

CSCI 5020 Assignment 4

Part 1

Chapter 11 Exercises 1, 2, 3, 4, and 5 on page 365.

Chapter 12 Exercises 1, 2, and 3 on page 384.

Chapter 13 Exercises 1, 3, and 5 on page 408.

The solutions to the chapter exercises in the book are provided through the student download and posted in D2L as well. For learning purpose, make sure you complete each exercise before checking the solution. **You do not need to turn in this part for grading.**

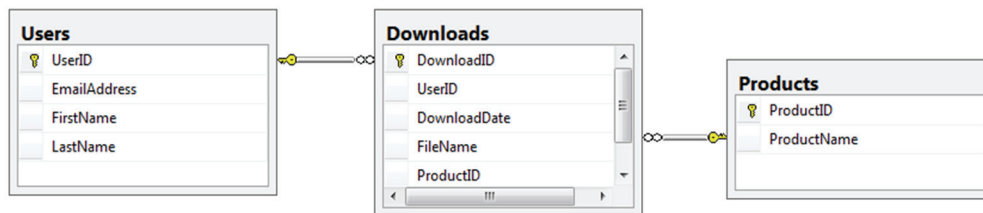
Part 2

Chapter 10 How to design a database

1. Create a database diagram that shows the relationships between the seven tables in the MyGuitarShop database. (The administrators table is not related to the other six tables.)

Chapter 11 How to create and maintain databases, tables, and sequences with SQL statements

1. Write a script that adds an index to the MyGuitarShop database for the zip code field in the Addresses table.
2. Write a script that implements the following design in a database named MyWebDB:



In the Downloads table, the UserID and ProductID columns are the foreign keys.

Include a statement to drop the database if it already exists.

Include statements to create and select the database.

Include any indexes that you think are necessary.

3. Write a script that adds rows to the database that you created in exercise 2.

Add two rows to the Users and Products tables.

Add three rows to the Downloads table: one row for user 1 and product 2; one for user 2 and product 1; and one for user 2 and product 2. Use the GETDATE function to insert the current date and time into the DownloadDate column.

Write a SELECT statement that joins the three tables and retrieves the data from these tables like this:

	email_address	first_name	last_name	download_date	filename	product_name
▶	johnsmith@gmail.com	John	Smith	2012-04-24 16:15:38	pedals_are_falling.mp3	Local Music Vol 1
	janedoe@yahoo.com	Jane	Doe	2012-04-24 16:15:38	tum_signal.mp3	Local Music Vol 1
	janedoe@yahoo.com	Jane	Doe	2012-04-24 16:15:38	one_horse_town.mp3	Local Music Vol 2

Sort the results by the email address in descending order and the product name in ascending order.

- Write an ALTER TABLE statement that adds two new columns to the Products table created in exercise 2.

Add one column for product price that provides for three digits to the left of the decimal point and two to the right. This column should have a default value of 9.99.

Add one column for the date and time that the product was added to the database.

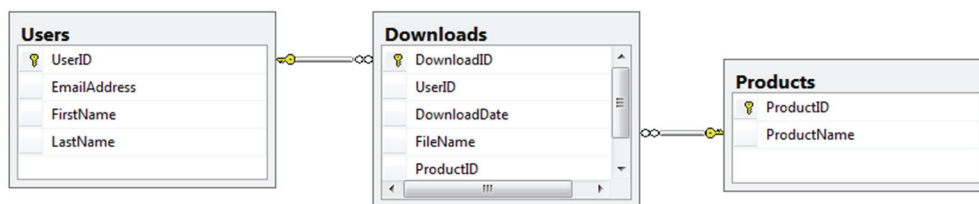
- Write an ALTER TABLE statement that modifies the Users table created in exercise 2 so the FirstName column cannot store null values and can store a maximum of 20 characters.

Code an UPDATE statement that attempts to insert a null value into this column. It should fail due to the not null constraint.

Code another UPDATE statement that attempts to insert a first name that's longer than 20 characters. It should fail due to the length of the column.

Chapter 12 How to use the Management Studio for database design

- Use the Management Studio to create a new database called MyWebDB using the default settings. (If the database already exists, use the Management Studio to delete it and then recreate it.)
- Use the Management Studio to create the following tables and relationships.



- Define the UserID column in the Users table, the ProductsID column in the Products table, and the DownloadID column in the Downloads table as primary keys and identity columns.
- In the Downloads table, set the UserID and ProductID column as the foreign keys.
- Define the columns so none of them allow null values.
- Use the Management Studio to create indexes for the foreign keys in the Downloads table.
- Use the Management Studio to create a unique index on the EmailAddress and ProductName columns and a regular index on the DownloadDate column.

Chapter 13 How to work with views

1. Create a view named CustomerAddresses that shows the shipping and billing addresses for each customer in the MyGuitarShop database.

This view should return these columns from the Customers table: CustomerID, EmailAddress, LastName and FirstName.

This view should return these columns from the Addresses table: BillLine1, BillLine2, BillCity, BillState, BillZip, ShipLine1, ShipLine2, ShipCity, ShipState, and ShipZip.

Use the BillingAddressID and ShippingAddressID columns in the Customers table to determine which addresses are billing addresses and which are shipping addresses.

HINT: You can use two JOIN clauses to join the Addresses table to Customers table twice (once for each type of address).

2. Write a SELECT statement that returns these columns from the CustomerAddresses view that you created in exercise 1: CustomerID, LastName, FirstName, BillLine1.
3. Write an UPDATE statement that updates the CustomerAddresses view you created in exercise 1 so it sets the first line of the shipping address to “1990 Westwood Blvd.” for the customer with an ID of 8.
4. Create a view named OrderItemProducts that returns columns from the Orders, OrderItems, and Products tables.

This view should return these columns from the Orders table: OrderID, OrderDate, TaxAmount, and ShipDate.

This view should return these columns from the OrderItems table: ItemPrice, DiscountAmount, FinalPrice (the discount amount subtracted from the item price), Quantity, and ItemTotal (the calculated total for the item).

This view should return the ProductName column from the Products table.

5. Create a view named ProductSummary that uses the view you created in exercise 4. This view should return some summary information about each product.

Each row should include these columns: ProductName, OrderCount (the number of times the product has been ordered), and OrderTotal (the total sales for the product).
6. Write a SELECT statement that uses the view that you created in exercise 5 to get total sales for the five best selling products.

Turn in screenshot for Chapter 10 and 12. Write each statement in separate script files (*.sql) for Chapter 11 and 13. **Zip all the files into a single file and submit it to the Dropbox.**