

A

Exercise Answers

Chapter 1

Exercise 1 Solution

A hierarchical database is based on a parent-child configuration in which data is decomposed into logical categories and subcategories that use records to represent the logical units of data. The model is similar to an inverted tree, like the one you see in a file management system. A network database is built on the hierarchical model. It enhances that model by allowing records to participate in multiple parent-child relationships. Both hierarchical and network databases require developers to program record navigation in their application. As a result, any changes to the database or application can result in complicated updates. A relational database represents a departure from the strict structure of hierarchical and network databases. The relational database is based on tables that are made up of columns and rows, and it supports complex relationship between tables, without the restrictions of the earlier models.

Exercise 2 Solution

Use the following CREATE TABLE statement to create the table:

```
CREATE TABLE Employees
(
    EmpID INT NOT NULL,
    EmpName VARCHAR(40) NOT NULL,
    PRIMARY KEY (EmpID)
)
ENGINE=MYISAM;
```

Exercise 3 Solution

Use the following PHP code:

```
$connection=mysql_connect($host,$user,$pw) or die ("Connection failed!");
```

Chapter 2

Exercise 1 Solution

You should download the following file:

```
mysql-max-4.0.20-pc-linux-i686.tar.gz
```

Exercise 2 Solution

You should use the following command:

```
rpm -i mysql-server-5.0.0-0.i386.rpm
```

Exercise 3 Solution

You should use the following command:

```
/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

Exercise 4 Solution

Run the MySQL Server Instance Configuration wizard after installing MySQL.

Chapter 3

Exercise 1 Solution

```
/usr/bin
```

Exercise 2 Solution

You should use the following command:

```
mysql -u myadmin -p
```

You can also use long option names rather than short option names in your command.

Exercise 3 Solution

You should add the following code to your configuration file:

```
[mysqladmin]  
host=system3  
user=mysqladmin  
user=pw1
```

You do not have to include the [mysqladmin] heading if it already exists. Simply place the rest of the commands beneath that heading.

Exercise 4 Solution

You should use the following command:

```
mysql -t < c:\mysql_files\users.sql > c:\mysql_files\users.txt
```

Exercise 5 Solution

You should use the following SQL statement:

```
SET PASSWORD FOR 'myadmin'@'%' = PASSWORD('pwl');
```

Chapter 4

Exercise 1 Solution

Columns, rows, and a primary key.

Exercise 2 Solution

To be in compliance with the first normal form, a relation must meet the following requirements:

- Each column in a row must be atomic. In other words, the column can contain only one value for any given row.
- Each row in a relation must contain the same number of columns. Given that each column can contain only one value, this means that each row must contain the same number of values.
- All rows in a relation must be different. Although rows might include the same values, each row, when taken as a whole, must be unique in the relation.

Exercise 3 Solution

In a one-to-many relationship, a row in the first relation can be related to one or more rows in the second relation, but a row in the second relation can be related to only one row in the first relation. In a many-to-many relationship, a row in the first relation can be related to one or more rows in the second relation, and a row in the second relation can be related to one or more rows in the first relation.

Exercise 4 Solution

You should apply the rules of normalization to the data.

Exercise 5 Solution

Many-to-many relationships are implemented in MySQL by adding a junction table between the two tables and creating a one-to-many relationship between the junction table and the first original table as well as a one-to-many relationship between the junction table and the second original table.

Chapter 5

Exercise 1 Solution

You should use the following SQL statement:

```
CREATE DATABASE NewDB
COLLATE latin1_general_ci;
```

Exercise 2 Solution

You should use the following SQL statement:

```
CREATE TABLE Bikes
(
  BikeID TINYINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  BikeName VARCHAR(40) NOT NULL
);
```

Exercise 3 Solution

You should use the following ALTER TABLE statement:

```
ALTER TABLE ModelTrains
ADD UNIQUE un_1 (ModelName);
```

Exercise 4 Solution

You should use the following ALTER TABLE statement:

```
ALTER TABLE ModelTrains
DROP INDEX un_1;
```

Exercise 5 Solution

You should use the following SQL statement:

```
SHOW TABLES;
```

Chapter 6

Exercise 1 Solution

You should use an SQL statement similar to the following:

```
INSERT INTO Books
VALUES (1001, 'One Hundred Years of Solitude');
```

You can also use the following SQL statement:

```
INSERT INTO Books (BookID, BookName)
VALUES (1001, 'One Hundred Years of Solitude');
```

Exercise 2 Solution

You should use an SQL statement similar to the following:

```
REPLACE INTO Books
SET BookID=1001, BookName='One Hundred Years of Solitude';
```

Exercise 3 Solution

You should use an SQL statement similar to the following:

```
UPDATE CDs
SET CDQuantity=CDQuantity+3;
```

Exercise 4 Solution

You should use an SQL statement similar to the following:

```
UPDATE CDs
SET CDQuantity=CDQuantity+3
WHERE CDName='Mule Variations';
```

Exercise 5 Solution

You should use an SQL statement similar to the following:

```
DELETE FROM CDs
WHERE CDID=1;
```

Chapter 7

Exercise 1 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT CDName, InStock, OnOrder, Reserved
FROM CDs;
```

Exercise 2 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT CDName, InStock, OnOrder, Reserved, InStock+OnOrder-Reserved AS Total
FROM CDs;
```

Exercise 3 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT CDName, InStock, OnOrder, Reserved, InStock+OnOrder-Reserved AS Total
FROM CDs
WHERE Department='Classical' AND InStock<20;
```

Exercise 4 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT Department, Category, COUNT(*) AS Total
FROM CDs
GROUP BY Department, Category WITH ROLLUP;
```

Exercise 5 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT Department, Category, COUNT(*) AS Total
FROM CDs
GROUP BY Department, Category WITH ROLLUP
HAVING Total<3;
```

Exercise 6 Solution

You should create the following `SELECT` statement to retrieve the information that you need:

```
SELECT CDName
FROM CDs
ORDER BY CDName DESC;
```

Chapter 8

Exercise 1 Solution

You should create the following SELECT statement to retrieve the information that you need:

```
SELECT ProdName, InStock, OnOrder, InStock+OnOrder AS Total
FROM Produce
ORDER BY ProdName;
```

Exercise 2 Solution

You should create the following SELECT statement to retrieve the information that you need:

```
SELECT ProdName, Variety, InStock, OnOrder
FROM Produce
WHERE InStock+OnOrder>=5000
ORDER BY ProdName;
```

Exercise 3 Solution

You should create the following SELECT statement to retrieve the information that you need:

```
SELECT ProdName, Variety, InStock
FROM Produce
WHERE InStock>=1000 AND (ProdName='Apples' OR ProdName='Oranges')
ORDER BY ProdName;
```

Exercise 4 Solution

You should create the following UPDATE statement to retrieve the information that you need:

```
UPDATE Produce
SET SeasonAttr=SeasonAttr | 2
WHERE ProdName='grapes';
```

Exercise 5 Solution

You should create the following SELECT statement to retrieve the information that you need:

```
SELECT ProdName, Variety, InStock
FROM Produce
WHERE ProdName REGEXP BINARY 'Ch'
ORDER BY ProdName;
```

Chapter 9

Exercise 1 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT ProdName, InStock,
       CASE ProdName
         WHEN 'Apples' THEN 'On Sale!'
         WHEN 'Oranges' THEN 'Just Arrived!'
         ELSE 'Fresh Crop!'
       END As Signage
FROM Produce
WHERE InStock>=1000
ORDER BY ProdName;
```

Exercise 2 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT ProdName, Variety, CAST(InStock AS CHAR) AS InStock_CHAR
FROM Produce
WHERE InStock>=1000
ORDER BY ProdName;
```

Exercise 3 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT CONCAT(ProdName, ' (', Variety, ')') AS ProduceVariety, InStock
FROM Produce
WHERE InStock>=1000 AND Variety IS NOT NULL
ORDER BY ProdName;
```

Exercise 4 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT UPPER(CONCAT(ProdName, ' (', Variety, ')')) AS ProduceVariety, InStock
FROM Produce
WHERE InStock>=1000 AND Variety IS NOT NULL
ORDER BY ProdName;
```

Exercise 5 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT Variety, OnOrder, DateOrdered, ADDDATE(DateOrdered, 4) AS DeliveryDate
FROM Produce
WHERE ProdName='Apples'
ORDER BY Variety;
```

Exercise 6 Solution

You should create a SELECT statement similar to the following:

```
SELECT ProdName, SUM(InStock) AS TotalInStock, SUM(OnOrder) AS TotalOrdered
FROM Produce
GROUP BY ProdName;
```

Chapter 10

Exercise 1 Solution

You should create a SELECT statement similar to the following:

```
SELECT BookTitle, CONCAT_WS(' ', AuthFN, AuthMN, AuthLN) As Author
FROM Books AS b, AuthorBook AS ab, Authors AS a
WHERE b.BookID=ab.BookID AND ab.AuthID=a.AuthID
ORDER BY BookTitle;
```

Exercise 2 Solution

You should create a SELECT statement similar to the following:

```
SELECT BookTitle, CONCAT_WS(' ', AuthFN, AuthMN, AuthLN) As Author
FROM Books AS b CROSS JOIN AuthorBook AS ab ON b.BookID=ab.BookID
CROSS JOIN Authors AS a ON ab.AuthID=a.AuthID
WHERE AuthLN='Toole' OR AuthLN='Thompson'
ORDER BY BookTitle;
```

Exercise 3 Solution

You should create a SELECT statement similar to the following:

```
SELECT BookTitle, CONCAT_WS(' ', AuthFN, AuthMN, AuthLN) As Author
FROM Books AS b RIGHT JOIN AuthorBook AS ab ON b.BookID=ab.BookID
RIGHT JOIN Authors AS a ON ab.AuthID=a.AuthID
ORDER BY BookTitle;
```

Exercise 4 Solution

You should create a SELECT statement similar to the following:

```
SELECT BookTitle, CONCAT_WS(' ', AuthFN, AuthMN, AuthLN) As Author
FROM Books AS b NATURAL RIGHT JOIN AuthorBook AS ab
NATURAL RIGHT JOIN Authors AS a
ORDER BY BookTitle;
```

Exercise 5 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT BookTitle
FROM Books
WHERE BookID IN
    (SELECT BookID FROM BookOrders WHERE Quantity>2)
ORDER BY BookTitle;
```

Exercise 6 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT OrderID, BookID, Quantity
FROM BookOrders
WHERE BookID=
    (SELECT BookID FROM Books WHERE BookTitle='Letters to a Young Poet');
```

Exercise 7 Solution

You should create a `SELECT` statement similar to the following:

```
(SELECT AuthLN FROM Authors)
UNION
(SELECT AuthLN FROM Authors2)
ORDER BY AuthLN;
```

Chapter 11

Exercise 1 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT ProdName, Variety, InStock
    INTO OUTFILE 'Apples.txt'
FROM Produce WHERE ProdName='Apples';
```

Exercise 2 Solution

You should create a `SELECT` statement similar to the following:

```
SELECT ProdName, Variety, InStock
    INTO OUTFILE 'Oranges.txt'
    FIELDS
        TERMINATED BY ','
        ENCLOSED BY '*'
FROM Produce WHERE ProdName='Oranges';
```

Exercise 3 Solution

You should create a CREATE TABLE statement similar to the following:

```
CREATE TABLE Produce2
(
  ProdName VARCHAR(40) NOT NULL,
  Variety VARCHAR(40) NULL,
  InStock SMALLINT UNSIGNED NOT NULL
)
SELECT ProdName, Variety, InStock
FROM Produce WHERE ProdName='Cherries';
```

Exercise 4 Solution

You should create an INSERT statement similar to the following:

```
INSERT INTO Produce2
SELECT ProdName, Variety, InStock
FROM Produce WHERE ProdName='Mushrooms';
```

Exercise 5 Solution

You should create a LOAD DATA statement similar to the following:

```
LOAD DATA INFILE 'Apples.txt'
INTO TABLE Produce2;
```

Exercise 6 Solution

You should create a LOAD DATA statement similar to the following:

```
LOAD DATA INFILE 'Oranges.txt'
INTO TABLE Produce2
FIELDS
  TERMINATED BY ','
  ENCLOSED BY '*';
```

Chapter 12

Exercise 1 Solution

You should use the following SQL statement to begin the transaction:

```
START TRANSACTION;
```

Exercise 2 Solution

You should use the following SQL statement to commit the transaction:

```
COMMIT;
```

Exercise 3 Solution

You should use the following SQL statement to roll back the transaction:

```
ROLLBACK;
```

Exercise 4 Solution

You should use the following SQL statement to create the savepoint:

```
SAVEPOINT save1;
```

Exercise 5 Solution

You should use the following SQL statement to roll back the transaction to the savepoint:

```
ROLLBACK TO SAVEPOINT save1;
```

Exercise 6 Solution

You should use the following SQL statement to turn the autocommit mode to off:

```
SET AUTOCOMMIT=0;
```

Exercise 7 Solution

You should use the following SQL statement to change the isolation level:

```
SET GLOBAL TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

Exercise 8 Solution

You should use the following SQL statement to lock the Produce and Orders tables:

```
LOCK TABLES Produce READ, Orders WRITE;
```

Chapter 13

Exercise 1 Solution

You should use the following command to retrieve the information that you need:

```
mysqladmin -u myadmin -p refresh status
```

Exercise 2 Solution

You should use the following SQL statement to retrieve the information that you need:

```
SHOW GLOBAL VARIABLES LIKE '%max%';
```

Exercise 3 Solution

You should use the following SQL statement to retrieve the information that you need:

```
SELECT @@query_cache_limit;
```

Exercise 4 Solution

You should use the following SQL statement to retrieve the information that you need:

```
SHOW STATUS LIKE '%cache%';
```

Exercise 5 Solution

You should use the following SQL statement to set the variable value:

```
SET MAX_TMP_TABLES=24;
```

Exercise 6 Solution

You should add the following command to the `[mysqld]` section of your option file:

```
log-bin
```

Exercise 7 Solution

You should use the following command to view the binary log file:

```
mysqlbinlog Server21-bin.000327
```

Chapter 14

Exercise 1 Solution

You should use the following SQL statement to create the user account and assign privileges:

```
GRANT SELECT, INSERT, UPDATE (PubName, City)  
ON Books.Publishers  
TO 'mgr1'@'localhost' IDENTIFIED BY 'mgr1pw';
```

Exercise 2 Solution

You should use the following SQL statement to change the password:

```
SET PASSWORD FOR 'mgr1'@'%' = PASSWORD('mgr2pw');
```

Exercise 3 Solution

You should use the following SQL statement to reload the privileges:

```
FLUSH PRIVILEGES;
```

Exercise 4 Solution

You should use the following SQL statement to display the privileges:

```
SHOW GRANTS FOR 'mgr1'@'%';
```

Exercise 5 Solution

You should use the following SQL statement to revoke the privileges:

```
REVOKE SELECT, INSERT, UPDATE (PubName, City)  
ON Books.Publishers  
FROM 'mgr1'@'%';
```

You can also use the following SQL statement to revoke the privileges:

```
REVOKE ALL PRIVILEGES, GRANT OPTION  
FROM 'mgr1'@'%';
```

Exercise 6 Solution

You should add the following SQL statement to remove the user account:

```
DROP USER 'mgr1'@'%';
```

Chapter 15

Exercise 1 Solution

You should consider defining an index on the ManfID column of the Parts table.

Exercise 2 Solution

You should use the following SQL statement to analyze the SELECT statement:

```
EXPLAIN SELECT PartID, PartName, ManfName
FROM Parts AS p, Manufacturers as m
WHERE p.ManfID = m.ManfID
ORDER BY PartName;
```

Exercise 3 Solution

You should try to add a `FORCE INDEX` clause to your `SELECT` statement, then use an `EXPLAIN` statement against the updated `SELECT` statement to see whether the `FORCE INDEX` clause improves the performance of the statement.

Exercise 4 Solution

The fastest way to insert the data is to use a `LOAD DATA` statement to insert the data from a text file.

Exercise 5 Solution

You should use the following SQL statement to delete the data:

```
TRUNCATE Parts;
```

Exercise 6 Solution

When setting up your columns, you should take into account the following guidelines: Use identical column types for compared columns, specify data types that have the correct length, define your columns as `NOT NULL` when appropriate, and consider defining your columns with the `ENUM` data type.

Exercise 7 Solution

To implement query caching in Linux, add following entry to the `[mysqld]` section of your option file:

```
query_cache_size=10M
```

If you're implementing query caching in Windows, set the existing `query_cache_size` system variable to 10M.

Chapter 16

Exercise 1 Solution

On Windows, you should use the following command to create a backup file:

```
mysqldump --flush-logs ProduceDB > c:\mysqlbackup\producedb001.sql
```

On Linux, you should use the following command to create a backup file:

```
mysqldump --flush-logs ProduceDB > /mysqlbackup/producedb001.sql
```

Exercise 2 Solution

On Windows, you should use the following command to create a backup file:

```
mysqldump --flush-logs ProduceDB Produce > c:\mysqlbackup\produce.sql
```

On Linux, you should use the following command to create a backup file:

```
mysqldump --flush-logs ProduceDB Produce > /mysqlbackup/produce.sql
```

Exercise 3 Solution

On Windows, you should use the following command to create a backup file:

```
mysqldump --flush-logs --databases ProduceDB > c:\mysqlbackup\producedb.sql
```

On Linux, you should use the following command to create a backup file:

```
mysqldump --flush-logs --databases ProduceDB > /mysqlbackup/producedb.sql
```

Exercise 4 Solution

On Windows, you should use the following command to restore the database:

```
mysql < c:\mysqlbackup\producedb.sql
```

On Linux, you should use the following command to restore the database:

```
mysql < /mysqlbackup/producedb.sql
```

Exercise 5 Solution

You should use the following SQL statement to grant the necessary privileges to the slave server:

```
GRANT REPLICATION SLAVE ON *.*  
TO 'rep_user'@'slave1'  
IDENTIFIED BY 'rep_pw';
```

Exercise 6 Solution

You should execute the following CHANGE MASTER statement on the slave server:

```
CHANGE MASTER TO  
  MASTER_HOST='master1',  
  MASTER_USER='rep_user',  
  MASTER_PASSWORD='rep_pw',  
  MASTER_LOG_FILE='master1-bin.000127',  
  MASTER_LOG_POS=79;
```

Chapter 17

Exercise 1 Solution

You should use a PHP statement similar to the following:

```
$myLink = mysql_connect("server1", "app_user", "app_pw");
```

Exercise 2 Solution

You should use a PHP statement similar to the following:

```
mysql_select_db("sales_db");
```

Exercise 3 Solution

You should use a PHP statement similar to the following:

```
$myResult = mysql_query($myQuery)  
or die("Error in query: " . mysql_error());
```

Exercise 4 Solution

You should use a PHP statement similar to the following:

```
while($row = mysql_fetch_array($myResult))  
{  
    $cdName = $row["CDName"];  
    $inStock = $row["InStock"];  
    printf($cdName, $inStock);  
}
```

Exercise 5 Solution

You should use a PHP statement similar to the following:

```
if(isset($_POST["user1"]))  
    $user1 = $_POST["user1"];
```

Exercise 6 Solution

You should use a PHP statement similar to the following:

```
header("Location: index.php");
```

Exercise 7 Solution

You should use a PHP statement similar to the following:

```
mysql_close($myLink);
```

Chapter 18

Exercise 1 Solution

You should use a Java statement similar to the following:

```
<%@ page import "java.sql.*" %>
```

Exercise 2 Solution

You should use a Java statement similar to the following:

```
String strBook = new String("The Open Space of Democracy");
```

You can also use a Java statement similar to the following:

```
String strBook = "The Open Space of Democracy";
```

Exercise 3 Solution

You should use a Java statement similar to the following:

```
Class.forName("com.mysql.jdbc.Driver").newInstance();  
conn = DriverManager.getConnection("jdbc:mysql://localhost/books", "usr1", "pw1");
```

Exercise 4 Solution

You should use a Java statement similar to the following:

```
ResultSet rs = stmt.executeQuery(selectSql);
```

Exercise 5 Solution

You should use a Java statement similar to the following:

```
while(rs.next())  
{  
    String cdName = rs.getString("CDName");  
    int inStock = rs.getInt("InStock");  
    System.out.println(cdName + ", " + inStock);  
}
```

Exercise 6 Solution

You should use a Java statement similar to the following:

```
int intValue = Integer.valueOf("530").intValue();
```

Exercise 7 Solution

You should use a Java statement similar to the following:

```
PreparedStatement ps = conn.prepareStatement(insertSql);
```

Chapter 19

Exercise 1 Solution

You should use a C# statement similar to the following:

```
<%@ Import Namespace="System.Data.Odbc" %>
```

Exercise 2 Solution

You should use C# statements similar to the following:

```
OdbcConnection odbcConn = new OdbcConnection(strConn);
```

Exercise 3 Solution

You should use C# statements similar to the following:

```
odbcConn.Open();
```

Exercise 4 Solution

You should use C# statements similar to the following:

```
OdbcCommand odbcComm = new OdbcCommand(selectSql, odbcConn);
```

Exercise 5 Solution

You should use C# statements similar to the following:

```
OdbcDataReader odbcReader = odbcComm.ExecuteReader();
```

Exercise 6 Solution

You should use a C# statement similar to the following:

```
<!-- #Include File="change.aspx" -->
```

Exercise 7 Solution

You should use a C# statement similar to the following:

```
Response.Redirect("new.aspx");
```