## Show for all Sci Fi books, book title, year written, brief synopsis, sorted by title

 ${\tt SELECT\ Books. YearWritten,\ Books. synopsis,\ Categories. category Type}$ 

FROM Categories

**INNER JOIN** 

(Books INNER JOIN BooksCategories ON Books.ID = BooksCategories.bookId\_FK)

ON Categories.ID = BooksCategories.catId\_FK

WHERE Categories.categoryType = 'Science Fiction'

ORDER BY title:

### 

## Show all categories for which there are books

SELECT Categories.categoryType, COUNT(\*) AS numberOfTitles

**FROM Categories** 

**INNER JOIN** 

(Books INNER JOIN BooksCategories ON Books.ID = BooksCategories.bookId\_FK)

ON Categories.ID = BooksCategories.catId FK

GROUP BY Categories.categoryType;

#### 

# Show for all mysteries, title and average rating

SELECT Books.title, ROUND(Avg(UserReviews.rating), 1) AS averageRating

**FROM** 

(Categories INNER JOIN (Books INNER JOIN BooksCategories ON Books.ID = BooksCategories.bookId FK)

ON Categories.ID = BooksCategories.catId FK)

**INNER JOIN UserReviews** 

ON Books.ID = UserReviews.bookld FK

WHERE (((Categories.categoryType)='Mystery'))

GROUP BY Books.title;

## 

Show for each author, first name, last name and average rating

SELECT Author.firstName, Author.lastName, ROUND(Avg([rating]), 1) AS AverageRating

FROM

(Books INNER JOIN

(Author INNER JOIN AuthorsOfBooks

ON Author.ID = AuthorsOfBooks.authorld FK)

ON Books.ID = AuthorsOfBooks.bookId FK)

LEFT JOIN UserReviews

ON Books.ID = UserReviews.bookId\_FK

GROUP BY firstName, lastName;

Create a statement to add to user table, the user named: Mike Zellers, Userid: mzellers, password: password

INSERT INTO Users (firstName, lastName, ID, [password])

VALUES ('Mike', 'Zellers', 'mzellers', 'password');

Write a SQL statement to delete all mysteries and everything about them.

DELETE DISTINCTROW Books.\*

FROM Books

INNER JOIN (Categories INNER JOIN BooksCategories

ON Categories.ID = BooksCategories.catId FK)

ON Books.ID = BooksCategories.bookld FK

WHERE Categories.categoryType = 'Mystery';

I wasn't entirely sure of which SQL syntax we are using but the above statement was developed in MS Access. The keyword DISTINCTROW may not be valid in other dialects.

## Is there anything that would keep this from working?

This would not work as intended (deleting companion records through cascading deletes) if the referential integrity was not properly set. This statement WILL NOT delete the corresponding Author, Category, and User Entries from the other side of the many-to-many tables.

Write a SQL statement to update author Mark Twain with his correct birth and death date.

**UPDATE** Author

SET dateOfBirth = '11/30/1835', dateOfDeath = '4/21/1910'

WHERE ID = 10

## What if there were two authors named Mark Twain?

The table of authors was designed to use an arbitrary ID number as the primary key. The above statement will select only the Mark Twain that has ID# 10. Although this would require the user to know the arbitrary ID#, an Application could be built to allow the user to pinpoint the Author they would like to update in a more intuitive fashion.