

PART 1: SQL Query

Abc

PART 6: DO ALL EXERCISE AT THE END OF EACH CHAPTER**PART 2: Chapter 5 – Manipulating Data**

1. The film club chairperson wants a list of members' names and addresses. She needs only the street and zip code. If either the Street column or the ZipCode column is NULL, make sure that the address is listed as No address details available for this member.

Coalesce() function must be used to prevent null values from being displayed. In the coalesce function the first value is the column name; the second is the message to be displayed if the column for that record contains null values. Below is a table with random information that uses these methods to display the message where data is null:

```

30 • insert into memberdetails(FirstName, LastName, Street, ZipCode) values
31 ('Micah', 'Lombard', '1032 Commonwealth Avenue', '14532'),
32 ('Kyle', 'Davies', '67 Main Street', '57323'),
33 ('James', 'Smith', '1032 Commonwealth Avenue', '86745'),
34 ('John', 'Lodder', '1032 Commonwealth Avenue', '78523'),
35 ('Katie', 'McKinnon', '1032 Commonwealth Avenue', '02564');
36
37 • select FirstName, LastName,
38 coalesce(Street, 'No address details available for this member'),
39 coalesce(ZipCode, '')
40 from memberdetails;
41
42 • select * from memberdetails;

```

	FirstName	LastName	Street	ZipCode
	James	Smith	1032 Commonwealth Avenue	32767
	John	Lodder	1032 Commonwealth Avenue	32767
	Katie	McKinnon	1032 Commonwealth Avenue	2564
	Kyle	Davies	67 Main Street	32767
	Micah	Lombard	1032 Commonwealth Avenue	14532
	Samantha	Stevenson	No address details available for this member	
	NULL	NULL	NULL	NULL

2. *This question is for MS SQL Server and IBM DB2 users. After speaking to a member at a club meeting last night, the film club chairperson wants to contact the member again but can't quite remember her name. She thinks it sounds something like Jilly Johns. Can you help track down the person using a SQL query?*

The following Query uses the Difference() function to match names that sound like *Jon Lofer*:

FirstName	LastName	Street	ZipCode
John	Lodder	1032 Commonwealth Avenue	32767

```

30 • insert into memberdetails(FirstName, LastName, Street, ZipCode) values
31 ('Micah', 'Lombard', '1032 Commonwealth Avenue', '14532'),
32 ('Kyle', 'Davies', '67 Main Street', '57323'),
33 ('James', 'Smith', '1032 Commonwealth Avenue', '86745'),
34 ('John', 'Lodder', '1032 Commonwealth Avenue', '78523'),
35 ('Katie', 'McKinnon', '1032 Commonwealth Avenue', '02564');
36
37 • select FirstName, LastName
38 from memberdetails
39 where difference(LastName, 'Lofer') >= 5 and difference(FirstName, 'Jon') >= 3;
40

```

PART 3: Chapter 6 – Manipulating Data in MySQL Database

1. `INSERT INTO `books` (BookID, BookName) values ('1001', 'One Hundred Years of Solitude');`
2. `SELECT REPLACE(BookName, 'One Hundred Years of Solitude', 'One Hundred Years of Solitude')
from books;`
3. `UPDATE cds SET CDQuantity=CDQuantity + 3;`
4. `UPDATE cds set CDQuantity='CDQuantity+3' where CDID=1;`
5. `DELETE FROM cds WHERE CDID=1;`

PART 4: Chapter 10 – Accessing Data in Multiple Tables

1. `SELECT BookName, CONCAT_WS(' ', AuthFirstName, AuthLN) As Author
FROM Books AS b, AuthorBook AS ab, Authors AS a
WHERE b.BookID=ab.BookID AND ab.AuthorID=a.AuthorID
ORDER BY BookName;`
2. `SELECT BookName, CONCAT_WS(' ', AuthFirstName, AuthLastName) As Author
FROM Books AS b CROSS JOIN AuthorBook AS ab ON b.BookID=ab.BookID
CROSS JOIN Authors AS a ON ab.AuthorID=a.AuthorID
WHERE AuthLastName='Toole' OR AuthLastName='Thompson'
ORDER BY BookName;`

3.

```
SELECT BookName, CONCAT_WS(' ', AuthFirstName, AuthLastName) As Author
FROM Books AS b RIGHT JOIN AuthorBook AS ab ON b.BookID=ab.BookID
RIGHT JOIN Authors AS a ON ab.AuthorID=a.AuthorID
ORDER BY BookName;
```
4.

```
SELECT BookName, CONCAT_WS(' ', AuthFirstName, AuthLastName) As Author
FROM Books AS b NATURAL RIGHT JOIN AuthorBook AS abNatural
RIGHT JOIN Authors AS a
ORDER BY BookName;
```
5.

```
SELECT BookName FROM Books
WHERE BookID IN
(SELECT BookID FROM BookOrders WHERE Quantity>2)
ORDER BY BookName;
```
6.

```
SELECT OrderID, BookID, Quantity
FROM BookOrders
WHERE BookID= (SELECT BookID FROM Books
WHERE BookName='Letters to a Young Poet');
```
7.

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

PART 5: Chapter 9 – More on Keys and Constraints

1. *The ever-useful “customer orders product” example again—there is always something new to discover. Design the table that will represent the Order class in Figure 9-14. Consider constraints, primary and foreign keys, and updating rules*

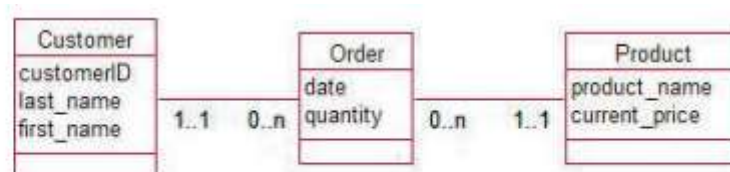


Figure 9-14. Model for customers placing orders

```
CREATE TABLE Order (order_num INT PRIMARY KEY,
product INT NOT NULL FOREIGN KEY REFERENCES Product,
customer INT NOT NULL FOREIGN KEY REFERENCES Customer,
quantity INT NOT NULL, date DATE NOT NULL,);
```

2. A car sales yard needs to keep information about makes and models of cars that are available, and also the registrations of the individual cars they have in stock. For example, Ford Siestas are available in sedans and hatchbacks, and they currently have a blue sedan with registration TC545 in stock. For some models you might be able to choose automatic or manual transmission and some come with different capacities (e.g., 1.5 l or 2.0 l). Think about the options available for setting up tables for this situation.

MAKE (manufactureName)

MODEL (modelName)

VERSION (versionID, trim, occupancy, transmission, price)

CAR (registration, color)

