

*DB2 9 Database  
Administration Workshop for  
Windows*

(Course code CF23)

**Instructor Exercises Guide**  
**with hints**

ERC 8.3

IBM certified course material

## Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	AIX 5L	Approach
CICS	ClearCase	DB2
DB2 Connect	DB2 Universal Database	Distributed Relational Database Architecture
DRDA	Encina	Enterprise Storage Server
Informix	iSeries	Lotus
MQSeries	MVS	MVS/ESA
OS/2	OS/390	OS/400
pSeries	pureXML	QMF
QBIC	QuickPlace	RAMAC
S/390	Tivoli	z/OS
zSeries	1-2-3	

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

## April 2007 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright International Business Machines Corporation 1999, 2007. All rights reserved.

**This document may not be reproduced in whole or in part without the prior written permission of IBM.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Trademarks</b> .....	<b>v</b>
<b>Instructor exercises overview</b> .....	<b>vii</b>
<b>Exercises configuration</b> .....	<b>ix</b>
<b>Exercises description</b> .....	<b>xi</b>
<b>Exercise 1. Starting your lab environment</b> .....	<b>1-1</b>
<b>Exercise 2. DB2 customization</b> .....	<b>2-1</b>
Section 1 - Exploring the CLP usage .....	2-3
Section 2 - Looking at Release Notes .....	2-5
<b>Exercise 3. Create an instance and explore the environment</b> .....	<b>3-1</b>
<b>Exercise 4. Creating databases and data placement</b> .....	<b>4-1</b>
Section 1 - Creating the Database .....	4-3
Section 2 - Creating Table Spaces .....	4-12
<b>Exercise 5. Create objects</b> .....	<b>5-1</b>
Section 1 - Create Tables .....	5-3
Section 2 - Create Indexes .....	5-13
Section 3 - Create Views .....	5-16
Section 4 - Create Alias .....	5-18
Section 5 - Add Referential Integrity .....	5-20
Section 6 - Add Check Constraints .....	5-24
Section 7 - Create a Trigger .....	5-26
Section 8 - OPTIONAL - Working with XML .....	5-28
<b>Exercise 6. Moving data</b> .....	<b>6-1</b>
Section 1 - Import Data in Tables .....	6-3
Section 2 - Create Exception Tables - ARTISTS, ALBUMS, STOCK .....	6-7
Section 3 - Backup Database .....	6-7
Section 4 - Load Data with INSERT option into CONCERTS .....	6-9
Section 5 - Load Data with REPLACE option into ARTISTS .....	6-11
Section 6 - Examining the SET INTEGRITY PENDING State .....	6-13
Section 7 - Understanding CHECK CONSTRAINT Enforcement .....	6-18
Section 8 - Understanding TRIGGER Enforcement .....	6-20
Section 9 - Using the db2look tool .....	6-22
Section 10 - OPTIONAL: Working with LOB Data .....	6-23
<b>Exercise 7. Backup and recovery</b> .....	<b>7-1</b>

Section 1 - Determining Log Configurations . . . . .	7-3
Section 2 - Backup/Restore Support with Circular Logging . . . . .	7-10
Section 3 - Backup/Restore Support with Archive Logging . . . . .	7-20
Section 4 - Resetting the Environment . . . . .	7-29
<b>Exercise 8. Investigating DB2 locking . . . . .</b>	<b>8-1</b>
Section 0 - Setup . . . . .	8-3
Section 1 - Basic Lock Monitoring . . . . .	8-3
Section 2 - Lock Timeouts . . . . .	8-7
Section 3 - Deadlocks . . . . .	8-12
Section 4 - Resetting the Environment . . . . .	8-15
<b>Exercise 9. Problem determination . . . . .</b>	<b>9-1</b>
Section 1 - Collecting information . . . . .	9-3
Section 2 - Miscellaneous . . . . .	9-7
Section 3 - Reset DIAGLEVEL . . . . .	9-10
<b>Exercise 10. Application performance tools . . . . .</b>	<b>10-1</b>
Section 1 - Access strategies and utilities . . . . .	10-3
Section 2 - Program samples . . . . .	10-11
<b>Exercise 11. Security . . . . .</b>	<b>11-1</b>
Section 1 - Default PUBLIC Privileges . . . . .	11-3
Section 2 - DB2 Authorities . . . . .	11-10
Section 3 - DB2 User and Group Privileges . . . . .	11-22
Section 4 - Resetting the Environment . . . . .	11-28
<b>Appendix A. UNIX vi Editor command summary . . . . .</b>	<b>A-1</b>
<b>Appendix B. SAMPLE database . . . . .</b>	<b>B-1</b>

# Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®	AIX 5L™	Approach®
CICS®	ClearCase®	DB2®
DB2 Connect™	DB2 Universal Database™	Distributed Relational Database Architecture™
DRDA®	Encina®	Enterprise Storage Server®
Informix®	iSeries™	Lotus®
MQSeries®	MVS™	MVS/ESA™
OS/2®	OS/390®	OS/400®
pSeries®	pureXML™	QMF™
QBIC®	QuickPlace®	RAMAC®
S/390®	Tivoli®	z/OS®
zSeries®	1-2-3®	

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



# Instructor exercises overview

The objectives of these exercises are to have the students create an instance and a database, and learn to perform administrative tasks on the database.

The instance must be created before the database can be created, and the database must be created and loaded prior to doing the rest of the exercises. The remaining exercises can be completed in almost any sequence.





# Exercises configuration

The labs are designed so all the tasks can be performed by using the student workstation as a DB2 GUI front end to the database server or using the DB2 Command Window. The database server resides on the Windows system.



## Exercises description

The students will learn how to administer a database server using either DB2's GUI interface or a DB2 Command Window.

**Table 0-1: Lab Guide Substitutions**

Description	Your Value	Substitution in Lab Guide
Administrator's Password		ibm2blue
Server hostname		IBMCLASS
Administrative Server		db2admin
Admin Server Password		ibm2blue
Windows User Name		student
Windows Password		ibm2blue
DB2 Drive Letter		C

### DO NOT CHANGE ANY OF YOUR PASSWORDS!

Each exercise in this course is divided into sections as described below.

**Objectives** — This section outlines the learning points of the lab.

**Reference Materials** — List of documents which you may find useful in performing the Lab.

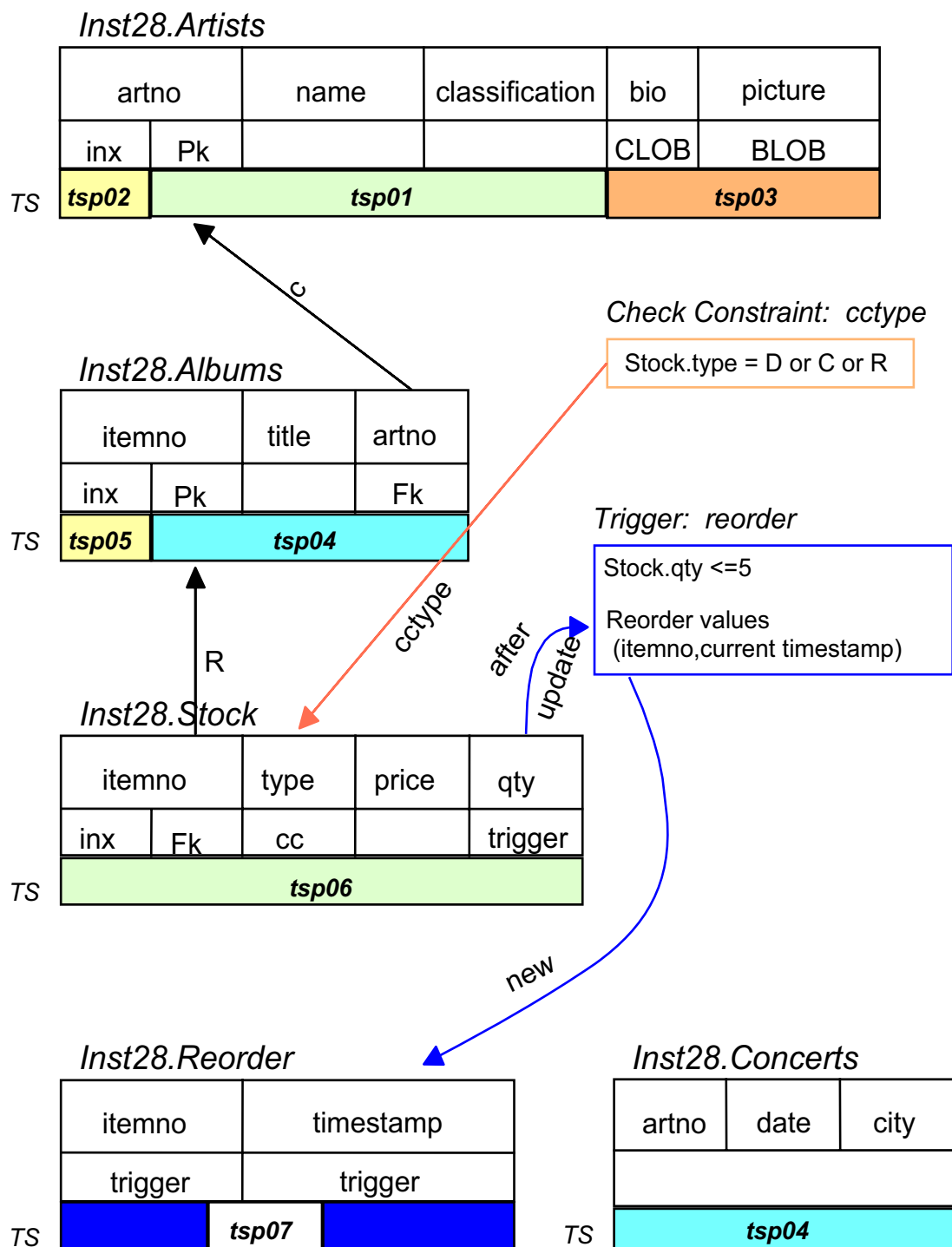
**Lab Exercise** — The step-by-step guide to the lab.

The exercise instructions include hints to help you perform the tasks requested or to provide answers to the questions posed in the exercise. In some cases, there are different solutions depending on the server platform you are using. In other cases, there are different solutions depending on whether you are using a local administration option through the telnet session, or using a remote administration option using the GUI workstation tools.

**Optional Exercises** — This section gives you additional exercises to perform relating to the unit of discussion. It is strictly optional and should be performed when you have completed the required exercises. The required exercises pertain to the most pertinent information provided in the unit. This section may help round out the hands-on experience for a related unit.

## Lab environment

# MUSICDB: Data Model and Objects



# Exercise 1. Starting your lab environment

*(with hints)*

## Estimated time

00:10

## What this exercise is about

This exercise covers the use of VMware Player. Your lab environment is a virtual machine image, used with the VMware Player product.

## What you should be able to do

At the end of the exercise, you should be able to:

- Find your DB2 VMware image
- Start your DB2 VMware image using VMware Player

## Introduction

You will use a VMware image to run labs within. There are several images available; you will need to start the image of your choice depending on the operating system you wish to run in. DB2 VMware lab images available are:

- If using DB2 server on Windows, you will use the Windows 2000 Pro image, in a single user environment.

## ***Instructor Notes:***

*Introduction* — In this lab the students will start their respective VMware images to be used for all future labs.

When using the provided Enterprise Linux (or open-source Linux) images, you should explain the relationship between the host system (generally Windows) and the VMware Linux image and the way to connect from the host Windows system to the Linux system.

The following topics should be discussed:

- How to understand a virtual machines (VM), using some logical-physical comparisons:
  - Virtual disk = A Windows file (one or more .vmdk files).
  - Virtual memory = A block of memory allocated from Windows to the VMware Player (the .nvram files are the BIOS and any overlayed memory).
  - Virtual devices (CD-ROM, Network Adapter, other) = Physical devices under the management of the VM.
  - VM configuration (.vmx file, a text file that contains the configuration of the VM).
- How to use VMware Player (used to run the image in the classroom) and/or VMware Workstation (used to create the image in the first place).
- How to make the virtual machine run with the full screen.
- How to escape your mouse from the VM to return to the host screen (Ctrl+Alt).
- How to boot the VM (Ctrl+Alt+Ins).

The same password — *ibm2blue* — is used for all logins (not a good practice on a production system, but very convenient in the classroom).

note that the VMware images have password protection to lock in the resources used by VMware. The password for this is — *ibm2blue*.

*Time for Lab* — 10 minutes.

*Things to Review at End of Lab* — Ensure each student has their VMware image running, and that they can navigate around their OS.

## Exercise instructions with hints

### Starting the VMware image

The VMware image you will be using is a Virtual Machine image of an operating system, run within the native Windows operating system of your computer.

- \_\_\_ 1. Logon to your student workstation with the username and password supplied by your instructor.  
\_\_\_\_\_
- \_\_\_ 2. Start a Linux or Windows VMware image on your computer.
  - \_\_\_ a. On your local computer, find the VMware Player icon and start it.
  - \_\_\_ b. The VMware Player window opens, which is a browser. Navigate on the browser to the location of your VMware image (your instructor will inform you of the location).  
\_\_\_\_\_
  - \_\_\_ c. Select the file with the **.vmx** extension, whose icon looks like this:



- \_\_\_ d. Click the **Open** button.
- \_\_\_ 3. This starts your Virtual Machine image. The image has preset parameters which will ensure that the native OS will not be overtaxed for resources.
- \_\_\_ 4. Logon to the system within your VM image. Your username and password for the local Linux or Windows system is:
  - **student**
  - **ibm2blue**
- \_\_\_ 5. Become familiar with the method of file handling and navigation for the system you are using.
  - \_\_\_ a. **Windows:** Use Windows Explorer (see above) to look at folders in your VMware image.
- \_\_\_ 6. Leave your Virtual Machine image running while we continue the class.
- \_\_\_ 7. Maximize your VMware image by clicking the Maximize icon on the upper right of your image window.

### END OF LAB





## Exercise 2. DB2 customization

*(with hints)*

### Estimated time

00:55

### What this exercise is about

This exercise will let you explore the CLP and DB2 GUIs, making you familiar with them using the DB2 default instance - DB2 - and the SAMPLE database.

### What you should be able to do

At the end of the lab, students should be able to:

- Use the CLP
- Use the different GUIs
- Know the possibilities of the CLP and the GUIs

### Introduction

Before class started, your instructor installed the DB2 product on the Database server that you are using. An Database Administration Server (DAS) was also created and started. Also the default DB2 instance with the SAMPLE database was created. You will now be able to familiarize yourself with the CLP and the GUIs.

The general approach taken in the classroom for DB2 9 is to use a VMware Image running on the same PC that you log into.

### *Instructor Notes:*

- *Introduction* — In this lab, the students have the chance to familiarize themselves with the CLP and the GUIs used by DB2. From their Windows systems, the students will define the UNIX server and the default instance through the DB2 Control Center.
- If using the AIX backend, please start the DAS.

*Time for Lab* — 55 minutes.

*Things to Review at End of Lab*

- Creating the instance must be performed by the Linux/UNIX root user.
- Each instance name is a Linux/UNIX user name.

- There can only be one active Administration Server on the database server.

From the remote administration system, both the Administration Server and the database instance must be defined.

## Exercise instructions with hints

The lab solutions using primarily commands are indicated as *Command Window*. The lab solutions using primarily the graphical tools are indicated as *GUI*.

### Section 1 - Exploring the CLP usage

**Note:** For Windows, please invoke a command window by navigating **Start > Programs > IBM DB2 > DB2COPY1 > Command Line Tools > Command Window**. This ensures your login ID has the appropriate DB2 environment set up (do NOT use Run > cmd).

Because you will use the default DB2 instance and the sample database, you must logon with the appropriate userid and password to be able to perform all operations.

- \_\_\_ 1. If using a Windows system, and you are not already logged on, log on now using the DB2 Administrator name. Use your Windows username and Windows password.

Hold down the Ctrl and Alt keys and press Delete to logon

Enter **student** as the Username

Enter **ibm2blue** as the Password

(Your login and password may differ, depending on the classroom environment and setup. The instructor will give you the username and password if these are different in your classroom.)

- \_\_\_ 2. First, try to get general information about commands that can be used in the CLP.

`db2 ?`

You will get a list with all possible commands. Scroll through this list to see the possibilities.

- \_\_\_ 3. Now examine what can be done with the get command.

`db2 ? get`

---

**You will receive a complete list of all “get” commands with the relevant syntax.**

- \_\_\_ 4. Next let's assume you receive an SQLSTATE=08003. How can you determine what this sqlstate means? Execute the correct command and explain the meaning.

---

**You can issue `db2 ? 08003`**

**SQLSTATE 08003 The connection does not exist.**

- \_\_\_ 5. Next let's examine the options set for your CLP session. Issue the correct command and determine which options are set to on.
-

*db2 ? options*

**You should see the options c, o, p and w set to ON. All others are set off.**

- \_\_\_ 6. Next let's create a script with the name **myscript**, which connects to the sample database, selects the columns empno and lastname from the table employee and lists the database manager configuration. Depending on whether you are working on the Windows station or a Linux/UNIX machine use the appropriate tool (for example, Notepad or vi) and save the created script.

*notepad myscript\_nn.txt*

*or*

*vi myscript\_nn.txt*

Where *nn* is your team number.

Enter these lines in the file:

```
db2start;  
connect to sample;  
select empno, lastname from employee;  
get dbm cfg;
```

- \_\_\_ 7. Now execute the created script using your CLP session and echo the current command.

*db2 -tvf myscript\_nn.txt*

- \_\_\_ 8. As there is a lot of information returned from the select statement and the database manager configuration, it might be helpful to store the information in a file called **myout** to be able to examine it step by step. Execute the script again and save the output in a file named **myout**.

*db2 -tvf myscript\_nn.txt -z myout\_nn*

- \_\_\_ 9. You can now examine the file using Notepad or vi depending on the operating system you use.

## Section 2 - Looking at Release Notes

- \_\_\_ 1. Look at the files located in the Readme subdirectory. They are located in the DB2 installation directory, under `Readme/en_US.iso88591` (or similar directory). These files contain information about the DB2 product that did not make it into the manuals. (Do not edit these files.) You should always read these files. They may contain information that is not included in the manuals.

Windows	<p>Navigate to the Information Center: <b>Start &gt; Programs &gt; IBM DB2 &gt; DB2COPY1 &gt; Information &gt; Information Center V9.</b></p> <p>In the search field, enter <b>Release Notes</b> and click on the <b>GO</b> button.</p> <p>Select <b>About the release notes</b> entry on the left panel.</p> <p>Click on the URL for English (or other languages) manuals.</p> <p>View the Release Notes in XHTML or PDF form.</p>
---------	---

- \_\_\_ 2. You can also examine the other files under **sqllib**.
- \_\_\_ 3. Select the **+** left of the SAMPLE database to explore what is under the database. You will see folders like Tables, Views, Aliases, and so forth.
- \_\_\_ 4. Next click the folder named TABLES. In the *Contents* pane, all tables are shown.
- \_\_\_ 5. Select the **EMPLOYEE** table. What is displayed in the *Detail* pane?

**Schema, Creator and number of columns as well as column details. Possible actions are: Open, Query, Show related Objects and Create New Table.**

- \_\_\_ 6. Now click the **OPEN** tab in the *Detail* pane. What is the result?

**Sample contents (select \*) of the table are shown. You have also the possibility to add or delete rows.**

- \_\_\_ 7. Close this window and click the **QUERY** tab. What happens now?

**A Command Editor window opens, an automatic connect to the sample database is established and a select \* from employee statement is prepared**

**but not executed. You might execute the query using the green arrow on the top left corner.**

- \_\_\_ 8. Leave the select \* line and add a line with the following: select empno from employee; and execute this using the green arrow on the top left corner or by pressing Ctrl and Enter. Is the execution of two SQL statements successful and are the results displayed under the Query results tab?
- \_\_\_\_\_
- \_\_\_\_\_

**Yes, the execution is successful. The results are displayed in the query results tab, but not in the lower communication area of the Commands tab.**

- \_\_\_ 9. Change back to the Control Center using the appropriate button, and examine the different options by first selecting database and then right-clicking the database.
- \_\_\_ 10. Repeat this examination for other objects like instance, table, and so forth.

**END OF LAB**

## Exercise 3. Create an instance and explore the environment

*(with hints)*

This lab is only for Linux/UNIX students.





## Exercise 4. Creating databases and data placement

*(with hints)*

### Estimated time

01:00

### What this exercise is about

During this lab, you will create your database and your table spaces. Also in this lab, you will select information on table spaces from the system catalog (SYSCAT) views.

### What you should be able to do

At the end of the lab, students should be able to:

- Create a database
- Create a table space
- Execute a script file to create multiple table spaces
- Access the SYSCAT views containing table space information
- List table space information
- List container information

### Introduction

This lab will step you through the creation of your database and the table spaces that will be needed to house the tables in your database. You will have an option in this lab to perform the functions using one of:

- command window to local Linux or Windows (referred to as non-GUI)
- using the graphical tools on your Linux or Windows system (referred to as GUI)

## ***Instructor Notes:***

*Introduction* — During this lab, the student will create their MUSICDB database and their table spaces. Students will also select information on table spaces from the system catalog (SYSCAT) views.

*Time for Lab* — 1 hour.

*Things to Review at End of Lab* — At the end of this lab, make sure every team has created their database and table spaces, and that all the objects are spelled correctly to avoid problems in other labs. If they have made mistakes, they should drop the objects and re-create them correctly. Make sure the students are able to use the Control Center and Command Editor or a DB2 Command Window to accomplish these tasks.

*Common Mistakes Students May Make* — Specifying incorrect values when creating or customizing their database or tablespaces will impact later labs.

## Exercise instructions with hints

The lab solutions using primarily commands are indicated as *Command Window*. The lab solutions using primarily the graphical tools are indicated as *GUI*.

### Section 1 - Creating the Database

You will be looking at the create database command and determining what would be produced if you ran it. **DO NOT create the database until the lab specifically instructs you to do so.**

- \_\_\_ 1. Creating a database can be done with the DB2 command **create database**. Use the online help facility to display the DB2 command syntax for create database.

Command Window	db2 ? create db   more
GUI	<p>___ From the <i>DB2 Control Center</i>, open the <i>Command Editor</i> (either choose the Tools option on the menu bar and then select Command Editor, or choose the Command Editor icon on the tool bar (4th from the left)).</p> <p>___ From the <i>DB2 Command Editor</i>, in the <i>Commands</i> box enter:</p> <p style="padding-left: 40px;"><b>? create database</b></p> <p>___ Choose <i>Selected</i> from the menu bar.</p> <p>___ Select <i>Execute</i> from the pull-down menu.</p>

- \_\_\_ 2. What information can you specify on the create database command?

---

**The name of the database, the location of the database, an alias name, the codeset and territory for storing the data, a collating sequence, a default extent size, automatic storage, and table space information can be specified on the create database command.**

- \_\_\_ 3. Based on what you learned in lecture, what table spaces are created when a database is created?

---

**SYSCATSPACE, USERSPACE1, and TEMPSPACE1**

**Depending on what you specify when you create the database, there may be a fourth tablespace created with a name of SYSTOOLSPACE. This is created if you specify *With Automatic Maintenance* instead of *Standard* when you create the database.**

- \_\_\_ 4. What type of table space, SMS or DMS, will your database use by default for these default table spaces?

---

**DMS table spaces for SYSCATSPACE and USERSPACE1, and SMS for TEMPSPACE1. Note that in DB2 UDB V8.2 and earlier the default table spaces are SMS.**

- \_\_\_ 5. Before we create our MUSICDB database, remember what you learned in lecture about what some of the defaults will be.
- What is the default path that the database will be created on?
  - What will be the default table space type (SMS or DMS) for the table spaces that will be created to house User Tables, Catalog Tables, and Temporary Tables?
  - What are the default Extent and Prefetch sizes?
  - What are the default Territory, Code Set and Collating Sequence values?

- 
- **The database will be created on the C:\DB2 path by default.**
  - **The default table space type for User Tables and Catalog Tables is DMS, and for Temporary Tables is SMS.**
  - **The default Extent and Prefetch sizes are 32 4KB pages each.**
  - **The default Territory and Code Set is dependent on your local system settings.**

- \_\_\_ 6. Create your database with a Database name of MUSICDB using the default settings. It will take a few minutes to create the database.

**Note:** The creation of the database may take several minutes. It might be a good time to take a break, while it is being created.

Windows	From your Command Window session, run the following command: <code>db2 create db musicdb</code>
---------	--

GUI	<p>From the <i>Control Center</i>, click the + next to your <b>DB2</b> instance, and then right-click the <i>Databases</i> folder and select <b>Create Database -&gt; Standard</b>.</p> <p>___ On the <i>Name</i> page, enter a database name of <b>MUSICDB</b>. The default is to let DB2 manage the storage, so make sure that radio button is set.</p> <p>___ On the <i>Region</i> page, make sure <b>Country/Region</b> is set to default. (If your instructor has told you to use a different language, please verify the change with your instructor.)</p> <p>___ Accept the values for <i>Territory</i> and <i>Code set</i>.</p> <p>___ Select the <b>System Collating Sequence</b> radio button.</p> <p>___ On the <i>Summary</i> page, click <b>Show Command</b>. Look at the generated command and <i>Close</i> the Show Command window.</p> <p>___ Click the <b>Finish</b> button.</p> <p>___ Wait a couple of minutes while your database is being created. You may receive an information message that you could launch the <i>Configuration Advisor</i> to tune your database. Do not use the <i>Configuration Advisor</i> at this time. Click the <b>No</b> button.</p> <p>___ Your database will appear under the <i>Databases</i> folder in the left pane of your <i>DB2 Control Center</i> panel. Select the + next to the <i>Database</i> folder to see your MUSICDB database listed.</p> <p><b>Note:</b> The first request that you make from the client to access information in the database (for example, if you look at the list of tables via the <i>DB2 Control Center</i>) will automatically bind a number of packages to support the client utilities. This may take a few moments; please be patient.</p>
-----	---

**Instructor Note:** In V8.2, it does not appear that the package bind occurs at first access; it may occur during database creation. It appears that the same happens with V9.1.

- \_\_\_ 7. You should now have your MUSICDB database created. The System Database Directory contains an entry for all databases known by this instance. Check the System Database Directory for an entry for the MUSICDB database by issuing the `list db directory` command.

Command Window	<code>db2 list db directory</code>
GUI	<p>___ From the <i>DB2 Command Editor</i>, enter:</p> <p style="padding-left: 40px;"><code>list db directory</code></p> <p>___ Press <b>Ctrl-Enter</b> to cause the command to be executed.</p>

\_\_\_ 8. What is the Database alias name and where did it come from?

---

**The Database alias name is MUSICDB. It defaulted to the database name since no alias was specified when the database was created.**

\_\_\_ 9. What does a Directory entry type of Indirect mean?

---

**A Directory entry type of Indirect means the database is located on this system, and that the local database directory can be found here. You will see this if you issue the list database directory command from your telnet window.**

\_\_\_ 10. What does a Directory entry type of Remote mean?

---

**A Directory entry type of Remote means the database is located on another system. You will see this if you issue the list database directory command from the Windows client.**

\_\_\_ 11. It's time to connect to your MUSICDB database. Check your current connection state with the `get connection state` command.

Command Window	<code>db2 get connection state</code>
GUI	From the <i>DB2 Command Editor</i> , enter: <code>get connection state</code> Select the green arrow icon under the <i>Commands</i> tab to execute the command.

\_\_\_ 12. What is the connection state?

---

**The connection state is *Connectable* and *Unconnected*.**

\_\_\_ 13. Connect to your MUSICDB database.

Command Window	<code>db2 connect to musicdb</code>
GUI	From the <i>DB2 Command Editor</i> , enter and execute: <code>connect to musicdb</code>

\_\_\_ 14. Did you get connected to your MUSICDB database?

**Yes, you should have gotten connected successfully. If you are using the Command Editor, notice that the name MUSICDB now appears in the Target box. If it does not appear in the Target box:**

**Click the Add button to the right of the Target box.**

**Select MUSICDB in the Available targets box.**

**Enter your instnn User ID and Password.**

**Click OK.**

**MUSICDB should now appear in the Target box.**

**If it still does not, select Command Editor from the menu bar, and select Shut Down DB2 Tools. Then restart your Control Center and Command Editor sessions again, and try reconnecting.**

\_\_\_ 15. Check your connection state again. What does it show?

Command Window	<code>db2 get connection state</code>
GUI	<p>From the <i>DB2 Command Editor</i>, delete the last statement you entered in the <i>Commands</i> box.</p> <p>Choose <i>Selected</i> from the menu bar and then select <i>History</i>.</p> <p>Select <i>get connection state</i> in the <i>Commands</i> box, and click the <b>Paste</b> button.</p> <p>Execute the command: <code>get connection state</code></p>

**The Database Connection State shows Connectable and Connected.**

\_\_\_ 16. Every database has its own Database Configuration file that contains information about the database and tuning parameters. Look at the Database Configuration file for your MUSICDB database.

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI	<p>___ Switch to your <i>DB2 Control Center</i> session.</p> <p>___ Select <i>MUSICDB</i> in the left pane under the <i>Databases</i> folder.</p> <p>___ Right-click and select <i>Configure Parameters</i> from the pop-up menu.</p> <p>___ Scroll through the list to see the parameters that can be changed.</p> <p>___ Click the <b>Cancel</b> button when finished.</p>

- \_\_\_ 17. Find the default values for two specific configuration parameters for your database, LOCKLIST and MAXLOCKS. Since there are a large number of configuration parameters, we can use *grep* to find the specific ones we want when using a local connection (the equivalent for DB2 on Windows is: ... | find /i "lock") — the option "i" means *case insensitive*.
- 

Command Window	db2 get db cfg for musicdb   find /i "lock"
GUI	Switch to the <i>DB2 Control Center</i> session looking for these two specific configuration parameters and their values: LOCKLIST and MAXLOCKS.

- \_\_\_ 18. Values of some of the parameters can be changed. Update the following parameters and specify the values shown.
- Change maxlocks to 20
  - Change num\_freqvalues to 12

Command Window	db2 update db cfg for musicdb using maxlocks 20 num_freqvalues 12
GUI	From the <i>Command Editor</i> , enter and execute: update db cfg for musicdb using maxlocks 20 num_freqvalues 12

- \_\_\_ 19. When do these database configuration file changes take effect?
- 

**For these configuration parameters, the change takes effect immediately.**

**Note that when you changed the value for MAXLOCKS, the default value for LOCKLIST is also changed — from automatic to an appropriate manual value.**

- \_\_\_ 20. Check to see if the *Current* and *Delayed Values* are the same for **maxlocks** and **num\_freqvalues**.

Command Window	Issue the following command from a telnet window: db2 get db cfg for musicdb show detail   more Scroll through the output.
----------------	--



GUI	Issue the following command from <i>Command Editor</i> : <code>get db cfg for musicdb show detail</code> Scroll through the output. (You may need to scroll to right to see both Current and Delayed Values.)
-----	---

\_\_\_ 21. Repeat Step #17 above, and record here the new values of LOCKLIST and MAXLOCKS.

\_\_\_ 22. Some default table spaces were created during creation of the database. List the table space information.

Command Window	<code>db2 list tablespaces   more</code>
GUI	From the <i>Command Editor</i> : <code>list tablespaces</code>

\_\_\_ 23. What are the table space names and what ID number is associated with the table space?

**SYSCATSPACE — ID 0**

**TEMPSPACE1 — ID 1**

**USERSPACE1 — ID 2**

**SYSTOOLSPACE — ID 3**

Since you queried the database size info, the fourth table space named SYSTOOLSPACE was created for you.

There are four cases that SYSTOOLSPACE will be automatically created on an active database starting with V8.2:

1. DB Summary View of the database is displayed in the Control Center, or similar information is displayed from the command line by issuing:  
`db2 "CALL GET_DBSIZE_INFO(?, ?, ?, -1)"`
2. Create a database with automatic maintenance.
3. Turn on automatic maintenance for a standard database (in the DB CFG file).
4. For a standard database without automatic maintenance, which has not been connected to through the Control Center, *hmon* (the health monitor) will create one when it starts evaluating health indicators (by default every 2 hrs).

Thus, eventually, a V9.1 database will have a minimum of four table spaces. The automatic statistics collection and reorganization features — available starting with DB2 UDB V8.2 — store working data in tables in your database. These tables are created in the SYSTOOLSPACE table space. The SYSTOOLSPACE table space is created automatically with default options. Storage requirements for these tables are proportional to the number of tables in the database and should be calculated as approximately 1 KB per table. If this is a significant size for your database, you may want to drop and re-create the table space yourself and allocate storage appropriately. The automatic maintenance and health monitor tables in the table space are automatically re-created. Any history captured in those tables is lost when the table space is dropped.

- \_\_\_ 24. Table space container information can be displayed with the `list tablespace containers` command. Use the Help facility to show the DB2 command syntax.

Command Window	<code>db2 "? list tablespace containers"</code>
GUI	<code>? list tablespace containers</code>

- \_\_\_ 25. Show the container information for table space ID 0. What type of container is this and where is it located?

Command Window	<code>db2 list tablespace containers for 0</code>
GUI	<code>list tablespace containers for 0</code>

There is one **file** container (DMS) associated with tablespace 0 and it is located at `C:\DB2\NODE0000\MUSICDB\T0000000\C0000000.CAT`

- \_\_\_ 26. List the names of the system catalog tables. What are these tables?

Command Window	<pre>db2 list tables for system   more -or- db2 "select substr(tabname,1,18) as tabname,       substr(definer,1,10) as definer,       case type when 'T' then 'T Table'             else 'Error' end as type from syscat.tables where type = 'T' and definer = 'SYSIBM' "   more</pre>
----------------	--

GUI	<pre>list tables for system -or- select tabname, definer,        case type when 'T' then 'T Table' else 'Error' end as type from syscat.tables where type = 'T' and definer = 'SYSIBM'</pre>
-----	--

**These are the tables that are stored in the SYSCATSPACE table space.**

**Note: *list tables for system* shows tables and views.**

- \_\_\_ 27. Get more detailed information for the table spaces and indicate which table spaces are set to automatic size increase.

Command Window	db2 get snapshot for tablespaces on musicdb
GUI	get snapshot for tablespaces on musicdb

**SYSCATSPACE, USERSPACE1, and SYSTOOLSPACE are set to automatic size increase, while TEMPSPACE1 is not.**

- \_\_\_ 28. Verify which default path containers are associated with the temporary table space and the default user table space.

Command Window	<pre>db2 list tablespaces db2 list tablespace containers for 1 db2 list tablespace containers for 2</pre>
GUI	<p>On the <i>Command Editor</i>, <i>Commands</i> tab, first make sure the Tool is configured for multiple commands.</p> <p>___ Select <i>Tools Settings</i> icon on the tool bar.</p> <p>___ Make sure the checkbox next to Use statement termination character is checked, and that the statement termination character specified is a semicolon.</p> <p>___ Close <i>Tools Setting</i> window.</p> <pre>list tablespaces; list tablespace containers for 1; list tablespace containers for 2</pre>

**C:\DB2\NODE0000\MUSICDB\T0000001\C0000000.TMP is the default file container for table space 1 (TEMPSPACE1).**

**C:\DB2\NODE0000\MUSICDB\T0000002\C0000000.LRG is the default file containers for table space 2 (USERSPACE1).**

\_\_\_ 29. Retrieve detailed container information from the Catalog tables. What is the container types for each container?

Command Window	db2 select tbsp_name, tbsp_id, container_name, container_id, container_type from sysibmadm.snapcontainer order by tbsp_id
GUI	Using the <i>Command Editor</i> , execute the following statement: select tbsp_name, tbsp_id, container_name, container_id, container_type from sysibmadm.snapcontainer order by tbsp_id

TBSP_NAME	TBSP_ID	CONTAINER_NAME	CONTAINER_ID	CONTAINER_TYPE
SYSCATSPACE	0		0	FILE_EXTENT_TAG
TEMPSPACE1	1		0	PATH
USERSPACE1	2		0	FILE_EXTENT_TAG
SYSTOOLSPACE	3		0	FILE_EXTENT_TAG

## Section 2 - Creating Table Spaces

Your MUSICDB database requires additional table spaces. You will create one of these table spaces. You will then create the remaining six table spaces via a script command file.

\_\_\_ 1. Create your first table space. It should have the following characteristics:

- Table space name is DMS01
- Table Space Type is Regular
- Buffer Pool should be IBMDEFAULTBP (which is also the default)
- Table Space management is DMS (also called *high performance*)
- Container size should be 1006 pages with 4 KB pages
- Container should be a File
- Container path and filename should be C:\dms\dms01 (Windows)
- Table space extent size and prefetch size should be 4

Command Window	cd (Windows) db2 "create regular tablespace dms01 managed by database using (file 'C:\dms\dms01' 1006) extentsize 4 prefetchsize 4"
----------------	---

GUI part 1	<p>From the <i>Control Center</i> right-click <i>Table Spaces</i> and choose <i>Create</i>.</p> <p>___ Use the <i>Create Table Space Wizard</i> to specify the characteristics as shown in the instructions above.</p> <p>___ On the <i>Name</i> page, indicate name of <b>dms01</b>. Click the <i>I want to manage my storage manually</i> radio button.</p> <p>___ On the <i>Type</i> page, indicate <i>Regular</i>.</p> <p>___ On the <i>Space Management</i> page, indicate <i>Database-managed space (high performance)</i>.</p> <p>___ On the <i>Containers</i> page, click <b>Add</b>; specify 4 KB pages, then a size of 1006. Indicate the file name as given in the instructions. (You can select the directory names for your <b>C:\</b> directory, but you will need to type in <b>dms\dms01</b> as the Container name since the dms directory does not yet exist.)</p> <p>___ Click <b>OK</b>.</p> <p>___ Ensure that on the <i>Read/Write</i> window, you indicate <i>Extent</i> size as 4 pages (four 4KB pages).</p> <p>___ Ensure that on the <i>Recovery</i> window, you uncheck the box: <i>Enable dropped table recovery</i>.</p>
GUI part 2	<p>After specifying all the characteristics, check your settings by clicking <i>Show SQL</i> on the <i>Summary</i> page before creating the table space. Your SQL should look like this:</p> <pre>(Windows): CREATE REGULAR TABLESPACE DMS01 PAGESIZE 4 K MANAGED BY DATABASE USING ( FILE 'C:\dms\dms01' 1006 ) EXTENTSIZE 4 OVERHEAD 10.5 PREFETCHSIZE 4 TRANSFERRATE 0.14 BUFFERPOOL IBMDEFAULTBP DROPPED TABLE RECOVERY OFF;</pre> <p>Click <b>Finish</b> to create the table space.</p>

\_\_\_ 2. Verify your new table space (DMS01) by listing table spaces.

Verify that the Allocated size is 1006 pages. Verify that the extent size and prefetch size are both 4. If not, then drop the table space and create it again.

Command Window	<pre>db2 connect to musicdb db2 list tablespaces show detail   more</pre>
GUI	<p>From the <i>Command Editor</i>:</p> <pre>list tablespaces show detail;</pre>

\_\_\_ 3. On your Windows Database Server, a script file named **crttblsp** contains SQL statements to create your additional table spaces.

Command Window	cd cd C:\labfiles\cf23
GUI	cd cd C:\labfiles\cf23

\_\_\_ 4. Execute the script file to create your remaining table spaces.

Make sure you have the following options set before executing the script:

- Auto commit should be enabled.
  - Execution should be stopped if there is an error.
  - Commands and statements should be echoed back to the screen.
  - A semicolon should be used as the termination character.
- 
- 

Command Window	Verify how to set the script options by examining the output from the following command: <code>db2 list command options   more</code> Execute the script with the following script options: <code>db2 -tv sf crtblsp</code>
GUI part 1	On the <i>Command Editor, Commands</i> tab: ___ Before loading and executing the file, check the Command Editor options to make sure you have the following script options: ___ Select <i>Tools Settings</i> from the icon bar. ___ Select the <i>Command Editor</i> tab. ___ Under <i>Execution and history</i> , make sure the checkbox for <i>Automatically commit SQL statements</i> and <i>Stop execution if errors occur</i> are checked. This turns on auto commit, and stops execution if an error occurs. ___ Close the <i>Tools Settings</i> window.

GUI part 2	<p>Selecting the file:</p> <p>___ Select <i>Selected</i> -&gt; <i>Open</i> from the menu bar.</p> <p>___ In the <i>File name</i> box, type in the complete directory and file name  <i>(Windows): C:\labfiles\cf23\crtblsp</i></p> <p>or, in the <i>Directories</i> list, select each directory by clicking twice and scroll to the bottom of the <i>Files</i> list until you reach a file named <b>crtblsp</b> and select it. Note that if you use the <i>Directories</i> list option, the <i>File name</i> box gets filled in with the <b>crtblsp</b> file name.</p> <p>___ Click OK.</p> <p>(If prompted, indicate <i>Yes</i> to discard your current script.)</p>
GUI part 3	Run the file by pressing <b>Ctrl-Enter</b> .

- \_\_\_ 5. Confirm that the additional six table spaces are present. Are your new table space names listed?
- 

Command Window	<code>db2 list tablespaces   more</code>
GUI	<p>From the <i>DB2 Control Center</i>, refresh table spaces.</p> <p>___ Right-click and select <i>Table Spaces</i>.</p> <p>___ Select <i>Refresh</i> from the pop-up menu.</p> <p>___ Check the <i>Object</i> pane for your new table spaces.</p>

- \_\_\_ 6. From your **command window**, change directories to **C:\dms** and do a list of the files.

```
cd C:\dms
dir
```

- \_\_\_ 7. What are these files?
- 

**The files are the containers for the DMS table spaces.**

- \_\_\_ 8. Display detailed information about the table spaces. You must be connected to the database first.

Command Window	<pre>db2 connect to musicdb db2 list tablespaces show detail   more</pre>
-------------------	---

GUI	From the <i>Command Editor</i> : connect to musicdb; list tablespaces show detail;
-----	--

- \_\_\_ 9. What ID numbers are associated with each of the table spaces? How many usable pages are there in the DMS table spaces? List a few differences between SMS and DMS table spaces.

---

---

---

---

---

---

---

**SYSCATSPACE — ID 0**  
**TEMPSPACE — ID 1**  
**USERSPACE1 — ID 2**  
**SYSTOOLSPACE — ID 3**  
**DMS01 — ID 4 — 1000**  
**DMS02 — ID 5 — 12**  
**DMS03 — ID 6 — 720**  
**DMS04 — ID 7 — 20**  
**DMS05 — ID 8 — 14**  
**DMS06 — ID 9 — 36**  
**SMS01 — ID 10**

**SMS table spaces do not use preformatted containers. Space will be allocated until the file system is full. DMS table spaces preallocate all space.**

- \_\_\_ 10. Show the container information for table space ID 4. What type of container is this and where it is located?

---

---

Command Window	db2 list tablespace containers for 4 show detail
GUI	list tablespace containers for 4 show detail

**This is a *file* container, and it is located at . C : \dms\dms01.**

- \_\_\_ 11. Show the container information for your SMS table space ID 10. What type of containers are being used and where are they?



Command Window	db2 list tablespace containers for 10 show detail
GUI	list tablespace containers for 10 show detail

**It is a *directory path* container located at C:\sms\sms01.**

- \_\_\_ 12. From your **command window**, change directories to SQLT0000.0 and list the directory.

```
cd
cd C:\sms01
dir
```

- \_\_\_ 13. Table data is stored in .DAT files, indexes in .INX files, and LOB info in .LB and .LBA files. And there is an SQLTAG.NAM file that holds overhead information about this container, and the tables that are found there.

Currently there are no tables in this table space.

- \_\_\_ 14. From your ssh/telnet session to the database server, create a table and an index on that table in this table space. And then list the contents of the directory again. What do you see now? Are the new files named after the table name?

```
db2 "create table t (i int) in sms01"
db2 "create index ii on t(i) "
ls
```

**You will see two new files: SQL00002.DAT and SQL00002.INX, one for the table and one for all indexes on that table.**

**No — the files not named after the table (t). Note the this first table is numbered 00002 and the file containing the index (and all indexes, if there were more than one) has the same numbering. The next table created would be named SQL00003.DAT.**

- \_\_\_ 15. Drop table t as it is no longer needed: db2 drop table t
- \_\_\_ 16. Table space information is accessible through a view with the name SYSCAT.TABLESPACES. The fields that contain table space information are:
- TBSPACE — Name of primary table space for this table
  - DEFINER — Authid of table space creator
  - TBSPACEID — Internal table space identifier
  - TBSPACETYPE — Type of table space. D for DMS or S for SMS.

- DATATYPE — Type of data that can be stored in the table space. L for long data only, A for all types of permanent data, or T for temporary tables only.

Issue a select from SYSCAT.TABLESPACES to answer the following questions.

Command Window	db2 "select substr(tbspace,1,18) as tbspace, substr(definer,1,10) as definer, tbspaceid, tbspacetype, datatype from syscat.tablespaces"
GUI	select tbspace, definer, tbspaceid, tbspacetype, datatype from syscat.tablespaces If the results do not show up on the <i>Query Results</i> tab, either change your select statement to the one shown in the Local instructions so you can see the fields without excessive scrolling, or exit the <i>Command Editor</i> , reopen it, and reissue your statement.

\_\_\_ 17. What is the authorization ID of table space definer for the default table spaces?

---

**DEFINER=SYSIBM**

\_\_\_ 18. Which of the various table spaces allows long data types?

---

**USERSPACE1, SYSTOOLSPACE, and DMS03 table spaces allow long data.**

\_\_\_ 19. Which table space only allows temporary table data?

---

**TEMPSPACE1 only allows temporary table data.**

\_\_\_ 20. Table space information for individual tables can be accessed through the SYSCAT.TABLES view. The fields that contain table space information are:

- TBSPACEID — Table space ID of primary table space for this table
- TBSPACE — Name of primary table space for this table
- INDEX\_TBSPACE — Table space containing the indexes for this table
- LONG\_TBSPACE — Table space containing LONG or LOB data for this table

List the table space information for the table SYSIBM.SYSTABLES.

Command Window	db2 "select substr(tabname,1,18) as tabname, tbspaceid, tbspace, index_tbspace, long_tbspace from syscat.tables where tabname='SYSTABLES' "
----------------	---

GUI	select tabname, tbspaceid, tbspace, index_tbspace, long_tbspace from syscat.tables where tabname='SYSTABLES'
-----	---

\_\_\_ 21. Is the SYSIBM.SYSTABLES table divided between different table spaces?

---



---



---

**No. But this is not something you can easily determine from the information that you are viewing.**

**In previous releases (V8.2 and earlier), the answer would be easier, since with those releases, catalog tables were stored in SYSCATSPACE and that table space was SMS. And, a table cannot be divided between table spaces if SMS is being used.**

\_\_\_ 22. Enter `connect reset` to break your database connection.

Command Window	<code>db2 connect reset</code>
GUI	<code>connect reset</code>

**END OF LAB**



## Exercise 5. Create objects

*(with hints)*

### Estimated time

01:00

### What this exercise is about

This exercise is an online lab which creates objects used in a database.

### What you should be able to do

At the end of the lab, students should be able to:

- Create tables
- Create indexes
- Create views
- Create an alias
- Add referential integrity constraints to a table
- Add check constraints to a table
- Add triggers to a table
- Access System Catalog information about objects
- Retrieve an XML document

### Introduction

Now that the database and table spaces have been created, the database objects needed for your MUSICDB database need to be created. The DB2 Control Center (GUI labs) or a DB2 statement will be used to create at least the first object of each type, such as the first table. Where there are multiple objects of the same type to be created, a script file will be executed to create all the remaining objects of a certain type.

## ***Instructor Notes:***

*Introduction* — This exercise is an online lab which creates the database objects for the MUSICDB database.

*Time for Lab* — 1 hour.

*Things to Review at End of Lab* — At the end of this lab make sure every team has created their database objects.

*Common Mistakes Students May Make* — Some students forget to grant select privilege on the new tables to public. This is done at the end of Section 1, and will affect later labs if they forget to do it.

If students are performing the GUI labs, SQL SELECT statements should be done, in general, from the Command Editor, Commands tab. By default, these results should show up on the Query Results tab. Because that output does not have the problem with the long column output, we removed the substr's in the select statement - to make the SQL easier to enter. On the Command Line interface, the substr's are still required.

The check scripts have the SQL in a file so this will also reduce the amount of typing required.

## Exercise instructions with hints

**Instructor Note:** The following instructions should be pointed out to students. Not every instruction is included in the exercises (like closing every window, or scrolling to find the action buttons). These comments are included to let the students know that these details are left out on purpose.

Throughout the exercise, if the DB2 Message Window appears, or if you are looking at a Show SQL or Show Command window, in many cases, the lab instructions will not specifically tell you to close the window. Read the message, look at the SQL or Command, and then click Close to close the window.

If a GUI window, other than the Control Center, appears to be hung, check to be sure that the Control Center does not have a message window open. If there is a message window open in the Control Center, it may impact the usability of the other GUI tools.

Commands are generally to be entered on the Command Editor, Commands tab. If you want to enter them on the DB2 Command Window (prefaced with db2), that will generally work as well.

In some cases, the window may need to be scrolled to access the Action buttons on the window (like OK, Cancel, or Apply). If you cannot see the Action button, scroll down the window to see them.

If, at any point in the exercises, when you are importing a new script into the Command Editor, Commands tab, you are prompted on whether you want to discard your current script, indicate Yes.

The lab solutions using primarily your telnet session and commands are indicated as *Command Window*. The lab solutions using primarily the GUI tools are indicated as *GUI*.

## Section 1 - Create Tables

In this section, you will be creating the tables needed for your MUSICDB database.

\_\_\_ 1. Create your ARTISTS table. The table should have the following characteristics.

- Table schema of **student**
- Table Name of artists
- Regular data to go in table space dms01
- Indexes to go in table space dms02
- Long data to go in table space dms03
- The column artno should be defined as a primary key
- Columns defined like the following:
 

```
(artno          smallint not null,
  name          varchar (50),
  classification char (1) not null,
  bio           clob (100K) logged compact,
  picture       blob (500k) not logged compact)
```

Command Window	<p>db2 connect to musicdb</p> <p>Do not press <i>Enter</i> until you key in the complete statement. Pressing <i>Enter</i> sends the statement to DB2 to be executed.</p> <pre>db2 "create table artists (artno          smallint not null,  name          varchar (50),  classification char (1) not null,  bio           clob (100K) logged compact,  picture       blob (500k) not logged compact,  primary key (artno))   in dms01   index in dms02   long in dms03"</pre>
GUI part 1	<ul style="list-style-type: none"> <li>___ Ensure that you have refreshed your list of table spaces in the <i>Control Center</i> and that you can see all of your table spaces before starting the process of creating your table.</li> <li>___ On the <i>Control Center</i>, from your <b>MUSICDB</b> database objects list, select <i>Tables</i> with your right mouse button, and choose <i>Create</i> from the pop-up menu.</li> <li>___ Select a table schema name of <b>student</b>.</li> <li>___ Enter a value of <b>artists</b> in the <i>Table</i> name box.</li> <li>___ Click <b>Next</b> to define Columns.</li> </ul>
GUI part 2	<p>On the <i>Columns</i> window, define the <b>artno</b> column.</p> <ul style="list-style-type: none"> <li>___ Click <b>Add</b> button.</li> <li>___ Enter a <i>Column name</i> of <b>artno</b>.</li> <li>___ Select pull-down arrow to the right of the <i>Data type</i> box and select <b>SMALLINT</b>.</li> <li>___ Indicate <b>Value generation</b> of <b>None</b>.</li> <li>___ Ensure that there is no checkmark to the left of <i>Nullable</i> to ensure that this is a not null column.</li> <li>___ Click <b>Apply</b>.</li> </ul>
GUI part 3	<p>Continue by defining the <b>name</b> column:</p> <ul style="list-style-type: none"> <li>___ Enter a <i>Column name</i> of <b>name</b>.</li> <li>___ Select <i>Data type</i> of <b>VARCHAR</b>.</li> <li>___ In the <i>Length</i> box, change the value to 50.</li> <li>___ Indicate <b>Value generation</b> of <b>None</b>.</li> <li>___ Add a checkmark to the left of <i>Nullable</i> to make the column nullable.</li> <li>___ Click <b>Apply</b>.</li> </ul>



GUI part 4	Define the <b>classification</b> column. ___ Enter a <i>Column name</i> of <b>classification</b> . ___ Select <i>Data type</i> of <b>CHARACTER</b> . ___ In the <i>Length</i> box, change the value to 1. ___ Indicate <i>Value generation</i> of <b>None</b> . ___ Remove the checkmark to the left of <i>Nullable</i> to make this a not null column. ___ Click <b>Apply</b> .
GUI part 5	Define the <b>bio</b> column. ___ Enter a <i>Column name</i> of <b>bio</b> . ___ Select <i>Data type</i> of <b>CLOB</b> . ___ Select pull-down arrow to the right of the <i>LOB unit</i> box. ___ Select <b>KBytes</b> . ___ Enter a value of 100 in the <i>Length</i> box. ___ Select check boxes to the left of the <i>Logged</i> and <i>Compact LOB</i> options, to add a checkmark to each. ___ Indicate <i>Value generation</i> of <b>None</b> . ___ Add a checkmark to the left of <i>Nullable</i> to make the column nullable. ___ Click <b>Apply</b> .
GUI part 6	Define the <b>picture</b> column. ___ Enter a <i>Column name</i> of <b>picture</b> . ___ Select <i>Data type</i> of <b>BLOB</b> . ___ Select pull-down arrow to the right of the <i>LOB unit</i> box. ___ Select <b>KBytes</b> . ___ Enter a value of 500 in the <i>Length</i> box. ___ Ensure the check box to the left of the <i>Compact LOB</i> option is checked to indicate that this column should be compact. The <i>Logged</i> box should be unchecked, indicating that changes to this column will not be logged. ___ Indicate <i>Value generation</i> of <b>None</b> . ___ Verify the check box to the left of <i>Nullable</i> is checked. ___ Click <b>OK</b> .

GUI part 7	<p>Carefully verify the column definitions before continuing:</p> <table><tr><th>Column name</th><th>Data type</th><th>Length</th><th>Nullable</th><th>LOB option</th></tr><tr><td>ARTNO</td><td>SMALLINT</td><td></td><td>No</td><td></td></tr><tr><td>NAME</td><td>VARCHAR</td><td>50</td><td>Yes</td><td></td></tr><tr><td>CLASSIFICATION</td><td>CHARACTER</td><td>1</td><td>No</td><td></td></tr><tr><td>BIO</td><td>CLOB</td><td>100 KBytes</td><td>Yes</td><td>Logged, Compact</td></tr><tr><td>PICTURE</td><td>BLOB</td><td>500 KBytes</td><td>Yes</td><td>Compact</td></tr></table> <p>___ Verify your columns are defined correctly as stated in the column requirements above. If not, select the appropriate column, and then select <i>Change</i> button to make your corrections.</p> <p>___ Warning: An error at this point will cause some of the following lab exercises to fail.</p> <p>___ Select the <b>Table Spaces</b> window.</p>	Column name	Data type	Length	Nullable	LOB option	ARTNO	SMALLINT		No		NAME	VARCHAR	50	Yes		CLASSIFICATION	CHARACTER	1	No		BIO	CLOB	100 KBytes	Yes	Logged, Compact	PICTURE	BLOB	500 KBytes	Yes	Compact
Column name	Data type	Length	Nullable	LOB option																											
ARTNO	SMALLINT		No																												
NAME	VARCHAR	50	Yes																												
CLASSIFICATION	CHARACTER	1	No																												
BIO	CLOB	100 KBytes	Yes	Logged, Compact																											
PICTURE	BLOB	500 KBytes	Yes	Compact																											
GUI part 8	<p>Define the table spaces to be used by the <b>ARTISTS</b> table.</p> <p>___ Select pull-down menu arrow to the right of the <i>Table space</i> box and select <b>DMS01</b> from the pull-down list.</p> <p>___ Click to put a check in the <i>Use separate index space</i> and then select the pull-down list arrow to the right of the <i>Index table space</i> box to select <b>DMS02</b> from the pull-down list.</p> <p>___ Click to put a check in the <i>Use separate long space</i> and then select pull-down list arrow to the right of the <i>Long table data</i> box to select <b>DMS03</b> from the pull-down list.</p> <p>___ Click <b>Next</b> to go to the <i>Keys</i> window.</p>																														
GUI part 9	<p>Define a primary key on the <b>artno</b> column.</p> <p>___ Click <i>Add Primary</i> button.</p> <p>___ Select <b>ARTNO</b> in the <i>Available columns</i> box and click the single &gt; button to add <b>ARTNO</b> to the <i>Selected columns</i> box. <b>(Note: only columns defined as NOT NULL are presented for selection since Primary Keys must be defined as NOT NULL.)</b></p> <p>___ Click <b>OK</b>.</p>																														
GUI part 10	<p>Go to the <i>Summary</i> window. Click <b>Show SQL</b>. The SQL should look like this:</p> <pre>CREATE TABLE STUDENT.ARTISTS ( ARTNO SMALLINT NOT NULL , NAME VARCHAR (50) , CLASSIFICATION CHARACTER (1) NOT NULL , BIO CLOB (100 K ) LOGGED COMPACT , PICTURE BLOB (500 K ) NOT LOGGED COMPACT , CONSTRAINT CC1044760857306 PRIMARY KEY ( ARTNO) ) IN DMS01 INDEX IN DMS02 LONG IN DMS03 ;</pre> <p>Note: the name of the constraint will vary.</p> <p>___ Click <b>Finish</b> to create the table.</p>																														

Command Editor	<pre>create table artists (artno          smallint not null,  name           varchar (50),  classification char (1) not null,  bio            clob (100K) logged compact,  picture        blob (500k) not logged compact,  primary key (artno))   in dms01   index in dms02   long in dms03</pre>
----------------	---

- \_\_\_ 2. Verify the creation of your ARTISTS table and the table description. **Important:** Ensure the definition of your table is correct. It could cause unexpected failures in later labs if your definitions are not done correctly.

Column name	Type schema	Type name	Length	Scale	Nulls
ARTNO	SYSIBM	SMALLINT	2	0	No
NAME	SYSIBM	VARCHAR	50	0	Yes
CLASSIFICATION	SYSIBM	CHARACTER	1	0	No
BIO	SYSIBM	CLOB	102400	0	Yes
PICTURE	SYSIBM	BLOB	512000	0	Yes

Command Window	<pre>db2 list tables for user db2 describe table artists</pre>
GUI	<pre>connect to musicdb; list tables for user; describe table artists;</pre>

- \_\_\_ 3. You will create the remainder of the tables required using a script file called crttables. For Windows, it is located in the C:\labfiles\cf23 directory. Examine the DB2 statements in the file and answer the following questions.

In which table space will both the indexes and data for table STOCK be placed? For CONCERTS? For REORDER?

---



---

Command Window	<pre>cd C:\labfiles\cf23 more crttables</pre>
----------------	---

GUI	___ Select <i>Commands</i> tab on the <i>Command Editor</i> panel. ___ <b>Selected</b> -> <b>Open</b> (menu bar) ___ Verify that the <i>System name</i> is your local workstation. ___ In the <i>Directories</i> list, select the <b>C:\labfiles\cf23</b> directory by clicking twice. ___ Scroll down the <i>Files list</i> until you reach a file named <b>crtables</b> and select it. ___ Note that the <i>Path</i> box gets filled in with the <b>crtables</b> file name. ___ Click <b>OK</b> .
-----	---

**The indexes and data for table STOCK will both be placed in table space dms06, for CONCERTS in dms04, and for REORDER in sms01.**

\_\_\_ 4. Execute the **crtables** script to create the tables.

Command Window	<b>db2 -tvf crtables</b>
----------------	--------------------------

\_\_\_ 5. Verify that all tables (ALBUMS, STOCK, CONCERTS, and REORDER) were

GUI	___ Change the connect statement to reflect your instance name. ___ Press <b>Ctrl-Enter</b> to execute.
-----	--

created successfully.

Command Window	<b>db2 list tables for user</b> <b>db2 describe table albums</b> <b>db2 describe table stock</b> <b>db2 describe table concerts</b> <b>db2 describe table reorder</b>
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: <b>list tables for user;</b> <b>describe table albums;</b> <b>describe table stock;</b> <b>describe table concerts;</b> <b>describe table reorder;</b>

\_\_\_ 6. Information about each column in a table can be found by accessing the SYSCAT.COLUMNS view. Enter an interactive SQL statement to select the columns TABNAME, COLNAME, and TYPENAME from SYSCAT.COLUMNS for all tables with the TABSCHEMA of **'student'**, order the data by TABNAME and COLNO, and pipe the output to a file named **tabchk.file**. The SQL can be found in a file named **tabchk.sql**.

Place the output in the **C:\labfiles\cf23** subdirectory.

Windows	<pre>cd C:\labfiles\cf23 db2 connect to musicdb db2 -tf tabchk.sql &gt; tabchk.file more tabchk.file  tabchk.sql contains: select tabname, colname, typename                       from syscat.columns where tabschema = user order by 1,2;</pre>
Command Window	<p>From your telnet or command window, enter the following commands:</p> <pre>db2 connect to musicdb db2 -tf tabchk.sql &gt; tabchk.file more tabchk.file  tabchk.sql contains: select tabname, colname, typename                       from syscat.columns where tabschema = user order by 1,2;</pre>
GUI	<p>Using the DB2 <i>Command Window</i> (make certain you are in the directory with the course files), enter the following command:</p> <pre>cd \cf2n db2 connect to musicdb user instnm using ibm2blue db2 -tf tabchk.sql &gt; tabchk.file more tabchk.file  tabchk.sql contains: select tabname, colname, typename                       from syscat.columns where tabschema = user order by 1,2;</pre>

\_\_\_ 7. Check your output against the `tabchk.master` file.

**Ensure that there are no differences between your output and the master output or the following labs may not work correctly!**

GUI	<pre>fc /w tabchk.file tabchk.master   more</pre> <p>If you see a message that indicates:  <b>FC: no differences encountered</b>  your table definitions match our master file.</p> <p>If you see any output, there are lines that do not match between the files. The first set of lines will indicate the information in your output, the second set of lines will indicate the information in the master output. The first column will indicate which table has the incorrect definition; the second column will indicate which column has the incorrect definition. Examine this information, drop the table with a problem, and go back to the step that defined it and define it correctly. Come back to the previous step and check it again.</p>
-----	--

- \_\_\_ 8. Information about a table's definition to a table space can be found by accessing the SYSCAT.TABLES view. Enter an SQL statement to select the columns TABNAME, TBSPACE, and INDEX\_TBSPACE from SYSCAT.TABLES for all tables with the TABSCHEMA of 'student', order the data by TABNAME, and pipe the output to a file named tbschk.file. The SQL can be found in a file named tbschk.sql.

Place the output in the C:\labfiles\cf23 subdirectory.

---



---

Command Window	<pre>db2 -tf tbschk.sql &gt; tbschk.file more tbschk.file</pre> <p>tbschk.sql contains: select tabname, tbSPACE, index_tbSPACE from syscat.tables where tabschema = user order by tabname</p>
GUI	<p>On a DB2 Command Window:</p> <pre>db2 -tf tbschk.sql &gt; tbschk.file more tbschk.file</pre> <p>tbschk.sql contains: select tabname, tbSPACE, index_tbSPACE  from syscat.tables where  tabschema = user order by tabname</p>

- \_\_\_ 9. Check your output against the tbschk.master file.

**Ensure that there are no differences between your output and the master output, or the following labs may not work correctly!**

GUI	<pre>fc /w tbschk.file tbschk.master   more</pre> <p>If you see a message that indicates: <b>FC: no differences encountered</b> your table definitions match our master file.</p> <p>If you see any output, there are lines that do not match between the files. The first set of lines will indicate the information in your output, the second set of lines will indicate the information in the master output. The first column will indicate which table has the incorrect definition; the second column will indicate the table space for the data; the third column will indicate the index table space. Examine this information, drop the table with a problem, and go back to the step that defined it and define it correctly. Come back to the step where you ran the select against SYSCAT.COLUMNS (4 steps ago) and check the columns and the table spaces again.</p>
-----	--

- \_\_\_ 10. Grant select privilege on tables that are owned by **student** to public by executing a script file, `grants` (an operating system command file) for local administration.

GUI	On a DB2 Command Window: <code>db2 -tvf grants</code>
-----	--



## Section 2 - Create Indexes

In this section, you will be creating the indexes needed by several of the tables.

- \_\_\_ 1. Now that your tables are created, you can create the indexes. Create an index called ITEM on the ITEMNO column in STOCK table.

Command Window	db2 "create index item on stock (itemno)"
GUI	<p>___ On the <i>Control Center</i>, from the <b>MUSICDB</b> objects list, right-click and select <i>Indexes</i>.</p> <p>___ Select <i>Create &gt; Index</i> from the pop-up menu.</p> <p>___ Indicate the following:</p> <p>___ Index schema of <b>student</b>.</p> <p>___ Index name of <b>item</b>.</p> <p>___ Table schema of <b>student</b>.</p> <p>___ Table name of <b>STOCK</b>.</p> <p>___ Select <b>ITEMNO</b> from the <i>Available columns</i> list and click the single &gt; button to add <b>ITEMNO</b> to the <i>Selected columns</i> list.</p> <p>___ Click OK.</p> <p>You may get a warning that "Some statistics are in an inconsistent state." This is because index statistics are automatically collected when an index is created through the Control Center. More about statistics collection will be addressed later in class.</p> <p>___ Click Close.</p> <p>___ Click Cancel.</p>

- \_\_\_ 2. Create a unique index called ITEMNO on the ITEMNO column in ALBUMS table.

Command Window	db2 "create unique index itemno on albums (itemno)"
----------------	---

GUI	<p>             ___ Using the DB2 Control Center, from the <b>MUSICDB</b> objects list, right-click and select <i>Indexes</i>.              ___ Select <i>Create &gt; Index</i> from the pop-up menu.              ___ Indicate the following:              ___ Index schema of <b>student</b>.              ___ Index name of <b>itemno</b>.              ___ Table schema of <b>student</b>.              ___ Table name of <b>ALBUMS</b>.              ___ Select <b>ITEMNO</b> from the <i>Available columns</i> list and click the single &gt; button to add <b>ITEMNO</b> to the <i>Selected columns</i> list.              ___ Select checkbox to the left of <i>Unique</i>.              ___ Click <b>OK</b>.              You may get a warning that "Some statistics are in an inconsistent state." This is because index statistics are automatically collected when an index is created through the Control Center. More about statistics collection will be addressed later in class.              ___ Click <b>Close</b>.              ___ Click <b>Cancel</b>.           </p>
-----	--

- \_\_\_ 3. Select the information from the catalog tables about your indexes. The columns you should select are the first 18 characters of TABNAME, UNIQUERULE, the first 18 characters of INDNAME, and the first 30 characters of COLNAMES from SYSCAT.INDEXES, where INDSHEMA is **student**, order by TABNAME and INDNAME.

If UNIQUERULE = "U", then only unique values are allowed.

If UNIQUERULE = "D", then duplicate values are allowed.

If UNIQUERULE = "P", then it is a Primary key.

Do you see your new indexes?

Command Window	<pre>db2 "select substr(tabname,1,18), uniquerule, substr(indname,1,18), substr(colnames,1,30) from syscat.indexes where indschema = user and uniquerule &lt;&gt; 'P' order by tabname, indname"</pre>
GUI	<p>On a DB2 Command Window:</p> <pre>db2 "select substr(tabname,1,18), uniquerule, substr(indname,1,18), substr(colnames,1,30) from syscat.indexes where indschema = user and uniquerule &lt;&gt; 'P' order by tabname, indname"</pre>

**Yes.**

**Instructor note:** The primary key created by the GUI has a schema name of the ID running the create table; the primary key created by the statement on the local session has a schema name of SYSIBM. The index name is also different. On the Command Interface, the index name is SQL plus numbers; through the Control Center create table, it is CC plus numbers.

- \_\_\_ 4. Use the select you executed in the previous step and route the output to `indchk.file`. The SQL can be found in a file named `indchk.sql`.

Check your results against `indchk.master`.

**Ensure that there are no differences between your output and the master output, or the following labs may not work correctly!**

GUI	<p>On a DB2 Command Window:</p> <pre>db2 -tf indchk.sql &gt; indchk.file type indchk.file fc /w indchk.file indchk.master   more</pre> <p><b>Instructor Note:</b> I made this a "type" rather than a "more" because the "more" wiped out some lines...</p> <p>If you see a message that indicates:  <b>FC: no differences encountered</b>  your table definitions match our master file.</p> <p>If you see any output, there are lines that do not match between the files. The first set of lines will indicate the information in your output, the second set of lines will indicate the information in the master output. The first column will indicate which table has the incorrect definition; the third column will indicate the index that has a problem. Examine this information, drop the index with a problem, and go back to the step that defined it and define it correctly. Come back to this step and check the indexes again.</p> <p><code>indchk.sql</code> contains: <code>select substr(tabname,1,18), uniquerule, substr(indname,1,18), substr(colnames,1,30) from syscat.indexes where indschema = user and uniquerule &lt;&gt; 'P' order by tabname, indname</code></p>
-----	--

## Section 3 - Create Views

In this section, you will create two views that will be used in a later lab.

- \_\_\_ 1. Create a view called music that will select title, classification, name from albums, artists where artists.artno = albums.artno.

Command Window	db2 "create view music as select title, classification, name from student.albums, student.artists where student.artists.artno = student.albums.artno"
GUI	<p>___ On the <i>Control Center</i>, from the <b>MUSICDB</b> objects list, right-click and select <i>Views</i>.</p> <p>___ Select <i>Create</i> from the pop-up menu.</p> <p>___ Indicate a View schema of <b>student</b>.</p> <p>___ Enter a name of <b>music</b> in the <i>View name</i> box.</p> <p>___ Click <i>Clear</i> button to the right of the SQL statement box.</p> <p>___ In the <i>SQL statement</i> box, enter: as select title, classification, name from student.albums, student.artists where student.artists.artno = student.albums.artno</p> <p>___ Click <b>OK</b>.</p>

- \_\_\_ 2. Create another view called inventory by executing the script file, crview.

Command Window	db2 -tvf crview
GUI	<p>___ On the <i>Commands</i> tab on the <i>Command Editor</i>:</p> <p>___ Select <i>Selected -&gt; Open</i> from the menu bar.</p> <p>___ Verify that the System name is your local workstation.</p> <p>___ In the <i>Directories</i> list, select the <b>C:\labfiles\cf23</b> directory by clicking twice.</p> <p>___ Scroll down the <i>Files</i> list until you reach a file named <b>crview</b> and select it. Alternatively, you can type in the full path name in the <i>File name</i> field.</p> <p>___ Click <b>OK</b>.</p> <p>___ Press <b>Ctrl-Enter</b> to execute.</p>

- \_\_\_ 3. Information about views can be found in the SYSCAT.VIEWS and SYSCAT.TABLES views.

Tables and views for the current user can be shown with a LIST TABLES statement. A TYPE of V is a view.

Issue one of the following statements and verify your views are listed.

- LIST TABLES statement
- OR
- Select columns TABSCHEMA, TABNAME, and TYPE from the SYSCAT.TABLES view with a TABSCHEMA of **student**
- OR
- Select columns VIEWSCHEMA, VIEWNAME from SYSCAT.VIEWS with DEFINER of **student**

**Note:** You can use the user special register; it resolves to the current authorization ID.

Command Window	<pre>db2 list tables for user OR db2 "select substr(tabschema,1,18) as tabschema,       substr(tabname,1,18) as tabname,       type       from syscat.tables where tabschema = user" OR db2 "select substr(viewschema,1,18) as viewschema,       substr(viewname,1,18) as viewname       from syscat.views where definer = user"</pre>
GUI	<p>Use the DB2 <i>Command Editor</i> to execute one of the following command sets:</p> <pre>connect to musicdb; list tables for user; OR select tabschema, tabname, type       from syscat.tables where tabschema = user; OR select viewschema, viewname       from syscat.views where definer = user;</pre>

**You should see your views listed.**

## Section 4 - Create Alias

In this section, you will be creating two aliases for a later lab.

- \_\_\_ 1. Create an alias called singers for the ARTISTS table, and an alias called emptystock for the REORDER table.

Command Window	<pre>db2 create alias singers for artists db2 create alias emptystock for reorder</pre>
GUI part 1	<p>Define the alias <b>singers</b>.</p> <p>___ On the <i>Control Center</i>, from the <b>MUSICDB</b> objects list, right-click and select <i>Aliases</i>.</p> <p>___ Select <i>Create</i> from the pop-up menu.</p> <p>___ Indicate the following:</p> <p>___ Alias schema of <b>student</b>.</p> <p>___ Alias name of <b>singers</b>.</p> <p>___ Object schema of <b>student</b>.</p> <p>___ Object name of <b>artists</b>.</p> <p>___ Select <i>Show SQL</i> button and look at the SQL statement.</p> <p>___ Click <b>OK</b>.</p>
GUI part 2	<p>Define the alias <b>emptystock</b>.</p> <p>___ Right-click <i>Aliases</i>, select <i>Create</i>.</p> <p>___ Indicate the following:</p> <p>___ Alias name of <b>emptystock</b>.</p> <p>___ Object name of <b>reorder</b>.</p> <p>___ Click <b>OK</b>.</p>

- \_\_\_ 2. Information about aliases can be found in the SYSCAT.TABLES view. Tables, views, and aliases for the current user can also be shown with the LIST TABLES statement. A TYPE of A is an alias.

Issue one of the following statements and verify your aliases are listed.

- LIST TABLES statement

OR

- Select columns TABNAME and TYPE from the SYSCAT.TABLES view with a TABSCHEMA of **student**.

Command Window	<pre>db2 list tables for user OR db2 "select substr(tabname,1,18), type       from syscat.tables where tabschema = user"</pre>
----------------	--

GUI	From the <i>Command Editor</i> : connect to musicdb; list tables for user; OR select tabname, type from syscat.tables where tabschema = user;
-----	---

## Section 5 - Add Referential Integrity

In this section, you will be adding referential integrity between the ARTISTS and ALBUMS tables and the ALBUMS and STOCK tables.

- \_\_\_ 1. Alter the ALBUMS table and define referential integrity relationships it has with other tables.
  - Add a primary key on the itemno column
  - Add a foreign key called fkartno on the artno column which references a primary key in the ARTISTS table
  - The delete rule between ARTISTS and ALBUMS should be delete cascade

Command Window	db2 "alter table albums primary key (itemno) add constraint fkartno foreign key (artno) references artists on delete cascade"
GUI part 1	___ Using the DB2 <i>Control Center</i> , from the MUSICDB objects list, left-click and select <i>Tables</i> . ___ In the <i>Contents</i> pane, scroll down until you reach the ALBUMS table. ___ Right-click and select ALBUMS and then select <i>Alter</i> from the pop-up menu.
GUI part 2	Define the primary key. ___ Select <i>Keys</i> tab. ___ Click the <i>Add Primary</i> button. ___ Select ITEMNO in the <i>Available columns</i> list. Click the single > button to add ITEMNO to the <i>Selected columns</i> box. ___ Click OK.
GUI part 3	Define the foreign key. ___ Click the <i>Add Foreign</i> button. ___ Indicate the following: ___ Foreign key name of <b>fkartno</b> . ___ Parent table Schema name of <b>student</b> . ___ Parent table Table name of <b>Artists</b> . ___ Note that ARTNO appears in the <i>Primary key</i> box. ___ Select ARTNO in the <i>Available columns</i> list and click the single > button to add ARTNO to the <i>Foreign key</i> box. ___ Indicate the following: ___ <i>Delete rule</i> (On delete) of <b>Cascade</b> ___ Click OK. ___ Click OK.



Command Editor	<pre>alter table albums primary key (itemno) add constraint fkartno foreign key (artno) references artists on delete cascade</pre>
----------------	--

**A primary key must be associated with a unique index. If an unique index already exists, it is used. If an unique index does not exist, DB2 creates one for you. If using the Control Center, the index name will be CC followed by a set of numbers. If using the command line interface, the index name will be SQL followed by a set of numbers.**

- \_\_\_ 2. Now alter the STOCK table to define its referential integrity relationships, by executing the script file, `crri`.

Command Window	<code>db2 -tvf crri</code>
GUI	<p>___ Select <i>Commands</i> tab on the <i>Command Editor</i> panel.</p> <p>___ Select <i>Selected -&gt; Open</i> from the menu bar.</p> <p>___ Verify that the System name is your local workstation.</p> <p>___ In the <i>Directories</i> list, select the <code>C:\labfiles\cf23</code> directory by clicking twice.</p> <p>___ Scroll down the <i>Files list</i> until you reach a file named <code>crri</code> and select it.</p> <p>___ Click <b>OK</b>.</p> <p>___ Press <b>Ctrl-Enter</b> to execute the script.</p>

- \_\_\_ 3. Information about referential integrity can be found by accessing the SYSCAT.REFERENCES view. Table dependencies can also be accessed through the SYSCAT.TABLES view.

Enter an interactive SQL statement to select CONSTNAME, TABNAME, REFTABSCHEMA, REFTABNAME, DELETERULE from the SYSCAT.REFERENCES view with a TABSCHEMA of `student`. You may also wish to use the SUBSTR function in your SQL statement to select the first 8 bytes of FK\_COLNAMES and PK\_COLNAMES.

The CONSTNAME is the name of the RI constraint defined at create time. This name may be used to alter or drop the constraint.

The DELETERULE states the action to take if a parent row is to be deleted and dependent rows also exist. If the DELETERULE = "N", it means Set Null. If the DELETERULE = "C", it means Cascade. If the DELETERULE = "R", it means RESTRICT.

Select TABNAME, PARENTS, and CHILDREN from the SYSCAT.TABLES view with a TABSCHEMA of `student`.

PARENTS is the number of referential constraints in which this table is a dependent.  
 CHILDREN is the number of referential constraints in which this table is a parent.

Command Window	<pre>db2 "select substr(constname,1,18),       substr(tabname,1,18),       substr(reftabschema,1,14),       substr(reftabname,1,14),       deleterule,       substr(fk_colnames,1,14),       substr(pk_colnames,1,14) from syscat.references where tabschema= user"  db2 "select substr(tabname,1,18), parents, children from syscat.tables where tabschema = user and (parents &gt; 0 or children &gt; 0) order by 2, 3 desc"</pre>
GUI	<p>From the DB2 Command Editor, Commands tab:</p> <pre>select constname, tabname, reftabschema, reftabname,       deleterule, fk_colnames, pk_colnames from syscat.references where tabschema= 'student';  select tabname, parents, children from syscat.tables where tabschema = 'student' and (parents &gt; 0 or children &gt; 0) order by 2, 3 desc;</pre>

- \_\_\_ 4. Run a select to retrieve the first 18 characters of TABNAME, and the PARENTS and CHILDREN columns from SYSCAT.TABLES where the TABSCHEMA is equal to your userid. Order the results by TABNAME and direct your output to a file named richk.file. The SQL can be found in a file named richk.sql.

Check your output against the richk.master file.

**Ensure that there are no differences between your output and the master output, or the following labs may not work correctly!**

GUI	<p>On a DB2 <i>Command Window</i>:</p> <pre>db2 -tf richk.sql &gt; richk.file more richk.file fc /w richk.file richk.master   more</pre> <p>If you see a message that indicates: <b>FC: no differences encountered</b> your table definitions match our master file.</p> <p>If you see any output, there are lines that do not match between the files. The first set of lines will indicate the information in your output, the second set of lines will indicate the information in the master output. The first column will indicate the table that has a different definition from our solution; the second column indicates how many relationships it is in as a dependent; the third column indicates how many relationships it is in as a parent.</p> <p>Examine this information, go back to the information in the lab that directed you on what you should set up, and alter the table with a problem (or its dependent) to correct the problem. Come back to this step and check it again.</p> <p>contents of <b>richk.sql</b>: select substr(tabname,1,18), parents, children from syscat.tables where tabschema = user order by tabname</p>
-----	--

## Section 6 - Add Check Constraints

In this section, you will be adding a check constraint to the STOCK table.

- \_\_\_ 1. Alter the STOCK table and add a check constraint to it. The check constraint should have the following characteristics.
  - It should be named **cctype**.
  - The business rule it should enforce is to only allow the values of 'D', 'C', or 'R' in the type column. **Make sure the alphanumeric literals are typed in uppercase letters.**

Command Window	<pre>db2 "alter table stock add constraint cctype check (type in ('D', 'C', 'R'))"</pre> <p>Note: 'D','C' and 'R' must be uppercase.</p>
GUI	<p>___ Using the DB2 <i>Control Center</i>, from the <b>MUSICDB</b> objects list, left-click and select <i>Tables</i>.</p> <p>___ In the <i>Contents</i> pane, scroll down until you reach the <b>STOCK</b> table.</p> <p>___ Right-click and select <b>STOCK</b> and select <i>Alter</i> from the pop-up menu.</p> <p>___ Select <i>Check Constraints</i> tab.</p> <p>___ Click the <b>Add</b> button.</p> <p>___ Enter a value of <b>cctype</b> in the <i>Check name</i> box.</p> <p>___ In the <i>Check condition</i> box, enter the line shown below:        Type in ('D','C','R')  <b>Note:</b> 'D', 'C', and 'R' must be uppercase. This indicate that the "type" column is only allowed to have the values of 'D', 'C', or 'R'.</p> <p>___ Click OK.</p> <p>___ Click OK.</p>
Command Editor	<pre>alter table stock add constraint cctype check (type in ('D', 'C', 'R'))</pre>

- \_\_\_ 2. Information about check constraints can be found by accessing the SYSCAT.CHECKS, SYSCAT.COLCHECKS, SYSCAT.TABCONST, and SYSCAT.TABLES views. Issue SQL to:

Select CONSTNAME, TABNAME, COLNAME from the SYSCAT.COLCHECKS view.

Select CONSTNAME, TABNAME, TYPE from the SYSCAT.TABCONST view. If the TYPE = "K", then it is a check constraint. If the Type = "P", then it is a primary key. If the Type = "F", then it is a foreign key.

Command Window	<pre>db2 "select substr(constname,1,18) as constname,       substr(tabname,1,15) as tabname,       substr(colname,1,15) as colname       from syscat.colchecks"  db2 "select substr(constname,1,18),       substr(tabname,1,15),       type       from syscat.tabconst"</pre>
GUI	<p>From the <i>Command Editor</i>:</p> <pre>select constname, tabname, colname from syscat.colchecks;  select constname, tabname, type from syscat.tabconst;</pre>

- \_\_\_ 3. To check your work, run a select statement that selects the first 100 characters of the TEXT column from SYSCAT.CHECKS where CONSTNAME = 'CCTYPE', and direct your output to a file named ckchk.file. The SQL can be found in a file named ckchk.sql.

Check your output against the ckchk.master file.

**Ensure that there are no differences between your output and the master output, or the following labs may not work correctly!**

GUI	<p>On a DB2 <i>Command Window</i>:</p> <pre>db2 -tf ckchk.sql &gt; ckchk.file more ckchk.file fc /w ckchk.file ckchk.master   more</pre> <p>If you see a message that indicates:</p> <p><b>FC: no differences encountered</b></p> <p>your table definitions match our master file.</p> <p>If you see any output, there are lines that do not match between the files. The first set of lines will indicate the information in your output, the second set of lines will indicate the information in the master output. If the differences are just spacing <b>between</b> the items (not within the quotes), then the differences are OK. If there are any other differences, you will need to drop the constraint you have defined, and define it again correctly. Then, come back to this step and check it again.</p> <p>ckchk.sql contains: select substr(text,1,100) from syscat.checks where constname = 'CCTYPE'</p>
-----	---

## Section 7 - Create a Trigger

In this section, you will be creating a trigger for the REORDER table. When an inventory item in the STOCK table falls below a quantity of 6, the REORDER table will have a row inserted.

- \_\_\_ 1. Create a trigger which has the following characteristics.
- It should be named reorder
  - It should fire after an update of the qty column on the STOCK table, if the new value of qty is  $\leq 5$
  - New should be referenced as n
  - The triggered action should insert the values n.itemno and current timestamp into the REORDER table
  - For each row mode db2sql
- 
- 

Command Window	<pre>db2 "create trigger reorder       after update of qty on stock       referencing new as n       for each row       mode db2sql       when (n.qty &lt;=5)       insert into reorder values (n.itemno, current timestamp) "</pre>
GUI part 1	<p>___ From the <b>MUSICDB</b> objects list, right-click and select <i>Triggers</i> and choose <i>Create</i> from the pop-up menu.</p> <p>___ Indicate the following:</p> <p>___ Trigger schema of <b>student</b>.</p> <p>___ Trigger name of <b>reorder</b>.</p> <p>___ Table schema of <b>student</b>.</p> <p>___ Table name of <b>Stock</b>.</p> <p>___ Select <i>After</i> radio button under <i>Time to trigger</i> action.</p> <p>___ Select <i>Update of Columns</i> radio button under <i>Operation</i> that causes the trigger to get executed.</p> <p>___ Select <b>QTY</b> in the <i>Update of columns</i> box.</p>

GUI part 2	Define the SQL statement to invoke when the trigger is fired. ___ Select <i>Triggered Action</i> tab. ___ Enter a value of <i>n</i> in the <i>Correlation name for the new rows</i> box. ___ Click the <i>Clear</i> button. ___ Enter the following SQL into the <i>Triggered Action</i> box: when (n.qty <= 5) insert into reorder values (n.itemno,current timestamp) ___ Click <i>Show SQL</i> and look at the SQL statement. ___ Click <i>OK</i> .
---------------	--

- \_\_\_ 2. Information about triggers can be found by accessing the SYSCAT.TRIGGERS and SYSCAT.TRIGDEP views. Enter interactive SQL to research the REORDER trigger:

Select TRIGNAME, TABNAME, and TRIGEVEN columns from the SYSCAT.TRIGGERS view. TRIGEVEN describes the event that fires the trigger as I for insert, D for delete, or U for update.

Select TRIGNAME, BTYPE, BSHEMA, BNAME columns from SYSCAT.TRIGDEP view. BTYPE and BSHEMA give the name of the objected depended on by the trigger. BTYPE describes the type of base object as A for alias, F for function instance, T for table, or V for view.

Command Window	<pre>db2 "select substr(trigname,1,18),         substr(tabname,1,18),         trigevent         from syscat.triggers where trigname = 'REORDER'"  db2 "select substr(trigname,1,18),         btype,         substr(bschema,1,14),         substr(bname,1,14)         from syscat.trigdep where trigname = 'REORDER'"</pre>
GUI	From the <i>Command Editor, Commands</i> tab: <pre>select trigname, tabname, trigevent         from syscat.triggers where trigname = 'REORDER'; select trigname, btype, bschema, bname         from syscat.trigdep where trigname = 'REORDER';</pre>

## Section 8 - OPTIONAL - Working with XML

In this section, you will be querying a table with an XML data column.

\_\_\_ 1. If the SAMPLE database has not yet been create, create it now.

Command Window	<code>db2samp1 -xml</code>
GUI	<p>___ From the <i>First Steps</i> window, select <i>Database Creation, Create SAMPLE Database</i>.</p> <p>___ Indicate the following:</p> <p>___ The <i>XML and SQL objects and data</i> radio button should be selected.</p> <p>___ Click OK.</p>

\_\_\_ 2. Connect to the SAMPLE database.

\_\_\_ 3. Use XQUERY and the **db2-fn:xmlcolumn** function to retrieve all of the XML documents from the customer table's info column.

---



---

Command Window	<code>db2 "xquery db2-fn:xmlcolumn('CUSTOMER.INFO')"</code>
GUI	<p><b>XML code is case sensitive, so be careful about how you enter commands.</b></p> <p>___ Using the <i>DB2 Command Editor</i>, connect to the <b>SAMPLE</b> database.</p> <p>___ In the upper pane, enter the following:</p> <p style="padding-left: 40px;"><code>xquery db2-fn:xmlcolumn('CUSTOMER.INFO');</code></p> <p>___ Execute the query by pressing <b>CTRL-Enter</b>.</p>

\_\_\_ 4. Now use XQUERY with SQL and **db2-fn:sqlquery** function to retrieve all of the XML documents from the customer table's info column.

---



---

Command Window	<code>db2 "xquery db2-fn:sqlquery('select info from customer')"</code>
----------------	--



GUI	<p>Note that you are executing an SQL query within the XML function. In this way, you can retrieve both XML data and relational data in the same query.</p> <p>___ Using the DB2 <i>Command Editor</i>, connect to the <b>SAMPLE</b> database.</p> <p>___ In the upper pane, enter the following:</p> <pre>xquery db2-fn:sqlquery('select info from customer');</pre> <p>___ Execute the query by pressing <b>CTRL-Enter</b>.</p>
-----	---

- \_\_\_ 5. Use XQUERY and the **db2-fn:sqlquery** function to retrieve, from the INFO column in the CUSTOMER table, all customers where the customer ID (column CID) equals 1002.

Command Window	<pre>db2 "xquery db2-fn:sqlquery('select info from customer where cid = 1002')"</pre>
GUI	<p>___ Using the DB2 <i>Command Editor</i>, connect to the <b>SAMPLE</b> database.</p> <p>___ In the upper pane, enter the following:</p> <pre>xquery db2-fn:sqlquery('select info from customer where cid = 1002');</pre> <p>___ Execute the query by pressing <b>CTRL-Enter</b>.</p>

- \_\_\_ 6. Use the DB2 Control Center and the XML Document View to display the CUSTOMER.INFO document in the first row of the table. View the document in both Tree view and Source view.

GUI	<ul style="list-style-type: none"><li>___ Using the <i>DB2 Control Center</i> connect to the <b>SAMPLE</b> database.</li><li>___ From the <b>SAMPLE</b> objects list, select <i>Tables</i>, then double-click the <b>CUSTOMER</b> table icon.</li><li>___ On the first row of the table in the <i>Open Table</i> panel, click the three dots (...) in the <b>INFO</b> column.</li><li>___ The <i>XML Document Viewer</i> panel opens, displaying the <i>Tree</i> view of the document. Note that you can expand the various elements to view the information.</li><li>___ Click the <i>Source</i> view to see the actual document code.</li><li>___ Click the <i>Preferences</i> button and select <i>Format Text</i>, then click <b>OK</b>. Note the structural view.</li></ul>
-----	--

**END OF LAB**

## Exercise 6. Moving data

*(with hints)*

### Estimated time

01:15

### What this exercise is about

This exercise is an online lab in which you load your tables with data using the load utility and import utility.

### What you should be able to do

At the end of the lab, students should be able to:

- Use the import utility to insert data from a file into a table
- Use the load utility to fast load data from a file into a table
- Create exception tables
- Manage check constraints, triggers, and check pending status

### Introduction

Now that the database objects have been created in your MUSICDB database, it is time to load data into the tables. This will be accomplished by using the import and load utilities.

**Note:** You will be using scripts to do some of the steps within this lab.

With some codeset and territory values, you may need to use the MODIFIED BY FORCEIN syntax on the Load or Import statements. Your instructor will let you know if this is a requirement for your environment.

## ***Instructor Notes:***

*Introduction* — This exercise is an online lab which loads the student's tables using both the load utility and the import utility.

*Time for Lab* — 75 minutes.

*Things to Review at End of Lab* — At the end of this lab, make sure the students were successful in getting all their tables loaded.

**Note:** Optional lab dealing with rowcount and restartcount was removed since restartcount is no longer a documented option, and Patti couldn't figure out how to do rowcount in the GUI in V8.

## Exercise instructions with hints

The lab solutions using primarily your telnet session and commands are indicated as *Command Window*. The lab solutions using primarily the GUI tools are indicated as *GUI*.

### Section 1 - Import Data in Tables

In this section, you will be using import to load data into tables. You will import data into the ARTISTS table. Then you will execute a script file to import data into the ALBUMS and STOCK tables.

- \_\_\_ 1. Import data into your ARTISTS table. For the IMPORT command, the data must be locally available to the machine from which you issue the import.

The import should specify the following:

- The import file name is `artists.exp`
- The import file is in ixf format
- Import in insert mode
- Write any messages to a file called `art.msg`

Command Window	<pre>db2 connect to musicdb db2 "? import"   more db2 "import from C:\labfiles\cf23\artists.exp of ixf messages C:\labfiles\cf23\art.msg insert into artists"</pre>
GUI part 1	<p>___ On the <i>Control Center</i>, from your MUSICDB database objects list, select <i>Tables</i> by clicking with your left mouse button.</p> <p>___ In the contents pane, scroll down the list of tables until you reach ARTISTS.</p> <p>___ Select ARTISTS with your right mouse button and select <i>Import</i> from the pop-up menu.</p>
GUI part 2	<p>Define the import parameters.</p> <p>___ Enter a name of <code>C:\labfiles\cf23\artists.exp</code> in the Import file box.</p> <p>___ Make sure the <i>Integrated Exchange Format</i> radio button is selected.</p> <p>___ Make sure the Import mode specified is <i>Insert</i>.</p> <p>___ Enter a file name of <code>C:\labfiles\cf23\art.msg</code> in the <i>Message file</i> box.</p> <p>___ Click <i>Show Command</i> and look at the syntax of the import command. (You may need to scroll down to select Show Command.)</p> <p>___ Close <i>Show Command</i> window.</p> <p>___ Click OK. You may need to scroll down to select OK.</p>

- \_\_\_ 2. Look at the file `art.msg` to determine how many rows were inserted and any error or warning messages that may have been generated.

Number of Rows Committed:

---

Command Window	<code>more &lt; art.msg</code>
----------------	--------------------------------

**Number of Rows Committed: 79**

**The warning message states:**

SQL3050W Conversions on the data will be made between the IXF file code "850" and the application code page "819". (The application code page that your message indicates may be different - it depends on the operating system where you are running the command.)

**No it is not a problem. The warning is just informing you that the `artists.exp` file was created with the export command from a table in a database on a different system that had a code page of 850, and that your application uses a code page of 819, so a conversion was done.**

- \_\_\_ 3. Now you will import data into the ALBUMS and STOCK tables using script files.

The file you will use to import data into the ALBUMS table is called `imp_albu`.

Command Window	<code>db2 -tvf imp_albu</code>
GUI	From the <i>Command Editor, Commands</i> tab: ___ Select <i>Selected</i> -> <i>Open</i> from the menu bar. ___ Find the file <code>C:\labfiles\cf23\imp_albu</code> and select it. Note that the <i>Path</i> box gets filled in with the <code>imp_albu</code> file name. ___ Click <b>OK</b> . ___ Press <b>Ctrl-Enter</b> to execute the script.

- \_\_\_ 4. Record the number of rows committed. Check `albums.msg` to see if any additional messages occurred.

Number of Rows Committed:

---

**Number of Rows Committed: 264**

**The messages state that 264 rows were committed successfully.**

- \_\_\_ 5. The file you will use to import data into the STOCK table is called `imp_sto`.

Command Window	<code>db2 -tvf imp_sto</code>
GUI	<p>From the <i>Command Editor</i>, <i>Commands</i> tab:</p> <p>___ Select <i>Selected</i> -&gt; <i>Open</i> from the menu bar.</p> <p>___ Find the file <code>C:\labfiles\cf23\imp_sto</code> and select it. Note that the <i>Path</i> box gets filled in with the <code>imp_sto</code> file name.</p> <p>___ Click OK.</p> <p>___ Press <b>Ctrl-Enter</b> to execute the script.</p>

- \_\_\_ 6. Record the number of rows committed. Check `sto.msg` to see if any additional messages occurred.

Number of Rows Committed:

---

**Number of Rows Committed: 778**

**The messages state that 778 rows were committed successfully.**

- \_\_\_ 7. For all the imports, where was the source file located that was being imported, on the Windows machine or on the UNIX Database Server?

---

**The source file was located on the machine where the IMPORT command was issued.**

- \_\_\_ 8. Remember that at the end of the previous lab, you created a trigger on the STOCK table that should put rows into the REORDER table if any rows were updated to have a quantity of less than or equal to 5 in the STOCK table.

Check to see if the trigger was fired during the Import utility. Explain what is returned.

Command Window	<pre>db2 connect to musicdb db2 "select * from stock where qty &lt;= 5" db2 "select * from reorder"</pre>
GUI	<p>From the <i>Command Editor</i>:</p> <pre>connect to musicdb; select * from stock where qty &lt;= 5; select * from reorder;</pre>

The import utility does fire triggers, however, the trigger that you defined was after UPDATE, not inserts. Our Import statements are inserts not updates so the trigger is not fired.



## Section 2 - Create Exception Tables - ARTISTS, ALBUMS, STOCK

In this section, you will be creating the exception tables that will be required to Load the ARTISTS, ALBUMS, and STOCK tables using the Load Utility. You will create the exception tables by executing a script file.

- \_\_\_ 1. It is normally a good idea to use exception tables during a LOAD operation. We have provided a script named `crexptab` that will create exception tables for ARTISTS, ALBUMS, and STOCK.

Make sure each table is created successfully.

Note that the table space SMSEXP is being created to store all of the exception tables. Also note that the exception tables mimic the base table plus two additional columns. What are the exception table names? What are the two additional columns and their attributes? Are they optional?

---



---

Command Window	<code>db2 -tvf crexptab</code>
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>crexptab</code> . ___ Run the script.

**ARTEXP is the exception table for ARTISTS, ALBEXP is the exception table for ALBUMS, and STOEXP is the exception table for STOCK.**

**The n+1 column is optional. It is called `ts` and must be a **TIMESTAMP**. The n+2 column is also optional. It is called `msg` and must be **CLOB (32 KB) or greater in length**. Both `ts` and `msg` are user-defined column names.**

## Section 3 - Backup Database

### IMPORTANT: Make a BACKUP of the DATABASE!

More details about database backup/restore comes in later lecture/lab. In Section 5, you will issue a LOAD REPLACE command. You should always complete a table space backup or full database backup before a load replace if you want the option of recovering the original table space. This backup can always be used during class to recover your database to this point in time.

- \_\_\_ 1. The backup will be done against your database server and will use a directory path of `c:\backup`. Using your **Command Window**, enter the following commands:

`cd`

```
mkdir backup
db2 force application all
db2 "backup db musicdb to C:\backup"
```

The backup will take several minutes to run.

- \_\_\_ 2. Verify the backup completed successfully by checking the messages that appear on your screen. What is its status?

---

**The job status should show Successful.**

**Only if errors occur later in lab and need to restore.**

**STOP!** Do not complete the restore unless errors occur in the following steps which cannot be recovered. If this happens, then complete the restore steps to restore your database from the backup taken in the previous step. Database Recovery will be discussed in detail at a later time.

Only if a restore is required, enter the following commands from a telnet window:

```
db2 force application all
db2 "restore database musicdb from C:\backup"
```

## Section 4 - Load Data with INSERT option into CONCERTS

In this section, you will be LOADING the CONCERTS table using the INSERT LOAD option.

### \_\_\_ 1. LOAD data into your CONCERTS table.

The load should specify the following:

- Work with files in the path `C:\labfiles\cf23`
- The load file name is `concerts.exp`
- The import file is in ixf format
- Load in insert mode
- Write any messages to a file called `concerts.msg`

Command Window	<pre>db2 connect to musicdb db2 "? load"   more db2 "load from C:\labfiles\cf23\concerts.exp of ixf messages C:\labfiles\cf23\concerts.msg insert into concerts"</pre>
GUI	<p>In the <i>Control Center</i>, from your <b>MUSICDB</b> database objects list, left-click and select <i>Tables</i>.</p> <p><b>Note: you may need to attach to your instance and/or connect to the database as the force application command in the previous section would have removed your connection.</b></p> <p>In the right pane, scroll down the list of tables until you reach <b>CONCERTS</b>. Right-click and select <b>CONCERTS</b> and then select <i>Load</i> from the pop-up menu. (You may need to refresh the list of tables if you do not see the <b>CONCERTS</b> table.)</p> <p>___ On the <i>Type</i> window, indicate <i>Append data to table</i></p> <p>___ On the <i>Files</i> window, indicate:</p> <p>___ <i>Input file format of Integrated Exchange Format (IXF)</i></p> <p>___ <i>Input file location of Server</i></p> <p>___ Full path and filename of input files of: <b>(Windows): C:\labfiles\cf23\concerts.exp</b></p> <p>___ Full path and filename to store progress messages of: <b>(Windows): C:\labfiles\cf23\concerts.msg</b></p> <p>___ On the <i>Schedule</i> window, indicate to <i>Run now</i> without saving task history.</p> <p>___ On the <i>Summary</i> window, view the command, and click <b>Finish</b> to run the command.</p>
Command Editor	<p>On the <i>Command Editor</i>, issue:</p> <pre>load client from C:\labfiles\cf23\concerts.exp of ixf messages C:\labfiles\cf23\concerts.msg insert into concerts</pre>

### \_\_\_ 2. Record the number of rows successfully loaded and committed from the messages file.

Number of Rows Committed:

---

**Number of Rows Committed: 10**

## Section 5 - Load Data with REPLACE option into ARTISTS

This section has you replace the data in the ARTISTS table using the LOAD utility.

- \_\_\_ 1. LOAD data into the ARTISTS table using a script file named `load_art`.

Command Window	<code>db2 -tvf load_art</code>
GUI	On the DB2 <i>Command Editor</i> , <b>Commands</b> tab: ___ Retrieve the script file <code>load_art</code> . ___ Run the script.

- \_\_\_ 2. The LOAD command loads rows into the ARTISTS table using a **different** IXF input file from the earlier IMPORT. Note that it uses the REPLACE option to replace the previously imported rows in the ARTISTS table. The artexp table is specified to contain exception rows for the ARTISTS table during the load with the FOR EXCEPTION OPTION. What type of violations will be stored in this table during the load?

**Unique key violations will be stored in the exception table during the load.**

- \_\_\_ 3. Record the number of rows committed.

Number of Rows Committed:

**Number of Rows Committed: 78**

- \_\_\_ 4. Check the ARTEXP table to see if any exception rows violated the unique key index on the ARTISTS table. Use the script file `selexp`.

Command Window	<code>db2 -tvf selexp   more</code>
GUI	On the DB2 <i>Command Editor</i> , <b>Commands</b> tab: ___ Retrieve the script file <code>selexp</code> . ___ Run the script.

**No rows are in the ARTEXP table because no rows were inserted with duplicate unique key values (ARTNO has a unique index since it is a primary key).**

\_\_\_ 5. Check the load\_art.msg file for messages.

---

---

Command Window	more load_art.msg
GUI	From a DB2 <i>Command Window</i> : more < load_art.msg

**The messages indicate that the load occurred successfully.**

## Section 6 - Examining the SET INTEGRITY PENDING State

In this section, you will learn how to clear SET INTEGRITY PENDING status from tables using the SET INTEGRITY command.

- \_\_\_ 1. Try to select from the ARTISTS, STOCK, and ALBUMS tables using the script `selstab`.

Command Window	<code>db2 -stvf selstab   more</code>
GUI	On the DB2 <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>selstab</code> . ___ Run the script.

- \_\_\_ 2. What SQL code did you get on the first select statement? Check the meaning of the SQL code using the online help facilities. What action do you need to take to clear this status?

**Received SQL0668 reason code 1 when trying to access the ARTISTS table.**

**You are not allowed to access the recently loaded table with DML because it is in a check pending state. The SQL0668 message tells you similar information. This indicates that you need to run the SET INTEGRITY command against the ARTISTS table.**

- \_\_\_ 3. Check the check constraints status of the tables by executing the script `listtbst` and record the status.

Table Name	STATUS	FK_CHECKED	CC_CHECKED
ARTISTS			

Command Window	<code>db2 -tvf listtbst</code>
GUI	On the DB2 <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>listtbst</code> . ___ Run the script.

**The ARTISTS table is in a check pending state.**

Table Name	STATUS	FK_CHECKED	CC_CHECKED
ARTISTS	C	Y	Y

- \_\_\_ 4. The SET INTEGRITY SQL statement can be used to check for referential constraint exceptions. Exception tables should be specified for all tables in a check pending state (in this case, just ARTISTS). Run the script `setcsts_art` to check for referential constraint exceptions in the ARTISTS table.

Command Window	<code>db2 -tvf setcsts_art</code>
GUI	On the DB2 Command Editor, Commands tab: ___ Retrieve the script file <code>setcsts_art</code> . ___ Run the script.

- \_\_\_ 5. What does the warning SQL3601 mean?

---



---

(telnet or Command Window) `db2 "? sql3601"`  
 or  
 (Command Editor) `? sql3601`

**SQL3601 indicates that the SET INTEGRITY statement caused one or more tables to be placed in the check pending status.**

- \_\_\_ 6. Check the check constraints status of the tables by executing the script `listtbst` and record the status.

Table Name	STATUS	FK_CHECKED	CC_CHECKED
ALBUMS			
STOCK			

Command Window	<code>db2 -tvf listtbst</code>
GUI	On the DB2 Command Editor, Commands tab: ___ Retrieve the script file <code>listtbst</code> . ___ Run the script.



**The ALBUMS and STOCK tables are in a check pending state.**

Table Name	STATUS	FK_CHECKED	CC_CHECKED
ALBUMS	C	N	Y
STOCK	C	N	Y

- \_\_\_ 7. The SET INTEGRITY SQL statement can be used to check for referential constraint exceptions. Exception tables should be specified for all tables in a check pending state (in this case, ALBUMS and STOCK). Run the script `setcsts_2` to check for referential constraint exceptions in the ALBUMS and STOCK tables.

Command Window	<code>db2 -tvf setcsts_2</code>
GUI	On the DB2 <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>setcsts_2</code> . ___ Run the script.

- \_\_\_ 8. Select from the tables to verify that the check pending status has been removed from the tables. Run the scripts `listtbst` and `selstab`.

Were there any constraint pending conditions?

Command Window	<code>db2 -tvf listtbst</code> <code>db2 -tvf selstab   more</code>
GUI	On DB2 <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>listtbst</code> . ___ Run the script. ___ Retrieve the script file <code>selstab</code> . ___ Run the script.

**You should see no constraints pending. You should be able to select from ARTISTS, STOCK, and ALBUMS.**

- \_\_\_ 9. Where are the rows that were moved out of the tables in order to get the table out of constraint pending? How can you see them?

**The rows would have been moved into the exception tables.**

- \_\_\_ 10. Examine the rows that were moved to your exception tables using the script file `selexp`.

Command Window	<code>db2 -tvf selexp</code>
GUI	On the DB2 Command Editor, Commands tab: <input type="checkbox"/> Retrieve the script file <code>selexp</code> . <input type="checkbox"/> Run the script.

- \_\_\_ 11. The rows that were moved from the STOCK table to STOEXP and from the ALBUMS table to ALBEXP were foreign key rows that did not have a matching parent key row in the ARTISTS table. You wish to carry this artist's work. Execute SQL to insert a row for item number 100, title of "Patti & Cart Wheels", classification of "S" into the ARTISTS table.

Command Window	<code>db2 "connect to musicdb"</code> <code>db2 "insert into artists (artno, name, classification) values (100, 'Patti &amp; Cart Wheels', 'S')"</code>
GUI	<code>connect to musicdb user;</code> <code>insert into artists (artno, name, classification) values (100, 'Patti &amp; Cart Wheels', 'S');</code>

- \_\_\_ 12. Now you wish to use the exception table rows in STOEXP and in ALBEXP to insert rows into the STOCK and ALBUMS tables, respectively. View the rows that are in the two exception tables by using the script called `selexp`. Note that you do not want to insert the N+1 and N+2 columns into the STOCK and ALBUMS tables. Examine the MSG column in the two exception tables.

---



---

**Note there are two rows in ALBUMS for ARTNO 100 with ITEMNO of 300 and 301. There is 1 row in STOCK for ITEMNO 300.**

- \_\_\_ 13. Run a script called `insexp` to insert the rows in the exception tables into the STOCK and ALBUMS tables. The script also selects from the ALBUMS and STOCK tables to see if the rows were inserted properly.

Command Window	<code>db2 -tvf insexp</code>
----------------	------------------------------

GUI	On the DB2 <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>insexp</code> . ___ Run the script.
-----	--

**The rows are inserted into the base tables.**

## Section 7 - Understanding CHECK CONSTRAINT Enforcement

- \_\_\_ 1. Previously you created a check constraint on the TYPE column in the STOCK table. Review the constraint requirements.

---



---

Command Window	db2 "select substr(text,1,30) from syscat.checks where tabname = 'STOCK'"
GUI	<p>On the <i>Control Center</i>, from your MUSICDB database objects list, right-click and select <i>Tables</i>.</p> <p>___ Refresh the list of tables.</p> <p>___ In the <i>Contents</i> pane, scroll down the list of tables until you reach <b>STOCK</b>.</p> <p>___ Right-click and select <b>STOCK</b> and then select <i>Alter</i> from the pop-up menu.</p> <p>___ Select <i>Check Constraints</i> tab.</p> <p>___ Note the check constraint requirement.</p> <p>___ Click <b>Cancel</b>.</p>

- \_\_\_ 2. The hot new band, Double Dare, has made a Music Video that they want you to carry. Connect to the database and attempt to insert stock data for this artist with an ITEMNO of 302, TYPE of V, PRICE of 100.00, and QTY of 20.

Why did you get the message SQL0545N?

---



---

Command Window	db2 "insert into stock values (302,'V',100.00,20) "
GUI	<p>On the <i>Command Editor</i>:</p> <p>insert into stock values (302,'V',100.00,20)</p>

**SQL0545 means the requested operation is not allowed because a row does not satisfy the check constraint .STOCK.CCTYPE.**

- \_\_\_ 3. Reissue the insert SQL statement with the TYPE = 'C'. You have decided to only carry the CD type.

---



---

Command Window	db2 "insert into stock values (302,'C',100.00,20) "
GUI	On the <i>Command Editor</i> : insert into stock values (302,'C',100.00,20)

**This insert should work since C is a valid type defined in the CHECK CONSTRAINT on the TYPE column in the STOCK table.**

## Section 8 - Understanding TRIGGER Enforcement

- \_\_\_ 1. Previously you created a trigger on the QTY column in the STOCK table. Review the trigger requirements, by selecting from the SYSCAT.TRIGGERS view.

Under what conditions would the trigger be fired?

---



---

Command Window	db2 "select substr(text,1,200) from syscat.triggers where tabname = 'STOCK'"  more
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: select substr(text,1,200) from syscat.triggers where tabname = 'STOCK'

**The trigger is fired when an update to the QTY column of the STOCK table causes the value to be equal to or less than 5. The action is to insert a row into the REORDER table.**

- \_\_\_ 2. The Double Dare CD, ITEMNO 302, entitled "I Dare You" is selling very fast. Update the STOCK table to reflect the current inventory of 3. Note that no messages occur when a trigger is fired.

---



---

Command Window	db2 "update stock set qty = 3 where itemno = 302"
GUI	On the <i>Command Editor</i> : update stock set qty = 3 where itemno = 302

**Update is successful, but no messages occur when the trigger is fired.**

- \_\_\_ 3. Query the REORDER table to see if the TRIGGER was fired during the previous update to the QTY in the STOCK table for ITEMNO 302.

---



---

Command Window	db2 "select * from reorder"
GUI	On the <i>Command Editor</i> : select * from reorder

```

ITEMNO    TIMESTAMP
-----    -
      302    yyyy-mm-dd-hh-mm-ss-nnnnnn

```

**When implementing triggers, you probably would implement a job that periodically queried the REORDER table and placed orders for new stock.**

- \_\_\_ 4. Previously you created an alias called EMPTYSTOCK for the REORDER table. Does it give the same results as querying directly against REORDER?

---



---

Command Window	db2 "select * from emptystock"
GUI	On the <i>Command Editor</i> : select * from emptystock

```

ITEMNO    TIMESTAMP
-----    -
      302    yyyy-mm-dd-hh-mm-ss-nnnnnn

```

**YES, the alias EMPTYSTOCK is a nickname for REORDER and gives the same results.**

## Section 9 - Using the db2look tool

In this section, you will use the db2look tool to extract the DDL for a table in the MUSICDB database.

- \_\_\_ 1. This is a command line tool you will use to extract the DDL for the ARTISTS table into a file. You would do this to create a look-alike table in another database, or to store the structure away for future use.

Windows	<pre>cd C:\labfiles\cf23 db2look -d musicdb -t artists -e -o artists.ddl</pre>
---------	--

Command Window	<pre>View this DDL : more artists.ddl</pre>
----------------	---

- \_\_\_ 2. Use the DB2 DESCRIBE tool and compare the results with the DDL file.

Command Window	<pre>cd db2 connect to musicdb db2 describe table artists</pre>
----------------	---



## Section 10 - OPTIONAL: Working with LOB Data

\_\_\_ 1. Load data including LOB columns into your ARTISTS table. The load should perform the following:

- The table you are loading is the ARTISTS table
- The file you are loading is `C:\labfiles\cf23\lob_in`
- The file format is del
- LOAD in INSERT mode
- Write any messages to a file called `lob.msg`
- The LOB data files are in `C:\labfiles\cf23\lobs`.

**Note:** These data files are called `queen.txt`, `queen.tif`, `beatles.txt`, and `beatles.tif`.

- "modified by lobsinfile" must be specified
- The load method to be used is `p(1,2,3,4,5)` - this specifies the numbers of the columns to be loaded
- The `lob_in` input file looks like the following:
- `42,"Beatles","P",beatles.txt,beatles.tif,`
- `68,"Queen","P",queen.txt,queen.tif,`

**Note:** The first three values are for the ARTNO, NAME, and CLASSIFICATION columns in ARTISTS table.

The `queen.txt` and `beatles.txt` are the text files where the data for the ARTISTS.BIO column which is a CLOB(100 KB) is stored.

The `queen.tif` and `beatles.tif` are the binary files where the data for the ARTISTS.PICTURE column which is a BLOB(500 KB) is stored.

Command Window	db2 "load from <code>C:\labfiles\cf23\lob_in</code> of del lobs from <code>C:\labfiles\cf23\lobs</code> modified by lobsinfile method <code>p(1,2,3,4,5)</code> messages <code>C:\labfiles\cf23\lob.msg</code> insert into artists"
GUI part 1	<p>On the DB2 Control Center, from your MUSICDB database objects list, left-click and select <i>Tables</i>.</p> <p>___ In the <i>Contents</i> pane, scroll down the list of tables until you reach <b>ARTISTS</b>.</p> <p>___ Right-click and select <b>ARTISTS</b> and then select <i>Load</i> from the pop-up menu.</p> <p>___ On the <i>Type</i> window, indicate <i>Append data to table</i>.</p>

GUI Part 2	___ On the <i>Files</i> window, indicate: ___ <i>Input file format of Delimited Text (DEL)</i> ___ <i>Input file location of Server</i> ___ Full path and filename of input files of: (Windows): C:\labfiles\cf23\lob_in ___ Full path and filename to store progress messages of: (Windows): C:\labfiles\cf23\lob.msg
GUI part 3	___ On the <i>Columns</i> window, check the check box to indicate <i>Use these directories to find large (LOB) object data</i> and enter directory name of: (Windows): C:\labfiles\cf23\lobs
GUI part 4	___ On the <i>Schedule</i> window, indicate <i>Run now without saving task history</i> . ___ On the <i>Summary</i> window, view the command, and click <b>Finish</b> to run the command.
Command Editor	(Windows): LOAD client FROM C:\labfiles\cf23\lob_in OF DEL LOBS FROM C:\labfiles\cf23\lobs MODIFIED BY LOBSINFILE METHOD P (1, 2, 3, 4, 5) MESSAGES C:\labfiles\cf23\lob.msg INSERT INTO ARTISTS (ARTNO, NAME, CLASSIFICATION, BIO, PICTURE);

- \_\_\_ 2. Look at the message file lob.msg. Near the bottom of the file, note that the SQL3509W message appears. Note that the number of rows deleted is two. From your telnet or DB2 Command Window, enter the command:

db2 ? SQL3509

- \_\_\_ 3. Did the two new rows get inserted into the ARTISTS table and replace the existing rows? Why not?

---



---

**SQL3509W The utility has deleted two rows from the table.**

**The rows were deleted from the table. The table ARTISTS has a primary key on the column ARTNO, and the LOAD just tried to insert a duplicate key value.**

- \_\_\_ 4. Why can't you use the REPLACE option on the LOAD?

---

**Because with the REPLACE option on the LOAD, you delete all rows and insert the new rows.**

- \_\_\_ 5. Is there an INSERT\_UPDATE option on the LOAD - similar to the IMPORT utility?

---

**No.**

- \_\_\_ 6. Instead of using load, let's use import. Import the data including LOB columns into your ARTISTS table, using the insert\_update option.

The import should specify the following:

- The file you are importing from is lob\_in
- **Note:** If you are doing remote administration, this is now a local file since we are importing and not loading.
- The import file is in del format
- IMPORT in INSERT\_UPDATE mode
- Write any messages to a file called imp\_lob.msg
- LOB data files are in the local directory - you must end the directory name with a / or \ - the syntax requires it.
- **Note:** These data files are called queen.txt, queen.tif, beatles.txt, and beatles.tif
- "modified by lobsinfile" should be specified
- The import method to be used is p(1,2,3,4,5) - this specifies the numbers of the columns to be loaded
- The lob\_in input file looks like the following:

```
42,"Beatles","P",beatles.txt,beatles.tif,
68,"Queen","P",queen.txt,queen.tif,
```

Command Window	db2 "import from C:\labfiles\cf23\lob_in of del lobs from C:\labfiles\cf23\lobs\ modified by lobsinfile method p(1,2,3,4,5) messages C:\labfiles\cf23\imp_lob.msg insert_update into artists"
GUI part 1	<p>On the DB2 Control Center, from your MUSICDB database objects list, left-click once and select <i>Tables</i>.</p> <p>___ In the <i>Contents</i> pane, scroll down the list of tables until you reach <b>ARTISTS</b>.</p> <p>___ Right-click and select <b>ARTISTS</b> and then select <i>Import</i> from the pop-up menu.</p>
GUI part 2	<p>Specify the file to be imported.</p> <p>___ Enter a name of C:\labfiles\cf23\lob_in in the <i>Import file</i> box.</p> <p>___ Select <i>Delimited ASCII format</i> radio button.</p> <p>___ Select pull-down arrow to the right of the <i>Import mode</i> box.</p> <p>___ Select <i>INSERT_UPDATE</i> from the list.</p> <p>___ Enter a file name of C:\labfiles\cf23\imp_lob.msg in the <i>Message file</i> box.</p>

GUI part 3	Specify the <i>LOB</i> information. ___ Select <i>Columns</i> tab. ___ Select checkbox next to <i>Use these directories to find large (LOB) object data</i> . ___ Enter a value of <b>C:\labfiles\cf23</b> in the <i>LOB paths</i> box.
GUI part 4	___ Click <b>OK</b> to execute the import. When the import is finished, select <b>Close</b> to clear the DB2 message.

- \_\_\_ 7. Check the results by executing the script file `selarti`. Are the two rows with the LOB data there?

---



---

Command Window	<code>db2 -tvf selarti</code>
GUI	On the <i>Command Editor</i> , <i>Command</i> tab: ___ Retrieve the script file <code>selarti</code> . ___ Run the script.

```

ARTNO      CLOB_BIO      BLOB_PICTURE
-----
42    Beatles -  x '49492A0044A10D0A0080'
68    Queen  -   x '49492A00DE350C00803F'

```

**Yes, the two rows were updated successfully. Note that the first 10 characters of the CLOB column are printing (that's what our select asked for), and the first 10 bytes (hex characters) of the BLOB column are printing. Let's explore how to read the CLOB data with a C program in the next steps.**

### The following only works on the Windows platform.

- \_\_\_ 8. From a DB2 Command Window session, look at the program `C:\labfiles\cf23\lobfile1.sqc`. The program creates a cursor and fetches the contents of the ARTISTS.BIO CLOB data from the MUSICDB database and writes the resume LOB data out as a file.

```
more < C:\labfiles\cf23\lobfile1.sqc
```

- \_\_\_ 9. The `lobfile1.sqc` program has already been precompiled, compiled and link-edited. You still need to bind it to your database (more information on all of this coming in the Application Programming topic). From the DB2 Command Window, enter the bind command shown below.

```
db2 connect to musicdb user student using ibm2blue
db2 bind C:\labfiles\cf23\lobfile1.bnd
```

- \_\_\_ 10. Execute the program lobfile1 from the cf23 directory in your DB2 Command Window session. Watch the messages on the screen, and record the filename where the resume is stored.

You will be prompted to supply the database name of MUSICDB, the USERID of student, the password of ibm2blue, and which artist's bio you which to see (choose 42 the first time you run the program). Record the file name the program responds with.

```
cd C:\labfiles\cf23
lobfile1
```

FILENAME: \_\_\_\_\_

```
Resume for ARTNO 42 is in file: res42.txt
```

- \_\_\_ 11. View the file you just created (res42.txt). From your Command Prompt session, enter:

```
more < res42.txt
```

---

Beatles -

The foursome started out in the early sixties with the name "Silver Beatles" but changed it to the Beatles. The original drummer in the band was Pete Best, but he was replaced by Ringo Starr before the band became famous. The foursome was composed of George Harrison, Ringo Starr, John Lennon and Paul McCartney.

The band was playing in Liverpool's Cavern nightclub when they were "discovered" by Brian Epstein, who became the band's manager. Many attribute the rise to popularity of the Beatles to Epstein's influence and marketing savy. Epstein himself did not live to see the later successes of the band.

Two of the most famous appearances of the band were at the Hollywood Bowl and on the Ed Sullivan show. At shows, the audiences screamed so loud that the boys from England couldn't hear themselves sing. The Beatles invented the vocal monitor, a speaker cabinet directed back toward the band, to overcome the loud audiences.

The Beatles dominated the airwaves through the sixties with smash albums and singles too numerous to name. However, in the late sixties the writing team of Lennon and McCartney began to have difficulties. Also contributing to the tension was the relative failure of Apple

Records, a label started by the Beatles. Ultimately, the band broke up in 1970. Their final album, "Let It Be", summarized the feelings of several band members.

- \_\_\_ 12. Reexecute the program `lobfile1`. This time specify the other ARTNO 68. Watch the messages on the screen, and record the filename where the resume is stored.

`lobfile1`

FILENAME: \_\_\_\_\_

Resume for ARTNO 68 is in file: `res68.txt`

- \_\_\_ 13. View the file you just created (`res68.txt`). From your Command Prompt session, enter:

`more < res68.txt`

---

Queen -

This band intermixed the dynamics of classically trained vocalists with rock musicians that tended to play on the harder side.

Freddie Mercury handled the lead vocals, and on albums would sing multiple parts. Brian May provided the main drive to the music with guitar riffs and power chords that identified the band.

## **END OF LAB**

# Exercise 7. Backup and recovery

*(with hints)*

## Estimated time

00:40

## What this exercise is about

This exercise is an online lab where the database MUSICDB will be used for backup and recovery. During this lab the students will prepare the database for archival logging, create a backup and restore and recover the database to the current as well as to a prior point in time.

## What you should be able to do

At the end of the lab, students should be able to:

- Create backup images of a database and of table spaces
- Restore a database from a backup image
- Roll forward a database or table spaces to perform full recovery
- Perform actions to reactivate databases in exception states

## Introduction

The database objects in your MUSICDB database have now been populated with data (through imports, loads, inserts, or updates). Now you will perform the tasks to enable yourself to re-create your data in case of loss or destruction of the data.

## *Instructor Notes:*

*Introduction* — During this lab, students will determine their logging configuration. They will then have a choice of doing Section 2 (Recovery Support with Circular Logging) or Section 3 (Recovery Support with Log Retention Logging). Time should not be provided for all students to do both Sections 2 and 3. All students should do Section 4.

*Time for Lab* — 40 minutes.

*Things to Review at End of Lab* — Make sure each student was successful in backing up and restoring their databases, and that they completed the last section of the lab where they reset their environment for the next lab.

*Common Mistakes Students May Make* — **Forgetting to reconfigure the database before executing this lab — Section 1/Step 1.**

**WARNING:** If during the lab, students should get an **SQL1042** return code (probably while trying to issue a ROLLFORWARD command), have them execute the following commands, then retry the ROLLFORWARD.

```
db2 FORCE APPLICATION ALL
```

```
db2 TERMINATE
```



## Exercise instructions with hints

If you are only interested in recovery in a circular logging environment, then do Sections 0, 1, 2, and 4. If you are only interested in recovery in an archive logging environment, then do Sections 0, 1, 3, and 4.

The lab solutions using primarily commands are indicated as *Command Window*. The lab solutions using primarily the graphical tools are indicated as *GUI*.

### Section 1 - Determining Log Configurations

During this section, you will determine the current logging options that are in effect and locate the log files.

- \_\_\_ 1. Before beginning this lab, you need to make sure some of your database configuration parameters are set to the values required by this lab. Do this by running the script `recovrst`.

Command Window	<code>db2 -tvf recovrst</code>
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: ___ Retrieve the script file <code>recovrst</code> . ___ Execute the script.

- \_\_\_ 2. Connect to MUSICDB to activate the database.

Command Window	<code>db2 connect to musicdb</code>
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: <code>connect to musicdb</code>

- \_\_\_ 3. Examine your db cfg file for MUSICDB.  
How large are the log files? Is this amount in pages or bytes?

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI Control Center	On the <i>Control Center</i> session: ___ Right-click and select <b>MUSICDB</b> in the left pane and then select <i>Configure Parameters</i> from the pop-up list. ___ Scroll to the <i>Logs</i> section.
Command Editor	<code>get db cfg for musicdb</code>

The log file size is given in 4KB pages via the parameter LOGFILSIZ. In this lab, the log files are fairly small — six 4KB pages per file. This small size would not be typical in an active installation. It is used to facilitate the class environment.

- \_\_\_ 4. How many primary log files will be allocated? When will these files be allocated?

---



---

Three log files that are considered primary will be allocated when the database is created. DB2 will increase the LOGFILSIZ by two pages for overhead, hence a total for three primary logfiles of (6+2) pages of 4KB makes  $(3 \times (6+2) \times 4KB) = 3 \times 32KB = 96KB$  of space, which will be allocated for the primary logs.

- \_\_\_ 5. How many secondary log files will be allocated? When will these files be allocated?

---



---

Secondary log files will be allocated when needed. This will occur if the primary logs become full due to uncommitted units of work. The number of secondary log files allowed is limited by the parameter LOGSECOND. In this case, a maximum of two secondary log files is permitted.

- \_\_\_ 6. For which type of logging is your database currently configured (log retention or circular)? What parameters provide this information?

---



---

The database is currently configured for circular logging. The LOGRETAIN and USEREXIT DB2 parameters designate whether log retention logging is being used or not. In your case, the values of both of these parameters is 0 (OFF), which means both parameters are turned off. Since both of these fields are set to OFF, log retention logging is not being used, therefore circular logging is being used. If one or both were set to 1 (ON), archival logging (log

retention logging) would have been used. Note that the configuration parameters LOGRETAIN and USEREXIT indicate the type of logging that will be used when all applications disconnect from the database and then one connects.

- \_\_\_ 7. Which types of recovery are supported by this type of logging — Crash, Version, or Roll forward?

**Crash and Version recovery are supported with this type of recovery. Roll forward recovery is not supported.**

- \_\_\_ 8. Where do the log files reside?

**The log files reside in the logpath. This is currently set to**

**C:\DB2\NODE0000\SQL00001\SQLOGDIR\**

- \_\_\_ 9. Assume the database manager was active. A short interruption of power made it necessary to perform a crash recovery. Will manual intervention be required to cause this crash recovery? What configuration parameter relates to this issue?

**On the GUI, check under the Recovery heading. Then click Cancel to close your Configure Parameters window.**

**The failure situation described requires a crash recovery. Since the AUTORESTART parameter is set to ON, this recovery will be automatic when the database is accessed. If this parameter was 0 (OFF), manual intervention would be required.**

- \_\_\_ 10. From your **Command Window** session, change your current directory to the log path, **but exclude the SQLOGDIR directory from your command.**

**Note:** Do NOT EDIT these files!

```
cd
cd C:\DB2\NODE0000\SQL00001
dir ..\MUSICDB
```

- \_\_\_ 11. Determine the contents of this directory.  
What are T0000000, T0000001, T0000002?

```
dir
```

**These are the names of the DMS table space containers for SYSCATSPACE, TEMPSPACE1, and USERSPACE1.**

- \_\_\_ 12. Is it appropriate to keep the log files in the same path? (Consider the type of recovery currently supported in your answer.)

---

**You will hear different answers to this question, and several may be correct. The answer needs to account for other considerations in the installation. An installation supporting primarily read only applications that populate decision support DB2 tables from extracts of other systems may satisfy logging requirements with the default. On the other hand, an installation with high change activity may determine it necessary to move the logs to a different file system, even if that installation does not elect to support roll-forward recovery.**

- \_\_\_ 13. Now, list the contents of SQLOGDIR.

```
cd SQLOGDIR
dir
```

- \_\_\_ 14. How many log files are in this directory? Does this match what you expected considering your configuration file? Why or why not?

---

**There are three log files in the directory. This matches the number of primary log files in the configuration. This is appropriate since primary log files are allocated when the database is created.**

- \_\_\_ 15. Increase the quantity in the STOCK table by 1 but do not commit the update. If you are not certain how to turn off auto-commit, check the solutions.

Command	db2 connect to musicdb
Window	db2 +c "update stock set qty = qty + 1"

GUI	<p>On the <i>Command Editor</i> session, <i>Commands</i> tab:</p> <p>___ Select <i>Tools Settings</i> from the icon bar.</p> <p>___ Select the <i>Command Editor</i> tab.</p> <p>___ Deselect <i>Automatically commit SQL statement</i>.</p> <p>___ Close the <i>Tools Settings</i> window.</p> <p>___ Enter the following:</p> <p style="padding-left: 40px;">connect to musicdb</p> <p>___ Click <b>Execute</b> icon.</p> <p>___ Enter the following:</p> <p style="padding-left: 40px;">update stock set qty = qty + 1</p> <p>___ Click <b>Execute</b> icon.</p>
-----	---

- \_\_\_ 16. If you encounter an error, go to the next question. If you did NOT encounter any error, decrease the quantity in the STOCK table by 1.

Command Window	db2 +c "update stock set qty = qty - 1"
GUI	update stock set qty = qty - 1

- \_\_\_ 17. What did you observe? What was the SQL code you received?

**The statement failed with an SQL code of -964. (You may see that the trigger is identified as causing the error.)**

- \_\_\_ 18. Issue a command to retrieve help on the SQL code. Why did the statement fail?

Command Window	db2 ? sql0964   more
GUI	? sql0964

**SQL0964C The transaction log for the database is full.**

**Cause: All space in the transaction log is being used.**

If a circular log with secondary log files is being used, an attempt has been made to allocate and use them. When the file system has no more space, secondary logs cannot be used.

The statement failed because you encountered a log file full condition.

Note that if you saw the trigger identified as causing the error, that the inserts being done by the trigger are also logged.

GUI	<p>If you are completing the GUI Remote Administration labs, reset your <i>Tools Settings</i> options in the <i>DB2 Command Editor</i> to enable <i>Automatically commit SQL statement</i>.</p> <p>On the <i>Command Editor</i> session, <i>Commands</i> tab:</p> <p>___ Select <i>Tools Settings</i> from the icon bar.</p> <p>___ Select the <i>Command Editor</i> tab.</p> <p>___ Select <i>Automatically commit SQL statement</i>.</p> <p>___ Close the <i>Tools Settings</i> window.</p>
-----	---

\_\_\_ 19. Since the unit of work could not complete successfully, issue a rollback.

**Note:** This particular problem should not occur in production environments during an attempt to update only 778 rows. The log file size on your lab databases have intentionally been set artificially low.

Command Window	db2 rollback
GUI	rollback

\_\_\_ 20. From your telnet session, determine the number of log files allocated at this point. How many are there?

dir

**There are five currently allocated.**

\_\_\_ 21. What accounts for the additional files?

**When circular logging is used and the primary log files fill up, secondary log files are allocated. The additional log files are secondary log files.**

\_\_\_ 22. Change to your home directory in your **Command** window.

```
cd \
```

\_\_\_ 23. From your telnet session, create a directory that will contain backups of your database:

```
mkdir restore
```

**Note:** In this lab environment, the directory path you will backup to is **C:\restore**. Normally, a scheme for isolating backups from the database supported by the backups would be preferable.

## Section 2 - Backup/Restore Support with Circular Logging

Either perform this section of the lab, if you wish to do circular logging, or Section 3 if you wish to do archival logging.

\_\_\_ 1.

Command Window	Open another <b>Command</b> window and connect to your <b>musicdb</b> database from the new window. <b>db2 connect to musicdb</b>
GUI	Shutdown the <i>DB2 Tools</i> and re-open the <i>Control Center</i> . Connect to your <b>MUSICDB</b> database from the <b>Command Window</b> session: <b>db2 connect to musicdb</b>

**Instructor Note:** The shutdown of CC was added in V8.2 to try to resolve some attachment issues later in this lab. I don't think it helped, so this can probably be removed in the next update.

\_\_\_ 2. Attempt to perform an online backup of your MUSICDB database to the directory of **C:\restore**. Were you successful in trying to specify an online backup? Explain why or why not.

Command Window	<b>db2 "backup db musicdb online to C:\restore"</b>
GUI Control Center	On the <i>Control Center</i> : ___ Right-click and select <b>MUSICDB</b> in the left pane and then select <i>Backup</i> from the pop-up list. ___ Indicate the following: ___ On the <i>Image</i> window, click <b>Add</b> and choose the <b>restore</b> directory in <b>C:\</b> . Click <b>OK</b> . ___ On the <i>Options</i> tab, attempt to select the <i>Online</i> radio button.
Command Editor	(Windows) <b>backup db musicdb online to C:\restore</b>

**No, you were not able to specify an online backup. In the GUI, DB2 prevented you from selecting the Online radio button when trying to do your backup. On the command line, an error message, SQL2413, was returned. An online backup allows other transactions to be changing the database simultaneously. Therefore, only a database configured for rollforward recovery can support an online backup. Otherwise, the integrity of the database would be threatened.**



\_\_\_ 3. Attempt to perform an offline backup instead.

Command Window	<code>db2 "backup db musicdb to C:\restore"</code>
GUI Control Center	<p>To accomplish the purposes of this lab, ensure that you deselect the <i>Quiesce Database</i> option.</p> <p>___ On the <i>Options</i> tab, make sure the <i>Offline</i> radio button has been selected. Deselect the check box <i>Quiesce the database before initiating the offline operation</i>.</p> <p>___ On the <i>Performance</i> tab, indicate the <i>Number of Buffers</i> as 2.</p> <p>___ On the <i>Schedule</i> tab, choose <i>Run now without saving task history</i>.</p> <p>___ Click the <b>Finish</b> button.</p>
Command Editor	<code>(Windows) backup db musicdb to C:\restore</code>

\_\_\_ 4. Did the backup complete successfully? What did the message say?

**The backup did not complete. The message said you can't perform a backup right now because the database is currently in use (SQL1035N).**

**It is possible in DB2 9 that the offline backup completed. This indicates that applications were forced off the database.**

\_\_\_ 5. In order to take an offline backup, do you think that exclusive use of the database is required? Who is currently connected to the database?

**An offline backup requires exclusive use of the database. This is the manner in which integrity is guaranteed. The application that is connected to the database is your other telnet session which previously connected to the database.**

\_\_\_ 6. Resolve the apparent problem by forcing all applications to end.

Command Window	<code>db2 force applications all</code>
----------------	---

\_\_\_ 7. Now try your backup again.

Command Window	<code>db2 "backup db musicdb to C:\restore"</code>
GUI Control Center	On the <i>Control Center</i> session ( <i>Backup wizard</i> should still be up): ___ Click <b>Finish</b> .
Command Editor	<code>(Windows) backup db musicdb to C:\restore</code>

**Note:** Be patient. A backup is a utility that may require some time to complete.

\_\_\_ 8. Was the backup successful this time?

---

**The backup was successful for this execution.**

**Instructor Note:** Because of the problems in V8.2, I removed this step from the lab. Test again when labs are tested on next fixpack. Appears successful with V9.1.

Review your output messages and the status of your backup job. Record the date and time for the backup image.

---

**Instructor Note:** These should be instructions used, but requires GUIs to be shut down in V8.2, so changed instructions.

Command Window and Command Editor	Look at the timestamp returned from the backup command.
-----------------------------------	---

GUI	<ul style="list-style-type: none"> <li>___ On your DB2 <i>Control Center</i> session, re-establish your attachment to the instance. Right-click <b>DB2</b> and select <i>Attach</i>.</li> <li>___ From your DB2 <i>Control Center</i> session, select <i>Journal</i> icon or click the <i>Tools</i> menu bar option and <i>Journal</i> from the pull-down list.</li> <li>___ From the <i>Journal</i>, select <i>Database history</i> button.</li> <li>___ Next to the <i>Database</i> field, click the button with three dots.</li> <li>___ Select your system, instance, and database.</li> <li>___ Click <b>OK</b>.</li> <li>___ If you receive an error about already being attached to an instance, just select the system, instance, and database again.</li> <li>___ If you still receive the error about already being attached to an instance, choose <i>Journal -&gt; Shut Down DB2 Tools</i>, bring the <i>Control Center</i> back up, attach to the instance, start the <i>Journal</i>, and try again.</li> <li>___ Sort based on <i>Start Date</i> (click the title at the top of the column). The backup should be the top task - you should be able to see the timestamp. The start time timestamp would be the one required if issued from a command line.</li> <li>___ Record the timestamp for the backup image.</li> <li>___ Close the <i>Journal</i> session and return to the DB2 <i>Control Center</i>.</li> </ul>
-----	---

- \_\_\_ 9. The timestamp for a backup image is necessary for restore if you are using the RESTORE command rather than the GUI DB2 Control Center, and multiple images reside in a given source.

When the backup is targeted to disk, the name of the file itself will reflect the timestamp. When tape or TSM managed devices are used, the header in the backup image will contain this information.

- \_\_\_ 10. Look at the backup information through the LIST BACKUP command. Remember that you have a previous backup from an early lab (LOAD lab). Issue the list backup command with the ALL keyword.

Command Window and GUI	<code>db2 list backup all for musicdb   more</code>
Control Center	You can also view this information if you are connected to the database in the lower right pane in the <i>Control Center</i> . The date and time of the latest backup shows in the information displayed.

**Instructor Note:** Previously the lab had this:

GUI	<pre>terminate attach to instnn user instnn using ibm2blue list backup all for musicdb This request may take a while to return information. Please be patient. Or you may prefer to use the telnet session to execute this command.</pre>
-----	---

But in V8.2, this hung forever or you had to shut down db2 tools at this point. Test again on a future fixpack. With V9.1, this warning is left here in case similar problems occur. Please report problems into the QuickPlace entry for CF28.

\_\_\_ 11. Is the timestamp available as part of this information?

---

**The timestamp is available as part of the information presented.**

\_\_\_ 12. Record the start timestamp of the backup.

---

**Start Time: 20040805111328 (Your actual timestamp will not match. This value is used for the lab solutions.)**

\_\_\_ 13. How many table spaces are included in this backup?

---

**This full database backup contains 11 table spaces.**

\_\_\_ 14. Instead of using the keyword ALL, it is possible to restrict the output of the LIST BACKUP command via a timestamp specification or by object name specification.  
Issue a request to get just the backups for today.  
You may wish to request online help for this command.

Command Window and GUI	On <b>Command Window</b> session, <pre>db2 list backup since <b>yyyymmddhh</b> for musicdb   more</pre>
------------------------	--

**Instructor Note:** Previously the lab had this:

GUI	<pre>list backup since <b>yyyymmddhh</b> for musicdb This request may take a while to return information. Please be patient.</pre>
-----	--

But in V8.2, this hung forever or you had to shut down db2 tools at this point. Test again on a future fixpack. With V9.1, this warning is left here in case similar problems occur. Please report problems into the QuickPlace entry for CF28.

- \_\_\_ 15. The history information regarding backups is maintained by the database manager and can become extensive if it is never removed. Use the `?` command to invoke the online help facility to identify a command that can be used to manage the amount of history information retained. Write the name of the command you believe would give you the capability to eliminate old history information.

Command Window	<code>db2 ?   more</code>
GUI	On the <i>Command Editor</i> : <code>? ?</code>

**The PRUNE HISTORY command can be used to manage backup history information.**

- \_\_\_ 16. From your **Command Window**, examine the file naming convention used for your backup.

```
cd \
cd restore
dir
```

You should see a file named

```
"MUSICDB.0.DB2.NODE0000.CATN0000.yyyymmddhhmmss.001".
```

**Note:** Your database name, instance name, and timestamp of the backup are all reflected in the file name.

- \_\_\_ 17. The backup image reflects the same data that is currently in your MUSICDB database since you have not issued any SQL to change data since the backup was taken. Select the artist's name and album titles for artist number (ARTNO) 77.

Command Window	<code>db2 connect to musicdb</code> <code>db2 "select name, title from artists a, albums b</code> <code>where a.artno = 77 and a.artno = b.artno"</code>
GUI	On the <i>Command Editor</i> : <code>connect to musicdb;</code> <code>select name, title from artists a, albums b</code> <code>where a.artno = 77 and a.artno = b.artno</code>

Record Name — **Livid IDs**

Record Title 1 — \_\_\_\_\_ **The First 13**Record Title 2 — \_\_\_\_\_ **Live at the Psychidelly**

- \_\_\_ 18. Change the name of the artist 77 to “Melanie and the Mechanics” and the album title of ITEMNO 261 to “Unmaterial Girl”.

Command Window	<pre>db2 "update artists set name = 'Melanie and the Mechanics' where artno = 77" db2 "update albums set title = 'Unmaterial Girl' where itemno = 261"</pre>
GUI	On the <i>Command Editor</i> : update artists set name = 'Melanie and the Mechanics' where artno = 77; update albums set title = 'Unmaterial Girl' where itemno = 261

- \_\_\_ 19. Execute a join between ARTISTS and ALBUMS for ARTNO 77 and document the data as it now appears. This reflects a change that occurred after your backup.

Command Window	<pre>db2 "select name, title from artists a, albums b where a.artno = 77 and a.artno = b.artno"</pre>
GUI	Erase the previous statement. On the menu bar, choose <i>Selected -&gt; History</i> . Pick the select statement: select name, title from artists a, albums b where a.artno = 77 and a.artno = b.artno Click <b>Paste</b> . Execute the statement.

Record Name — \_\_\_\_\_ **Melanie and the Mechanics**Record Title 1 — \_\_\_\_\_ **The First 13**Record Title 2 — \_\_\_\_\_ **Unmaterial Girl**

- \_\_\_ 20. Assume the disk containing the MUSICDB database failed. You now have the task to restore the backup image that was made.

Retrieve help on the DB2 restore command.

Command Window	<code>db2 ? restore</code>
GUI	<code>? restore</code>

**Note:** In reality, a manual should be consulted before continuing, unless the online documentation is used as a reminder of available parameters on the RESTORE command. In our particular case, many of the options are not required. The backup image contains the target database alias and target directory, the default buffer size will be appropriate, and the date/time at which the backup was taken is only necessary when multiple backups exist in the FROM directory/device and restoring from the command line.

\_\_\_ 21. First make sure there are no applications are connected to the database.

Command Window	<code>db2 force application all</code>
GUI	<code>attach to DB2;</code> <code>force application all;</code>

\_\_\_ 22. Restore the database from the `C:\restore` directory.

Command Window	___ Enter the following command: <code>db2 restore db musicdb from C:\restore</code> ___ Reply <code>y</code> to the prompt.
Command Editor	<code>(Windows) restore db musicdb from C:\restore without prompting</code>

**Instructor Note:** In V8.2, this required shutting down the Control Center due to it being attached to incorrect instance. I thought this was ugly, so I changed it to Command Editor; try again in a new fixpack. Warning kept here for V9.1.

GUI	<p>On the <i>Control Center</i> session:</p> <ul style="list-style-type: none"> <li>___ Right-click and select <b>MUSICDB</b> in the left pane and then select <i>Restore</i>.</li> </ul> <p>Specify the restore parameters:</p> <ul style="list-style-type: none"> <li>___ On the <i>Introduction</i> window, select <i>Restore to an existing database</i>.</li> <li>___ On the <i>Available Images</i> window, select Database backup image that corresponds with your backup image taken in this lab (make sure the date/time matches what you documented before). Click the single &gt; button to move the image to the <i>Selected backup images</i> box.</li> </ul> <p><b>Note: If you get a message that you are attached to an incorrect instance, Cancel the Restore Data Wizard, Shut Down the Control Center, bring it up again and retry the task.</b></p> <ul style="list-style-type: none"> <li>___ On the <i>Schedule</i> window, select <i>Run now without saving task history</i>.</li> <li>___ On the <i>Summary</i> window, click <i>Show Command</i> to view the command that has been generated. Click <b>Finish</b> to start the restore process.</li> <li>___ When the DB2 Message appears, verify that the command completed successfully. Click <b>Close</b>.</li> </ul>
-----	--

\_\_\_ 23. Connect to MUSICDB and execute the join between ARTISTS and ALBUMS for ARTNO 77.

Command Window	<p>db2 connect to musicdb</p> <p><b>Note: If the connect fails the first time, since you previously forced all applications to end, try it again.</b></p> <p>db2 "select name, title from artists a, albums b where a.artno = 77 and a.artno = b.artno"</p>
GUI	<p>terminate</p> <p>connect to musicdb</p> <p><b>Note: If the connect fails the first time, since you previously forced all applications to end, try it again.</b></p> <p>Select name, title from artists a, albums b where a.artno = 77 and a.artno = b.artno</p>

\_\_\_ 24. Does the data reflect the updates made, or is the original data restored?



---

**The output reflects the original data. The backup has been successfully restored.**

- \_\_\_ 25. Assume that you are not in a lab environment. You made your backup three days ago and thousands of transactions have since occurred. Would your current recovery strategy be acceptable?
- 

---

**Normally, it would not be acceptable to lose three days worth of work in an active database system. Such a system would benefit from a recovery strategy that supports roll forward recovery.**

## Section 3 - Backup/Restore Support with Archive Logging

- \_\_\_ 1. Update the database configuration file to indicate that log retention logging is to be used (recovery).

Command Window	<code>db2 update db cfg for musicdb using logretain recovery</code>
GUI	<p>On the <i>Control Center</i>:</p> <p>___ Right-click and select <b>MUSICDB</b> in the left pane, and then select <i>Connect</i> from the pop-up list.</p> <p>___ Right-click and select <b>MUSICDB</b> in the left pane, and then select <i>Configure Parameters</i> from the pop-up list.</p> <p>___ In the <i>Logs</i> area, find the <b>LOGRETAIN</b> keyword. Click to the right of the keyword (where it currently says <b>No</b>), and then click the button with three dots.</p> <p>___ On the <i>Change Database Configuration Parameter</i> window, change the value to <b>RECOVERY</b>. Click <b>OK</b>.</p> <p>___ On the <i>Database Configuration</i> window, click <b>OK</b>.</p> <p>___ Read the message displayed, then select <b>Close</b>.</p>
Command Editor	<p>On the <i>Command Editor</i>:</p> <code>update db cfg for musicdb using logretain recovery</code>

- \_\_\_ 2. Some changes to the db cfg file (including changes to LOGRETAIN) do not take affect until the database is deactivated and becomes active again. Since the database was activated by the first user connect, all applications must end to deactivate the database. Try issuing a DB2 terminate command.

Command Window	<code>db2 terminate</code>
GUI	<p>On the <i>Command Editor</i>,</p> <code>terminate</code>

- \_\_\_ 3. To find out for sure if log retention logging is in effect now, look at the Log retain status indicator in the db cfg.

**Note:** One parameter indicates the current setting of LOGRETAIN. However, the value of ON does not necessarily mean that log retention logging is currently being used. What is actually being used is indicated by the parameter, "Log retain for recovery status" indicator. Be certain to examine both parameters when

investigating the type of logging that is currently used, and the type that will be used. (In many cases, these parameters will match.)

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI	On the <i>Command Editor</i> : <b>attach to DB2</b> <code>get db cfg for musicdb</code>

\_\_\_ 4. Did the terminate command change anything concerning the parameters?

---



---

**The terminate command had no effect on the parameters. The logretain status is still off. (If using DB2 9, this parameter is changed dynamically, therefore you may not need to do Steps 5 and 6).**

\_\_\_ 5. Try forcing all applications off the database.

Command Window	<code>db2 force application all</code>
GUI	<code>force application all</code>

\_\_\_ 6. Did the Log retain status indicator value change to RECOVERY?

---

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI	<code>get db cfg for musicdb</code>

**The Log retain for recovery status is still set to NO.**

\_\_\_ 7. Connect to musicdb database. Was the connection successful? What does the SQL message indicate? Why does the database manager require a new backup?

---



---



---

Command Window	db2 connect to musicdb
GUI	connect to musicdb

**The connection was not successful. The database is in a backup pending status. Enabling or disabling log retention logging establishes a new recovery point for the database. The database manager requires an offline backup of your database to establish this new recovery point.**

\_\_\_ 8. Did the attempt to connect to the database cause log retention to be activated?

\_\_\_\_\_

Command Window	db2 get db cfg for musicdb   more
GUI	get db cfg for musicdb

**The attempt to connect to the database did cause the log retention change request to be active. This is indicated by the log retain status indicator equal to 1 or RECOVERY.**

\_\_\_ 9. Create an offline backup of your database and record the timestamp.

Record Timestamp - \_\_\_\_\_

Command Window	db2 "backup db musicdb to C:\restore"
GUI	(Windows): backup db musicdb to C:\restore;

**Timestamp — 20040805121328 (Your actual timestamp will not match. This value is used for the lab solutions.)**

**Note:** When the database configuration is changed regarding log retain, an **offline** backup is required. From this point forward, both online and offline backups would be permitted.

\_\_\_ 10. Complete a series of SQL statements to change data in the ARTISTS table and record points in time associated with these changes.

\_\_\_ a. Change the name of ARTNO 77 to "Melanie and the Mechanics".

\_\_\_ b. Record the time.

- \_\_\_ c. Change the name of ARTNO 77 to "Bill Pellett and the Comets".
- \_\_\_ d. Record the time.

Use `values current time`, an SQL statement that displays the current time, or `values current time - current timezone`, an SQL statement that displays the current time in UT time (Universal Time, often designated as GMT), to determine the approximate time of each change.

Command Window	<pre>db2 connect to musicdb db2 "update artists set name = 'Melanie and the Mechanics' where artno = 77" db2 "values current time - current timezone"</pre>
GUI	<p>On the <i>Command Editor</i>:</p> <pre>connect to musicdb; update artists set name = 'Melanie and the Mechanics' where artno = 77; values current time</pre>

Record time associated with ***Melanie and the Mechanics*** - \_\_\_\_\_  
**12:17:36 (your value will be different)**

Command Window	<pre>db2 "update artists set name = 'Bill Pellett and the Comets' where artno = 77" db2 "values current time - current timezone"</pre>
GUI	<pre>update artists set name = 'Bill Pellett and the Comets' where artno = 77; values current time</pre>

Record time associated with ***Bill Pellett and the Comets*** - \_\_\_\_\_  
**12:17:54 — (your value will be different)**

- \_\_\_ 11. View the db cfg again.

What is the value of the first active log file?

\_\_\_\_\_

Command Window	<pre>db2 get db cfg for musicdb   more</pre>
----------------	--

GUI	<p>On the <i>Control Center</i>:</p> <ul style="list-style-type: none"> <li>___ Right-click and select <b>MUSICDB</b> in the left pane, and then select <i>Configure Parameters</i> from the pop-up list.</li> <li>___ Find the <i>Logs</i> section and look at the <b>LOGHEAD</b> parameter.</li> <li>___ Click <b>Cancel</b>.</li> </ul> <p>-or-</p> <p>On the <i>Command Editor</i>:</p> <pre>get db cfg for muscidb</pre>
-----	---

**The value for first active log file is S0000000.LOG. This will be the file used to support logging for the next application.**

\_\_\_ 12. Restore your backup.

Command Window	db2 "restore db musicdb from C:\restore without prompting"
GUI	<pre>attach to DB2; force application all; (Windows): restore db musicdb from C:\restore without prompting;</pre>

\_\_\_ 13. Did the restore fail?

**If Section 2 was completed before Section 3, then the *from* directory contains more than one backup image for the database and this command would have failed. Since no explicit taken at timestamp was provided, all backups for the database are candidates.**

\_\_\_ 14. If the directory containing the backup image you wish to restore contains other images for the same database alias, you must specify which image to use with the **TAKEN AT** option on the restore command. If the previous restore failed, issue the restore command indicating which backup you wish to be restored via the **TAKEN AT** option.

Command Window	db2 restore db musicdb from C:\restore taken at <i>timestamp</i> without prompting
GUI	(Windows): restore db musicdb from C:\restore taken at <i>timestamp</i> without prompting

Where **timestamp** matches the value you wrote down after the latest backup completed. **DO NOT** specify any type of formatting for the timestamp. It is a string of numbers.

**If you were using the DB2 Control Center, you can pick the image you want to restore.**

- \_\_\_ 15. Earlier in this lab, you attempted to connect to the database when it was in a backup pending status. You received a message during the connect to indicate this status. There is another method to determine the state of the database. Examine the db cfg again. Check the Recovery Status values.

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI	<code>get db cfg for musicdb</code>

- \_\_\_ 16. Is the database in any pending state?

\_\_\_ **Look at the Backup pending indicator and the Rollforward pending indicator.**

**The database is in a rollforward pending state.**

- \_\_\_ 17. This state will occur if a restore is done to an online backup or a restore is done to an offline backup without specifying the WITHOUT ROLLING FORWARD phrase. In order to clear this flag and apply the log records associated with the **Melanie and the Mechanics** update, perform a ROLLFORWARD DATABASE. You want to rollforward to the timestamp associated with the Melanie and the Mechanics update. You also want the database removed from the rollforward pending state upon completion of the command.

Command Window	Ensure you convert the time you recorded to the equivalent CUT time, and issue the following command: <code>db2 "rollforward db musicdb to yyyy-mm-dd-hh.mm.ss.nnnnnn and stop" (Use 000000 for nnnnnn.)</code>
----------------	--

GUI	<p>On the <i>Control Center</i>:</p> <p>___ Right-click and select <b>MUSICDB</b> in the left pane and then select <i>Roll-forward</i> from the pop-up list. (If you get a message that the selected database is not in roll-forward pending, choose to <i>Shutdown DB2 Tools</i>, and restart the <i>Control Center</i>.)</p> <p>Specify the parameters to perform a roll forward.</p> <p>___ Select radio button to the left of <i>Rollforward to a point in time - Local</i>.</p> <p>___ The date in the <i>Rollforward to transaction</i> box should be accurate.</p> <p>___ Change the time associated with that date to match the time you documented for the <b>Melanie and the Mechanics</b> update. First update the hour, then cursor to the right and update the minutes, then cursor to the right and update the seconds.</p> <p>___ On the <i>Final State</i> window, Select the <i>Complete the roll forward and return to the active state</i> radio button.</p> <p>___ Click <b>Finish</b>.</p> <p>___ Read the DB2 message, then select <b>Close</b>.</p>
-----	--

- \_\_\_ 18. Check the db cfg again to see if the Rollforward pending status was removed. Was it?

Command Window	<code>db2 get db cfg for musicdb   more</code>
GUI	<p><b>attach to DB2</b></p> <p>get db cfg for musicdb</p>

**The Rollforward pending indicator in the db cfg is now set to NO.**

- \_\_\_ 19. Connect to MUSICDB database.
- Check the contents of the row you updated earlier (ARTNO 77).

Command Window	<code>db2 connect to musicdb</code> <code>db2 "select name from artists where artno = 77"</code>
GUI	<p>connect to musicdb</p> <p>select name from artists where artno = 77</p>

- \_\_\_ 20. What is the value of the name returned? Why?



**Melanie and the Mechanics is the name returned. This value corresponds to the name that was present at the point in time provided during the roll forward process. The logs were applied up to this point and no further. The log records associated with the Bill Pellett and the Comets update were never applied.**

**Note:** It is typical to roll forward to end of logs to apply all available log records.

- \_\_\_ 21. Even though the database manager is configured for roll forward recovery, you are not required to apply log records.

Restore the database backup and indicate that you do not want to roll forward.

Command Window	<pre>db2 "restore db musicdb from C:\restore taken at timestamp without rolling forward without prompting"</pre> <p><b>Note:</b> You must substitute the appropriate syntax. The timestamp on restore has no formatting (- or .) characters.</p>
GUI	<p>force application all (Windows): restore db musicdb from C:\restore taken at timestamp without rolling forward without prompting</p> <p><b>Note:</b> You must substitute the appropriate syntax. The timestamp on restore has no formatting (- or .) characters.</p>

- \_\_\_ 22. Check the pending status of your database by looking at the db cfg.

Is the database in any pending state? Did the value returned to you match what you expected?

---



---

Command Window	<pre>db2 get db cfg for musicdb   more</pre>
GUI	get db cfg for musicdb

**The database is not in any pending state. The use of without rolling forward bypasses the second stage of recovery.**

- \_\_\_ 23. Connect to MUSICDB database.

Issue a SELECT statement against the artists row that was previously updated (ARTNO 77).

Did the value returned to you match what you expected?

---

---

Command Window	db2 connect to musicdb db2 "select name from artists where artno = 77"
GUI	connect to musicdb select name from artists where artno = 77

**The value returned from the SELECT statement matches the original value because no log records were applied.**

**Note:** You can only specify the phrase without rolling forward if you are restoring from an offline backup.

## Section 4 - Resetting the Environment

Everyone **MUST COMPLETE** Section 4.

Complete the following tasks to ensure that your environment is set properly for the remaining lab exercises:

- \_\_\_ 1. Execute the cleanup script.

For local administration, this is an operating system command file named `cleanrec`.

For remote administration, this is a DB2 script named `clrec`.

Command Window	<code>cd C:\labfiles\cf23</code> <code>db2 -tvf clrec</code>
GUI	On the <i>Command Editor</i> , <i>Commands</i> tab: ___ Select <i>Selected</i> -> <i>Open</i> from the menu bar. ___ Select the script <code>clrec</code> (from <code>C:\labfiles\cf23</code> ). ___ Click <b>OK</b> . ___ Select <i>Execute</i> icon.

- \_\_\_ 2. Examine your db cfg and verify that your database is now configured for circular logging.

Command Window	<code>db2 connect to musicdb</code> <code>db2 get db cfg for musicdb   more</code>
GUI	<code>connect to musicdb</code> <code>get db cfg for musicdb</code>

*Log retain for recovery status* should be equal to NO.

**END OF LAB**



# Exercise 8. Investigating DB2 locking

*(with hints)*

## Estimated time

00:45

## What this exercise is about

In this exercise, you will use the system monitor to analyze locking occurring on your database. You will use the DB2 Control Center, DB2 Command Editor, and DB2 Command Window to represent transactions accessing your database.

## What you should be able to do

At the end of the lab, students should be able to:

- Use the Snapshot Monitor to perform basic lock monitoring
- Distinguish between lock timeouts and deadlocks
- Update database configuration parameters related to timeouts and deadlocks

## Introduction

This lab has three sections that will provide you with the opportunity to investigate and monitor locking that occurs within your database. The first section presents tasks necessary to use the Snapshot Monitor in order to perform basic lock monitoring. The second section highlights the concept of lock timeouts. The third section illustrates the concept of deadlocks.

A fourth section is included that will enable you to reset your lab environment.

**You should complete the lab sections in the order that they are presented. If you do not finish the first three sections, be sure to complete the fourth section to ensure that your database is in a state that can be used to complete other labs for the course.**

**Note:** In this lab there are a couple of unique items that you should pay special attention to:

Pay attention to which ID you are supposed to be on for each step.

- You will be turning off autocommit in CLP by using the '+c' option. Be careful — if you issue a CLP SQL statement without the '+c' option, work that was being held will be committed.

**Note:** If you enter a CLP statement that is invalid, it will be processed as an SQL statement (and you'll get an error). This will commit any previous work that you may have held.

## ***Instructor Notes:***

*Introduction* — In these exercises, the students will complete basic lock monitoring, investigate lock timeout, and investigate deadlocks.

*Time for Lab* — 45 minutes.

*Things to Review at End of Lab* — Ensure the students understand the difference between timeout and deadlock.

*Common Mistakes Students May Make* — Students may forget to turn off autocommit as indicated. They may also forget to RESET their environment during Section 4, which could cause problems in later labs.

## Exercise instructions with hints

The lab solutions using primarily commands are indicated as *Command Window*. The lab solutions using primarily the graphical tools are indicated as *GUI*.

### Section 0 - Setup

#### \_\_\_ 1. For Windows:

Open the first DB2 Command Window.

Select Start > Programs > IBM DB2 > DB2COPY1 > Command Line Tools > Command Window.

Open a second and third Command Window.

Within the first DB2 Command Window, type **db2cmd** and Enter. Do this again, so that you have three Command Windows open.

In one Command Window, change the window title to **student**.

Type:

```
title student
cd C:\labfiles\cf23
```

In the second Command Window, change the window title to **db2admin**.

Type:

```
title db2admin
cd C:\labfiles\cf23
```

In the third Command Window, change the window title to **monitor**.

Type:

```
title monitor
cd C:\labfiles\cf23
```

### Section 1 - Basic Lock Monitoring

#### \_\_\_ 1. From your **monitor** window, execute the following commands to ensure that no connections exist to your instance:

```
db2 force application all
```

#### \_\_\_ 2. From your **monitor** window, grant update privilege against the STOCK table to **db2admin**.

```
db2 connect to musicdb
db2 grant update on stock to user db2admin
```

- \_\_\_ 3. From your **monitor** window, examine the current setting of the database monitor switches by issuing the following:

```
db2 get monitor switches
```

- \_\_\_ 4. What is the current setting of every switch?

---

**The monitor switches should all be off except timestamp which is on by default.**

- \_\_\_ 5. Will the system database monitor provide **ANY** information concerning locks?

---



---



---

**The system monitor will always provide some information that is considered basic. Some locking information is included in the basic monitoring group.**

- \_\_\_ 6. From your **db2admin** window, connect to the musicdb database and update a single row in the STOCK table. **DO NOT** commit the change. Use the CLP command option of **+c**. The +c CLP command option must be used to prevent automatically committing an SQL statement. **Follow your lab instructions carefully!**

```
db2 connect to musicdb user db2admin using ibm2blue
```

```
db2 +c "update student.stock set qty = qty - 1 where itemno = 200  
and type = 'D' "
```

- \_\_\_ 7. From your **monitor** window, list the agents associated with the two CLP applications you currently have connected/attached to this instance.

```
db2 list applications show detail
```

Record the results below. You will find the Application Handle useful during subsequent portions of the lab.

```
student Application Handle = _____
```

```
db2admin Application Handle = _____
```

- \_\_\_ 8. From your **monitor** window, examine the database monitor elements related to locking by issuing the following:

```
db2 get snapshot for locks on musicdb > lock1.out
```

```
notepad lock1.out
```

- \_\_\_ 9. Are any locks currently held?

---



---



---



The first part of the lock snapshot provides the total number of locks held. The number should be greater than zero since **db2admin** has an outstanding unit of work in progress. The number of locks held will be variable, but general statements concerning lock volume can be made using this parameter.

- \_\_\_ 10. How many applications are currently connected?

Two applications should be connected. One is the agent associated with the **db2admin** window and the other is the agent associated with the **student** window. Recall that agents associated with CLP windows remain until the **db2 terminate** command is issued.

- \_\_\_ 11. What agents are currently holding locks? What applications are associated with this/these agents?

The only agent holding locks is the one associated with the **db2admin** window. The lock snapshot consists of a general information block and detailed information blocks for each connected application. Only the detailed information block for the **db2admin** application shows held locks.

- \_\_\_ 12. Is there ever a case where a row lock is possessed without a corresponding lock on the table which contains the row?

Row locks are never obtained unless a supporting table lock is possessed first.

- \_\_\_ 13. What lock modes are shown for tables under which a row is locked?

The lock modes of IX and IS are used to support the underlying row locks obtained. In other situations, a SIX lock may be used as well.

- \_\_\_ 14. How many different tables are being locked by the agent?

**Based on the order in which you perform steps, you may have different answers to this question.**

**When the labs were developed, one table was being locked by the agent. It was the **STUDENT.STOCK** table.**

**The number of tables you see locked may differ depending on sequence of lab steps and commits that may have been completed automatically for you.**

**Note:** You may also see one or more Internal Plan Locks. Any execution of SQL, whether it is static or dynamic, must be done while holding a Plan (or Package) lock. This ensures that no one can drop the package that is being executed.

Other information that will be displayed:

The Lock Name is an internal binary lock name that serves as a unique identifier for locks. Lock Attributes is a set of bits that relates to the current attribute of the lock, like *Wait for availability*, *Lock by RR scan*, and *Update/delete row lock*. Lock Release flags are a set of bits that relate to the possible release flag settings, like *Locks by SQL compiler*, *Non-unique*, *untracked locks*.

- \_\_\_ 15. It may be that you will see catalog tables being locked. Why might catalog tables be accessed?

---

---

---

**The user, **db2admin**, is accessing the database through the dynamic application Command Line Processor. Various authorization checks are done via the catalog tables in order to verify that the user has the privilege to perform the desired tasks and that a valid package exists. It is not necessary to understand every catalog table that will be accessed for a given application. The key point is that share locks can be obtained on the catalogs during normal operation. Therefore, applications that require exclusive locks on catalog tables (such as those issuing DDL) may encounter contention.**

- \_\_\_ 16. Exit the Notepad session.

## Section 2 - Lock Timeouts

- \_\_\_ 1. From your **student** window, issue the following SQL statement:

```
db2 connect to musicdb user student using ibm2blue
db2 "select * from stock where itemno = 198"
```

- \_\_\_ 2. What happened and why? (Recall that you currently have a row where itemno = 200 locked by the **db2admin** application.)

---



---



---

**Three result rows were returned by the application. Even though the usernn application has locks on the STOCK table and a row, there is no contention with the **student** application, which is acquiring IS locks on the table and S locks on rows. The rows accessed are not the same rows, therefore, contention is not experienced. This demonstrates the compatibility of IS and IX table locks.**

- \_\_\_ 3. From your **student** window, issue the following SQL statement:

```
db2 "select * from stock where itemno = 200"
```

- \_\_\_ 4. What happened and why? (Recall that you currently have a row where itemno = 200 locked by the **db2admin** application)

---



---



---

**In this case, the **student** application query does not complete. It is waiting for S locks on the same rows that the usernn application has already placed X locks upon. The two accesses to the data are not compatible.**

- \_\_\_ 5. Did table locking cause the contention problem?

---



---



---

**Table locking did not cause the contention problem. If contention occurred at the table level, the previous select would not have returned any data. If proof of such a conclusion were needed, a lock snapshot could be obtained to show the IS lock obtained by the **student** application at the table level.**

- \_\_\_ 6. You have not done anything in the lab environment to change the isolation level in the **student** window. Consider this and your results from the past two SQL statements issued from the **student** window. If given a choice between Uncommitted

Read and Cursor Stability, which one do you think is the default isolation level for the **student** window?

---

---

---

**The behavior of the **student** window indicates the isolation is Cursor Stability. If Uncommitted Read were the default, the attempt to select the rows where itemno = 200 would have executed.**

**Indeed, the isolation level by default is Cursor Stability.**

- \_\_\_ 7. From your **db2admin** window, issue the following to release the row lock.

```
db2 rollback
```

- \_\_\_ 8. Did the waiting application continue?

---

---

---

**The **student** application continued and retrieved the rows in the STOCK table. The rollback statement released the exclusive lock that was causing the lock wait. The **student** select request was able to proceed by obtaining share locks on the rows as they were read.**

- \_\_\_ 9. From your **monitor** window, examine the database configuration parameter concerning lock timeout by issuing:

```
db2 "get db cfg for musicdb" | more
```

- \_\_\_ 10. Check the value of the locktimeout parameter. If you had decided to get a cup of coffee while the original X lock was held on the row your other application was trying to read, would the situation have changed, even if you took 15 minutes to consume mass quantities of liquid refreshments?

---

---

---

**In the given situation, no change would occur. Lock waits do not *time out* if the default configuration value for locktimeout (-1) is used, like it is here.**

- \_\_\_ 11. From your **monitor** window, change the locktimeout parameter to 90 seconds by issuing the following commands: Also, update the monitor switches to turn lock and uow on.

```
db2 terminate
db2 force application all
```

```
db2 update db cfg for musicdb using locktimeout 90
db2 update monitor switches using lock on uow on
db2 connect to musicdb
```

- \_\_\_ 12. How much time is designated by the locktimeout parameter?

The parameter is specified in seconds, so 90 seconds will be the length of the timeout period. This is not necessarily the appropriate setting. Such will vary from installation to installation. However, this setting will provide enough time for you to complete some snapshot monitoring before the timeout period expires.

- \_\_\_ 13. What occurs by updating the monitor switch for lock to on? (The uow functional group setting to on is used in a later step.)

Greater detail of information will be available when the switch is on. In this case, we will be exploiting some of the finer level of detail to determine the actual lock involved, and the application possessing it, when timeouts and deadlocks are being analyzed.

- \_\_\_ 14. From the **student** window, you decided that the work being performed by the **student** user needs an isolation level of RR. What command could be used to change your isolation level for the **student** window to Repeatable Read? Document your command below and attempt to execute it. Were you successful?

```
db2 terminate
db2 change sqlisl to RR
```

(The keyword isolation could be used instead of sqlisl.)

- \_\_\_ 15. Enter the following from the **student** window where you now have isolation level set to Repeatable Read.

```
db2 connect to musicdb user student using ibm2blue
db2 +c "select * from stock where itemno = 200"
```

- \_\_\_ 16. From your monitor window, obtain a snapshot and state the number of row locks against the STOCK table that the **student** application holds.

```
db2 get snapshot for locks on musicdb > lock2.out
notepad lock2.out
```

**Four row locks are held against the STOCK table. Repeatable read states that share locks for all rows evaluated must be held until commit or roll back. Because of the algorithm used by DB2 to ensure the integrity of indexes, one lock more than the actual number of rows meeting the WHERE clause is obtained.**

- \_\_\_ 17. Exit the Notepad session.

**The next two steps require you to obtain a snapshot before your 90 second timeout period expires.**

- \_\_\_ 18. From your **db2admin** window, issue the following:

```
db2 connect to musicdb user db2admin using ibm2blue
db2 +c "update student.stock set qty = qty + 1 where itemno = 200 "
```

- \_\_\_ 19. It appears that **db2admin** is in a lock wait. Investigate this condition by getting another snapshot from your monitor window.

```
db2 get snapshot for locks on musicdb > lock3.out
notepad lock3.out
```

- \_\_\_ 20. Analyze the portion of the snapshot that is related to the **db2admin** application. Can you suggest the object that is causing the contention?

---

---

---

**The information retrieved from the snapshot identifies that a row lock is causing the contention. This means the X lock required has not been granted. The repeatable read application (**student**) has the row locked with a share lock (S) which is not compatible with the X lock requested.**

**Notice that the text oriented section of the snapshot concerning the usernn application also identifies the competing application and the object of contention. This information is available because the LOCK monitor switch was turned on. Otherwise, this information is not collected.**

**Note:** You may also see an Internal Variation Lock. Any execution of dynamic SQL is done via an entry in the SQL cache known as a variation. In order to ensure that the variation remains valid during the execution, Internal Variation Locks are acquired on behalf of the entry.

- \_\_\_ 21. Exit the Notepad session.
- \_\_\_ 22. If your **db2admin** application is still waiting, sit back and relax for a minute or so. Eventually, something should happen.

What message is returned to the **db2admin** application and why?

---

---

---

**The application received a -911 SQLCODE and a reason code of 68. This indicates that the timeout period for lock wait was exceeded.**

- \_\_\_ 23. Roll back the unit of work in the **student** window to release the share locks.

`db2 rollback`

- \_\_\_ 24. Is it necessary to roll back the update attempted by the **db2admin** application? Why or why not?

---

---

---

**It is not necessary to roll back the update attempt of the **db2admin** application because the timeout detected causes a roll back.**

## Section 3 - Deadlocks

- \_\_\_ 1. Examine the deadlock check time parameter currently active for your database by issuing the following from your monitor window.

```
db2 get db cfg for musicdb | more
```

- \_\_\_ 2. What is the value and how much time does this designate?

---

---

---

**The value of dlchktme is 10000. This translates to 10 seconds since the value is specified in milliseconds.**

- \_\_\_ 3. Issue the following series of SQL statements. **Pay particular attention to the window from which you submit the statements and the order of the statements.** Execute steps **a** and **b**, then for Steps **c** and **d** below, enter the SQL for Step **c**, but don't execute. Then enter the SQL for Step **d** in its window, but don't press Enter. Now go back to Step **c**'s window and execute. Then press Enter from Step **d**'s window. This will insure that the timeout parameter you have set will not interfere with the intention of the exercise. **Remember... we want the isolation level set to Repeatable Read in our instnn window.**

- a. **student** => db2 +c "select \* from stock where itemno = 100"
- b. **db2admin** => db2 +c "update **student**.stock set qty = qty + 1 where itemno = 200"
- c. **student** => db2 +c "select \* from stock where itemno = 200"
- d. **db2admin** => db2 +c "update **student**.stock set qty = qty + 1 where itemno = 100"

- \_\_\_ 4. What did you observe? Why did this occur?

---

---

---

**The database manager detected a deadlock. The SQLCODE of -911 is returned to the application chosen by the database manager as the deadlock victim. The reason code associated with the SQLCODE is a 2.**

**The deadlock resulted because of a classic access order problem. The **student** application obtained S locks on the rows with ITEMNO = 100 in Step a, and the **db2admin** application obtained X locks on the rows with ITEMNO = 200. In Step c, the **student** application attempted to obtain S locks on the rows that were held by **db2admin**. The request is queued. When the **db2admin** application attempts to obtain X locks on the rows already locked by the**



**student application in Step d, a deadlock results. This is because neither application could proceed to commit or roll back the logical unit of work.**

- \_\_\_ 5. Obtain a snapshot from your **monitor** window. This time, request an applications snapshot.

```
db2 get snapshot for applications on musicdb > appl.out
notepad appl.out
```

Look for the item UOW completion status for each application.

- \_\_\_ 6. What is the status for the application associated with the **student** window? Why is this the case?

---



---



---

**The UOW completion status for the **student** application is blank. This is due to the fact that the unit of work is not completed. The **student** application has not issued a commit or a roll back, nor has it been rolled back due to timeout or deadlock.**

- \_\_\_ 7. What is the status for the application associated with the **db2admin** window? Why is this the case?

---



---



---

**The status for the **db2admin** application is Rolled back - Deadlock Victim. This status indicates the action taken by the database manager to relieve the deadlock situation detected.**

- \_\_\_ 8. **Exit the Notepad session.**
- \_\_\_ 9. Release the locks held by the **student** window application.

```
db2 rollback
```

- \_\_\_ 10. Highlight the difference between deadlock and timeout by entering the following series of commands and SQL statements:

- student** => db2 +c "select \* from stock where itemno = 100"
- db2admin** => db2 +c "update **student**.stock set qty = qty + 1 where itemno = 200"
- student** => db2 +c "select \* from stock where itemno = 200"

- \_\_\_ 11. Wait for the database manager to take some action. What action did the database manager take? Why is this the case?

---

---

The database manager timed out the **student** application.

A deadlock situation does not exist unless the two applications can not possibly proceed without intervention. Therefore, the lock wait of the **student** application resulting from the third SQL statement is simply a queued request. When the timeout period expired before servicing the request, the **student** application received a -911 SQLCODE with reason code 68 - indicating a timeout.

- \_\_\_ 12. Roll back the changes requested by the **db2admin** window.

`db2 rollback`

- \_\_\_ 13. Make a general statement about the relationship of the configuration parameters `dlchktime` and `locktimeout`.

---

The parameter `dlchktime` should resolve to a time period that is less than the `locktimeout` parameter. This allows the database manager to distinguish between deadlocks and timeouts, and can provide the DBA with valuable information concerning application interaction.

## Section 4 - Resetting the Environment

Perform the following tasks to reset your system for the remaining labs.

- \_\_\_ 1. Exit your **db2admin** window by typing:  
`exit`
- \_\_\_ 2. Exit your **student** window by typing:  
`exit`
- \_\_\_ 3. From your **monitor** window, execute the following:  
`db2 revoke update on stock from db2admin`  
`db2 update monitor switches using lock off uow off`  
`db2 force application all`
- \_\_\_ 4. Exit from your **monitor** window:  
`exit`

**END OF LAB**



## Exercise 9. Problem determination

*(with hints)*

### Estimated time

00:30

### What this exercise is about

From time to time, the Database Administrator will need to troubleshoot and fix problems that develop. A collection of tools are available in DB2 to help determine problems, and find a resolution to the problems.

You will learn to use several DB2 tools to help determine and resolve database problems.

### What you should be able to do

At the end of the lab, students should be able to:

- Collect information for problem analysis and resolution
- Use error logs for basic problem analysis
- Describe the function of EXPLAIN and use this facility to assist basic analysis
- Use a series of basic commands to better work with connections and sessions

### Introduction

This exercise emphasizes the content of the lecture as well as explores additional areas in greater depth. You will use a variety of tools to investigate common problems, and affect a resolution to the problems.

## ***Instructor Notes:***

*Introduction* — This lab requires the students to work with several DB2 tools to resolve problems in the database.

*Time for Lab* — 30 minutes.

*Things to Review at End of Lab* — ensure that DIAGLEVEL is set to 3.

*Common Mistakes Students May Make* —

## Exercise instructions with hints

The lab solutions using primarily commands are indicated as *Command Window*. The lab solutions using primarily the graphical tools are indicated as *GUI*.

### Section 1 - Collecting information

You will be looking at message logs and the Activity Monitor in this section.

- \_\_\_ 1. So that you will not have too much information to look through, we will clear the db2diag.log file.

Windows	<pre>cd %DB2PATH%\DB2 ren db2diag.log db2diag.log.old</pre>
---------	---

- \_\_\_ 2. Change the DIAGLEVEL instance parameter to 4 to collect more information.

Command Window	<pre>db2 update dbm cfg using diaglevel 4 db2top force db2start</pre>
GUI	<p>___ From the <i>DB2 Control Center</i>, expand the <i>All Systems</i> folder until you get to the <b>DB2</b> instance.</p> <p>___ Right-click your instance, and select <i>Configure parameters</i> from the pop-up menu.</p> <p>___ Select <b>DIAGLEVEL</b>, and change the parameter from 3 to 4.</p> <p>___ Click <b>OK</b>.</p> <p>___ Right-click your instance, and select <i>Stop</i> from the pop-up menu.</p> <p>___ Click <b>OK</b> on the warning window.</p> <p>___ Acknowledge the DB2 Message.</p> <p>___ To restart your instance, right-click your instance, and select <i>Start</i> from the pop-up menu.</p> <p>___ Acknowledge the DB2 Message.</p> <p><b>This action of stopping and starting the instance causes the DIAGLEVEL parameter to be changed immediately.</b></p>

- \_\_\_ 3. Now cause some entries to be placed in the log file.
- Stop and restart the database server.

Command Window	db2stop force db2start
GUI	<p>___ From the DB2 Control Center, expand the All Systems folder until you get to the DB2 instance.</p> <p>___ Right-click your instance, and select Stop from the pop-up menu.</p> <p>___ Click OK on the warning window.</p> <p>___ Acknowledge the DB2 Message.</p> <p><b>This prevents any database connections within that instance. If you had connections, they were disconnected.</b></p> <p>___ To restart your instance, right-click your instance, and select Start from the pop-up menu.</p> <p>___ Acknowledge the DB2 Message.</p> <p><b>This will allow users to connect to any databases within that instance.</b></p>

\_\_\_ 4. View the db2diag.log file.

- Search the db2diag.log file for the db2stop process. Note that you will have several db2stop entries.
- Likewise, search for the db2start process.
- Can you determine the exact time that the server was stopped?
- What other kinds of information is available?
- Where there any applications that had to be forced off?
- What is the DB2 process left running after starting the instance?

---



---



---

Windows	notepad %DB2PATH%\DB2\db2diag.log
---------	-----------------------------------



GUI	<p>— From the DB2 Control Center, open the Command Editor (either choose the <i>Tools</i> option on the menu bar and then select <i>Command Editor</i>, or choose the <i>Command Editor</i> icon on the tool bar (4th from the left)).</p> <p>— Choose <i>Selected</i> from the menu bar.</p> <p>— Choose <i>Open</i> from the menu bar, and navigate to: (Windows): C:\IBM\sqlllib\DB2</p> <p>— Select <b>db2diag.log</b>.</p> <p>— View the <b>db2diag.log</b> information in the <i>Command Editor Command</i> window.</p>
-----	---

**Yes, you can see the exact time and date that the server was acted upon.**

**Process ID: PID, Thread ID: TID, process name PROC, instance name: INSTANCE, node ID: NODE. The number bits used in the system, and the version of DB2 running, and level of Fix Pack applied. Some operating system parameters, such as: number of CPUs, Physical Memory(MB), Virtual Memory(MB), and Swap Memory(MB).**

**Search for the Force keyword. The number of applications needed to be forced may vary, depending on who was connected when the db2stop command was executed.**

**The last process left running after starting the DB2 instance is db2star2 or db2syscs, depending on your operating system type.**

- \_\_\_ 5. Now let's cause some violation to occur, and view the db2diag.log file to see the results.
- Force all applications off the MUSICDB database.
  - In a terminal window, change the name of the dms02 tablespace to dms02.old.
  - Delete the db2diag.log file.
  - Connect to the MUSICDB database.
  - Can you determine the exact error condition we just imposed?
  - What is the first error message you see?
  - What process is having the difficulty?
  - Is there an SQL error code you can look up?

---



---



---

Windows	<pre>db2 force applications all cd C:\dms ren dms02 dms02.old cd %DB2PATH%\DB2 del db2diag.log db2 connect to musicdb more db2diag.log</pre>
---------	--

**Yes, we just caused a container to be not accessible.**

**The first error message you see is:**

```
MESSAGE : ZRC=0x8402001E=-2080243682=SQLB_CONTAINER_NOT_ACCESSIBLE
"Container not accessible"
```

**The process having problems is db2agent (Linux/UNIX), or db2syscs (Windows).**

**Yes, you can look up the SQL error code -290.**

```
db2 "? SQL290"
```

\_\_\_ 6. Replace the dms02 table space.

- Force all applications off the MUSICDB database.
- In a terminal window, change the name of the dms02.old table space to dms02.
- Delete the db2diag.log file.
- Connect to the MUSICDB database.

Windows	<pre>db2 force applications all cd C:\dms ren dms02.old dms02 cd %DB2PATH%\DB2 del db2diag.log db2 connect to musicdb</pre>
---------	---

## Section 2 - Miscellaneous

Some other information gathering tools will be used here, including the LIST APPLICATIONS command, the db2set command, and the db2pd command.

\_\_\_ 1. Use the LIST APPLICATIONS command to view the applications using the MUSICDB database.

- Remove all applications from your instance.
- Use the LIST APPLICATIONS command to view application in the MUSICDB database.
- Are there any applications connected to MUSICDB?
- Connect to the MUSICDB database.
- Re-run the LIST APPLICATIONS command. Now are there any applications connected to the MUSICDB database? If so, How many, and what are they?

---



---



---

Command Window	<pre>cd db2 force applications all db2 list applications for db musicdb show detail &gt; listapps1.out more listapps1.out db2 connect to musicdb db2 list applications for db musicdb show detail &gt; listapps2.out more listapps2.out</pre>
----------------	---

GUI	<p>___ In the <i>Command</i> window of the <i>Command Editor</i>, force applications all.</p> <p>___ In the <i>Command</i> window, enter the command:</p> <pre>db2 list applications for db musicdb show detail</pre> <p>___ Now in the <i>Command Editor</i>, connect to the MUSICDB database.</p> <pre>connect to musicdb</pre> <p>___ In the <i>Command</i> window, re-run the command:</p> <pre>db2 list applications for db musicdb show detail</pre>
-----	--

**No, there are no applications connected to the database.**

**After connecting, there are 3 (or 4 for Windows) applications connected. They are db2taskd, db2stmm, and db2bp (Windows will have the java application connected also).**

- \_\_\_ 2. The db2set command is a command line command, issued at the operating system level.
- Execute the db2set command, showing all profile variables.
  - List the variables found.

---

---

---

Command Window	db2set -all
----------------	-------------

**For Linux/UNIX the following variables are set: an instance variable - DB2COMM, and global variables - DB2SYSTEM, DB2INSTDEF, DB2ADMINSERVER.**

**For Windows, the following variables are set: environment - DB2PATH, instance variables - DB2ACCOUNTNAME, DB2INSTOWNER, DB2PORTRANGE, DB2INSTPROF, DB2COMM, global variables - DB2\_EXTSECURITY, DB2\_DOCPORT, DB2\_DOCHOST, DB2SYSTEM, DB2PATH, DB2INSTDEF, DB2ADMINSERVER.**

**Your results may vary!**

- \_\_\_ 3. The db2pd command is a command line command, issued at the operating system level.
- Execute the db2pd command for the musicdb database, and put the output in a file.
  - For the applications connected to MUSICDB, what kind of transactions are they performing?
  - Are there any locks imposed?
  - How many buffer pools are there? How many buffer pool pages are being used?
  - What table space has the largest number of usable pages?
  - What table space has the largest number of used pages?
  - Have any of the tables been reorganized?

---

---

---

Windows	<pre>cd C:\labfiles\cf23 db2pd -db musicdb &gt; stat.out notepad stat.out</pre>
---------	---

**There are currently five applications connected to MUSICDB, two of which are in a UOW waiting state, performing READ transactions.**

**No locks are taken.**

**There are currently five buffer pools.**

**There are a total of 182 buffer pool pages being used.**

**The dms01 table space has the largest number of usable pages. The SYSCATSPACE has the largest number of used pages.**

**None of the tables have been reorganized.**

**Your results may vary!**

## Section 3 - Reset DIAGLEVEL

Reset the DIAGLEVEL parameter back to its default value of **3**. It is best to only use the value of 4 when troubleshooting. You should not leave it at the value of 4 because the log file will fill up much faster, and it would make viewing the log file more difficult.

\_\_\_ 1. Reset the DIAGLEVEL value back to **3**, then stop and start the instance.

Command Window	<pre>db2 update dbm cfg using diaglevel 3 db2stop force db2start</pre>
GUI	<p>___ From the DB2 Control Center, expand the <i>All Systems</i> folder until you get to the <b>DB2</b> instance.</p> <p>___ Right-click your instance, and select <i>Configure parameters</i> from the pop-up menu.</p> <p>___ Select <b>DIAGLEVEL</b>, and change the parameter from 4 to 3.</p> <p>___ Click <b>OK</b>.</p> <p>___ Right-click your instance, and select <i>Stop</i> from the pop-up menu.</p> <p>___ Click <b>OK</b> on the warning window.</p> <p>___ Acknowledge the DB2 Message.</p> <p>___ To restart your instance, right-click your instance, and select <i>Start</i> from the pop-up menu.</p> <p>___ Acknowledge the DB2 Message.</p> <p><b>This action returns the DIAGLEVEL parameter to its default value immediately.</b></p>

## END OF LAB

# Exercise 10. Application performance tools

(with hints)

## Estimated time

00:30

## What this exercise is about

During this exercise, you will use a few of the available DB2 tools and utilities to assist you in evaluating and resolving some application performance problems.

## What you should be able to do

At the end of the lab, students should be able to:

- Use the Visual Explain tool to perform basic access path analysis
- Use the RUNSTATS utility to update statistics in the DB2 catalog tables
- Use the REORGCHK utility to evaluate whether tables need to be REORGed or not
- REORG a table that has become unclustered

## Introduction

This exercise contains several components to emphasize the content of the lecture as well as explore additional areas in greater depth. This exercise will have you performing most of the tasks from the Control Center.

## Instructor Notes:

*Introduction* — This lab requires the students to use the DB2EXPLN applet that comes with DB2 to externalize their application's access strategy. They will also invoke and use the REORGCHK, REORG, and RUNSTATS utilities.

*Time for Lab* — 30 minutes.

*Things to Review at End of Lab* — Make sure the students were able to use all the tools in the lab successfully.

*Common Mistakes Students May Make* — Not specifying different output file names each time they run DB2EXPLN or REORGCHK, will result in their previous output files getting overlaid or appended to.



## Exercise instructions with hints

### Section 1 - Access strategies and utilities

This exercise will use the Visual Explain GUI tool on the Windows client. If you run Xwindows in your own database environment, you can run the Visual Explain GUI tool on the server. The same content (but not as easy to read) can be found using the text-based db2exfmt tool.

- \_\_\_ 1. Start a DB2 Command Window.
- \_\_\_ 2. Set up for the lab by creating the explain tables. The DDL for creating the explain tables can be found in `sqlllib/misc/EXPLAIN.DDL`, but the DB2 provided DDL will attempt to put the tables into your DMS01 table space which does not have enough space. A customized version of the file can be found in your `labfiles\cf23` directory that stores the tables in USERSPACE1; it is named `explain.ddl`.

```
cd C:\labfiles\cf23
db2 connect to musicdb
db2 -tvf explain.ddl
```

- \_\_\_ 3. Edit the application file `myapp.sqc`.
 

```
strcpy(userid, "student");
strcpy(password, "ibm2blue");
strcpy(dbname, "musicdb");
```
- \_\_\_ 4. Populate the explain tables for the application, `myapp.sqc`.
 

```
db2 prep myapp.sqc explain yes explsnap yes
```

- \_\_\_ 5. Now, on your Control Center session, ensure that you are connected to the MUSICDB database.

**On the Control Center, right-click your MUSICDB database. Choose Connect. Enter your userid and password.**

- \_\_\_ 6. On the Control Center, open the Show Explained Statements History window.

**Right-click your MUSICDB database. Choose Show Explained Statements History.**

- \_\_\_ 7. You should see at least one package, the one for MYAPP that you just created through your `Command Window` session. Scroll to the right to see all of the columns of information. This application happens to have just one SQL statement (thus only one line in the Explained Statements History). What is the statement?

---

**As you scroll to the right, you will see a column entitled, Query text. The statement can be found there. Also, if you highlight the line, and choose**

**Statement > Show SQL Text from the menu bar, you will see the SQL statement.**

**The statement text is:**

```
DECLARE PTC CURSOR FOR SELECT ARTNO, TITLE FROM ALBUMS WHERE ARTNO > 70  
ORDER BY ARTNO
```

- \_\_\_ 8. When you created the package earlier in this exercise, you were told to use the options:

```
explain yes explsnap yes
```

EXPLAIN YES captures explain data for use with the db2exfmt tool.

EXPLSNAP YES captures explain data for use with Visual Explain.

Looking at the output on the Explained Statements History, can you see information that tells you that Visual Explain could be used for this statement?

---

**The Explain snapshot column has a value of “Yes”, indicating that an explain snapshot (which is required for the data to be displayed using Visual Explain) is available for this statement.**

- \_\_\_ 9. Look at the Visual Explain output for this statement.

**Right-click the package name, MYAPP, and choose Show Access Plan.**

Look at the Visual Explain graph starting at the bottom and moving to the top. What table is being accessed?

---

**ALBUMS table.**

- \_\_\_ 10. How is the ALBUMS table being accessed - by index or table scan?

---

**The ALBUMS table is being accessed via a table scan (TBSCAN).**

- \_\_\_ 11. Is a sort required? If so, what is the sort supporting?

---

**A sort is required in this case. The ORDER BY function requires the output to be sequenced by the value of the column ARTNO.**

- \_\_\_ 12. Is this access strategy appropriate? (Consider that the only index on the ALBUMS table is based on the column ITEMNO.)
-

Since the application requests only the rows with **ARTNO > 70** and a sorted result by the **ARTNO** value, and since no index exists that includes the **ARTNO** column, the optimizer chose to do a Table Scan followed by a sort.

- \_\_\_ 13. Assume you have decided to create an index on the **ARTNO** column of the **ALBUMS** table to support ordering requirements as well as support joins to the **ARTISTS** table. (After all, **ARTNO** on the **ALBUMS** table is a foreign key.) Create an index called **artno** on the **artno** column in the **ALBUMS** table.

Command Window	db2 "create index artno on albums (artno) "
GUI	<p>In the <i>Objects</i> pane of the <i>DB2 Control Center</i>, right-click and select <i>Indexes</i>.</p> <p>___ Select <i>Create &gt; Index</i> from the pop-up menu.</p> <p>___ Indicate the following:</p> <p>___ Index name of <b>artno</b>.</p> <p>___ Table schema of <b>STUDENT</b>.</p> <p>___ Table name of <b>ALBUMS</b>.</p> <p>___ Select <b>ARTNO</b> from the <i>Available columns</i> box and click the &gt; button to move <b>ARTNO</b> into the <i>Selected columns</i> box.</p> <p>___ If the <i>Unique</i> checkbox is checked, remove the checkmark by selecting it.</p> <p>___ Click <b>OK</b>.</p> <p>You may get a warning that "Some statistics are in an inconsistent state." This is because index statistics are automatically collected when an index is created through the <i>Control Center</i>. More about statistics collection will be addressed later in class.</p> <p>___ Click <b>Close</b>.</p> <p>___ Click <b>Cancel</b>.</p>

- \_\_\_ 14. Issue a **REBIND** so that a new access path is considered. Document your command below. Remember to connect to your database before running the rebind.

```
db2 connect to musicdb
db2 rebind package myapp
```

- \_\_\_ 15. Did the rebind use the original **.bnd** file on your workstation?

---

---

---

**REBIND does not use a bind file. It simply causes the database manager to optimize the SQL statements in the current package. The timestamp value on the resultant package will not change. REBIND can be useful when NO changes have been made to the source code and it is desirable to go through the optimization process because of new information available to the optimizer.**

- \_\_\_ 16. On the Explained Statements History window, refresh the view. Do you see a new version of the explain data? How can you tell which is the newest version?

---

**On Explained Statements History, choose View -> Refresh on the menu bar.**

**Another version of MYAPP appears. You can tell the latest one by the one with Latest Bind as Yes, or you can look at the Explain Time.**

- \_\_\_ 17. Look at the access strategy for the package with Latest Bind = Yes.

Did the access strategy change? Even though you have completed REBIND before running explain, can you positively state that the access strategy is appropriate?

---

---

---

**On Explained Statements History, right-click the copy of MYAPP with Latest Bind = Yes, and choose Show Access Plan. You may need to adjust your view by zooming out or using the overview window to see the whole plan.**

**The access strategy has changed. The new access plan shows that the optimizer chose to access the new index on ARTNO. You will see a sort and a RIDSCN (RID Scan) - this identifies List Prefetch access. There are now two sorts performed: one to sort the selected Index entries in RID sequence and a second sort to return the data rows in order of the ARTNO value. The optimizer chose the List Prefetch because an accurate estimate of the efficiency of using the new ARTNO index to access the ALBUMS table requires statistics to be gathered. At this time, the statistics for the index on ARTNO have definitely not been gathered since the index was just created. Furthermore, it is not yet determined if any other statistics have been updated.**

- \_\_\_ 18. Since the index on ARTNO was just created, you can state that the statistics for this index have not been gathered (though, CREATE INDEX has an option to COLLECT STATISTICS, we did not specify that). A simple way to check if the table statistics

have been updated is to query the SYSSTAT.TABLES view. Issue the SQL statement from your telnet session:

```
db2 "select card from sysstat.tables where tabname = 'ALBUMS' "
```

\_\_\_ 19. What value was returned and what does it indicate?

**Either a -1 value or a 0 value was returned. -1 indicates that statistics have not been collected for the table. If you have been using the GUI tools, 0 may be returned because statistics were automatically collected when the table was created through the Control Center.**

\_\_\_ 20. Another technique to check on whether statistics have been gathered is to use REORGCHK. Run the following command from your **Command Window** session to display the results of the applied formulas against your tables **without** executing RUNSTATS first.

```
cd C:\labfiles\cf23
```

```
db2 -r rchk1.out reorgchk current statistics on table user
```

Examine the chk.out file. How can you tell from the output that statistics need to be gathered on all your user tables, as well as their indexes?

```
more rchk1.out
```

**The formulas results are all dashes (-). This is how REORGCHK indicates that the formulas are not valid because there is no valid input. This occurs for the table-level formulas as well as the index-level formulas.**

\_\_\_ 21. Execute RUNSTATS on your ALBUMS table and also request the greatest amount of statistics possible including those for indexes.

Command Window	db2 "runstats on table student.albums with distribution and detailed indexes all"
----------------	---

GUI	<p>From the DB2 Control Center, select <i>Tables</i> in the <i>Objects</i> pane.</p> <p>___ Right-click and select <b>ALBUMS</b> and then select <i>Run Statistics</i> from the pop-up menu.</p> <p>___ Indicate:</p> <p>___ On the <i>Column</i> tab, <i>Collect basic statistics on all columns</i>.</p> <p>___ On the <i>Index</i> tab, <i>Collect statistics for all indexes</i>.</p> <p>___ On the <i>Schedule</i> tab, <i>Run now without saving task history</i>.</p> <p>___ Click <b>OK</b>.</p>
-----	--

- \_\_\_ 22. Execute REORGCHK again from your **Command Window** session, but only against the ALBUMS table, to examine the formulas against the statistics you just gathered for the ALBUMS table.

Although you would also gather statistics on your other tables in a production environment, this is not necessary for the lab exercise.

```
cd C:\labfiles\cf23
```

```
db2 -r rchk2.out reorgchk current statistics on table student.albums
more rchk2.out
```

- \_\_\_ 23. Are any of the formulas out of bounds? If any are out of bounds, does that mean that you should perform a REORG?

---



---



---

**None of the formulas are out of bounds. Even if some of the formulas were out of bounds, this does not mean that you must REORG. The output from REORGCHK are recommendations. In fact, you may have more stringent requirements for the space utilization or clustering of some tables. In those cases, you may choose to REORG tables that pass all of the formulas used by REORGCHK. In any case, REORGCHK is a good tool to display the results of the formulas and provide general recommendations.**

- \_\_\_ 24. Is the table in any particular clustering sequence?

---



---



---

**The table is clustered according to both indexes on the ALBUMS table. The data that was loaded into this table has an inherent relationship between the ITEMNO and ARTNO because items were assigned to artists in sequence. This type of relationship will not likely exist for all tables for an installation, but**

**may exist for some. The key is that there will be times when a table can ONLY be clustered according to one index.**

- \_\_\_ 25. From your **Command Window** session, rebind the application, and then look at the new access plan.

```
db2 rebind myapp
```

**On Explained Statements History, choose View -> Refresh on the menu bar.**

**On Explained Statements History, right-click the copy of MYAPP with Latest Bind = Yes, and choose Show Access Plan. You may need to adjust your view by zooming out or using the overview window to see the whole plan.**

- \_\_\_ 26. Did the access strategy change? Can you propose a reason why or why not? Is this access strategy appropriate?

---



---



---

**The access strategy has changed. Now that the optimizer has statistics available for the ALBUMS table and its indexes, the ARTNO index is being used to perform the query without the need for any sorts. The data is clustered by the ARTNO values, so the ARTNO index provides an efficient method of access the table and retrieving the data in sequence by the ARTNO value.**

- \_\_\_ 27. Assume the size of the ALBUMS table was much greater than it actually is, and also assume that the table was only clustered according to the index on ITEMNO. What would you need to do to change the clustering so that the data was in the same order as the index on ARTNO?

---



---

**A REORG on the ALBUMS table would be required. At the time of the REORG, the index on ARTNO could be specified.**

- \_\_\_ 28. Reorganize the ALBUMS table according to the index on ARTNO. You should specify TEMPSPACE1 as the table space used to support the REORG.

---



---

Command Window	db2 reorg table student.albums index student.artno use tempSPACE1
----------------	---

GUI	<p>From the DB2 Control Center, select <i>Tables</i> in the <i>Objects</i> pane.</p> <p>___ Right-click and select <b>ALBUMS</b> and then select <i>Reorganize</i> from the pop-up menu.</p> <p>(If the Progress window appears and more than a few seconds elapse before the "Reorganize Table - ALBUMS" window appears, close the Progress window, shutdown the Control Center, and re-open it.)</p> <p>___ Indicate:</p> <p>___ On the <i>Options</i> tab:</p> <p>___ Select the check box to indicate <i>Reorg using existing index</i>, <b>STUDENT.ARTNO</b>.</p> <p>___ Use the <i>Offline</i> option.</p> <p>___ Select the check box to indicate <i>Use temp space for temporary data</i>, and indicate <b>TEMPSPACE1</b>.</p> <p>___ On the <i>Schedule</i> tab, choose <i>Run now without saving task history</i>.</p> <p>___ Click <b>OK</b>.</p>
-----	--

- \_\_\_ 29. What other steps would be necessary if you were attempting to optimize the statement in the myapp application and you wanted to reflect the result of your REORG? (You need not actually perform these steps.)

---



---



---



---

**After completing REORG, RUNSTATS should be executed to update the statistics. REBIND should then be completed to request a new package to be created.**

- \_\_\_ 30. Close all of your access plan graph windows and the Explained Statements History window.
- \_\_\_ 31. The key points of this section were **NOT** to evaluate the actual access strategies determined against the small lab objects, but to gain understanding regarding the importance and relationships of RUNSTATS, REORG, and REBIND.
- \_\_\_ 32. If you want to explore further, from the Control Center, you can select the database, and from the pop-up menu, select **Explain SQL...** to explain statements that you enter dynamically. You can also examine the options on the Access Plan Graph window. More information on this tool and performance analysis is provided in the Performance Workshop course.



## Section 2 - Program samples

This exercise will familiarize you with the various sample programs and files supplied with DB2. This simple exercise does not require a compiler, but if you needed to work with some of these sample programs, you would need to compile them.

\_\_\_ 1. Navigate to the SQLLIB/samples/c directory.

Windows	<code>cd C:\IBM\SQLLIB\samples\c</code>
---------	---

\_\_\_ 2. Prep one of the files, **getlogs.sqc**.

Command Window	<code>db2 prep getlogs.sqc bindfile</code>
----------------	--

\_\_\_ 3. View the prepared file, **getlogs.c**.

Command Window	<code>more getlogs.c</code>
----------------	-----------------------------

**END OF LAB**



# Exercise 11.Security

*(with hints)*

## Estimated time

01:00

## What this exercise is about

This lab contains several sections. You should attempt to complete the first three sections as time permits, in sequence. Section 4 is considered optional if this is the last lab of the class. However, if this lab is not being done as the last lab, then you should complete Section 4 regardless of the number of other sections completed in order to reset your lab environment.

## What you should be able to do

At the end of the lab, students should be able to:

- Describe the default privileges available to PUBLIC
- Administer DB2 authorities (SYSADM, SYSCTRL, SYSMANT, and DBADM) and describe the functional capabilities of each authority
- Grant privileges to individuals and groups
- Demonstrate when group privileges are applicable and when individual privileges are required

## Introduction

DB2 has the capability to control access to its data by granting privileges to authorized individuals and groups of individuals. Blocks of privileges can also be provided to users through the use of various authority levels. In addition, authentication plays an integral part in the overall security scheme of DB2.

In the exercises that follow, you will have the opportunity to investigate these areas concerning DB2 Security. You will also have several optional opportunities if you wish to go beyond the objectives of the exercises.

## ***Instructor Notes:***

*Introduction* — Students should strive to complete the first three sections of the lab. Section 4 is optional, if this is the last lab of the class. If this unit has been taught earlier in the class, and there are more labs to be done, then Section 4 is required. **Even if a student does not finish the first three sections, they should complete Section 4 before continuing with any other labs (assuming this lab was not the last lab in the class).**

*Time for Lab* — 1 hour.

*Things to Review at End of Lab* — The difference between explicitly granted privileges and implicitly inherited privileges through group membership and authority levels (SYSADM, DBADM, SYSTCL, and so forth).

*Common Mistakes Students May Make* — Students may fail to properly connect or log in as the appropriate user name.

## Exercise instructions with hints

The lab solutions using primarily your ssh/telnet session and commands are indicated as *Command Window*. The lab solutions using primarily the GUI tools are indicated as *GUI*.

### Section 1 - Default PUBLIC Privileges

- \_\_\_ 1. From the DB2 Control Center, determine the current database authorization that all users possess against the database by default.

**From the DB2 Control Center, right-click and select the MUSICDB database.**

**Select Authorities.**

**Select Group tab.**

- \_\_\_ 2. What tasks can anyone that has an ID that has been authenticated successfully do against the MUSICDB database?

---



---



---

**By default, PUBLIC has the capability to CREATE TABLES in the database, ADD new PACKAGES to the database, CONNECT to the database, and CREATE NEW SCHEMAS IMPLICITLY.**

- \_\_\_ 3. The NOFENCE column contains an international *not* sign. What does this mean? Can any user create a fenced user-defined function? Is this an exposure?

---



---



---



---

**Since the value for the NOFENCE column is not a check, this means that the PUBLIC is NOT permitted to create NOT FENCED user-defined functions. NOT FENCED functions execute without the protection of a firewall between the function code and the database agent code. Therefore, defining such functions is considered a privilege.**

**Another privilege associated with user-defined functions is CREATE EXTERNAL ROUTINE. This is also not given to PUBLIC by default.**

- \_\_\_ 4. The LOAD column contains an international not sign. Who can load data?

---



---

**SYSADM, DBADM, or someone with LOAD privilege on the database, and**

- (1) insert privilege on the table if performing a LOAD INSERT, or**
- (2) insert and delete privileges on the table if performing a LOAD REPLACE, and**
- (3) insert privilege on the exception table.**

- \_\_\_ 5. Can you tell that this grant is done automatically by the database manager during some process? How could you find out this information?

---

**No, the GRANTOR does not appear in this display. To see the full grant, display the contents of the SYSCAT.DBAUTH view.**

**Cancel the Authorities window.**

**Select Views.**

**Select SYSCAT.DBAUTH view, right-click and select Open.**

**Look at the GRANTOR column for GRANTEE PUBLIC.**

**Click Close, when finished.**

**The GRANTOR of SYSIBM indicates that this authorization grant is done by the database manager itself. In fact, this particular grant is completed during the creation of the database.**

- \_\_\_ 6. Establish a session from which you will connect to the database as a different user.

Windows	Open a DB2 Command Window. Enter title <code>db2admin</code> .
---------	---

- \_\_\_ 7. Connect to the musicdb database as `db2admin`.

`db2 connect to musicdb user db2admin using ibm2blue`

- \_\_\_ 8. Recall that you created several table spaces during a prior lab and you determined and placed specific tables in those table spaces as part of your design. Assume that you are `db2admin` and you have just connected. You now want to create a test table in order to play with some SQL recursive query syntax that you have just learned.

Issue the following SQL statements to determine the table spaces that have been defined on the system.

`db2 "select substr(tbspace,1,18) from syscat.tablespace"`

- \_\_\_ 9. Did the SQL statement complete successfully? Why or why not?

---

---

---

**The SQL statement completed successfully because PUBLIC is granted the SELECT privilege to the system catalog tables by default. Therefore, db2admin was able to obtain the information regarding the table space names defined on the production system.**

- \_\_\_ 10. As db2admin, you decide that the DMS04 table space looks as good as any to contain your test table. Issue the following SQL statement to create your test table.

```
db2 "create table test1 (partno integer, subpart integer) in DMS04"
```

Did the statement complete successfully? What does this imply?

---

---

---

**The statement did not complete successfully. db2admin does not have USE privilege on the table space DMS04.**

- \_\_\_ 11. As db2admin, you ask your good friend, the corporate spy, who happens to have DBADM privilege, to grant you USE privilege on DMS04.

Put on your alter-ego of corporate spy, and on your student session, grant USE on DMS04 to db2admin.

```
db2 connect to musicdb user student using ibm2blue
```

```
db2 grant use of tablespace dms04 to user db2admin
```

- \_\_\_ 12. As db2admin, try creating your test table again. Issue the following SQL statement to create your test table.

```
db2 "create table test1 (partno integer, subpart integer) in DMS04"
```

Did the SQL statement complete successfully? What does this imply?

---

---

---

**The SQL statement completed successfully. With USE privilege on the table space, a user with CREATETAB privilege can create tables in that table space.**

- \_\_\_ 13. Assume db2admin used table test1 for some period of time. You should now assume the role of the system administrator (the good guy, not the corporate spy). You have been analyzing a problem concerning the space available for the DMS04 table space. As part of your analysis, you need to know the table space ID for DMS04.

What is the table space ID for DMS04? Document it below.

---

---

Command Window	db2 "list tablespaces"   more
GUI	<p>From the DB2 Control Center, select Views under the MUSICDB database in the Object tree.</p> <p>___ Right-click and select TABLESPACES view.</p> <p>___ Select Open from the pop-up menu.</p> <p>___ When finished, click Close.</p>

### The DMS04 table space is 7.

- \_\_\_ 14. You now decide to find out just what tables have been created in DMS04. Execute a SELECT statement that will show you what tables are in the table space ID (TBSPACEID) of DMS04.

Command Window	<pre>db2 connect to musicdb db2 "select substr(tabschema, 1, 18) as tabschema, substr(tabname, 1, 18) as tabname from syscat.tables where tbspaceid = 7"</pre>
GUI	<p>In the Command Editor, Commands tab.</p> <pre>connect to musicdb user student using ibm2blue select tabschema, tabname from syscat.tables where tbspaceid = 7</pre>

- \_\_\_ 15. What do you discover? When you originally estimated the size requirements for DMS04, did you anticipate this?

---



---



---



---

The DMS04 table space contains three tables. Two of these tables, **STUDENT.ALBUMS** and **STUDENT.CONCERTS** were placed in this table space after careful design of the system as well as sizing of the table space was considered. The table **DB2ADMIN.TEST1** was not considered in the design process. It is, in fact, an unwelcome guest.

- \_\_\_ 16. As **student**, drop the table **DB2ADMIN.TEST1** to remove it from the database.

Command Window	<pre>db2 connect to musicdb db2 drop table db2admin.test1</pre>
----------------	---



GUI	<p>From the DB2 Control Center, select <i>Tables</i> under the <b>MUSICDB</b> database in the <i>Object</i> tree.</p> <p>___ Scroll down the contents pane until you find the <b>TEST1</b> table. IF IT DOES NOT APPEAR IN THE LIST:</p> <p>___ Right-click and select <i>Tables</i> under the <b>MUSICDB</b> database in the <i>Object</i> tree.</p> <p>___ Select <i>Refresh</i> from the pop-up menu.</p> <p>___ Right-click and select <b>TEST1</b> table and then select <i>Drop</i> from the pop-up menu.</p> <p>___ Click OK.</p>
-----	--

\_\_\_ 17. Why were you able to drop a table in a schema different than your own?

---



---



---

**The ID performing the drop is a system administrator. A SYSADM has the highest level of authority on every database in the instance. A user with DBADM authority could also drop the table, as could the owning ID.**

\_\_\_ 18. Certainly, you do not want your production system exposed to object creation from users creating test objects. List the steps you should take in order to begin securing the production environment.

---



---



---

**In this particular case, we saw that, originally, **db2admin** was prevented from creating objects in DMS04 because **USE** privilege on the table space had not been granted. **USE** privilege allows you to control who has the ability to use a table space for storing data for their tables. However, if the corporate spy had not granted **USE** privilege to **db2admin**, **db2admin** could still have created their table in **USERSPACE1** (which has **USE** privilege given to **PUBLIC** by default), which still would have had impact on the storage of our system, and which we may want to maintain control over.**

**Production systems generally need to be controlled explicitly. In order to begin securing the environment, **REVOKE** of the **PUBLIC** privilege to **CONNECT**, **CREATETAB**, and **BINDADD** would be generally necessary. You may also wish to revoke the **USE** privilege on **USERSPACE1**.**

**Some installations would also consider revoking the **SELECT** authority to the catalog tables and views from **PUBLIC** and administering such privileges explicitly.**

**This is not meant to imply that such a strategy is appropriate in all situations. Your installation may have different requirements - for example, you may want PUBLIC to retain the CONNECT authority but not have any CREATETAB or BINDADD privilege.**

\_\_\_ 19. Revoke the default database privileges that have been granted to PUBLIC.

Command Window	<code>db2 "revoke bindadd, connect, createtab on database from public"</code>
GUI	<p>From the DB2 <i>Control Center</i>, select + sign to the left of <i>User and Group Objects</i> under the <b>MUSICDB</b> database in the <i>Object tree</i>.</p> <p>___ Select <i>DB Groups</i> under <i>User and Group Objects</i>.</p> <p>___ Right-click and select <b>PUBLIC</b> group from the <i>Contents</i> pane and then select <i>Change</i> from the pop-up menu.</p> <p>___ Select checkboxes to the left of <i>Connect to database</i>, <i>Create tables</i>, and <i>Create packages</i>, to uncheck them.</p> <p>___ Click <b>OK</b>.</p>

\_\_\_ 20. From your **db2admin** session, attempt to submit the CREATE TABLE statement again.

```
db2 "create table test1 (partno integer, subpart integer) in DMS04"
```

\_\_\_ 21. Is the user able to create a table?

---

**The user cannot create a table because the CREATETAB privilege has been revoked.**

\_\_\_ 22. Is **db2admin** still connected to the database? Does this mean the revoke of the CONNECT authority was not completed?

---



---

**The **db2admin** connection is still active, otherwise the message concerning the error would have been related to connect privilege not possessed. However, revoke did remove the connect authorization from PUBLIC.**

**Connection authorization is checked when a user establishes the connection. At the time **db2admin** connected to the database, the authorization to connect was granted to PUBLIC.**

\_\_\_ 23. Assume you had a business reason to revoke connect from a user and then ensure that any connections established by the user were disconnected. You have already

completed the revoke. Document and complete the remaining steps required.  
(Consult the solutions if you are not sure of your process.)

---



---



---

Command Window	<pre>db2 list applications</pre> <p>Note the application handle associated with the <code>db2admin</code> application.</p> <pre>db2 "force application (nnn)"</pre> <p>Where "nnn" is the application handle associated with the <code>db2admin</code> application.</p>
GUI	<p>From the <i>Control Center</i>, list the applications currently connected.</p> <ul style="list-style-type: none"> <li>___ From the <i>DB2 Control Center</i>, right-click and select <code>DB2</code> instance in the <i>Object tree</i> of the <i>Objects</i> pane, and then select <i>Applications</i> from the pop-up menu.</li> <li>___ Right-click and select application associated with <code>db2admin</code>.</li> <li>___ Click <b>Force</b>.</li> <li>___ Click <b>Close</b>.</li> </ul>

- \_\_\_ 24. From your `db2admin` session, attempt to submit the CREATE TABLE statement again.

```
db2 "create table test1 (partno integer, subpart integer) in DMS04"
```

- \_\_\_ 25. The result of the attempt leads you to believe the connection was lost or was terminated. Submit the following from the `db2admin` session.

```
db2 connect to musicdb user db2admin using ibm2blue
```

- \_\_\_ 26. Was the connect successful?
- 
- 

**The connect was not successful. DB2ADMIN does not have the CONNECT privilege. Therefore, the system administrator successfully revoked the connect privilege and disconnected the `db2admin` session.**

- \_\_\_ 27. Exit from the `db2admin` window.

## Section 2 - DB2 Authorities

- \_\_\_ 1. As **student**, you have been performing sysadm tasks since the creation of the instance. However, in the past week, you worked over 80 hours in order to manage massive changes and unforeseen problems with the system, and some of those hours were well after midnight. You decide to make **admin** a sysadm to relieve some of the strain.

From your **Command Window** session, look at the database manager configuration file and record the sysadm group name.

---

```
db2 get dbm cfg | more
```

**The current sysadm group is .**

**The current sysadm group is empty.**

- \_\_\_ 2. To add new group names and users, you will need to have Windows **Administrator** privileges. As the user student, you have Administrator privileges.
- \_\_\_ 3. Create a new Windows user, named **admin**, that belongs to the **admin** group.

Windows	<p>Create the <b>admin</b> user and group.</p> <p>___ On your Windows desktop, right click on the <i>My Computer</i> icon and select <i>Manage</i>. This opens a <i>Computer Managment</i> window.</p> <p>___ Expand the <i>Local Users and Groups</i> folder, and right click on the <i>Groups</i> folder. Select <i>New Group</i>. This opens a <i>New Group</i> window.</p> <p>___ Enter <b>admin<b>db</b>2</b> as the new group name.</p> <p>___ Click the <b>Create</b> button.</p> <p>___ Click the <b>Close</b> button.</p> <p>___ Right click on the <i>Users</i> folder. Select <i>New User</i>. This opens a <i>New User</i> window.</p> <p>___ Enter <b>admin</b> as the new user name.</p> <p>___ Enter <b>ibm2blue</b> as the password in both password locations.</p> <p>___ Uncheck the box labeled <i>User must change password at next login</i>.</p> <p>___ Check the box labeled <i>Password never expires</i>.</p> <p>___ Click the <b>Create</b> button.</p> <p>___ Click the <b>Close</b> button.</p> <p>___ Right click on the <b>admin</b> user (in the left pane), and select <i>Properties</i>.</p> <p>___ Click on the <i>Member Of</i> tab.</p> <p>___ Click the <b>Add</b> button.</p> <p>___ Select the <b>admin<b>db</b>2</b> group.</p> <p>___ Click the <b>Add</b> button.</p> <p>___ Click the <b>OK</b> button.</p> <p>___ You should see that the <b>admin<b>db</b>2</b> group has been added to the <b>admin</b> user.</p> <p>___ Click the <b>OK</b> button.</p>
---------	--

- \_\_\_ 4. What IDs currently act as sysadm? How would you provide the sysadm capability to additional users?

---



---



---

The output from the command shows that only one user is currently a member of the sysadm local group - **student**. Only this user can currently perform tasks requiring the sysadm authority level. Additional users can be given the privileges associated with the sysadm level of authority by adding them to the **admin**db**2** group.

- \_\_\_ 5. Complete the necessary steps to make **admin** a system administrator.

Do this by executing the following set of commands:

Windows	<p>___ On your Windows desktop, right-click the <i>My Computer</i> icon and select <i>Manage</i>. This opens a <i>Computer Managment</i> window.</p> <p>___ Expand the <i>Local Users and Groups</i> folder, and right-click the <b>admin</b> user (in the left pane), and select <i>Properties</i>.</p> <p>___ Click the <i>Member Of</i> tab.</p> <p>___ Click the <b>Add</b> button.</p> <p>___ Select the <i>Administrators</i> group.</p> <p>___ Click the <b>Add</b> button.</p> <p>___ Click the <b>OK</b> button.</p> <p>___ You should see that the <b>Administrators</b> group has been added to the <b>admin</b> user.</p> <p>___ Click the <b>OK</b> button.</p>
---------	--

- \_\_\_ 6. Have you made **admin** a sysadm? What types of tasks could this user now complete?

---



---



---

**The ID **admin** is now a sysadm. This ID could complete all tasks related to the database manager, including interrogating and changing all the data, creating and dropping any object, running any utility, controlling authorization, and so on.**

- \_\_\_ 7. Establish a session from which you will connect to the database as **admin**.

GUI	<p>Open a DB2 <i>Command Window</i>.</p> <p>Enter title <b>admin</b>.</p>
-----	---

- \_\_\_ 8. From the **admin** session, prove that **admin** has the appropriate authority. Recall in the previous section that the database privileges have been REVOKED from PUBLIC.

Complete the following:

```
db2 connect to musicdb user admin using ibm2blue
db2 select name from student.artists
db2 grant delete on student.artists to public
db2 revoke delete on student.artists from public
db2 attach to student user admin using ibm2blue
```

```
db2 get snapshot for locks on musicdb
```

- \_\_\_ 9. Did any of the commands or SQL statements fail due to authorization errors? Could this ID even drop some of the **student**'s objects? (**DO NOT** attempt to test this case.) What is your opinion regarding the number of users that should be placed in the sysadm group? What does this imply about coordination between database and operating system security personnel?

**Nothing failed. The user with sysadm can do any database task, including dropping any object.**

**The number of users placed in the sysadm group is normally limited, especially in a production environment. Otherwise, the security of the database is exposed.**

**The IDs that belong to a group are managed by operating system security personnel. If these individuals are NOT the same as those managing database privileges, a high degree of coordination is required. As a database administrator, and possibly a sysadm, you should be greatly concerned about the security mechanisms your installation has in place to control group membership.**

- \_\_\_ 10. Determine your current authorizations for your **admin** ID by issuing:

```
db2 get authorizations
```

- \_\_\_ 11. What, if any, item in the list is designated as YES? What does this mean?

**Indirect SYSADM authority is designated as YES. This means that the user is a member of the sysadm group. Also Indirect IMPLICIT\_SCHEMA is designated as YES. This means the user can create a schema implicitly by creating an object using a CREATE statement with a schema name that does not already exist. SYSIBM becomes the owner of the implicitly created schema, and PUBLIC is given the permission to create objects in this schema.**

**Indirect means that the authority is possessed through membership in a group, while direct means that an explicit GRANT has been executed to provide the authority.**

- \_\_\_ 12. From the **student** window, determine your database authorizations for your **student** ID.

```
db2 connect to musicdb
db2 get authorizations
```

- \_\_\_ 13. Notice that several Direct authorities are designated as YES. Does this mean that **student** has greater capability than **admin**? For example, can only **student** create a not fenced user-defined function?

---

---

---

**The Direct authorities result from the grants done by the system (SYSIBM) when the database was created. They can actually be viewed through SYSCAT.DBAUTH. However, this does not mean that **student** has greater capability than **admin**. Both of these IDs belong to the sysadm group. Remember that the sysadm authority encompasses all privileges on the system. Therefore, **admin** and **student** can be considered peers. The Direct authority for the **student** could be considered redundant when sysadm authority is also possessed.**

- \_\_\_ 14. Since the **admin** ID is a sysadm, it can update the database manager configuration. **From the **admin** window, update the configuration so that the sysctrl group is set to **ctrlldb2** and the sysmaint group is set to **maintdb2**.**

```
db2 update dbm cfg using sysctrl_group ctrlldb2 sysmaint_group maintdb2
```

- \_\_\_ 15. Was **admin** allowed to update the database manager configuration file?

---

---

---

**Yes, the statement completed successfully.**

- \_\_\_ 16. List the database manager configuration file to verify the changes have been made by issuing the following command:

```
db2 get dbm cfg | more
```

- \_\_\_ 17. What enabled you, from your **admin** window, to issue DB2 commands against your **DB2** instance?

---

**The attach to command issued in a previous step.**



- \_\_\_ 18. On your Command Window session, create two new groups by the names of **ctrlldb2** and **maintdb2**. Make user **ctrl** a member of the **ctrlldb2** group and user **maint** a member of the **maintdb2** group.

Windows Step 1	<p>Create the <b>ctrlldb2</b> group and the <b>maintdb2</b> group.</p> <p>___ On your Windows desktop, right-click the <i>My Computer</i> icon and select <i>Manage</i>. This opens a <i>Computer Managment</i> window.</p> <p>___ Expand the <i>Local Users and Groups</i> folder, and right-click the <i>Groups</i> folder. Select <i>New Group</i>. This opens a <i>New Group</i> window.</p> <p>___ Enter <b>ctrlldb2</b> as the new group name.</p> <p>___ Click the <b>Create</b> button.</p> <p>___ Enter <b>maintdb2</b> as the new group name.</p> <p>___ Click the <b>Create</b> button.</p> <p>___ Click the <b>Close</b> button.</p>
-------------------	--

<p>Windows Step 2</p>	<p>Create the <b>ctrl</b> user and the <b>maint</b> user.</p> <p>__ Right-click the <i>Users</i> folder. Select <i>New User</i>. This opens a <i>New User</i> window.</p> <p>__ Enter <b>ctrl</b> as the new user name.</p> <p>__ Enter <b>ibm2blue</b> as the password in both password locations.</p> <p>__ Uncheck the box labeled <i>User must change password at next login</i>.</p> <p>__ Check the box labeled <i>Password never expires</i>.</p> <p>__ Click the <b>Create</b> button.</p> <p>__ Enter <b>maint</b> as the new user name.</p> <p>__ Enter <b>ibm2blue</b> as the password in both password locations.</p> <p>__ Uncheck the box labeled <i>User must change password at next login</i>.</p> <p>__ Check the box labeled <i>Password never expires</i>.</p> <p>__ Click the <b>Create</b> button.</p> <p>__ Click the <b>Close</b> button.</p> <p>__ Right-click the <b>ctrl</b> user (in the left pane), and select <i>Properties</i>.</p> <p>__ Click the <i>Member Of</i> tab.</p> <p>__ Click the <b>Add</b> button.</p> <p>__ Select the <b>ctrl<b>db</b>2</b> group.</p> <p>__ Click the <b>Add</b> button.</p> <p>__ You should see that the <b>ctrl<b>db</b>2</b> group has been added to the <b>ctrl</b> user.</p> <p>__ Click the <b>OK</b> button.</p> <p>__ Right-click the <b>maint</b> user (in the left pane), and select <i>Properties</i>.</p> <p>__ Click the <i>Member Of</i> tab.</p> <p>__ Click the <b>Add</b> button.</p> <p>__ Select the <b>maint<b>db</b>2</b> group.</p> <p>__ Click the <b>Add</b> button.</p> <p>__ Click the <b>OK</b> button.</p> <p>__ You should see that the <b>maint<b>db</b>2</b> group has been added to the <b>maint</b> user.</p> <p>__ Click the <b>OK</b> button.</p>
---------------------------	---

\_\_ 19. Make the changes effective by completing the following from your **student** session.

```
db2 terminate
db2stop force
db2start
```

\_\_ 20. As **admin**, check the current contents of the dbm cfg to see if the update for SYSCTRL\_GROUP and SYSMANT\_GROUP that **admin** did was effective.

```
db2 attach to DB2 user admin using ibm2blue
db2 get dbm cfg show detail | more
```

Check the sysctrl\_group and sysmaint\_group parameters.

**Yes, the updates were done. The sysctrl\_group and sysmaint\_group parameters show values of CTRLDB2 and MAINTDB2 respectively.**

- \_\_\_ 21. Establish two sessions, one for ctrl and the other for maint.

GUI	Open two DB2 Command Windows and title each window with the user's name.
-----	--

- \_\_\_ 22. From the ctrl session, determine your authorizations.

```
db2 connect to musicdb user ctrl using ibm2blue
db2 get authorizations
```

- \_\_\_ 23. Does the SYSCTRL authority imply that you have the capability to CONNECT to any database?

---



---



---

**Since the connect executed and direct or indirect authority to connect is not listed as one of the authorizations possessed by the user, it is implied that the SYSCTRL authority provides connect capability. As stated in the lecture, SYSCTRL has other capabilities as well.**

- \_\_\_ 24. From the maint session, determine your authorizations.

```
db2 connect to musicdb user maint using ibm2blue
db2 get authorizations
```

- \_\_\_ 25. Does the SYSMANT authority imply that you have the capability to CONNECT to any database?

---



---



---

**Since the connect executed and direct or indirect authority to connect is not listed as one of the authorizations possessed by the user, it is implied that the SYSMANT authority provides connect capability. As stated in the lecture, SYSMANT has other capabilities as well.**

- \_\_\_ 26. From your ctrl session (SYSCTRL), determine the agent associated with the maint session. You may need to attach to your instance first.

```
db2 attach to DB2 user ctrl using ibm2blue
db2 list applications
```

- \_\_\_ 27. As the SYSCTRL, attempt to force the agent associated with user **maint** (SYSMAINT).

```
db2 "force application (n)"
```

Where **n** represents the appl handle you determined from the prior command.

- \_\_\_ 28. What did you observe? Which authority, SYSCTRL or SYSMAINT, appears to possess a greater level of authority?

---

---

**The command was successful. But just because ctrl forced off maint, does not prove ctrl (SYSCTRL) possesses a greater level of authority. However, from lecture you should have learned that SYSCTRL does possess a higher level of authority than SYSMAINT. Indeed, one must be SYSADM or SYSCTRL to issue the FORCE APPLICATION command.**

- \_\_\_ 29. From your **ctrl** session, attempt to submit some SQL statements.

```
db2 "create table test1 (col1 char(1))"
db2 drop table student.reorder
db2 "select * from student.music"
```

- \_\_\_ 30. Were any of the statements successful? **DO NOT DO ANY OTHER GRANTS AT THIS TIME!** Do you think the same could be said if the same statements were attempted from the ID possessing sysmaint authority?

---

---

---

**None of the SQL statements executed. All encountered authorization errors. The sysctrl authority provides a great deal of privileges, but these privileges do not extend to accessing data objects. The sysmaint authority, being a true subset of the sysctrl authority, cannot do anything that the sysctrl cannot do.**

- \_\_\_ 31. Try one more SELECT statement from the sysctrl user of **ctrl**.

```
db2 select name from student.artists
```

- \_\_\_ 32. Why did this SQL statement work? Is the sysctrl authority regarding table access table specific?

---

---

---

**The SELECT statement worked because the privilege to select from the INSTnn.ARTISTS table has been granted to PUBLIC.**

**Privileges associated with the sysctrl authority are not table-specific - they are instance-oriented.**

- \_\_\_ 33. Illustrate the reason the previous SELECT statement was successful, by entering the following from **ctrl**:

```
db2 "select grantee from syscat.tabauth where tabname = 'ARTISTS'"
```

- \_\_\_ 34. Issue a **db2 terminate** from both the **ctrl** and **maint** sessions.
- \_\_\_ 35. You have decided that you would like to make **dba** a **DBADM** at your installation. From your **student** session, make **dba** a DBADM.

Command Window	On your <b>student</b> session: db2 connect to musicdb db2 grant dbadm on database to user dba
GUI	From the DB2 Control Center, right-click and select MUSICDB database in the Object tree of the Objects pane. ___ Select Authorities from the pop-up menu. ___ Click the Add User button. ___ Select userid dba from the pull-down menu. ___ Click the OK button. ___ On the User tab, select user dba. ___ Select pull-down arrow to the right of the DBADM Authorities box and select Yes from the pull-down menu. ___ Click the OK button.

- \_\_\_ 36. Check the database authorization catalog view to see the result of your grant.

```
db2 "select * from syscat.dbauth" | more
```

Notice it shows **dba** has DBADM authority and that it was granted by **STUDENT**.

- \_\_\_ 37. How do you distinguish the internal DBADM grant given to the database creator from other DBADM grants, since the privilege list is identical?

**The internal grant completed when the database is created is identified by the grantor column containing SYSIBM.**

\_\_\_ 38. Establish a new session for **dba**.

Command Window	Establish a <b>Command Window</b> sessions for <b>dba</b> .
GUI	Open a DB2 <i>Command Window</i> and title it with the user's name.

\_\_\_ 39. From the **dba** session, connect to database musicdb.

```
db2 connect to musicdb user dba using ibm2blue
```

\_\_\_ 40. As the new dbadm, attempt to execute the following series of SQL statements.

```
db2 select name from student.artists
db2 "create table test1 (col1 char(1))"
db2 "create table student.dlc (col1 char(1))"
db2 drop table student.dlc
```

\_\_\_ 41. Did any of the SQL statements fail? Notice that you have created and dropped objects in another user's schema. Can any user with the authority to create tables complete such a task?

---

---

---

**None of the SQL statements failed because the dbadm authority granted to **dba** permits the **dba** ID to perform all the requested functions. The capability to create and drop objects in another user's schema is limited to the dbadm and sysadm authorities.**

\_\_\_ 42. The SELECT statement is just a simple example of a Data Manipulation Language (DML) statement privilege given to DBADM. Can you make a broader statement concerning DBADM and DML?

---

---

---

**The SELECT statement used as an example is just one of the DML statements that can be executed by a DBADM. Users with such authority can execute any DML statement (SELECT, INSERT, UPDATE, and DELETE) on ANY table in the database. Furthermore, they have CONTROL over every table in the database. Therefore, they could GRANT privileges with the GRANT OPTION to other users, DROP objects, and ALTER objects.**

- \_\_\_ 43. In addition to wide-sweeping SQL authority, dbadm has the capability to run certain utilities against the database, such as REORG and RUNSTATS. However, you do not need to demonstrate such capabilities at this time.

Recall that sysctrl had the privilege to issue the force command but did not have the privileges associated with data access. What if your desire was to create an ID with dbadm authority in addition to sysctrl authority - how would you accomplish the desired result?

**It is possible to place the user ID in the sysctrl group and then grant that ID DBADM authority as well.**

## Section 3 - DB2 User and Group Privileges

- \_\_\_ 1. As **dba**, who is a DBADM, insert the following rows in the **STUDENT.ARTISTS** table.

```
db2 "insert into student.albums values('DB2 Connect is My Life', 99, 310) "
```

```
db2 "insert into student.albums values('Rustle Up', 99, 311) "
```

- \_\_\_ 2. Recall that **maint** is a sysmaint and has no explicit SQL privileges. Grant connect database and delete table privileges to **maint**, from your **dba** session as follows:

```
db2 grant connect on database to user maint
```

```
db2 grant delete on student.albums to user maint
```

- \_\_\_ 3. From your **maint** session, connect to the database as **maint**. Attempt to delete a row from the **STUDENT.ALBUMS** table.

```
db2 connect to musicdb user maint using ibm2blue
```

```
db2 delete from student.albums where itemno = 310
```

The scenario shows that a DBADM has the authority to manipulate objects in another schema, and grant access to the data in such objects as well.

- \_\_\_ 4. Assume that **student** has discovered that the delete capability against his or her table has been granted by **dba**. This is considered outrageous, so **student** elects to remove the DBADM authority from **dba**.

Revoke DBADM privilege from **dba**.

```
db2 revoke dbadm on database from dba
```

- \_\_\_ 5. In the meantime, **dba** has heard that **student** is mad about the deletion of a row and tries to put it back. From the **dba** window, issue the SQL statement:

```
db2 "insert into student.albums values('DB2 Connect is My Life', 99, 310) "
```

- \_\_\_ 6. Why was this SQL statement unsuccessful?

---

---

---

**Since **dba** is no longer a dbadm, the authority to perform DML against any table has been lost.**

- \_\_\_ 7. What if **dba** had created tables during the time DBADM authority was possessed - would **dba** still have authority over those objects?

---

---

---



**Losing dbadm authority does not impact the control privileges held by virtue of being the creator of an object.**

- \_\_\_ 8. Prove (or disprove) your response to the prior question by submitting the following from the **dba** session. Recall **dba** created table test1.

```
db2 "select * from test1"
db2 drop table test1
```

- \_\_\_ 9. Recall that **maint** was granted delete authority by **dba** when the ID possessed the DBADM authority. Is this user still permitted to delete from the **STUDENT.ALBUMS** table? Find out by submitting the following SQL statement from the **maint** session.

```
db2 delete from student.albums where itemno = 311
```

- \_\_\_ 10. What does the result indicate?

---



---



---

**Privileges granted by a dbadm are not revoked simply because the dbadm authority is revoked.**

- \_\_\_ 11. In order to try to be forgiven by **student**, **dba** wants to revoke the grant previously given to **maint**. From the **dba** session, issue:

```
db2 revoke delete on student.albums from maint
```

- \_\_\_ 12. Even though **dba** originally did the grant, is this ID able to revoke the privilege?

---



---

**The **dba** privilege to CONTROL the **STUDENT.ALBUMS** table has also been lost when the DBADM authority was revoked. Therefore, the user cannot perform any operations against this table.**

- \_\_\_ 13. **STUDENT** decides to check on the authorization status of **dba**, as well as revoke the delete privilege from **maint**. Do these from the DB2 Control Center as follows:

Check the database privileges:

- \_\_\_ From the *DB2 Control Center*, select **Views** under the *MUSICDB* database in the *Object* tree.
- \_\_\_ Right-click and select **DBAUTH** view and then select **Open** from the pop-up menu.
- \_\_\_ Note the privileges **dba** still possesses.
- \_\_\_ Click **Close** button.

Check and revoke table privileges:

- \_\_\_ From the *DB2 Control Center*, select **Tables** under the *MUSICDB* database in the *Object* tree. Right-click **Tables** and select **Refresh**.
- \_\_\_ Right-click and select **ALBUMS** table and then select **Privileges** from the pop-up menu.
- \_\_\_ Select user **maint**.
- \_\_\_ Select pull-down arrow for the DELETE Privileges box and select **No** from the pull-down menu.
- \_\_\_ Click **OK** button.

\_\_\_ 14. Does **dba** possess any database privileges?

---

---

---

**ID **dba** still has the database privileges CONNECT, CREATETAB, NOFENCEAUTH, BINDADD, IMPLSCHEMAAUTH, LOADAUTH, EXTERNALROUTINEAUTH, and QUIESCECONNECTAUTH. If these privileges need to be revoked from the user, they would need to be done so explicitly.**

\_\_\_ 15. Did the REVOKE of delete privilege successfully complete? If so, why could **student** revoke the grant completed by **dba**?

---

---

---

**The REVOKE executed successfully. An individual with dbadm authority is permitted to revoke another user's grant.**

\_\_\_ 16. DB2 security can be managed in part via group authorization. Begin to investigate this capability by granting insert privilege on the ALBUMS table to the group staff.

---

Command Window	On your <b>student</b> session: <b>db2 grant insert on table albums to group staff</b>
----------------	---

GUI	<p>From the DB2 Control Center, select <i>Tables</i> under the <b>MUSICDB</b> database in the object tree.</p> <ul style="list-style-type: none"> <li>___ Right-click and select <b>ALBUMS</b> table and then select <i>Privileges</i> from the pop-up menu.</li> <li>___ On the <i>Group</i> tab, click the <b>Add Group</b> button.</li> <li>___ Select <i>Group Staff</i> from the pull-down menu.</li> <li>___ Click the <b>OK</b> button.</li> <li>___ Select <i>Staff Group</i>.</li> <li>___ Select pull-down arrow associated with the <i>INSERT Privileges</i> box and select <i>Yes</i> from the pull-down menu.</li> <li>___ Click the <b>OK</b> button.</li> </ul>
-----	--

- \_\_\_ 17. The **dba** user is a member of the staff group. You have now given the privilege to insert into the ALBUMS table to the staff group.

Can all of the users in the staff group insert into the ALBUMS table, or do they require an additional privilege as well?

---



---



---

**The members of the staff group could issue dynamic insert statements against the ALBUMS table, provided they also had the capability to CONNECT to the database.**

- \_\_\_ 18. Recall that **dba**, previously a DBADM, still retains connect privilege. Also, recall that this user was not able to insert into the ALBUMS table after the DBADM authority was revoked. Attempt an insert now that the privilege has been granted to a group of which **dba** is a member. Be sure to issue the SQL statement from the **dba** session.

```
db2 "insert into student.albums values('DB2 Connect is My Life', 99, 310)"
```

You should see that the statement executes successfully.

- \_\_\_ 19. The **dba** ID also retained the privilege to BINDADD. With this in mind, assume that **dba** attempts to bind a program that contains the insert statement previously executed as a **static** statement.

Would such a bind attempt be successful? Why or why not?

---



---



---

**The attempt to bind a static statement would not be successful. Group privileges are NOT considered when validating the authorization requirements of static SQL statements, except for the group PUBLIC.**

- \_\_\_ 20. From the DB2 Control Center, determine the current privileges granted against the ALBUMS table.

From the *DB2 Control Center*, select **Tables** under the *MUSICDB* database in the *Object* tree.

- \_\_\_ Right-click **Tables** and select **Refresh**.
- \_\_\_ Right-click and select **ALBUMS** table and then select **Privileges** from the pop-up menu.
- \_\_\_ Note the Privileges **STUDENT** has.
- \_\_\_ Select **Group** tab.
- \_\_\_ Note the Privileges **PUBLIC** has.
- \_\_\_ Note the Privileges **STAFF** has.
- \_\_\_ Click **Cancel** button.

- \_\_\_ 21. Which group ID has the privilege to select from the table? Who belongs to this group?

---

---

---

**The group PUBLIC has SELECT privilege against the ALBUMS table. Any user that can CONNECT to the database will have the privileges associated with the group PUBLIC for purposes of group authorization.**

- \_\_\_ 22. Since you have already determined that **dba** can still connect to the database, determine if the **PUBLIC** group is functional by issuing the following SQL statement from the **dba** session.

```
db2 "select * from student.albums"
```

- \_\_\_ 23. Can you state that the privilege is only present because of the grant to the **PUBLIC** group?

---

---

---

**Based on the results you saw when looking at the privileges on the ALBUMS table, the only grant that could permit the **dba** ID to select from the ALBUMS table is the grant to **PUBLIC**.**

- \_\_\_ 24. Close the **dba** session by entering:

```
db2 terminate
exit
```

- \_\_\_ 25. Close the **maint** session by entering:

```
db2 terminate
exit
```

- \_\_\_ 26. Close the **adm** session by entering:

```
db2 terminate
exit
```

- \_\_\_ 27. Close the **ctrl** session by entering:

```
db2 terminate
exit
```

- \_\_\_ 28. From the DB2 Control Center, revoke the insert privilege from the staff group.

From the *DB2 Control Center*, select the **+** sign to the left of User and Group Objects under the *MUSICDB* database in the *Object* tree.

- \_\_\_ Select **DB Groups** under *User and Group Objects*. Right-click **DB Groups** and select **Refresh**.
  - \_\_\_ Right-click and select **Staff Group** and then select **Change** from the pop-up menu.
  - \_\_\_ Select **Table** tab.
  - \_\_\_ Click the **Revoke All** button.
  - \_\_\_ Click the **OK** button.
- \_\_\_ 29. Delete the row inserted by **dba**.

Enter the SQL statements:

```
db2 connect to musicdb user student using ibm2blue;
db2 delete from albums where itemno = 310
```

## Section 4 - Resetting the Environment

This section only needs to be performed if this is not the last lab you are working on.

- \_\_\_ 1. Grant the following privileges to PUBLIC.

Enter the SQL statement:

```
db2 grant connect, createtab, bindadd on database to public
```

***END OF LAB***

## Appendix A. UNIX vi Editor command summary

<b>Edit a new file</b>	vi pgmname.sqc
<b>Save Program and Return to System</b>	:wq -or- ZZ
<b>Save Program and Stay in Edit</b>	:w
<b>Cancel Changes &amp; Return to System Return to System</b>	:q!
<b>Enter insert mode</b>	i
<b>Return to command mode</b>	ESC
<b>Add a blank line BEFORE current line</b>	O
<b>Add a blank line AFTER current line</b>	o
<b>Undo last command</b>	u
<b>Repeat last command</b>	.
<b>Turn on line numbers</b>	:set nu
<b>Move display UP one screen</b>	Ctrl-b
<b>Move display DOWN one screen</b>	Ctrl-f
<b>Move display UP 'n' lines</b>	##Ctrl-u
<b>Move display DOWN 'n' lines</b>	##Ctrl-d
<b>Move display LEFT</b>	h -or- left arrow
<b>Move display RIGHT</b>	l -or- right arrow
<b>Move cursor up one line</b>	k -or- up arrow
<b>Move cursor down one line</b>	j -or- down arrow
<b>Move cursor to end of line</b>	\$
<b>Move cursor to end of line and enter insert mode</b>	A
<b>Move cursor to start of line</b>	^
<b>Move to TOP of program</b>	H
<b>Move to TOP of program</b>	1G
<b>Move to BOTTOM of program</b>	G
<b>Split line at cursor</b>	Position cursor under character in insert mode and press enter
<b>Join lines</b>	Place cursor at end of first line, then press J
<b>Replace/Overtyping to end of line</b>	R
<b>Delete word</b>	dw
<b>Delete line</b>	dd
<b>Delete from cursor to end of line</b>	D

<b>Copy line(s)</b>	:startline#,endline# co taretline#
<b>Move line(s)</b>	:startline#,endline# m taretline#
<b>Copy lines from another file and insert after cursor</b>	:r filename
<b>Make a copy of the file you are editing</b>	:w filename
<b>Search DOWN for next occurrence of text</b>	/text
<b>Search UP for next occurrence of text</b>	?text
<b>Go to a particular line</b>	:line# G
<b>Delete character at cursor</b>	x
<b>Copy line to buffer</b>	Y
<b>Pull line from buffer</b>	P
<b>Change word</b>	cw



# Appendix B. SAMPLE database

## SAMPLE database

### SAMPLE: Data Model and Objects (1 of 5)

#### DEPARTMENT

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
Pk		Inx	Fk	Fk
USERSPACE1				

#### EMPLOYEE

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO
Pk				Inx	Fk
USERSPACE1					

HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY
USERSPACE1					

BONUS	COMM
USERSPACE1	

#### EMP\_PHOTO

EMPNO	PHOTO_FORMAT	PICTURE	EMP_ROWID
Pk	Fk	Pk	
USERSPACE1			

#### EMP\_RESUME

EMPNO	RESUME_FORMAT	RESUME	EMP_ROWID
Pk	Fk	Pk	
USERSPACE1			

## SAMPLE: Data Model and Objects (2 of 5)

To: Department(deptno)

To: Employee(empno)

### PROJECT

PROJNO	PROJNAME	DEPTNO	RESPEMP	PRSTAFF
Pk		Fk	Inx	Fk
USERSPACE1				

PRSTDATE	PRENDATE	MAJPROJ
		Fk
USERSPACE1		

### PROJACT

PROJNO	ACTNO	ACSTAFF	ACSTDATE	ACENDATE
Pk	Fk	Pk		Pk
USERSPACE1				

### EMPPROJACT

EMPNO	PROJNO	ACTNO	EMPTIME	EMSTDATE	EMENDATE
	Fk	Fk		Fk	
USERSPACE1					

## SAMPLE: Data Model and Objects (3 of 5)

*CUSTOMER*

CID	INFO	HISTORY
Pk	Inx	
IBMDB2SAMPLEXML		

*ACT*

ACTNO		ACTKWD	ACTDESC
Inx	Pk	Fk	Inx
USERSPACE1			

*PURCHASEORDER*

POID	STATUS	CUSTID	ORDERDATE	PORDER	COMMENTS
Pk		Fk		Inx	
IBMDB2SAMPLEXML					

*IN\_TRAY*

RECEIVED	SOURCE	SUBJECT	NOTE_TEXT
USERSPACE1			

*ORG*

DEPTNUMB	DEPTNAME	MANAGER	DIVISION	LOCATION
USERSPACE1				

*CL\_SCHED*

CLASS_CODE	DAY	STARTING	ENDING
USERSPACE1			

## SAMPLE: Data Model and Objects (4 of 5)

### STAFF

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
USERSPACE1						

### SALES

SALES_DATE	SALES_PERSON	REGION	SALES
USERSPACE1			

### EMPMDC

EMPNO	DEPT	DIV
IBMDB2SAMPLEREL		

### SUMMARY TABLE ADEFUSR

workdept	no_of_employees
FROM employee	

## SAMPLE: Data Model and Objects (5 of 5)

### PRODUCT

PID	NAME	PRICE	PROMOPRICE
Pk			
IBMDB2SAMPLEXML			

PROMOSTART	PROMOEND	DESCRIPTION
		Inx
IBMDB2SAMPLEXML		

### SUPPLIERS

SID	ADDR
Pk	
IBMDB2SAMPLEXML	

### INVENTORY

PID	QUANTITY	LOCATION
Pk		
IBMDB2SAMPLEXML		

### CATALOG

NAME	CATALOG
Pk	
IBMDB2SAMPLEXML	

### PRODUCTSUPPLIER

PID	SID
Pk	Pk
IBMDB2SAMPLEXML	

## **SAMPLE: Data Model and Objects - Aliases**

DEPT FOR DEPARTMENT  
EMP FOR EMPLOYEE  
EMP\_ACT FOR EMPPROJECT  
EMPACT FOR EMPPROJECT  
PROJ FOR PROJECT



