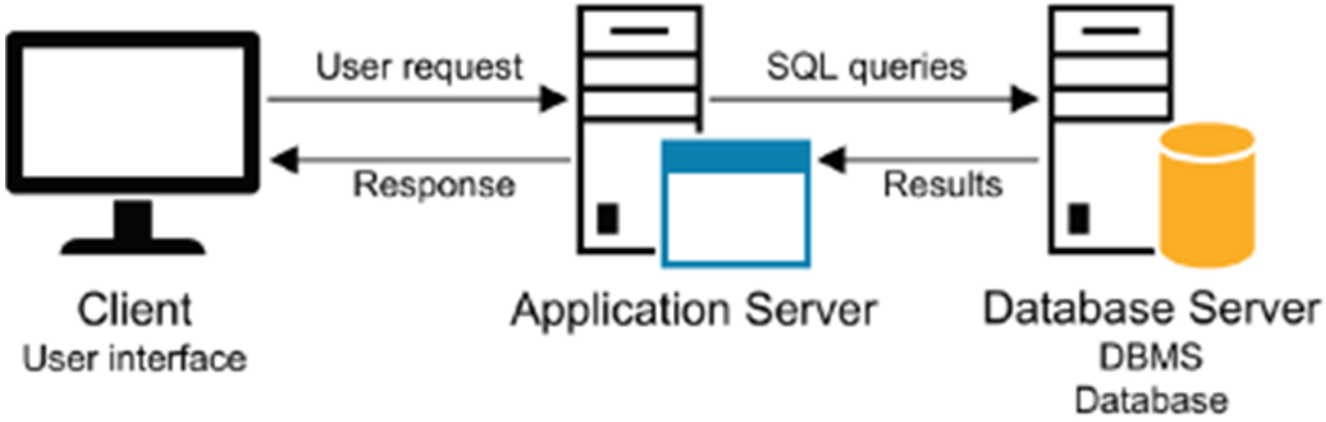
Client software

- Application software
- Data access API (application programming interface)
- The client software does the *front-end processing*

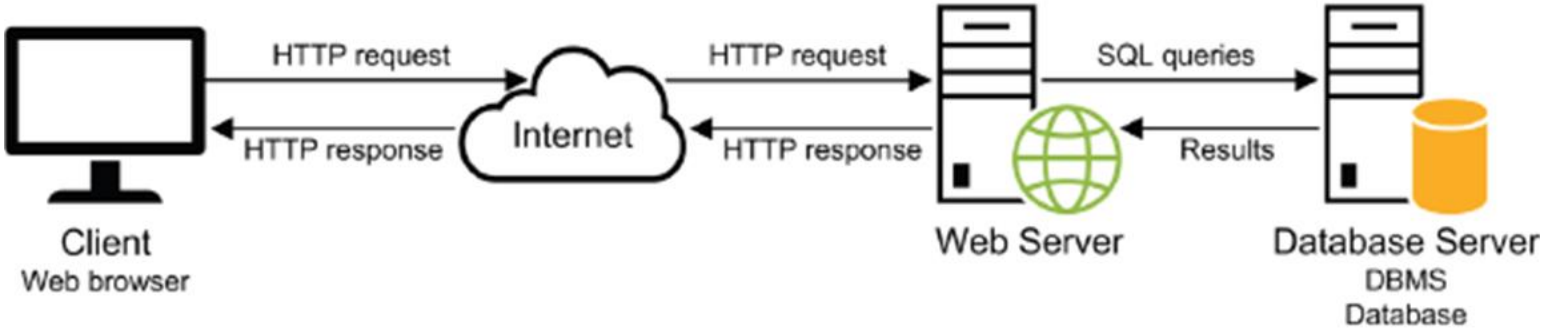
The SQL interface

- *SQL* stands for *Structured Query Language*, which is the standard language for working with a relational database.
- The application software communicates with the DBMS by sending SQL queries through the data access API.
- When the DBMS receives a query that requests data, it processes the query and returns the requested data (the query results) to the client.

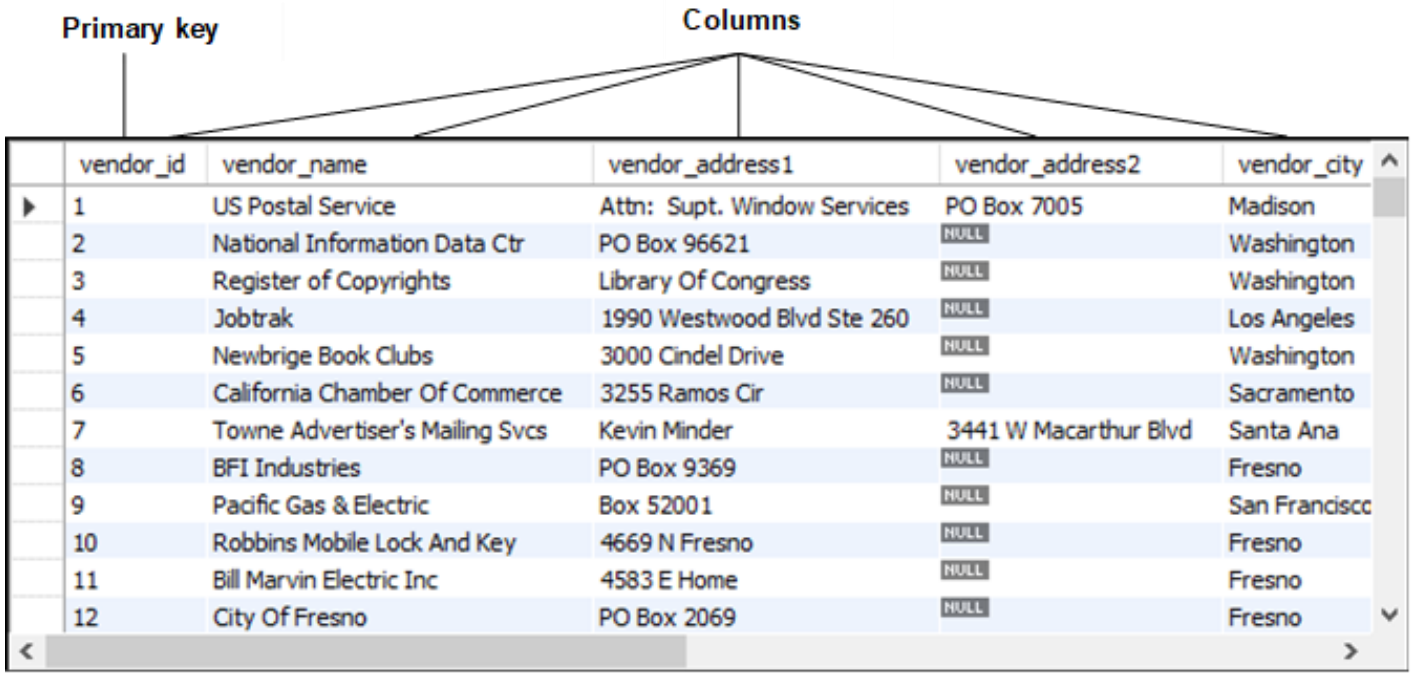
A networked system with an application server



A simple web-based system



The Vendors table in an Accounts Payable (AP) database

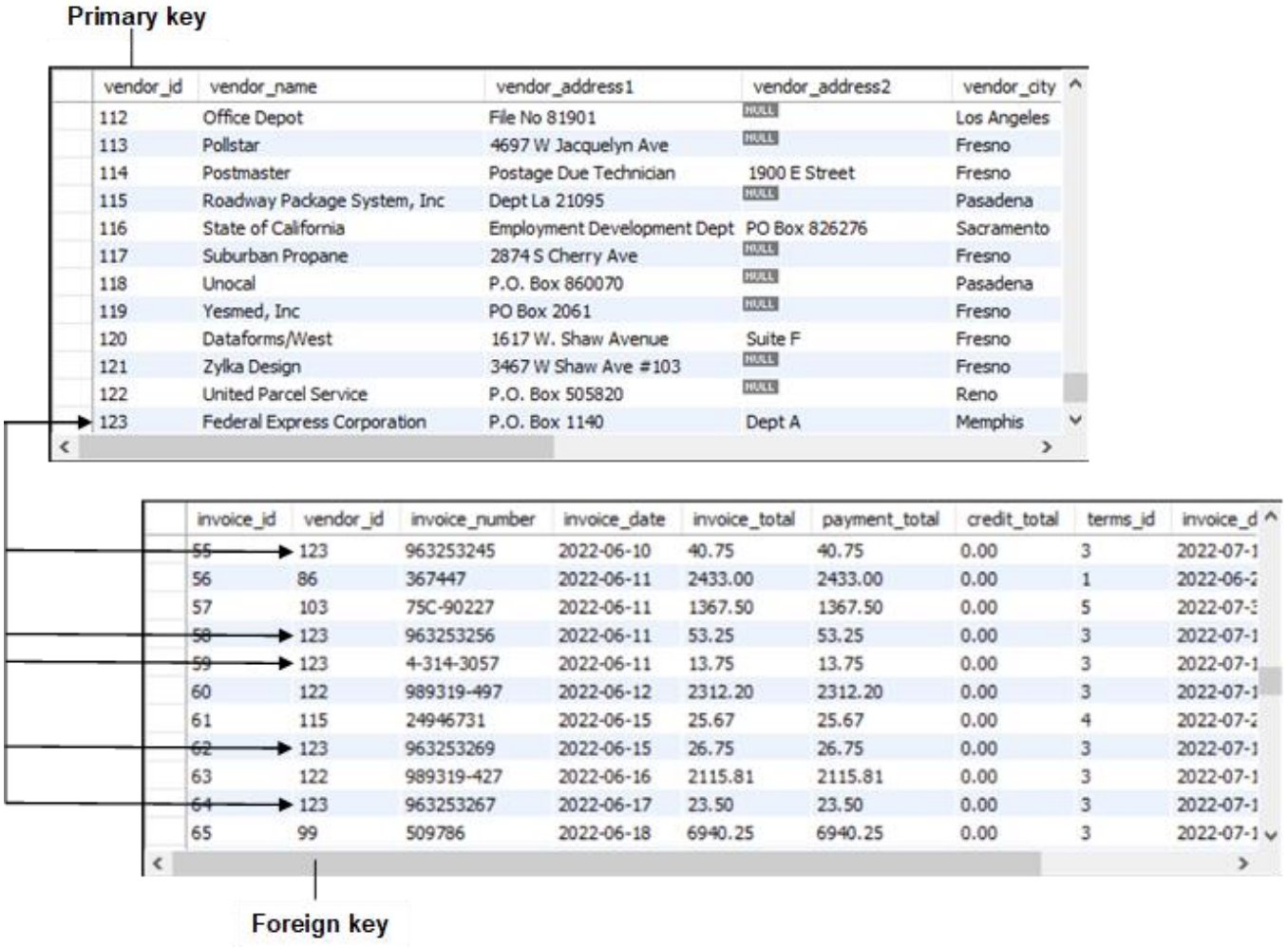


vendor_id	vendor_name	vendor_address1	vendor_address2	vendor_city
1	US Postal Service	Attn: Supt. Window Services	PO Box 7005	Madison
2	National Information Data Ctr	PO Box 96621	NULL	Washington
3	Register of Copyrights	Library Of Congress	NULL	Washington
4	Jobtrak	1990 Westwood Blvd Ste 260	NULL	Los Angeles
5	Newbrige Book Clubs	3000 Cindel Drive	NULL	Washington
6	California Chamber Of Commerce	3255 Ramos Cir	NULL	Sacramento
7	Towne Advertiser's Mailing Svcs	Kevin Minder	3441 W Macarthur Blvd	Santa Ana
8	BFI Industries	PO Box 9369	NULL	Fresno
9	Pacific Gas & Electric	Box 52001	NULL	San Francisco
10	Robbins Mobile Lock And Key	4669 N Fresno	NULL	Fresno
11	Bill Marvin Electric Inc	4583 E Home	NULL	Fresno
12	City Of Fresno	PO Box 2069	NULL	Fresno

Terms to know about database tables

- Relational database
- Table
- Column
- Row
- Cell
- Primary key
- Composite primary key
- Non-primary key (unique key)
- Index

The relationship between two tables



vendor_id	vendor_name	vendor_address1	vendor_address2	vendor_city
112	Office Depot	File No 81901	NULL	Los Angeles
113	Pollstar	4697 W Jacquelyn Ave	NULL	Fresno
114	Postmaster	Postage Due Technician	1900 E Street	Fresno
115	Roadway Package System, Inc	Dept La 21095	NULL	Pasadena
116	State of California	Employment Development Dept	PO Box 826276	Sacramento
117	Suburban Propane	2874 S Cherry Ave	NULL	Fresno
118	Unocal	P.O. Box 860070	NULL	Pasadena
119	Yesmed, Inc	PO Box 2061	NULL	Fresno
120	Dataforms/West	1617 W. Shaw Avenue	Suite F	Fresno
121	Zylka Design	3467 W Shaw Ave #103	NULL	Fresno
122	United Parcel Service	P.O. Box 505820	NULL	Reno
123	Federal Express Corporation	P.O. Box 1140	Dept A	Memphis

invoice_id	vendor_id	invoice_number	invoice_date	invoice_total	payment_total	credit_total	terms_id	invoice_d
55	123	963253245	2022-06-10	40.75	40.75	0.00	3	2022-07-1
56	86	367447	2022-06-11	2433.00	2433.00	0.00	1	2022-06-2
57	103	75C-90227	2022-06-11	1367.50	1367.50	0.00	5	2022-07-3
58	123	963253256	2022-06-11	53.25	53.25	0.00	3	2022-07-1
59	123	4-314-3057	2022-06-11	13.75	13.75	0.00	3	2022-07-1
60	122	989319-497	2022-06-12	2312.20	2312.20	0.00	3	2022-07-1
61	115	24946731	2022-06-15	25.67	25.67	0.00	4	2022-07-2
62	123	963253269	2022-06-15	26.75	26.75	0.00	3	2022-07-1
63	122	989319-427	2022-06-16	2115.81	2115.81	0.00	3	2022-07-1
64	123	963253267	2022-06-17	23.50	23.50	0.00	3	2022-07-1
65	99	509786	2022-06-18	6940.25	6940.25	0.00	3	2022-07-1

Terms to know about table relationships

- Foreign key
- One-to-many relationship
- One-to-one relationship
- Many-to-many relationship

The columns of the Invoices table

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
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)	☐	☑	☐	☐	☐	☐	☐	☐	'0.00'
credit_total	DECIMAL(9,2)	☐	☑	☐	☐	☐	☐	☐	☐	'0.00'
terms_id	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
invoice_due_date	DATE	☐	☑	☐	☐	☐	☐	☐	☐	
payment_date	DATE	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

Column Name: Data Type:

Charset/Collation: Default:

Comments: Storage: Virtual Stored

Primary Key Not Null Unique

Binary Unsigned Zero Fill

Auto Increment Generated

Common MySQL data types

CHAR, VARCHAR

INT, DECIMAL

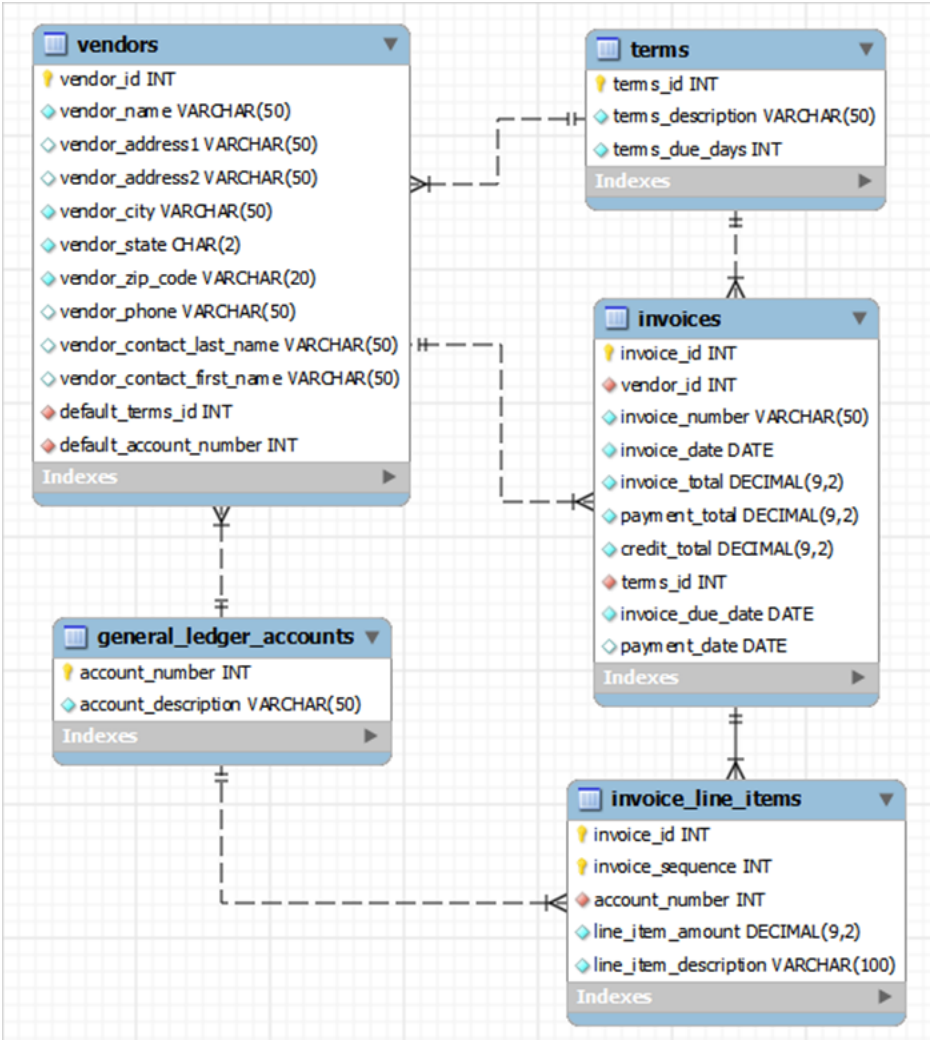
FLOAT

DATE

Terms to know about columns

- Data type
- Null value
- Default value
- Auto increment column

An EER diagram for the AP database



How knowing “standard SQL” helps you

- Basic SQL statements are the same for all *dialects*.
- Once you know one dialect, you can easily learn others.

How knowing “standard SQL” does not help you

- Most applications require modification when moved to another database.

A comparison of four relational databases

Oracle

Released in 1979.

Runs on Unix, z/OS, Windows, Linux, and macOS.

Typically used for large, mission-critical systems that run on one or more Unix servers.

DB2

Released in 1985.

Runs on OS/390, z/OS, AIX, Unix, Windows, Linux, and macOS.

Typically used for large, mission-critical systems that run on legacy IBM mainframe systems using the z/OS or OS/390 operating system.

A comparison of four relational databases (cont.)

SQL Server

Released in 1987.

Runs on Windows and Linux.

Typically used for small- to medium-sized systems that run on one or more Windows or Linux servers.

MySQL

Released in 2000.

Runs on Unix, Linux, Windows, and macOS.

A popular *open-source database* that runs on all major operating systems and is commonly used for web applications.

SQL statements used to work with data (DML)

```
SELECT
INSERT
UPDATE
DELETE
```

SQL statements used to work with database objects (DDL)

```
CREATE DATABASE
CREATE TABLE
CREATE INDEX

ALTER TABLE
ALTER INDEX

DROP DATABASE
DROP TABLE
DROP INDEX
```

A statement that creates a new database

```
CREATE DATABASE ap
```

A statement that selects the current database

```
USE ap
```

A statement that creates a new table

```
CREATE TABLE invoices
(
  invoice_id         INT             PRIMARY KEY
                     AUTO_INCREMENT,
  vendor_id          INT             NOT NULL,
  invoice_number     VARCHAR(50)     NOT NULL,
  invoice_date       DATE            NOT NULL,
  invoice_total      DECIMAL(9,2)    NOT NULL,
  payment_total      DECIMAL(9,2)                DEFAULT 0,
  credit_total       DECIMAL(9,2)                DEFAULT 0,
  terms_id           INT             NOT NULL,
  invoice_due_date   DATE            NOT NULL,
  payment_date       DATE,
  CONSTRAINT invoices_fk_vendors
    FOREIGN KEY (vendor_id)
    REFERENCES vendors (vendor_id),
  CONSTRAINT invoices_fk_terms
    FOREIGN KEY (terms_id)
    REFERENCES terms (terms_id)
)
```

A statement that adds a new column to a table

```
ALTER TABLE invoices
ADD balance_due DECIMAL(9,2)
```

A statement that deletes the new column

```
ALTER TABLE invoices
DROP COLUMN balance_due
```

A statement that creates an index on the table

```
CREATE INDEX invoices_vendor_id_index
  ON invoices (vendor_id)
```

A statement that deletes the new index

```
DROP INDEX invoices_vendor_id_index
  ON invoices
```

The Invoices base table

invoice_id	vendor_id	invoice_number	invoice_date	invoice_total	payment_total	credit_total	terms_id	invoice_du
1	122	989319-457	2022-04-08	3813.33	3813.33	0.00	3	2022-05-08
2	123	263253241	2022-04-10	40.20	40.20	0.00	3	2022-05-10
3	123	963253234	2022-04-13	138.75	138.75	0.00	3	2022-05-1
4	123	2-000-2993	2022-04-16	144.70	144.70	0.00	3	2022-05-16
5	123	963253251	2022-04-16	15.50	15.50	0.00	3	2022-05-16
6	123	963253261	2022-04-16	42.75	42.75	0.00	3	2022-05-16
7	123	963253237	2022-04-21	172.50	172.50	0.00	3	2022-05-2
8	89	125520-1	2022-04-24	95.00	95.00	0.00	1	2022-05-0

A SELECT statement that retrieves and sorts selected columns and rows

```
SELECT invoice_number, invoice_date, invoice_total,
    payment_total, credit_total,
    invoice_total - payment_total - credit_total
    AS balance_due
FROM invoices
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY invoice_date
```

The result set defined by the SELECT statement

invoice_number	invoice_date	invoice_total	payment_total	credit_total	balance_due
39104	2022-07-10	85.31	0.00	0.00	85.31
963253264	2022-07-18	52.25	0.00	0.00	52.25
31361833	2022-07-21	579.42	0.00	0.00	579.42
263253268	2022-07-21	59.97	0.00	0.00	59.97
263253270	2022-07-22	67.92	0.00	0.00	67.92
263253273	2022-07-22	30.75	0.00	0.00	30.75
P-0608	2022-07-23	20551.18	0.00	1200.00	19351.18
9982771	2022-07-24	503.20	0.00	0.00	503.20
134116	2022-07-28	90.36	0.00	0.00	90.36

A SELECT statement that joins data

```
SELECT vendor_name, invoice_number, invoice_date,
       invoice_total
FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
WHERE invoice_total >= 500
ORDER BY vendor_name, invoice_total DESC
```

The result set defined by the SELECT statement

vendor_name	invoice_number	invoice_date	invoice_total
Bertelsmann Industry Svcs. Inc	509786	2022-06-18	6940.25
Cahners Publishing Company	587056	2022-06-30	2184.50
Computerworld	367447	2022-06-11	2433.00
Data Reproductions Corp	40318	2022-06-01	21842.00
Dean Witter Reynolds	75C-90227	2022-06-11	1367.50
Digital Dreamworks	P02-3772	2022-05-21	7125.34
Federal Express Corporation	963253230	2022-07-07	739.20
Ford Motor Credit Company	9982771	2022-07-24	503.20
Franchise Tax Board	RTR-72-3662-X	2022-05-25	1600.00

Terms to know about SQL

- Query
- Base table
- Result table (result set)
- Calculated value
- Join
- Inner join
- Outer join

A statement that adds a row to the Invoices table

```
INSERT INTO invoices
  (vendor_id, invoice_number, invoice_date,
   invoice_total, terms_id, invoice_due_date)
VALUES
  (12, '3289175', '2022-07-18', 165, 3, '2022-08-17')
```

A statement that changes the value of a column for one row

```
UPDATE invoices
SET credit_total = 35.89
WHERE invoice_number = '367447'
```

A statement that changes the values in a column for multiple rows

```
UPDATE invoices
SET invoice_due_date
    = DATE_ADD(invoice_due_date, INTERVAL 30 DAY)
WHERE terms_id = 4
```

A statement that deletes a selected invoice from the Invoices table

```
DELETE FROM invoices
WHERE invoice_number = '4-342-8069'
```

A statement that deletes all paid invoices from the Invoices table

```
DELETE FROM invoices
WHERE invoice_total - payment_total - credit_total = 0
```

A SELECT statement that's difficult to read

```
select invoice_number, invoice_date, invoice_total,
payment_total, credit_total, invoice_total - payment_total -
credit_total as balance_due from invoices where
invoice_total - payment_total - credit_total > 0 order by
invoice_date
```

A SELECT statement that's easy to read

```
SELECT invoice_number, invoice_date, invoice_total,
    payment_total, credit_total,
    invoice_total - payment_total - credit_total
    AS balance_due
FROM invoices
WHERE invoice_total - payment_total - credit_total > 0
ORDER BY invoice_date
```

A SELECT statement with a block comment

```
/*
Author: Joel Murach
Date: 8/22/2023
*/
SELECT invoice_number, invoice_date, invoice_total,
    invoice_total - payment_total - credit_total
    AS balance_due
FROM invoices
```

A SELECT statement with a single-line comment

```
-- The fourth column calculates the balance due
SELECT invoice_number, invoice_date, invoice_total,
    invoice_total - payment_total - credit_total
    AS balance_due
FROM invoices
```

Coding recommendations

- Capitalize all keywords.
- Use lowercase for the other code.
- Separate the words in names with underscores.
- Start each clause on a new line.
- Break long clauses into multiple lines.
- Indent continued lines.
- Use comments only for code that is difficult to understand.

Note

- Line breaks, white space, indentation, and capitalization have no effect on how MySQL processes statements.