DBA Certification Course

(Summer 2008)

Chapter 1: DB2 Products and Tools

- DB2 History
- DB2 UDB Editions
- Accessing DB2, DB2 Tools
- Instances, Databases, Schemas, Files
- OLTP vs. OLAP (Data Warehouses)

Objectives

After completing this chapter, you should be able to:

- name different editions of DB2
- select a DB2 edition based on given requirements
- connect with a DB2 server
- name some further tools and their purpose
- explain differences between OLTP and OLAP applications (characteristics of data warehouses)

Literature (1)

- Raul Zikopoulos: Which distributed edition of DB2 9.5 is right for you?

  [http://www.ibm.com/developerworks/db2/library/techarticle/dm-0611zikopoulos/]
- Raul Zikopoulos: DB2 9 Fundamentals exam 730 prep, Part 1: DB2 planning

  [http://www.ibm.com/developerworks/edu/dm-dw-db2-cert7301.html]
- IBM - DB2 for Linux UNIX and Windows - Product Overview

  [http://www-306.ibm.com/software/data/db2/9/]

Literature (2)

- Clara Liu, Raul Chong, Dwaine Snow, Sylvia Qi: Understanding DB2: Learning Visually with Examples

  IBM Press/Pearson, 2005, ISBN 0-13-185916-1, 895 pages.
- For further literature, see Chapter 0 (Syllabus).

Overview

1. DB2 History
2. DB2 Editions
3. DB2 Tools
4. Instances, Databases, Directory Structure
5. OLTP vs. OLAP (Data Warehouses)

DB2 History (1)

- 1970: Codd proposes the relational model
- 1973–76: SEQUEL (later SQL)
- 1976: System/R [Research Prototype]

  The project was started in 1973 in the IBM Reserach Lab in San Jose (California). Also Ingres appeared in 1976. Oracle was the first commercial system, it appeared in 1979.
- 1981: SQL/DS (Data System) [commercial]
- 1983: DB2 for MVS, Version 1

  The "2" indicates that relational databases were considered the successor of the older hierarchical DBMS (like IMS: Information Management System).

DB2 History (2)

- 1987: Database Manager (DBM) in OS/2 Extended Edition v1.0.

  UNIX versions were developed later by porting the OS/2 version.
- 1996: DB2 UDB (Universal Database) V5

  "Universal Database": (1) besides traditional table data also new multimedia data (audio, video, text) can be stored ("any kind of data"), (2) it supports a wide range of platforms (OS/2, Windows, AIX, HP-UX, Solaris). (3) it supports arbitrary applications (including Web).
- 2002: DB2 UDB v8.1
- 2004: DB2 UDB v8.2 for Linux/UNIX/Windows
- 2006: DB2 9 ("Viper")

Overview

1. DB2 History
2. DB2 Editions
3. DB2 Tools
4. Instances, Databases, Directory Structure
5. OLTP vs. OLAP (Data Warehouses)

DB2 Editions: Overview

- DB2 has a common code base on Windows, Linux, UNIX (about 90% of the code are in common, 10% OS-specific).
- DB2 Everyplace Edition (for mobile devices) and Apache Derby (Cloudscape) have different code bases, but the SQL API is fully compatible.
- DB2 for mainframes (z/OS, i5/OS) has different code bases, and the SQL is 95% compatible.
- There is also a common Web services framework, and a common client (includes even Informix).

DB2 Everyplace

- For small mobile devices (occasionally connected)
- Only 200K / 350K (9.5) large
- “Black box database”: 100% self managing
- APIs e.g.: ODBC, JDBC, .NET, DB2 CLI.
- table-level and wire-level encryption (includes SSL)
- No triggers or stored procedures
- Can synchronize data with central DBMS

  Comes in two Editions: DB2 Everyplace Database edition has no synchronization server, the DB2 Everyplace Enterprise Edition has it. The synchronization server also works with non-IBM databases.

Apache Derby

- Formerly: IBM Cloudscape

  At Linux World 2004, IBM donated the IBM cloudscape code to the Apache Software Foundation.
- Open source, written in Java.
- 2MB .JAR File, can be used in own applications at no cost (even for profit).
- The SQL API is fully compatible with DB2 for Linux, UNIX, and Windows (→ easy transition).
- Has triggers and stored procedures.
- Can act as a synchronization client.

DB2 Personal Edition

- For Windows and Linux on Intel and AMD.

  Licensed by installation (each PC needs one license): 489 $.
- Does not accept client requests from other computers (i.e. does not act as a server on the net).

  Remote administration is possible: Administration client can connect.
- Better than Everyplace/Derby if 100% compatibility with distributed DB2 data servers is required.
- SQL replication can be used to synchronize data with remote servers.
- Includes e.g. “pureXML”.

DB2 Express-C

- "Free to develop, deploy, distribute"
- Windows, Linux (32 Bit and 64 Bit)

  APIs for Java, .NET, PHP, Ruby on Rails
- Uses at most two CPU cores and 2GB of RAM
- Unwarrented, no support, no FixPaks/Packs (patches), only current version (updated from time to time). There is a forum (community).
- Clusting servers for high availablity and using replication services is not allowed.
- Includes pureXML, no possibility to buy add-ons.

DB2 Express-C FTL

- FTL: Fixed-Term License: $3080 per year.
- Now one gets support, maintenance, regular updates at the normal DB2 release schedule.
- Uses up to four CPU cores, 4 GB RAM.
- One can cluster data servers for high availability.
- One can use SQL replication.
- Cannot buy feature packs (add-ons).

  But support includes free extenders (DB2 Spatial Extender, DB2 Net Search Extender). Without the FTL, it is not guaranteed to work.

DB2 Express (1)

- "Simple, secure, and inexpensive"

  Entry level, fully supported data server, for small and medium businesses (SMBs) and ISVs. DB2 Express is an edition of DB2, while Express-C is formally only a version of DB2.
- Only for machines with up to 4 CPU cores, uses at most 4GB. Only Windows, Linux, Solaris x86.

  In DB2 9, it could not be installed on a machine with more than 4GB. In version 9.5, it uses at most 4 GB even on larger machines.
- pureXML costs extra. Old XML extender included.

  E.g. XQuery and native XML support is only available if one buys the pureXML feature pack. In contrast, Express-C contains pureXML.

DB2 Express (2)

- High availability feature available (costs extra).

  Online table reorganisation. HADR ("High Availability Disaster Recovery") provides redundancy across the entire stack, MTTR ("Mean-Time To Repair") is measured in seconds. Two node cluster license for Tivoli System Automation (TSA) high availability services (includes support for automated failover).
- Autonomic managebility features (included).

  SMART intiative (self-managing and resource tuning). Self Tuning Memory Manager, Configuration Advisor, Design Advisor (advisors are included in every edition of DB2 9, also Express-C), autonomic maintenance plans, automated backup tuning and throttling, ...
- Homogeneous federation feature pack (costs extra)

  Allows to easily write joins between tables on different DB2 servers.

DB2 Express (3)

- Licensing/Prices:
  - ◇ Authorized User: 175 $ (min. 5 = 875 $).

    An authorized user is an identifiable person, not a DB user, and not the maximal number of users that are connected at the same time. Authorized users must be counted before multiplexing/connection concentration software. If a person has access to two DB servers, he/she needs two licenses. If there are three work shifts a day, each of five people, 15 licenses are needed (but long term movements of licenses between people are possible). Pure XML Feature and High Availablity Feature: each 93 $ per user.
  - ◇ Processor (e.g. Intel Xeon, 2 Cores): 5175 $

    The price per Processor Value Unit (PVU) is 51.75 $. A normal CPU has 100 PVUs. Intel Xeon and AMD Opteron count 50 PVUs per Core. DB2 Express can be installed only on machines with at most 200 PVUs. High Availablity Feature: 26.50 $ per PVU.

DB2 Workgroup Edition (1)

- "Secure, powerful, cost effective"

  Very similar to Express Edition, but for larger computers.
- Windows, Linux, UNIX (AIX, Solaris, HP-UX)
- Up to 4 CPUs (400 PVUs), uses up to 16 GB RAM.
- High Availability feature is included.

  As of Version 9.5. For Version 9, it had to be paid extra.
- 12.000 $ for Dual Core Intel Xeon / AMD Opteron

  Price per PVU: 120 $. A SUN UltraSPARC IV: 100 PVUs per Core. License per user: 421 $ (min. 5). (Prices might be before addition of High Availability Feature.) Earlier: Workgroup Server Edition licensed by user, Workgroup Server Unlimited Edition (WSUE) by CPU.

DB2 Workgroup Edition (2)

- Add-ons (at extra cost): pureXML, Query Optimization, Homogeneous Federation Feature.

  The query optimization feature might be important for Business Intelligence applications (OLAP): Materialized Query Tables (MQT), multi-dimensional clustering (MDC) tables, query parallelism. Note that one cannot buy this feature for the Express Edition.

  The pureXML Feature and the Performance Optimization Feature (old name of Query Optimization Feature) and the High Availability Feature (now included) cost 185 $/user (probably 53 $ per PVU).
- The idea is that one can bring it near to the Enterprise Edition by buying additional features, but one has to pay only for features that one really needs.

  But the scalability is restricted (CPUs, RAM).

DB2 Enterprise Edition (1)

- "Unmatched scalability, resiliency, and security"
- Windows, Linux, UNIX — no CPU/RAM limits

  Of course, one has to pay for additional CPUs.
- Includes HADR (High Availability Disaster Recovery), TSA (Tivoli System Automation), table partitioning, parallelism, MDC (multi-dimensional clustering), MQTs (materialized query tables).

  In the workgroup edition one has to pay extra for the query optimization feature (includes parallelism, MDC, MQT), one cannot get table partitioning there.

DB2 Enterprise Edition (2)

- 38.600 $ for Dual Core Intel Xeon / AMD Opteron

  The authorized user license costs 995 $ per user (min. 5).

- Add-ons: pureXML, Homogeneous Federation, Advanced Access Control, Geodetic Data Management, Performance Optimization, Storage Opt.

  (1) Advanced Access Control: Tables/rows can be assigned labels, label of user must match. (2) Geodetic Data Management: Respects curvature of the earth (in contrast to free Spatial Extender). (3) Performance Optimization: Contains Workload manager (WLM, replaces older Query Patroller / DB2 Governor, permits e.g. to control CPU usage), DB2 Performance Expert (reports to monitor DB activity). (4) Storage Optimization: Row level compression, Backup compr.

DB2 Connect

- Provides connectivity to mainframe/midrange databases from Windows/Linux/UNIX platforms.

  The editions for LUW (Linux, UNIX, Windows) are called the distributed editions of DB2. The Enterprise Edtion of DB2 contains a few complementary DB2 Connect licenses.

  Looks like a client to the DB2 server on the mainframe system (z/OS, i5/OS), and like a server to the client on Windows, Linux, UNIX. The communication standard with the mainframe is DRDA (Distributed Relational Database Architecture).

  There are also several versions of DB2 Connect, e.g. Personal Version (one client), Enterprise Version (several clients).

- In the opposite direction (from host/mainframe) to workstation, DB2 Connect is not required.

DB2 Clients

- Runtime Client

  Supports applications that need to communicate with a DB2 server.

- Java Common Client (JCC)

  For communication of JDBC/SQLJ applications with a DB2 server (for mainframes, DB2 Connect is still needed). Fully redistributable, 2MB. Part of the application, needs no explicit client installation.

- DB2 9 Client Lite

  As JCC, but for CLI/ODBC applications.

- DB2 9 Client

  Includes Runtime Client, tools for remote administration (formerly known as Administration client), and tools for application development (formerly known as application development client).

DB2 Add-on Tools

- DB2 Change Management Expert
- Data Archive Expert
- DB2 High Performance Unload
- DB2 Performance Expert
- DB2 Recovery Expert
- DB2 Table Editor
- DB2 Test Database Generator
- DB2 Web Query Tool

Overview

1. DB2 History
2. DB2 Editions
3. DB2 Tools
4. Instances, Databases, Directory Structure
5. OLTP vs. OLAP (Data Warehouses)

How to Access DB2

- DB2 Enterprise Edition (Linux) is installed on `bach.informatik.uni-halle.de`
- With `ssh` from any computer, one can at least use the text-based tools.
- For using the graphical tools, one needs an X server (e.g. on Linux, Solaris).

  The command “`ssh -X bach.informatik.uni-halle.de`” automatically redirects the X display. If that should not work (insecure X communication!): (1) For computers with several displays (e.g. `haydn`), find out your one: `echo $DISPLAY` (e.g. “`:4.0`”). (2) Use `xhost` ⟨`user`⟩`@bach` to allow requests from `bach`. (3) Use `ssh` for remote login to `bach`. (4) Set the display on `bach` with `export DISPLAY=haydn:4.0`.

Command Line Processor (1)

- On UNIX and Linux, there are two ways to use a text-based interface:
  - ◇ In a normal shell, prefix each command with “`db2`”, e.g.

```
brass@bach:~> db2 connect to SAMPLE
   Database Connection Information

   Database server        = DB2/LINUX 9.1.0
   SQL authorization ID   = BRASS
   Local database alias   = SAMPLE

brass@bach:~> db2 "select * from X"
```

  - ◇ Using interactive mode (see next slide).

Command Line Processor (2)

- Interactive mode is started with the command `db2` without parameters (except possibly options):

```
brass@bach:~> db2
   Command Line Processor for DB2 ADCL 9.1.0
   ...
db2 => connect to sample
   ...
db2 => select * from X
   ...
db2 => quit
```

  `quit` ends interactive mode, but does not stop the connection (one can pose further queries in non-interactive mode). `terminate` (see next slide) stops the connection (and also ends interactive mode).

Command Line Processor (3)

- There is a background/back-end process (`db2bp`) that manages the open database connection.
- To terminate the background process, use

  `db2 terminate`
- One can also disconnect from the database with

  `db2 disconnect sample`

  This only works when there is no active unit of work (transaction). Connections can be also stopped with `connect reset` (works even when there is an active transaction). But only `terminate` is a guaranteed way to stop the background process (important for server parameter changes). Closing a window/session might leave the back-end process alive.

Command Line Processor (4)

- Outside interactive mode, the shell might interpret characters like “>” (output redirection). Therefore, it is safer to enclose the SQL query in `"..."`.
- By default, the SQL query ends with the line end (using “;” as in Oracle gives a syntax error).
- One can use continuation lines by escaping the line end with “\”.
- One can choose to use a terminator character (by default “;”): `db2 -t`

  One can also define this character (e.g. `@`): `db2 -td@`

Command Line Processor (5)

- By default, commands are immediately commited (autocommit mode). One can stop this with “`+c`”:

  `db2 +c insert into X values (5)`

  But the next command without “`+c`” will commit everything before. The next command that fails will rollback everything before. Note that “`+`” turns off an option, “`-`” turns it on (`c` is on by default).
- To see the values of all options:

  `db2 list command options`
- To change auto-commit to off for this session:

  `db2 update command options using c off`

  To change the default (for all sessions), use `db2set db2options="+c"`.

Command Line Processor (6)

- In interactive mode, one can execute operating system commands by prefixing them with “!”.
- A list of all possible commands is printed with “?”.

  One can also get help for commands, e.g. “`? list applications`”, or for error messages, e.g. “`? SQL0104N`”.
- CLP also has a history function:
  - ◇ `h`: prints the last commands.
  - ◇ `r 3`: runs command 3 from the list (`r`: last).
  - ◇ `e 3`: calls an editor for command 3 from the list.

    `vi` on Linux/UNIX, Notepad on Windows. Can be selected with `db2set DB2_CLP_EDITOR=...`

Command Line Processor (7)

- Unter Windows, the built-in shell (MS-DOS Command Prompt) could not be used.
- Therefore, there is a "Command Window" that looks like an MS-DOS Prompt, but permits to execute DB2-commands by prefixing them with "`db2`".
- As usual, if one enters "`db2`" without parameters, one gets into interactive mode.
- Instead, one can also immediately start the Command Line Processor (CLP).

Command Line Processor (8)

- One can write a series of commands to a file, and execute that file (script, batch file).
- This can be a standard operating system shell script (with every command prefixed with "`db2`"), or it can be a CLP command script.

  A shell script can have parameters, a CLP command script not.
- A CLP script (e.g., in file "`xyz`") is executed with

  ```
  db2 -t -vf xyz
  ```

  `-t` turns the statement termination character "`;`" on, `-v` displays the executed commands, `-f` defines the input file. Comments in the CLP script are marked with "`--`" and extend to the end of the line.

DB2 System Commands

- DB2 CLP commands (and SQL statements) are entered into the Command Line Processor.
- Another category of commands are DB2 System Commands, which are independent programs to be executed outside the CLP, e.g.
  - ◇ `db2start/db2stop`: Start/Stop instance.
  - ◇ `db2ilist`: List all instances.
  - ◇ `db2sampl`: Create sample database.
  - ◇ `db2set`: Set DB2 Registry variable.

    Most commands have an option `-h` to display parameter syntax.

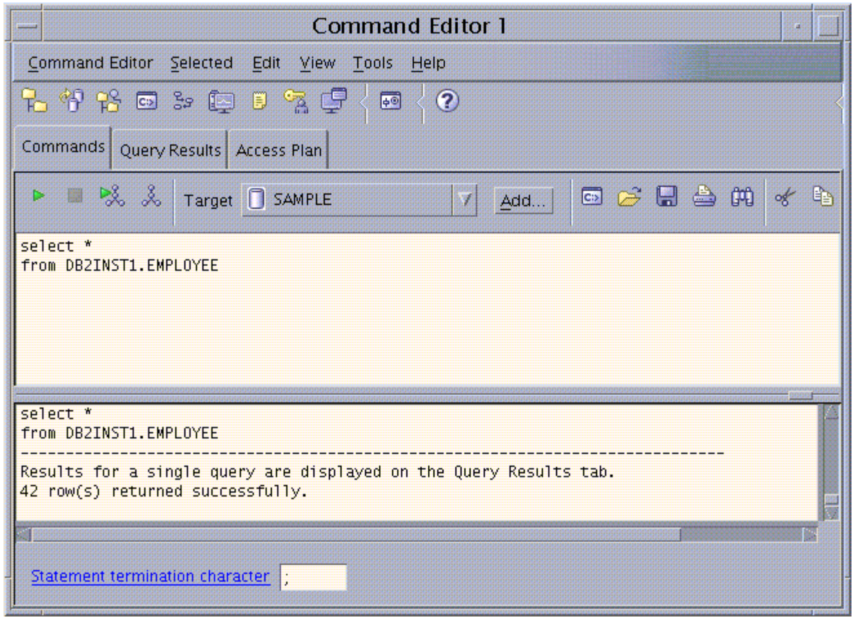
Command Editor

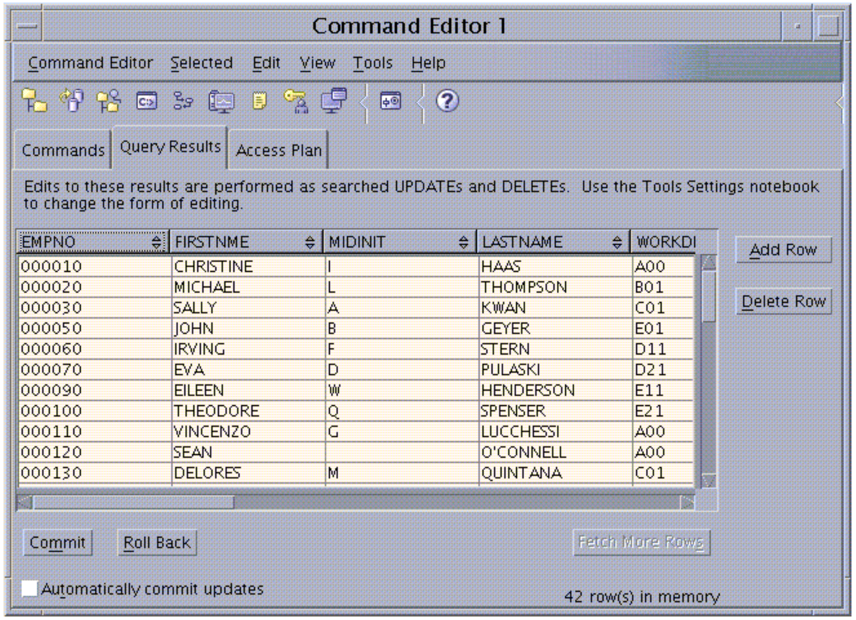
- A graphical tool for entering SQL statements is the Command Editor (start with `db2ce`). Three Tabs:
  - ◇ Commands

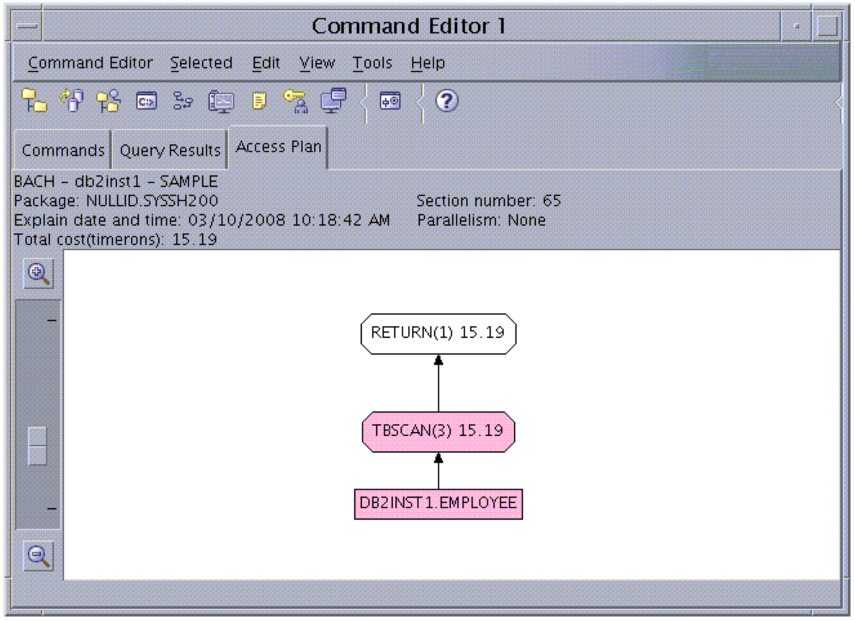
    In the upper part one can enter the command (also distributed over several lines), in the lower part error messages and other feedback (e.g. number of selected rows) are shown. Note that the feedback scrolls down, it is not deleted when the next command is entered. To execute a command, press the green arrow.
  - ◇ Query Results

    Tabular view of the query result. One can edit the displayed rows.
  - ◇ Access Plan

    Shows a tree view of the access plan (if one presses not the simple green arrow to execute the query, but the one with a tree).





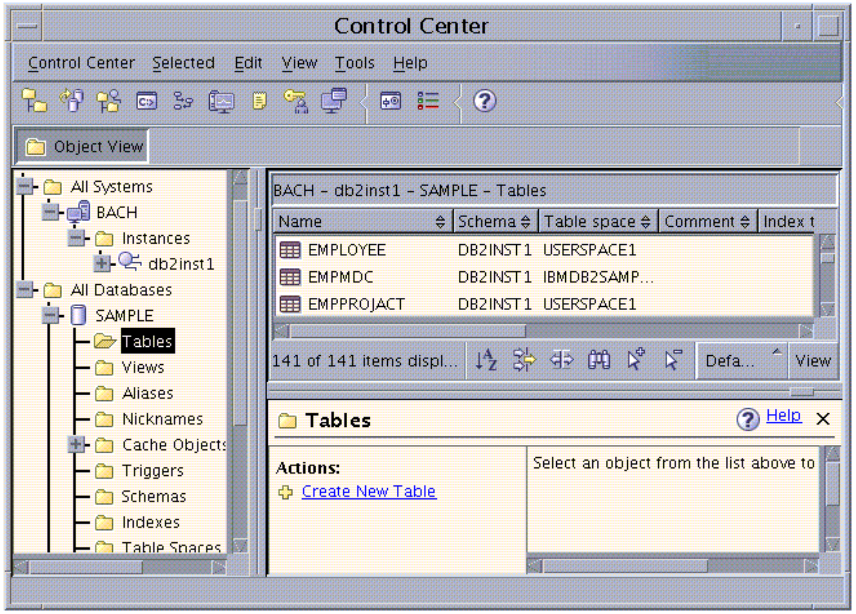


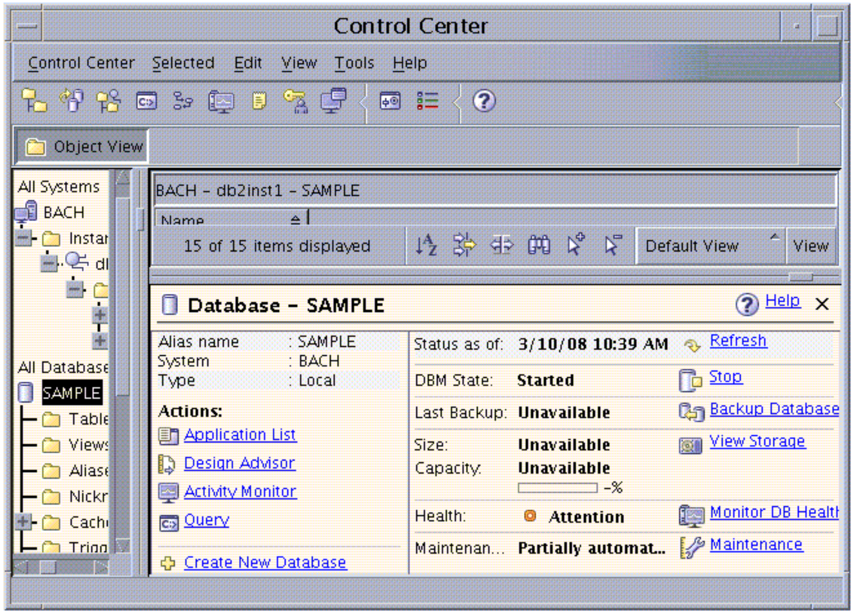
Control Center (1)

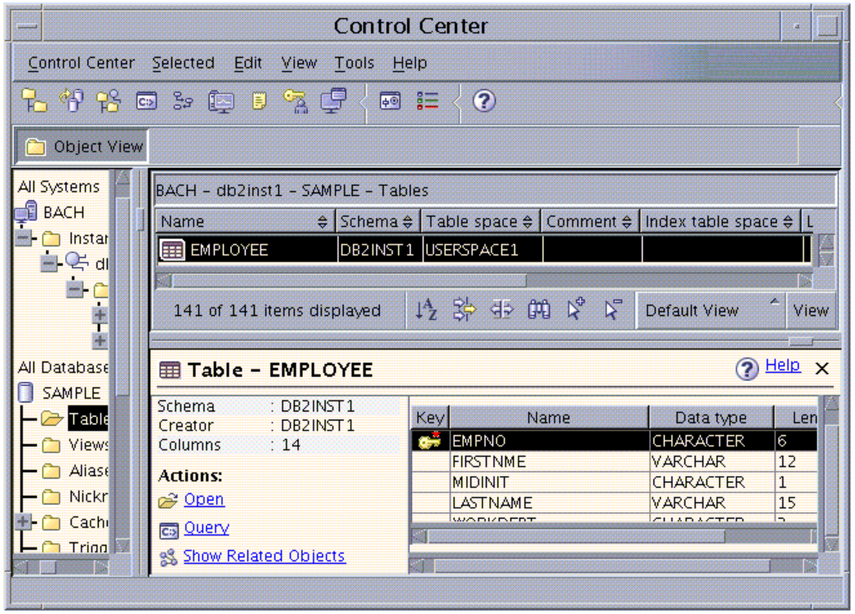
- The Control Center (`db2cc`) is the most important graphical tool for the database administrator.
- For instance, it permits to
  - ◇ create/alter/drop databases, table spaces, tables, views, indexes, triggers, and schemas.
  - ◇ load, import, export data.
  - ◇ gather statistics.
  - ◇ back up or restore databases or table spaces.
  - ◇ configure and tune databases and instances.
  - ◇ manage applications.

Control Center (2)

- On the left side, an object tree is displayed, with the usual possibilities to expand/collapse subtrees.
- The "Contents Pane" (upper part of the right side) shows a tabular view of the objects in the folder selected in the object tree.
- The "Object Details Pane" (lower part of the right side) shows information and possible actions for the object selected in the object tree or contents pane.
- If one right-clicks on an object, a large selection of possible actions is displayed in a pop-up menu.







Tool Functions (1)

- The Control Center (and other tools) contain
  - ◇ Wizards: Take the user step by step through a complex task, asking for certain information in each step.
  - ◇ Advisors: Like a wizard, but with more "intelligence" (more complex algorithms).

    E.g. Configuration Advisor (for selecting configuration parameters for an instance / a database), Design Advisor (for selecting indexes based on a given workload).
  - ◇ Notebooks: Like a wizard, but only a single page.

    So one does not need to remember the complex command syntax.

Tool Functions (2)

- "Show SQL / Show Command": Displays the command that the graphical user interface generates.
- "Generate DDL": This shows the SQL command that would be necessary for regenerating the object.

  E.g., if a table is selected, the "`CREATE TABLE`" command is shown (can also be used to clone tables). Of course, the contents of the table (data) is not regenerated. The GUI calls the program "`db2look`".
- "Show Related": Displays related objects.

  E.g. when a table is selected, one can see the views that access the table (indirect dependencies are not shown). One can also see indexes, aliases, triggers, tablespaces, user-defined functions (UDFs), and user-defined types (UDTs) for the table.

Tool Functions (3)

- For tabular displays of data, there is a menu button "View" in the lower right corner.
  - ◇ It permits to show the information differently sorted/grouped, or only a subset of the rows or columns (with a filter).
  - ◇ One can also define own filters.
  - ◇ There are also buttons to switch back to the default view, to search in the table, to sort it, to select all entries, and to deselect all entries.

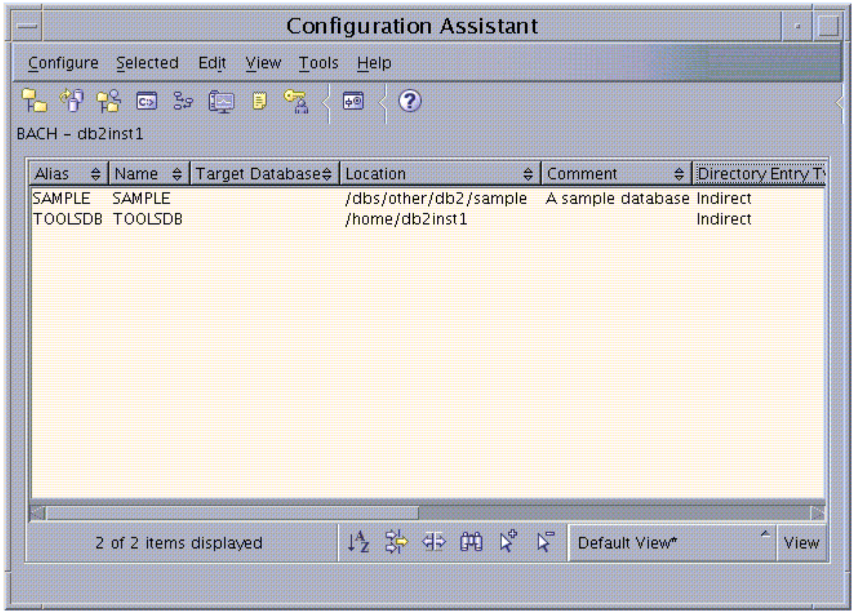
    The sort order can also be selected in the column heading.

Configuration Assistant

- The main purpose of the Configuration Assistant (`db2ca`) is the client-side configuration.

  However, one can also edit e.g., the instance (DBM) configuration parameters, and the DB2 Registry.
- The process to make a database known at a client is called "to catalog the database".

  There is also a discovery service to retrieve information about databases in the network.
- Configuration profiles permit to easily duplicate the configuration on other clients.



Configuration Parameters (1)

- There are a lot of configuration parameters that can be set at different places:
  - ◇ DBM Configuration (for an instance).

    An instance (Database Manager) can manage several databases.
  - ◇ DB Configuration (for a database).
  - ◇ The DB2 Registry (or “DB2 Profile Registry”).

    The DB2 Registry has nothing to do with the Windows Registry. Variables in the DB2 Registry can be set at different levels, e.g. global-level (for all instances on a server), instance-level.
  - ◇ Operating System environment variables.

    E.g. `DB2INSTANCE` selects the current instance.

Configuration Parameters (2)

- The DBM configuration can be displayed with

  `db2 get dbm cfg`
- E.g. to set the parameter `INTRA_PARALLEL` to `YES`:

  `db2 update dbm cfg using INTRA_PARALLEL YES`
- The DB configuration for database `SAMPLE` can be displayed with

  `db2 get db cfg for SAMPLE`
- Example for changing a DB config. parameter:

  `db2 update db cfg for SAMPLE using MINCOMMIT 3`

Configuration Parameters (3)

- To view the contents of the DB2 registry, use:

  `db2set -all`.
- All possible registry variables are shown with:

  `db2set -lr`.
- Example for setting a variable for instance `db2inst1`:

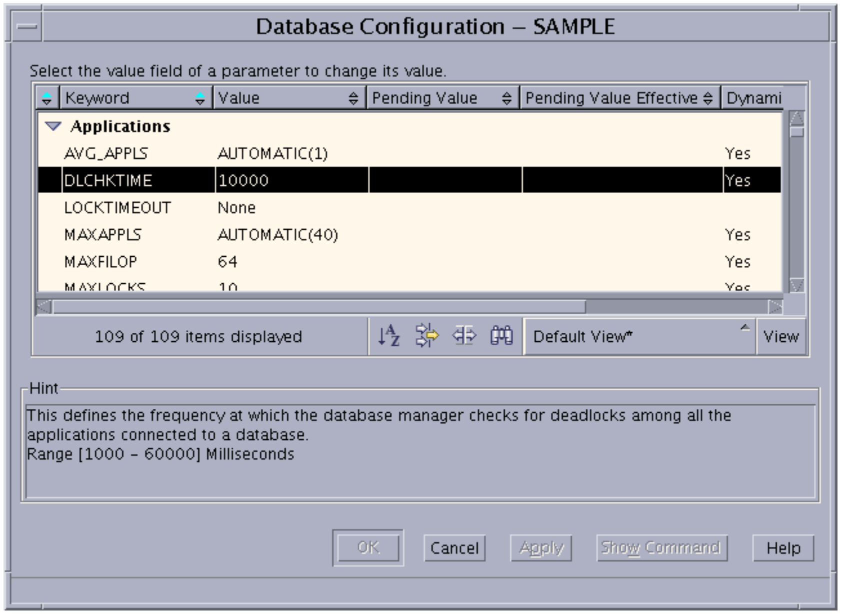
  `db2set DB2COMM=tcpip -i db2inst1`
- Example for setting a variable at global level:

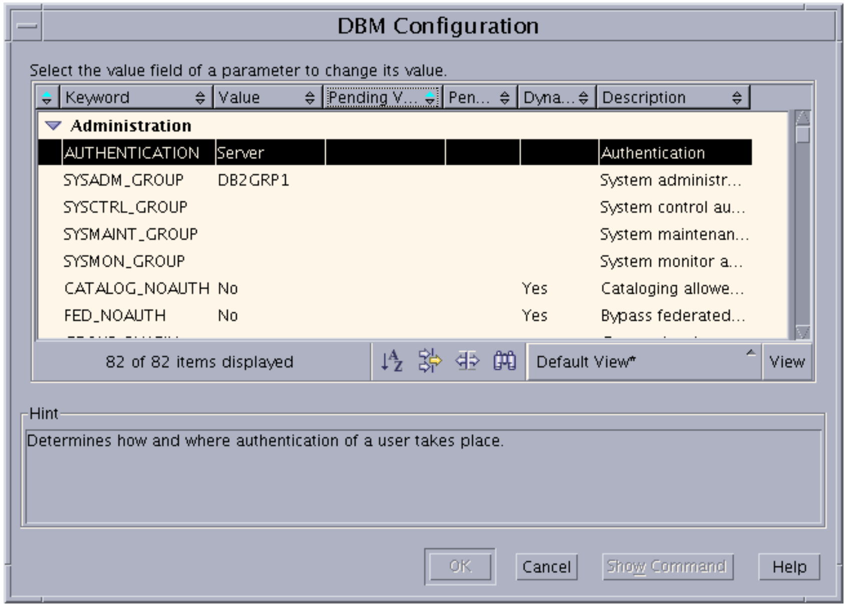
  `db2set DB2INSTPROF="C:\IBM DB2\SQLLIB" -g`

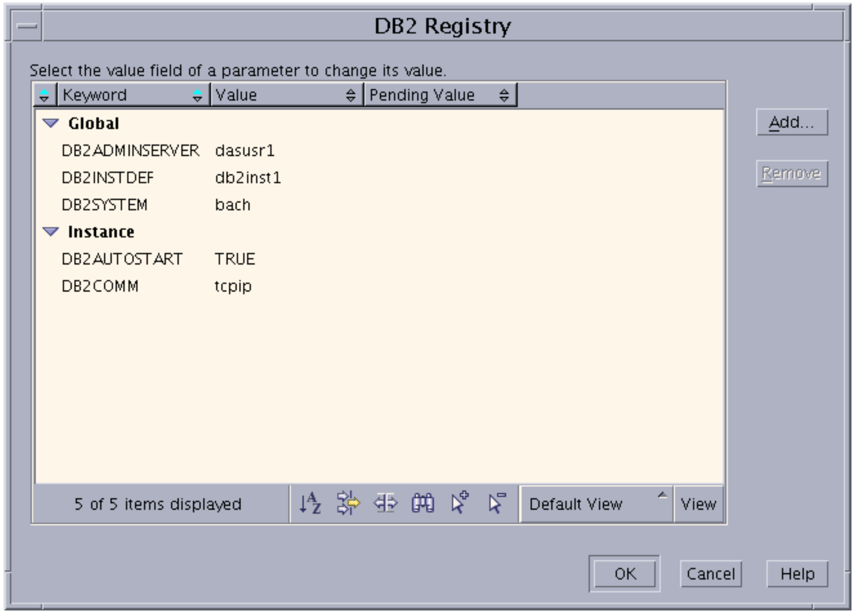
  If the same variable ist set at global and at instance level, the instance level takes precedence.

Configuration Parameters (4)

- Some parameters are dynamic, others are not. In that case, the change takes effect only if
  - ◇ the instance is stopped and started again (DBM).
  - ◇ all connections to the database are closed (DB).
- Of course, there is also a graphical interface for viewing/changing configuration parameters, available from
  - ◇ the Control Center (DBM and DB Config), and
  - ◇ the Configuration Assistant (DBM config and Registry).







Health Center

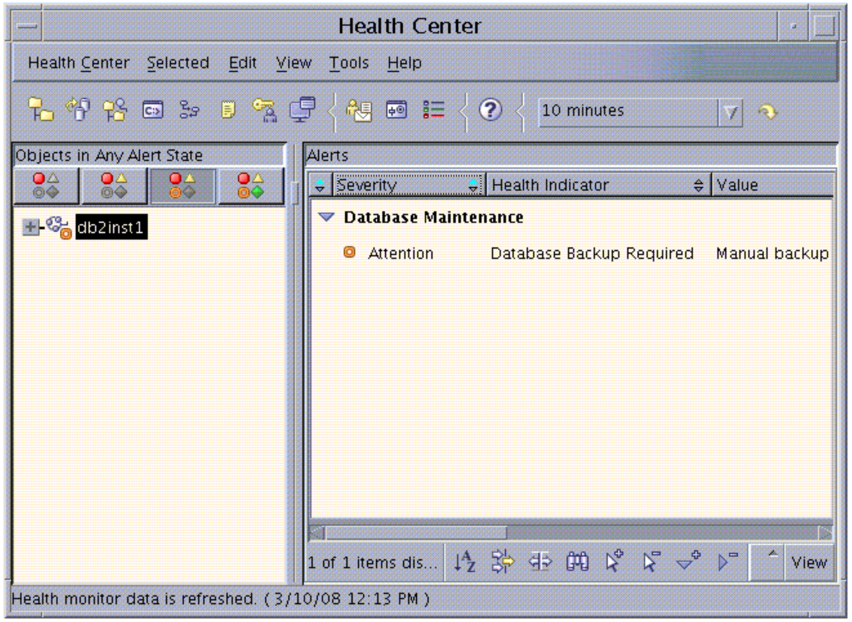
- The health center shows alerts for problems with the database and recommends solutions.

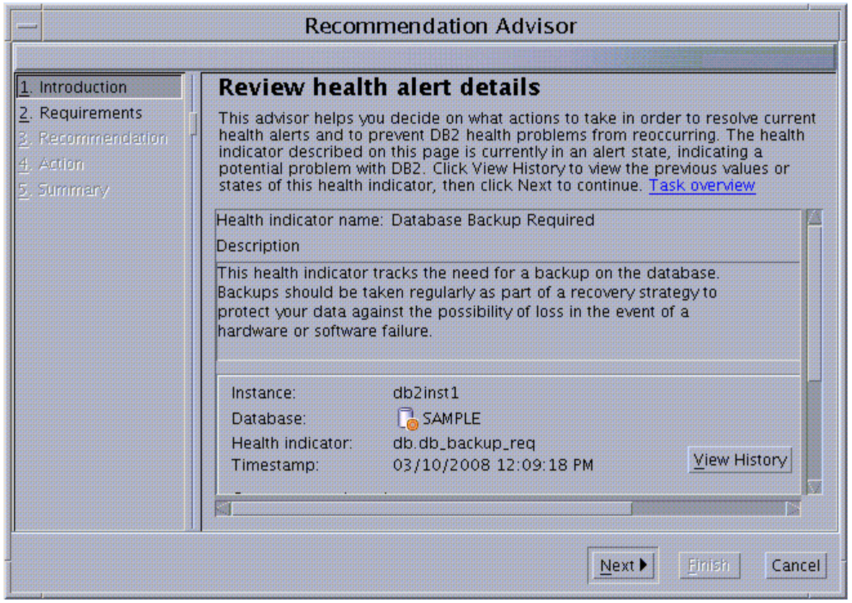
  By defining thresholds for problem indicators (e.g., free disk space), the problem might be detected before it actually occurs.

- The problems have different severity levels (Alarm, Warning, Attention, Normal).

  E.g. the freshly created database should be backed up. This is an alert of level “attention”. The recommended action is to take an offline backup, or to set up the database for automatic backups (in order to avoid this alert in future).

- One can define email and pager contacts to which a message is sent for alert of a certain severity level.





Tool Summary (1)

Windows Menu Structure:

- Command Line Tools:
  - ◇ Command Line Processor
  - ◇ Command Window
  - ◇ Command Editor
- Development Tools:
  - ◇ Development Center
  - ◇ Project Deployment Tool
- (continued on next two slides . . .)

Tool Summary (2)

- General Administration Tools:
  - ◇ Control Center
  - ◇ Journal
  - ◇ Replication Center
  - ◇ Task Center
- Information:
  - ◇ Check for DB2 Updates
  - ◇ Information Center

Tool Summary (3)

- Monitoring Tools:
  - ◇ Activity Monitor
  - ◇ Event Analyzer
  - ◇ Health Center
  - ◇ Indoubt Transaction Manager
  - ◇ Memory Visualizer
- Set-up Tools:
  - ◇ Configuration Assistant
  - ◇ First Steps
  - ◇ Register Visual Studio Add-Ins

Overview

Instances (1)

- "A DB2 instance [database manager instance] is a logical context in which DB2 commands and functions are executed."
- A DB2 instance is a set of processes and shared memory that make up a running DB2 system.
- An instance can manage several databases.

  Different instances can contain databases with the same name. E.g., one could have one instance for development, and one for production.
- Several distinct instances can run on the same server machine.

Instances (2)

- When DB2 is installed, by default an instance called "`db2inst1`" (Linux/UNIX) or "`DB2`" (Windows) is created.
- All existing instances are shown with the command `db2ilist`.
- An instance is selected with the environment variable `DB2INSTANCE`.
- One can see the current instance with

  `db2 get instance`

Instances (3)

- The selected instance is started with

  `db2start`

  To start an instance in "quiesced mode" for administration purposes (normal uses cannot connect), use "`db2start ADMIN MODE`". To start an instance on another server, use "`db2start REMOTE` ⟨instance name⟩".
- To terminate all connections and stop an instance:

  `dbstop force`
- Under UNIX, to ensure that an instance is automatically started after every system restart, use

  `db2iauto -on` ⟨instance name⟩

  To stop the autostart, use `-off`. Under Windows: Services panel.

Instances (4)

- To create an instance (needs root rights):

  `db2icrt -u` ⟨fenced user ID⟩ ⟨instance name⟩

  User defined functions and stored procedures can be executed in a different process under the "fenced user ID", if the administrator thinks that they are not completely reliable. The instance name must be the name of an existing UNIX user ID. On windows, the minimal syntax is `db2icrt` ⟨instance name⟩ and it is not required that there is corresponding Windows user.
- To delete an instance (needs root rights):

  `db2drop -f` ⟨instance name⟩
- To start a graphical tool for instance creation:

  `db2isetup`

Instances (5)

- Under UNIX, each instance corresponds to a UNIX user ID. In the home directory of this user, symbolic links are stored to the executable code of the DB2 programs (installed in a different directory).
- When a Fix Pack is installed (or one wants to change the DB2 installation directory), it might be necessary to update these links. This is done with

  `db2iupdt` ⟨instance name⟩

  This program must be executed in the installation directory. To migrate an instance from a previous release: `db2imigr` ⟨instance name⟩

Databases

- "A collection of interrelated or independent data items [e.g., tables] stored together to serve one or more applications."
- Databases are independent units. Unless one has the federation feature, one cannot pose a query to join tables from different databases.
- Each database has its own logs, bufferpools, table spaces (e.g., `SYSCATSPACE`, `TEMPSPACE1`, `USERSPACE1`).

  Logs contain information about changes to the databases needed for rollback/recovery. Bufferpools are main memory areas to cache disk blocks. Table space are storage containers for the data on the disk.

Schemas (1)

- Schemas are different name spaces within a database. Different schemas can contain tables with the same name.
- Usually, each user has a schema with the same name as the user ID, but there can be more schemas.

  There is a distinction between `OWNER` and `DEFINER`: Most schemas are owned by `SYSIBM`, but the `DEFINER` has the right to drop the schema (when it is empty).
- Several users can have the right to create, alter and drop tables within a schema.

  So although there is one official owner, several users can work with the schema as if it were their schema.

Schemas (2)

- Tables are globally identified with

  ⟨`Schema name`⟩`.`⟨`Table name`⟩
- If one has the necessary access rights, it is no problem to join tables in different schemas.

  Whereas it is not so easy to join tables in different databases (needs optional software, the federation feature).
- If one writes only ⟨`Table name`⟩, the current user name is added as name of the schema.

  With the `IMPLICIT_SCHEMA` privilege, a schema is automatically created (and the right to create tables in it is given to public), when a table in it is created. This also holds for the "personal" schema!

Data Dictionary (1)

- Tables (and views) for current user:

  `db2 list tables for user [show detail]`
- Tables for a given schema (or user):

  `db2 list tables for schema ⟨Name⟩ [show detail]`
- Tables for system schemas (e.g. `SYSCAT`, `SYSIBM`):

  `db2 list tables for system [show detail]`
- All Tables:

  `db2 list tables for all [show detail]`

Data Dictionary (2)

- Columns of a table:

  db2 describe table ⟨Name⟩ [show detail]

- Columns of a query result:

  db2 describe output ⟨Query⟩

- Indexes of a table (note that an index is automatically created for each key):

  db2 describe indexes for table ⟨Name⟩
  [show detail]

Data Dictionary (3)

- Besides these CLP commands, one can also use SQL to directly query the data dictionary, e.g.

```
SELECT TABSCHEMA, TABNAME
FROM   SYSCAT.TABLES
WHERE  OWNER = 'BRASS'
```

- The schema SYSCAT contains views that are the recommended way to access the data dictionary.

  The base tables are stored in the schema SYSIBM.

- E.g., to get information about schemas:

```
SELECT SCHEMANAME, OWNER, DEFINER
FROM   SYSCAT.SCHEMATA
```


Tablespaces

- A tablespace is something like a logical disk: For each table, one can define in which tablespace it should be stored.

  db2 list tablespaces [show detail]

- When a database is created, three tablespaces are automatically created:
  - ◇ SYSCATSPACE (catalog),
  - ◇ TEMPSPACE1 (system temporary space),
  - ◇ USERSPACE1 (default user tablespace).