Objectives

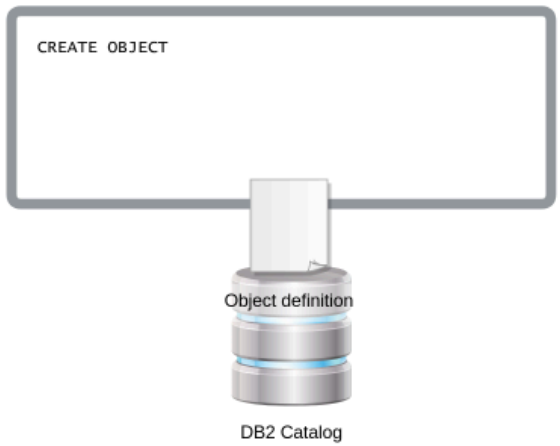
Viewing Database Objects - The Catalog

The Db2 Catalog contains information used by Db2 to manage objects within its subsystem.

In this module you will examine the structure of the Db2 Catalog and discover how you can display data stored within it.

After completing this module, you will be able to:

- Identify Tables and Views Associated With the Db2 Catalog
- Use SQL to Display Db2 Catalog Tables
- Use IBM Data Studio to Display Db2 Catalog Information



When an object is created in a Db2 system its definition is stored in the Db2 Catalog. This is a relational database maintained by Db2 itself.

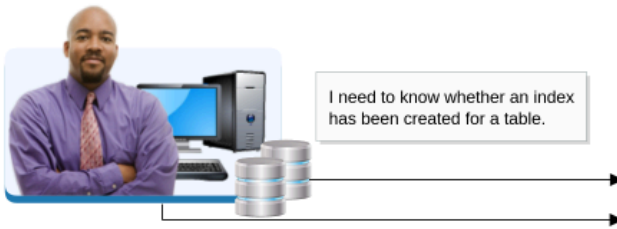
Information on every table and column, and all the other objects, is stored in the catalog. This information can be retrieved with a SELECT statement.

Click **Play** to see how the Db2 Catalog is used.

The DBA can use SQL statements to change the value of attributes for existing catalog indexes, sequences, and table spaces, and can add indexes to any of the catalog tables.

The Db2 Catalog consists of numerous tables created under the SYSIBM schema, or creator. Most of these tables have a read-only status with their content only updated automatically by Db2 when there is a change to a Db2 object.

While the general Db2 user will usually not need to bother too much with the Db2 catalog, the DBA will have access to SQL code, tools and utilities that enable them to gather statistics, and update some of the Db2 catalog tables, ensuring continued availability and performance.



I need to know whether an index has been created for a table.

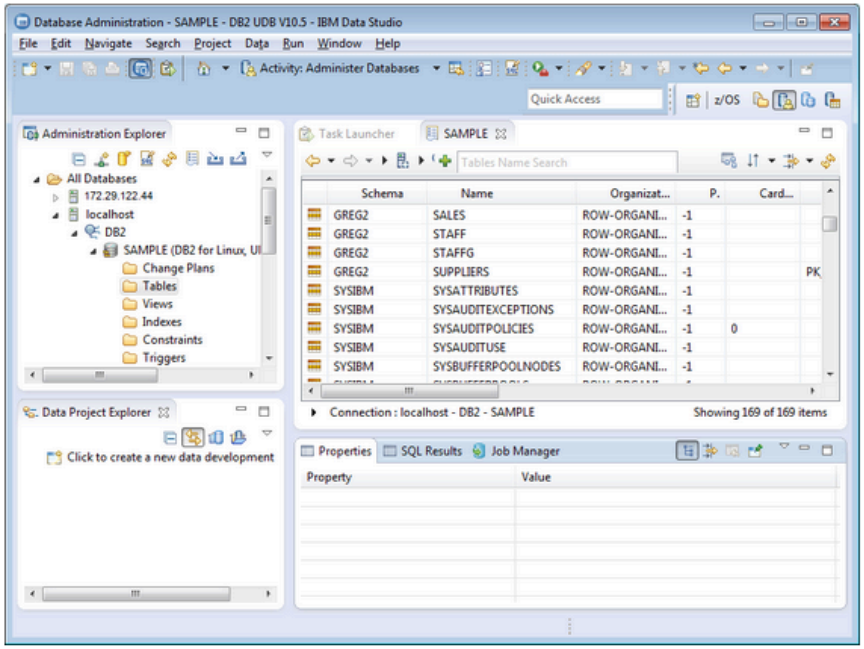
I need to find the minimum and maximum allowable values for a specific column in a table.

BUFFERPOOL	SYSIBM.SYSBUFFERPOOLS
STORAGE GROUP	SYSIBM.SYSSTOGROUP
TABLESPACE	SYSIBM.SYSTABLESPACE
DATABASE	SYSIBM.SYSDATABASE
TABLE	SYSIBM.SYSTABLES
INDEX	SYSIBM.SYSINDEXES
COLUMNS	SYSIBM.SYSCOLUMNS
CHECK CONSTRAINTS	SYSIBM.SYSCHECKS

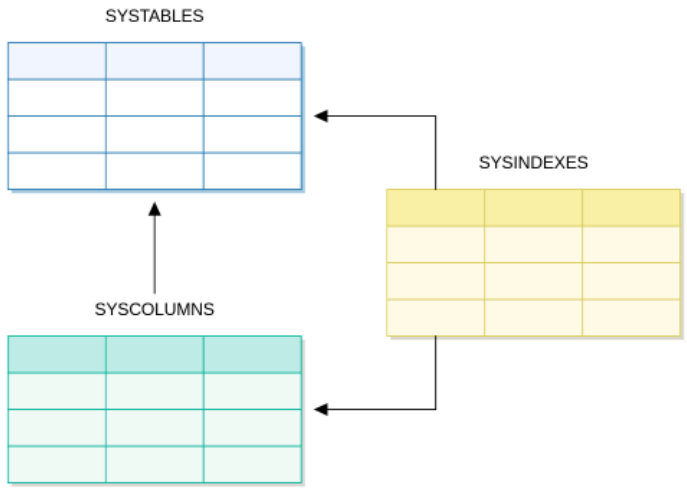
Details associated with the Db2 objects that we have created, altered or dropped in earlier modules will have been recorded in one of the Db2 Catalog tables mentioned here. As a DBA you may need to enquire on some of this information.

BUFFERPOOL	SYSIBM.SYSBUFFERPOOLS
STORAGE GROUP	SYSIBM.SYSSTOGROUP
TABLESPACE	SYSIBM.SYSTABLESPACE
DATABASE	SYSIBM.SYSDATABASE
TABLE	SYSIBM.SYSTABLES
INDEX	SYSIBM.SYSINDEXES
COLUMNS	SYSIBM.SYSCOLUMNS
CHECK CONSTRAINTS	SYSIBM.SYSCHECKS

The Db2 Catalog is obviously going to be slightly different depending on the version of Db2 you are running and platform it is running on. For example, data is stored differently on the mainframe and the location of the Db2 catalog in this environment is the DSNDB06 database.



The IBM Data Studio screen here shows a list of tables in the Sample Db2 Windows database, many of which are Db2 catalog tables, with schema SYSIBM.



Like any relational database, there are relationships between the catalog tables. The relationships allow you, Db2, or some third party tools to build a relational structure that can be spanned to find and view information about Db2.

For example, every column defined in SYSCOLUMNS refers to the table in SYSTABLES that contains it.

In addition to this, an index in SYSINDEXES refers to the table in SYSTABLES that the index is on and the columns in SYSCOLUMNS that are indexed.

```
SELECT * FROM SYSIBM.SYSTABLES
  WHERE CREATOR = 'SYSIBM'
---------+---------+---------+---------+---------
NAME
---------+---------+---------+---------+---------
DBDR
IPLIST
IPNAMES
LOCATIONS
LULIST
LUMODES
LUNAMES
MODESELECT
SCTR
SPTR
SYSAUDITPOLICIES
SYSAUTOALERTS
SYSAUTOALERTS_OUT
```

Simple SQL code can be used to display the content of the DB Catalog. In this example, only those tables that have the SYSIBM value in the CREATOR column are shown.

Customer

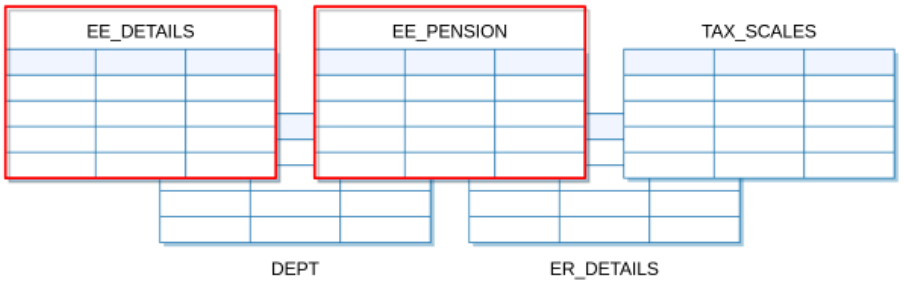
Name_ID	Official_Name	Last_Update

```
SELECT *
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME = 'CUSTOMER'
```

The two most useful tables for application programmers are:

- SYSIBM.SYSTABLES
- SYSIBM.SYSCOLUMNS

This example shows the SQL to return all the details of every column in every CUSTOMER table in the system.

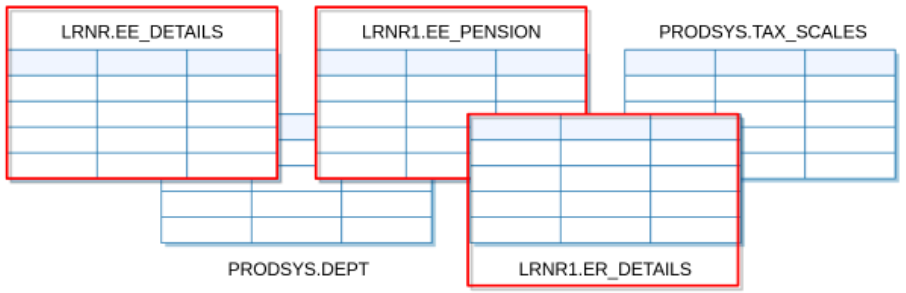



```
SELECT * FROM SYSIBM.SYSTABLES WHERE NAME LIKE 'EE%'
```

The following is another example:

```
SELECT * FROM SYSIBM.SYSTABLES WHERE NAME LIKE 'EE%'
```

This could be used to find the names of all tables that begin with EE.

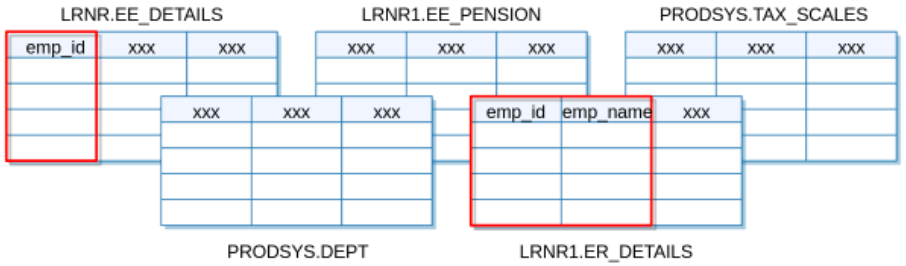



```
SELECT * FROM SYSIBM.SYSTABLES WHERE CREATOR = USER
```

Now look at this example:

```
SELECT * FROM SYSIBM.SYSTABLES WHERE CREATOR = USER
```

This would return the table definitions created by the person who runs the query. USER is a special variable that contains the userid of the person executing the query.



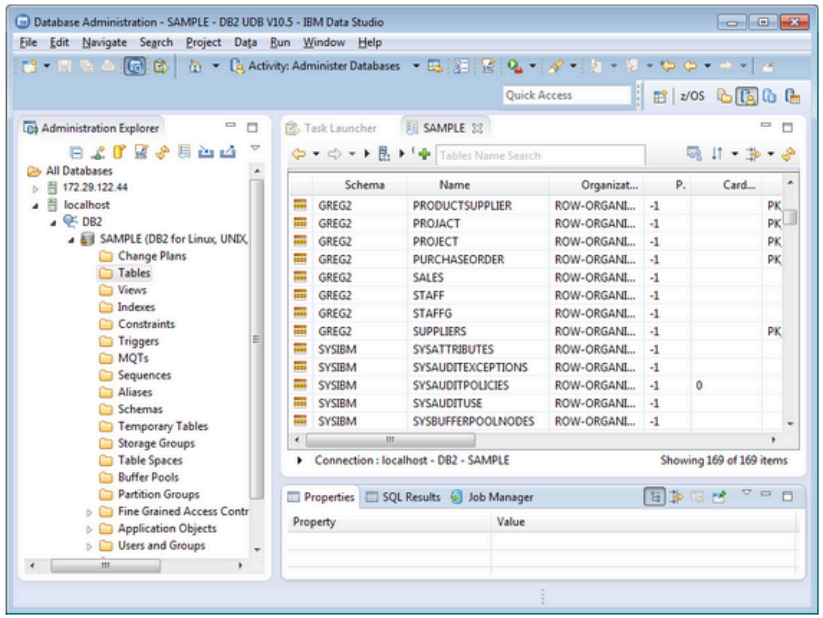

```
SELECT * FROM SYSIBM.SYSCOLUMNS WHERE NAME LIKE 'EMP%'
```

Another useful SQL is:

```
SELECT * FROM SYSIBM.SYSCOLUMNS WHERE NAME LIKE 'EMP%'
```

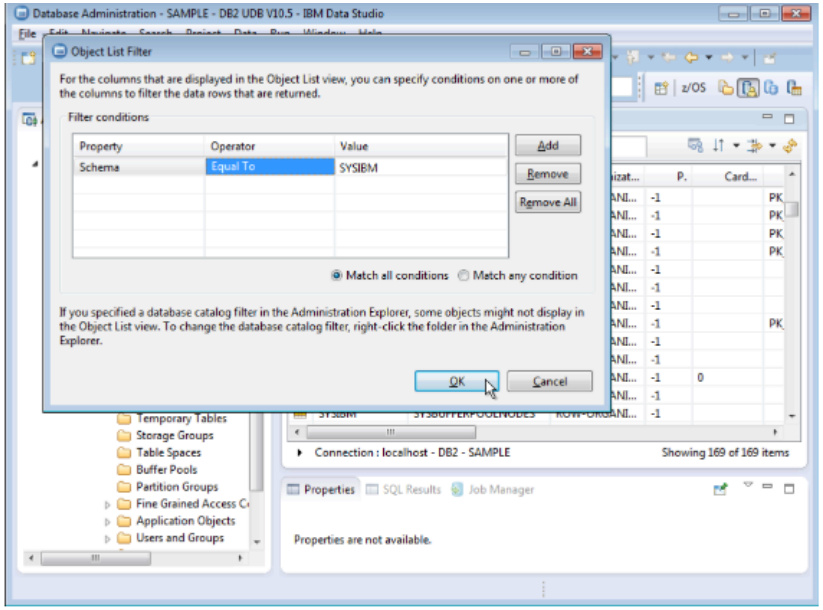
This would identify all columns with a certain pattern, or a name pattern, and also the table they are in.

It is worth spending some time familiarizing yourself with the catalog tables as they contain an exact description of the Db2 system.



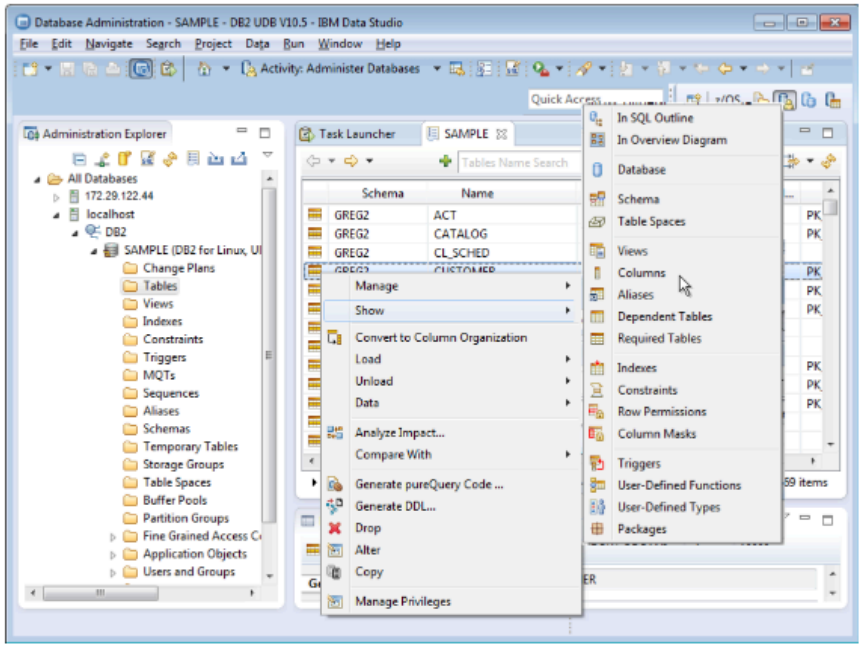
If you think that having to write SQL against the catalog to find out about the structure of your database is difficult, you are correct. Fortunately, there are many software tools that can be used to display Db2 catalog information.

The IBM Data Studio that we have been looking at throughout the last few modules can be used for this purpose. In this example, clicking the Tables folder on the left of the screen, displays a list of tables in the SAMPLE database on the right.



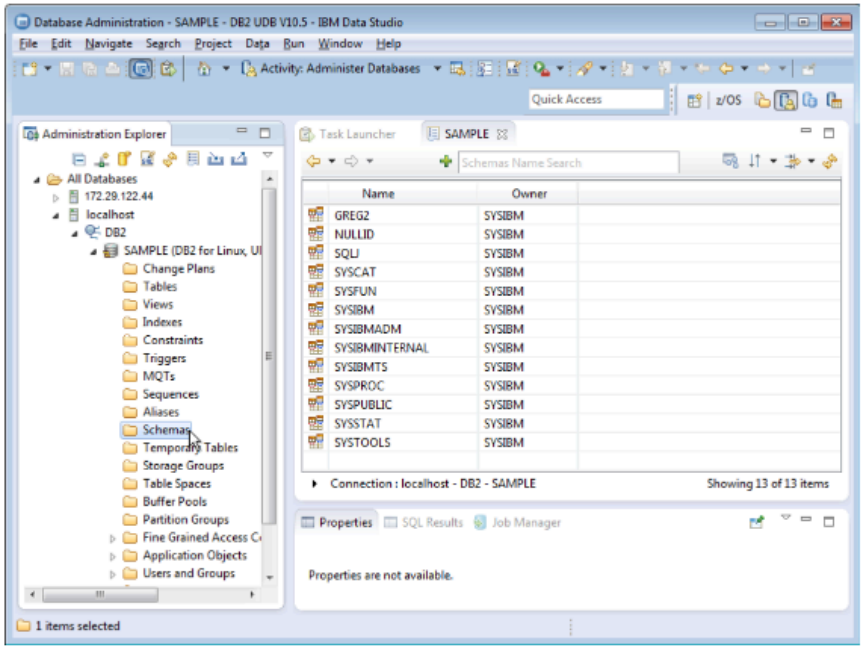
IBM Data Studio provides features to make it easy to filter displays, without having to know SQL.

Click **Play** to see an example of how the list is limited to the SYSIBM schema.



Using IBM Data Studio, it is easy to view information about each column in a table, including its name, type, default value, and if it is a key.

Click Play to see how to view column detail.



Using Db2 Data Studio, it is easy to view information about other Db2 resources such as buffer pools, indexes, schema and table spaces.

Click Play to see how this is done.