Skills
Network

Hands-on Lab: Db2 System Tables

Estimated time needed: 20 minutes

In this lab, you will learn how to access and work with system tables in the Db2 relational database management system. Every Db2 database contains system catalog tables that store metadata and definitions of the database objects such as tables, views, indexes, etc.

Objectives

After completing this lab, you will be able to use Db2 on Cloud to:

- Explore the system catalog tables of your Db2 database
- Query the system catalog to retrieve metadata from tables in your database
- Query the system catalog to retrieve column metadata

Software Used in This Lab

In this lab, you will use IBM Db2 Database. Db2 is a Relational Database Management System (RDBMS) from IBM, designed to store, analyze and retrieve data efficiently.

IBM **Db2**

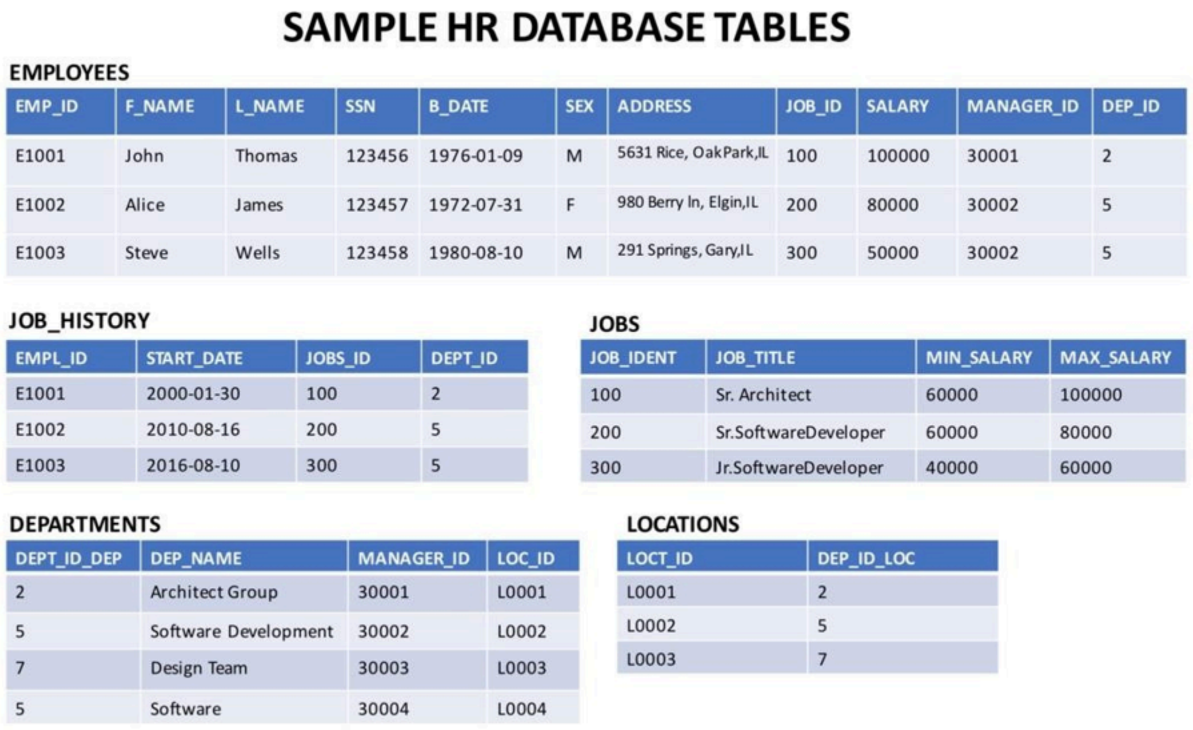
To complete this lab, you will use a Db2 database service on IBM Cloud. If you did not complete the following lab earlier, you may not have access to Db2 on Cloud and should complete that lab before starting this lab.

- Hands-on Lab : Sign up for IBM Cloud and Create Db2 service instance

Database Used in This Lab

The database used in this lab is an internal database. You will be working on a sample HR database. This HR database schema consists of 5 tables called EMPLOYEES, JOB_HISTORY, JOBS, DEPARTMENTS and LOCATIONS. Each table has a few rows of sample data. The following diagram shows the tables for the HR database:

|



SAMPLE HR DATABASE TABLES

EMPLOYEES

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

JOB_HISTORY

EMPL_ID	START_DATE	JOBS_ID	DEPT_ID
E1001	2000-01-30	100	2
E1002	2010-08-16	200	5
E1003	2016-08-10	300	5

JOBS

JOB_IDENT	JOB_TITLE	MIN_SALARY	MAX_SALARY
100	Sr. Architect	60000	100000
200	Sr.SoftwareDeveloper	60000	80000
300	Jr.SoftwareDeveloper	40000	60000

DEPARTMENTS

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
2	Architect Group	30001	L0001
5	Software Development	30002	L0002
7	Design Team	30003	L0003
5	Software	30004	L0004

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

Exercise 1: Load the Database into Db2

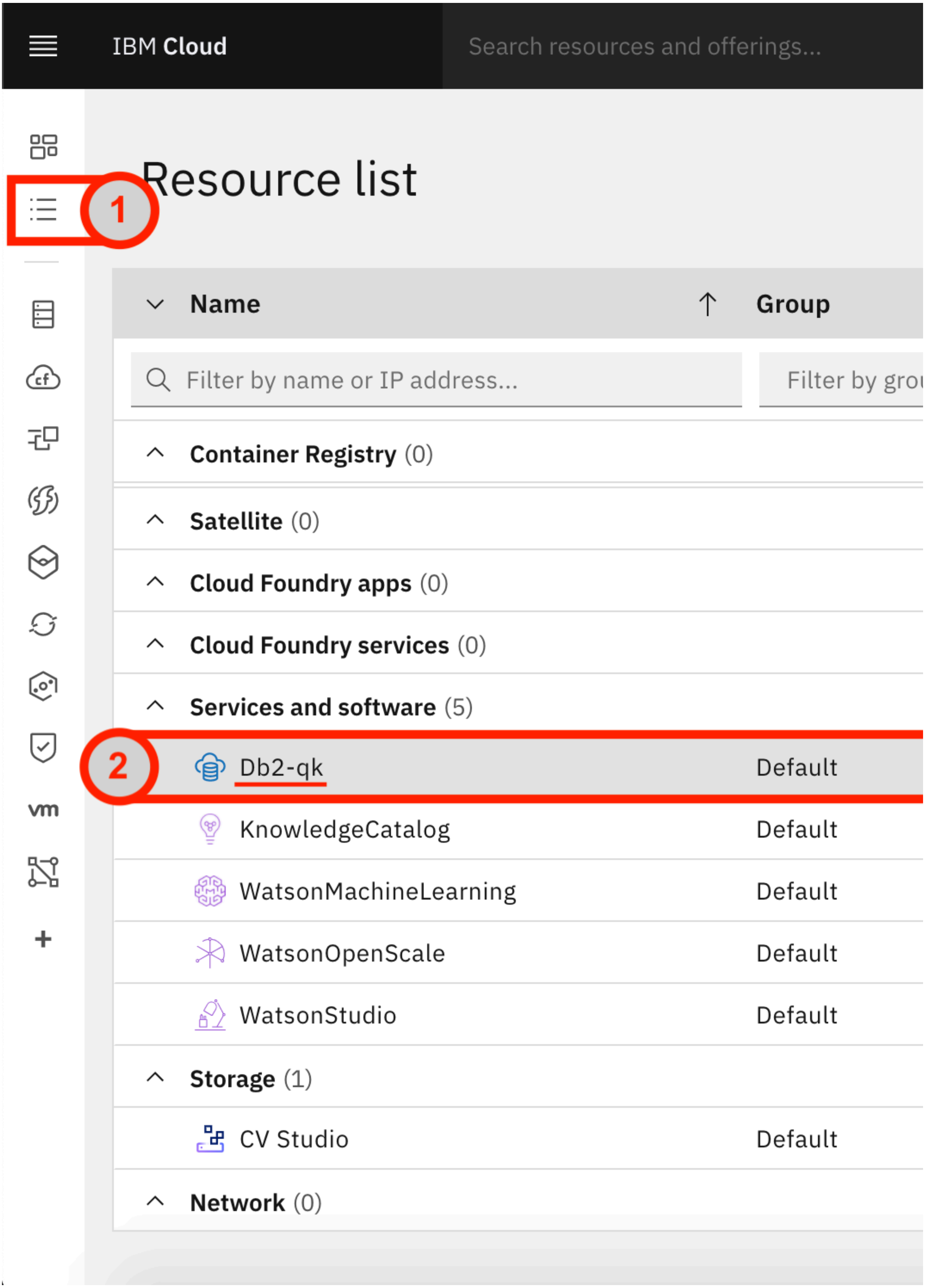
To get started, let’s open up Db2 on IBM Cloud and create the sample HR database.

1. Download the sample HR database by right-clicking the following link and saving as HR_Database_Create_Tables_Script.sql: Sample HR Database

2. Open up and log in to the IBM Cloud.

3. In left pane, click the “Resource list” button.

4. Under the “Services and software” subsection, select the Db2 database. It will be titled some variation of “Db2-*xx*“ where *xx* is some combination of letters and numbers.



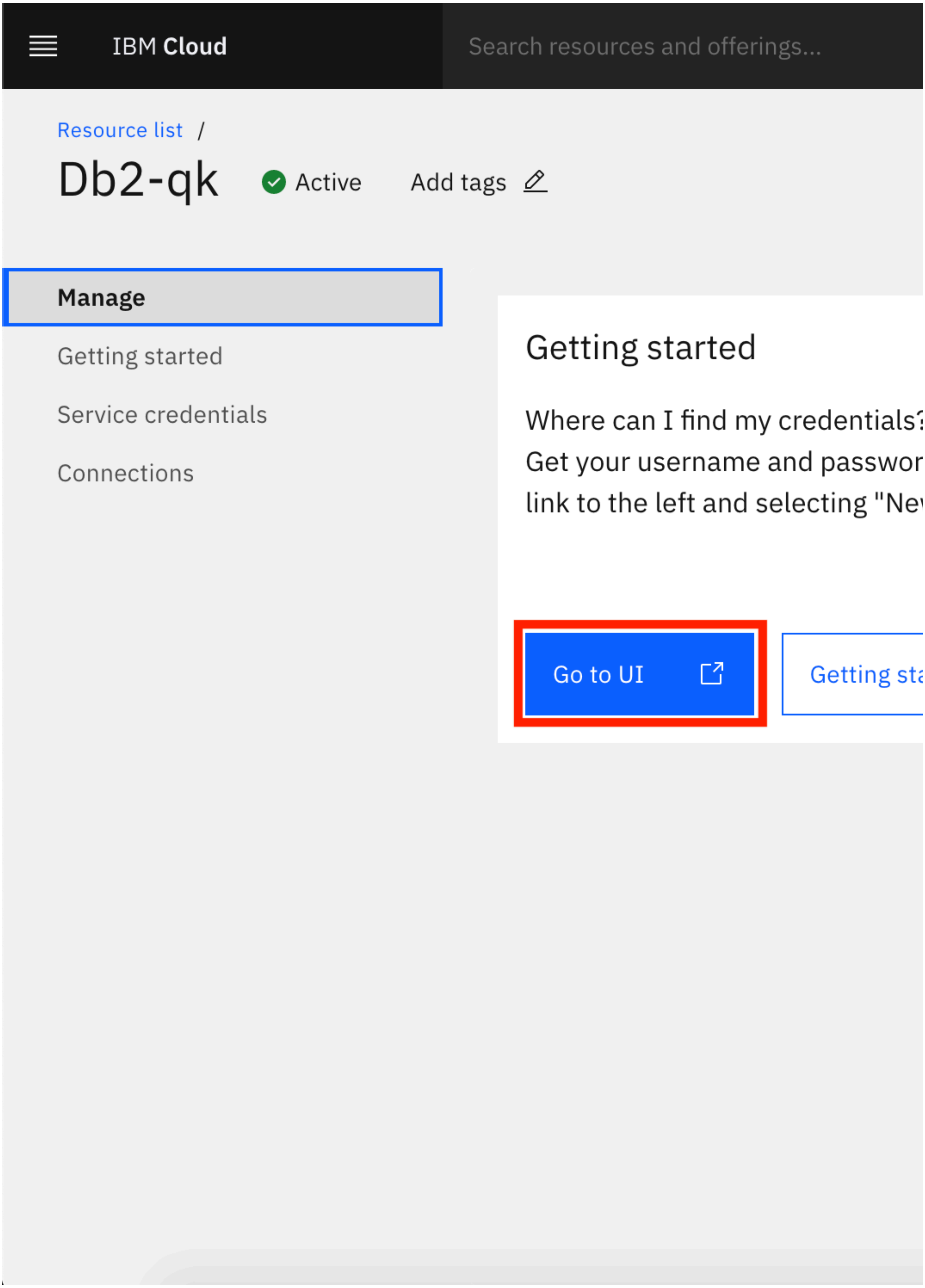
IBM Cloud

Search resources and offerings...

Resource list

Name	Group
Filter by name or IP address...	Filter by grou
Container Registry (0)	
Satellite (0)	
Cloud Foundry apps (0)	
Cloud Foundry services (0)	
Services and software (5)	
Db2-qk	Default
KnowledgeCatalog	Default
WatsonMachineLearning	Default
WatsonOpenScale	Default
WatsonStudio	Default
Storage (1)	
CV Studio	Default
Network (0)	

5. Click the “Go to UI” button to open the user interface for Db2.

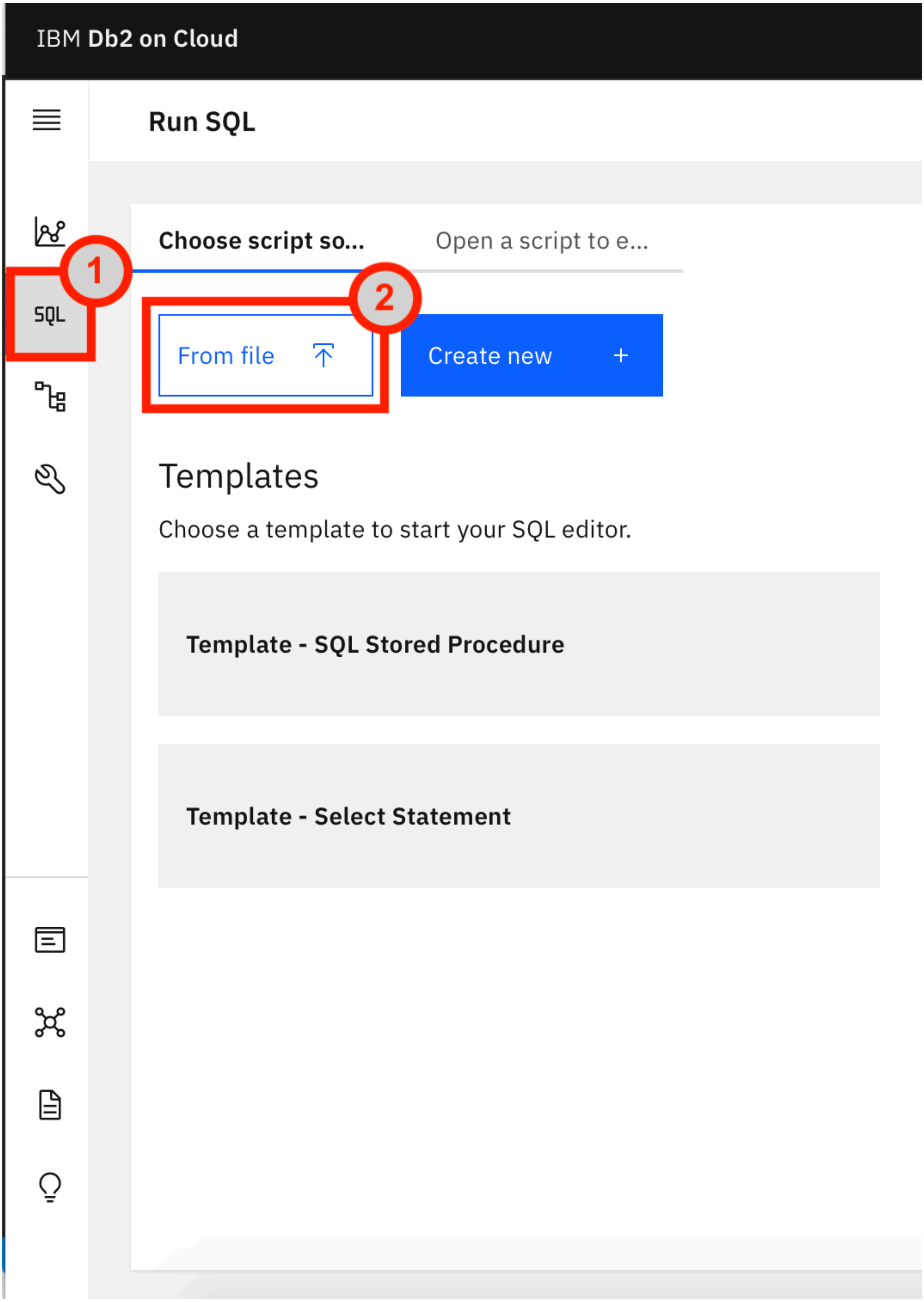


Getting started

Where can I find my credentials
Get your username and passwor
link to the left and selecting "Ne

6. To upload the sample HR database you are working with, click the "Run SQL" button on the menu tab in the left pane.

7. Click the “From file” button.



Templates

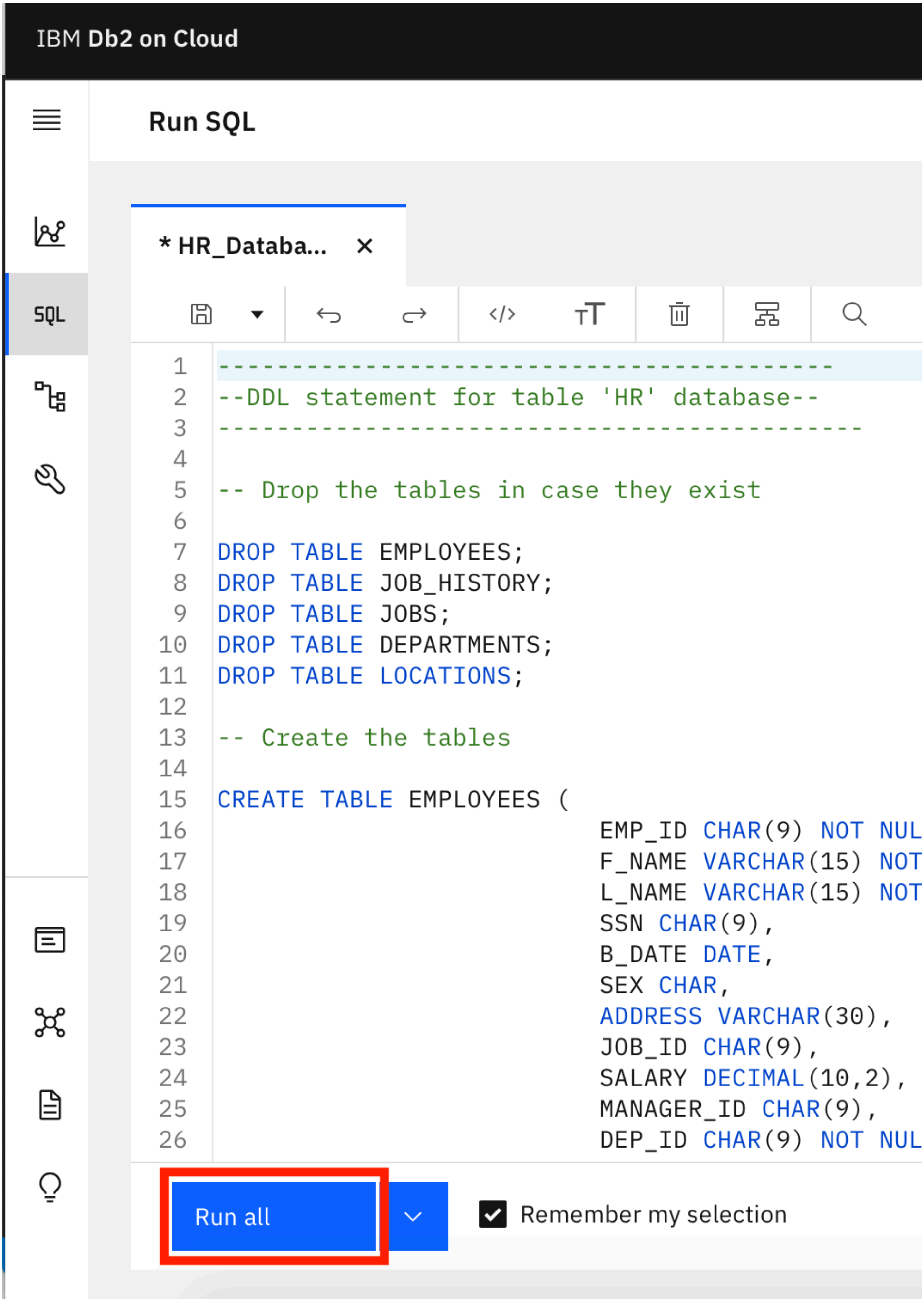
Choose a template to start your SQL editor.

Template - SQL Stored Procedure

Template - Select Statement

8. Upload the `HR_Database_Create_Tables_Script.sql` file you downloaded in Step 1.

9. Click the "Run all" button.



```
------------------------------------------
--DDL statement for table 'HR' database--
--------------------------------------------

-- Drop the tables in case they exist

DROP TABLE EMPLOYEES;
DROP TABLE JOB_HISTORY;
DROP TABLE JOBS;
DROP TABLE DEPARTMENTS;
DROP TABLE LOCATIONS;

-- Create the tables

CREATE TABLE EMPLOYEES (
                          EMP_ID CHAR(9) NOT NUL
                          F_NAME VARCHAR(15) NOT
                          L_NAME VARCHAR(15) NOT
                          SSN CHAR(9),
                          B_DATE DATE,
                          SEX CHAR,
                          ADDRESS VARCHAR(30),
                          JOB_ID CHAR(9),
                          SALARY DECIMAL(10,2),
                          MANAGER_ID CHAR(9),
                          DEP_ID CHAR(9) NOT NUL
```

After executing this SQL code, your Db2 server will now have a database containing several tables.

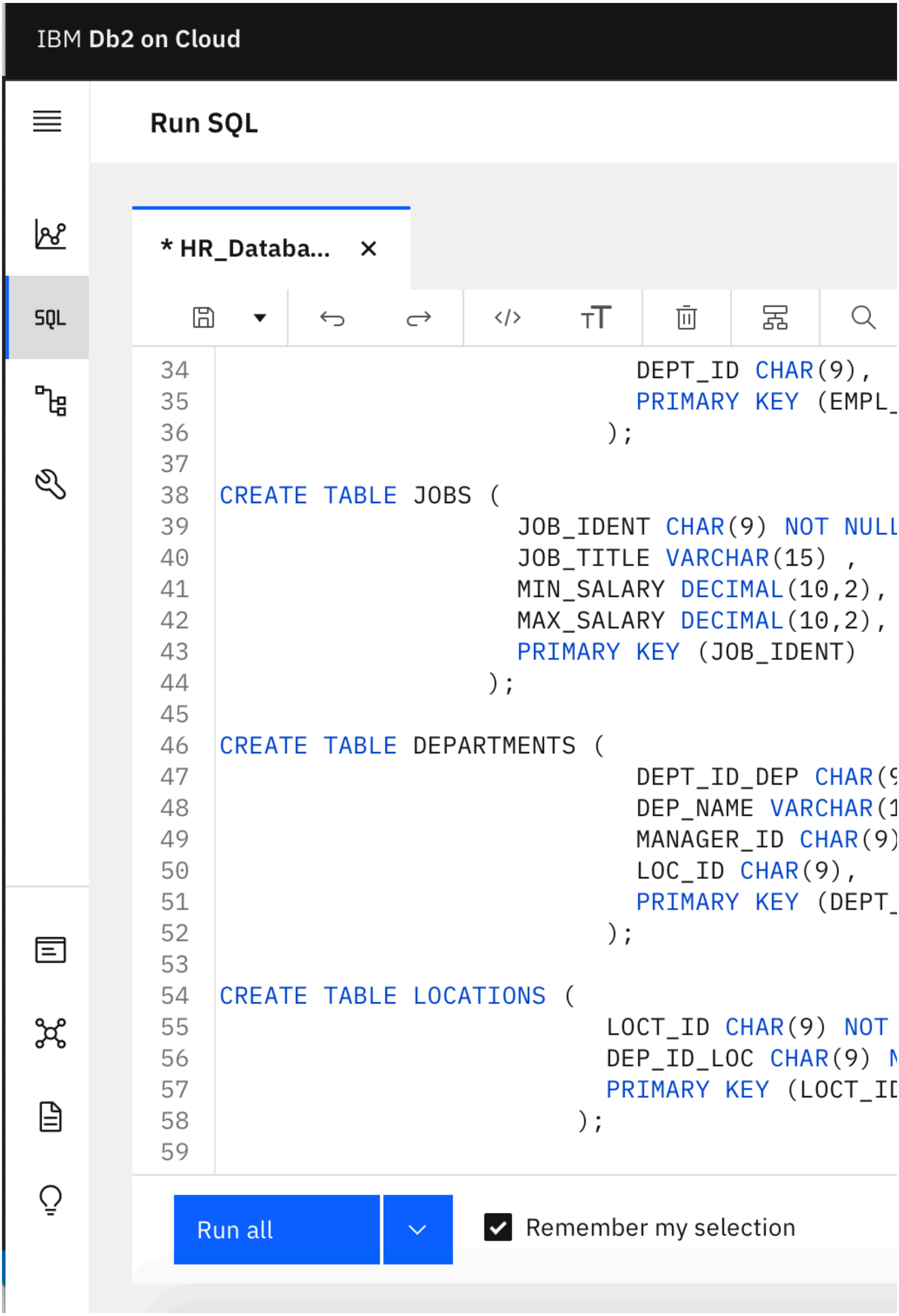
Exercise 2: Access the Db2 System Tables

Every Db2 database has a **SYSTABLES** catalog table. From the official IBM Db2 documentation, the SYSTABLES table contains one row for each table, view, or alias. Each table row indicates whether the object that it describes is a table, view, or alias, its name, who created it, the database that it belongs to, the table space it belongs to, and other information. The SYSTABLES table also has a REMARKS column in which you can store your own information about the table in question.

In essence, the SYSTABLES catalog table contains metadata about all other database objects in the Db2 server. Let’s go ahead and see how you can access some of this useful metadata.

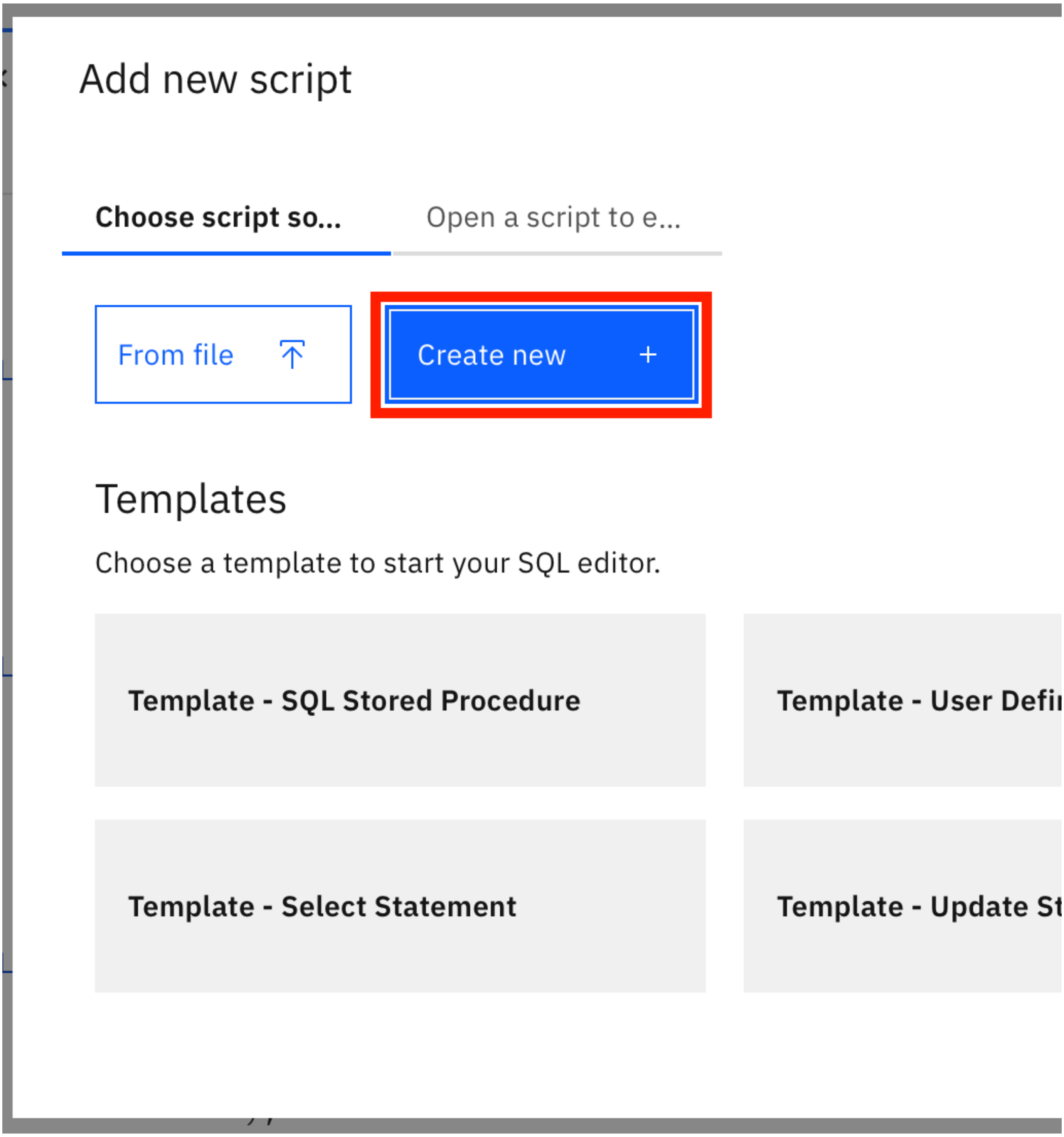
Task A: Perform a Simple Query on SYSTABLES

1. Click the “Add new script” button.



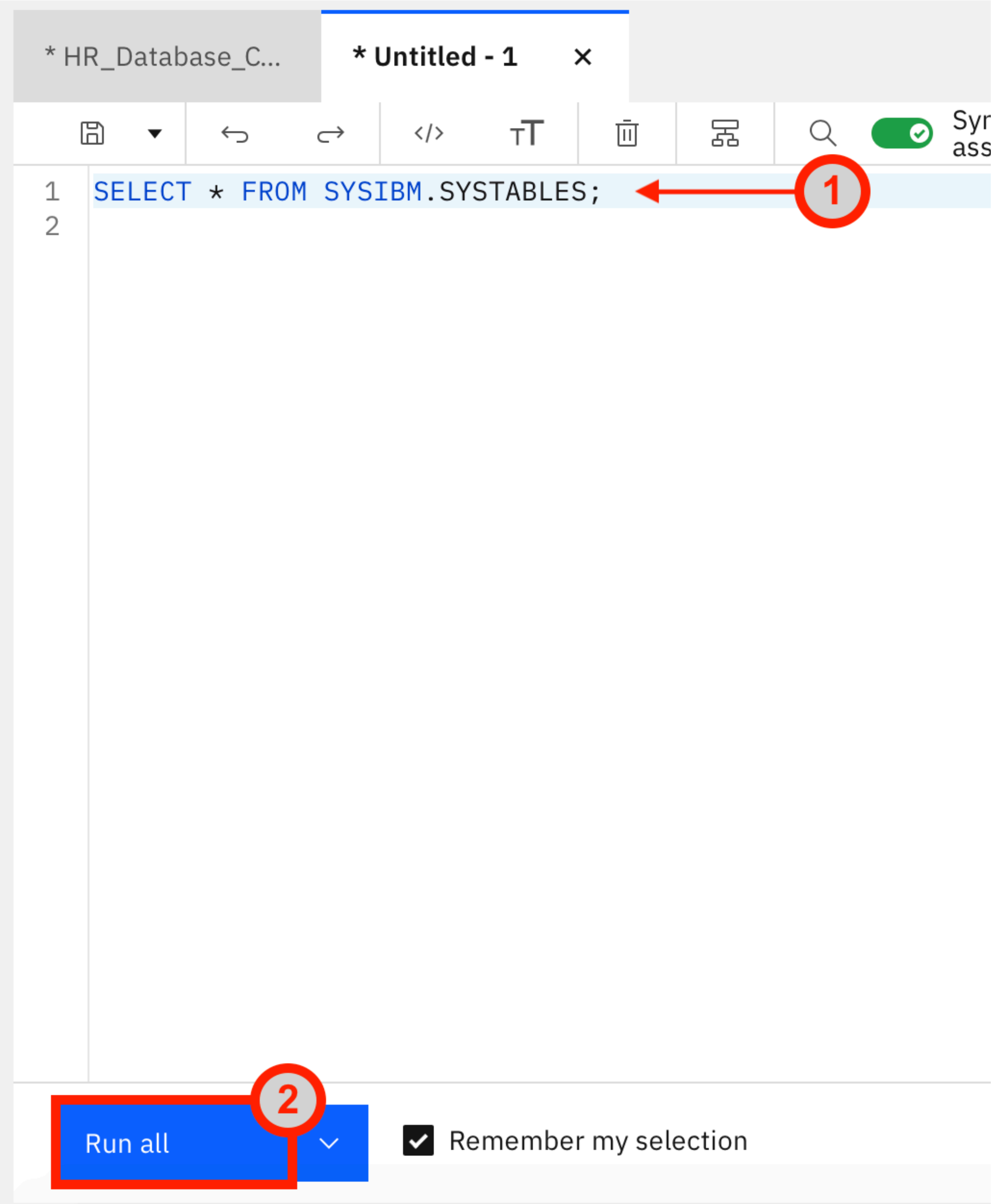
```
                            DEPT_ID CHAR(9),
                            PRIMARY KEY (EMPL_
                          );

CREATE TABLE JOBS (
                    JOB_IDENT CHAR(9) NOT NULL
                    JOB_TITLE VARCHAR(15) ,
                    MIN_SALARY DECIMAL(10,2),
                    MAX_SALARY DECIMAL(10,2),
                    PRIMARY KEY (JOB_IDENT)
                  );

CREATE TABLE DEPARTMENTS (
                            DEPT_ID_DEP CHAR(9
                            DEP_NAME VARCHAR(1
                            MANAGER_ID CHAR(9)
                            LOC_ID CHAR(9),
                            PRIMARY KEY (DEPT_
                          );

CREATE TABLE LOCATIONS (
                          LOCT_ID CHAR(9) NOT
                          DEP_ID_LOC CHAR(9) N
                          PRIMARY KEY (LOCT_ID
                        );
```

2. Click “Create new” to open an empty SQL editor.



3. Let’s start off with the most basic query on the `SYSTABLES` table by taking a look at the whole table. Enter the following code into the SQL editor and click the “Run all” button.

```
SELECT * FROM SYSIBM.SYSTABLES;
```



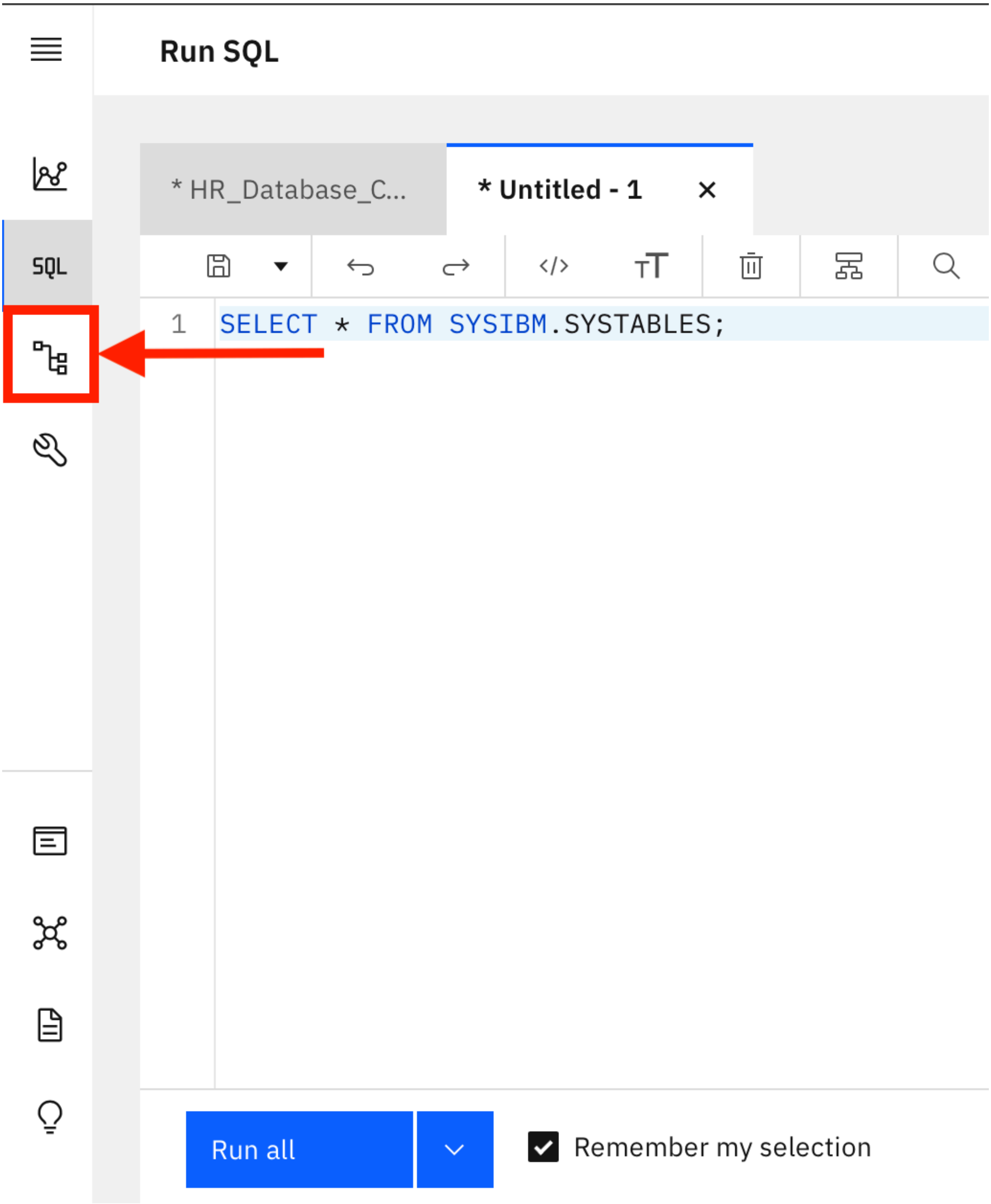
```
SELECT * FROM SYSIBM.SYSTABLES;
```

The result of the query is shown on the right-hand side of the screen. There are numerous results with only the first few displayed. You can see the first few columns showing the name of the table, its creator, and the type of the database object (T for table). The first five shown are all system tables and you can see that the creator is `SYSIBM`.

Task B: Query SYSTABLES by CREATOR

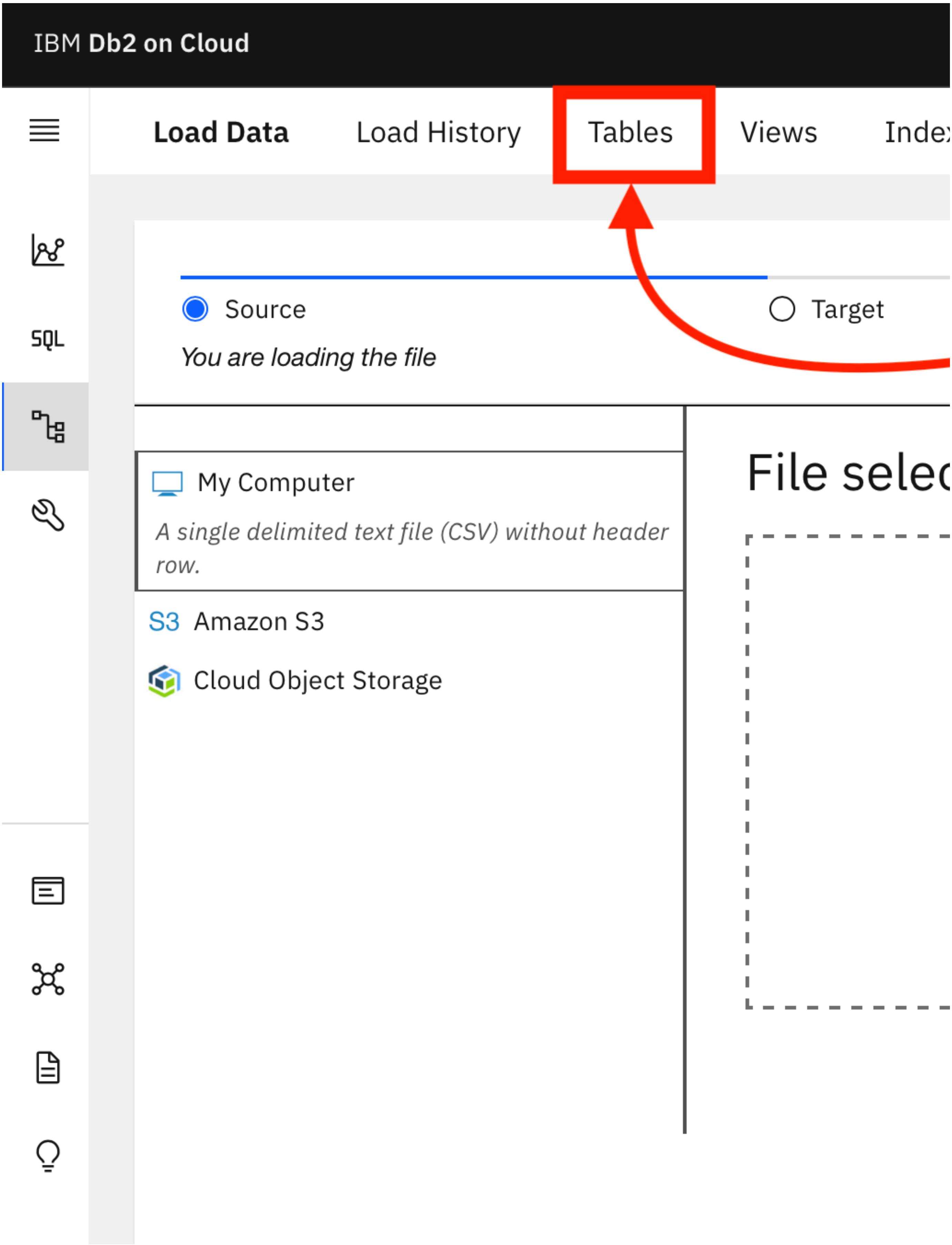
Let's do a slightly more advanced query on `SYSIBM.SYSTABLES` to retrieve the metadata for the tables you created for your sample HR database. One way to do this is to query the table based on the `CREATOR`column, which stores the schema that the given table belongs to.

1. First, you need to find the schema name for your Db2 database. In the left pane, click the "Data" button.

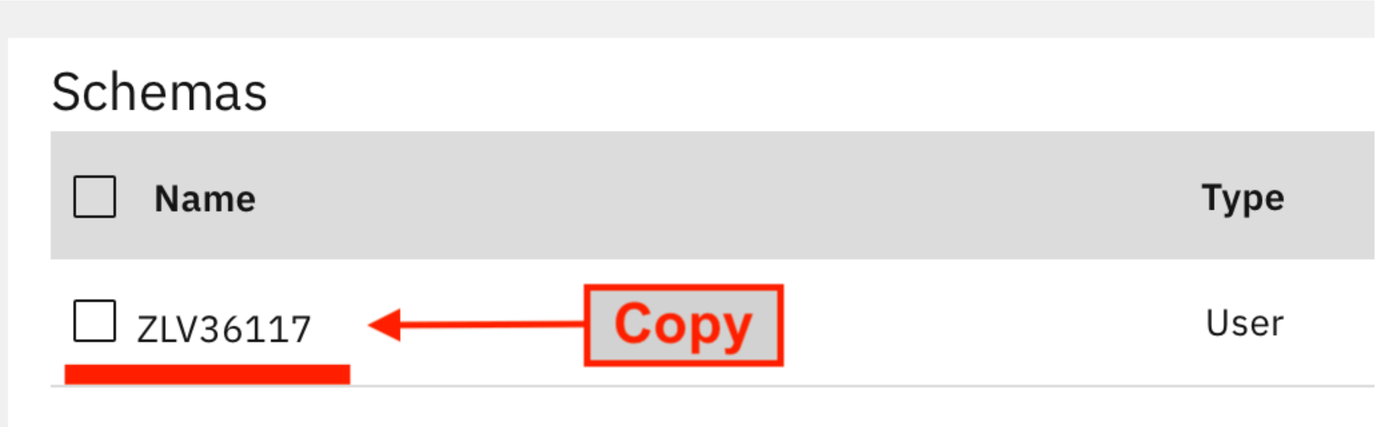


```
SELECT * FROM SYSIBM.SYSTABLES;
```

2. Click the “Tables” button located in the top menu bar.



3. Copy the name you see under the "Name" field and either write it down or paste it in a text editor of your choice. This is the schema name. In the screenshot below, this name is `ZLV36117`. In your case, it will certainly be a different name.



Schemas

□ Name	Type
□ ZLV36117	User

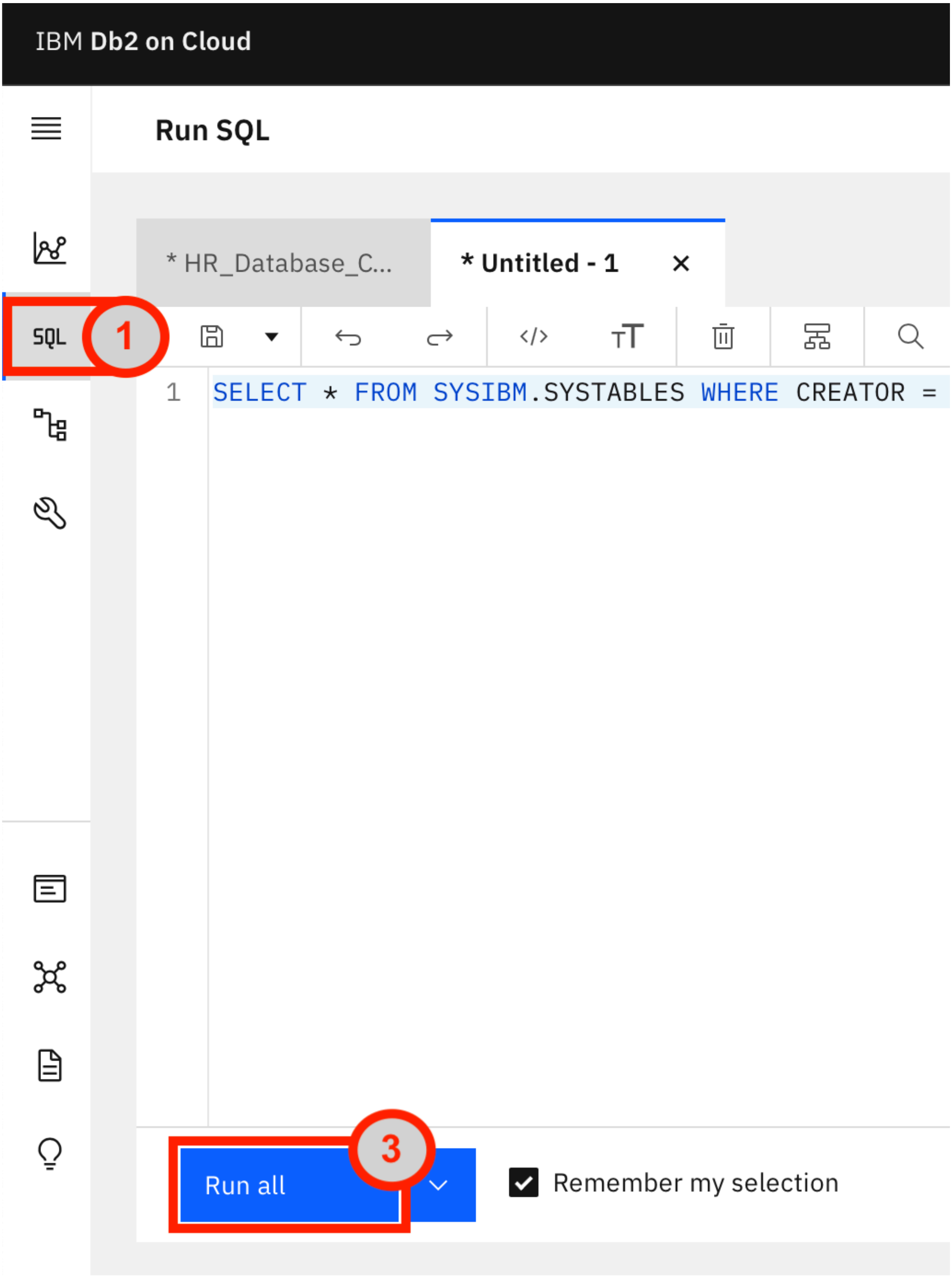
Copy

4. In the left pane, click the “SQL” button.

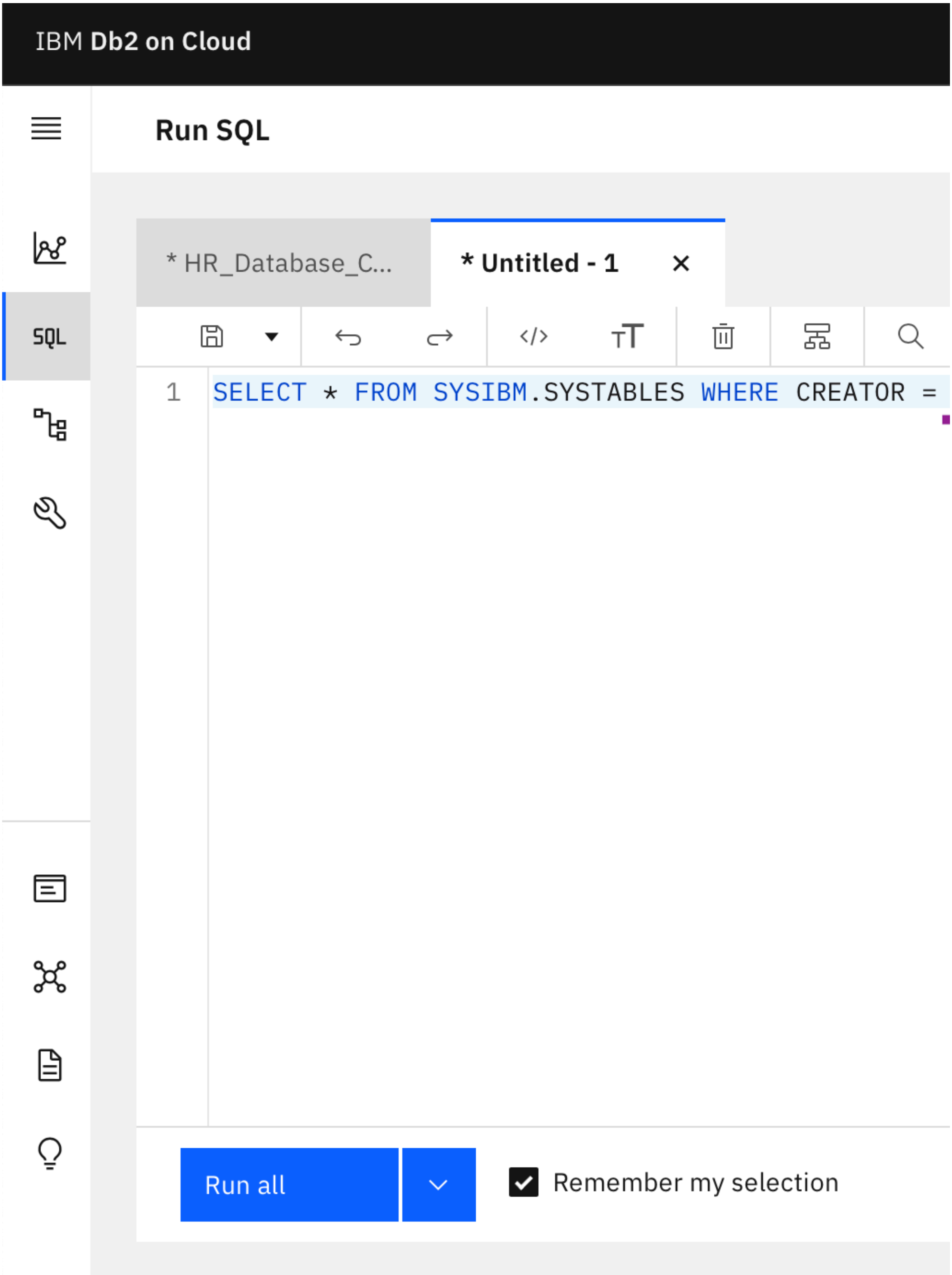
5. Enter the following SQL command to query the `SYSTABLES`, replacing the part within the single quotes with the name you copied in Step 3:

```
SELECT * FROM SYSIBM.SYSTABLES WHERE CREATOR = '<insert name here>';
```

6. Finally, click the “Run all” button to run the SQL script you created.



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



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

Again, on the right-hand side of the screen you can see the results of the query you executed. You can see the five tables that belong to the sample HR database and that their type is "T" for table.

Task C: Try Updating a System Table

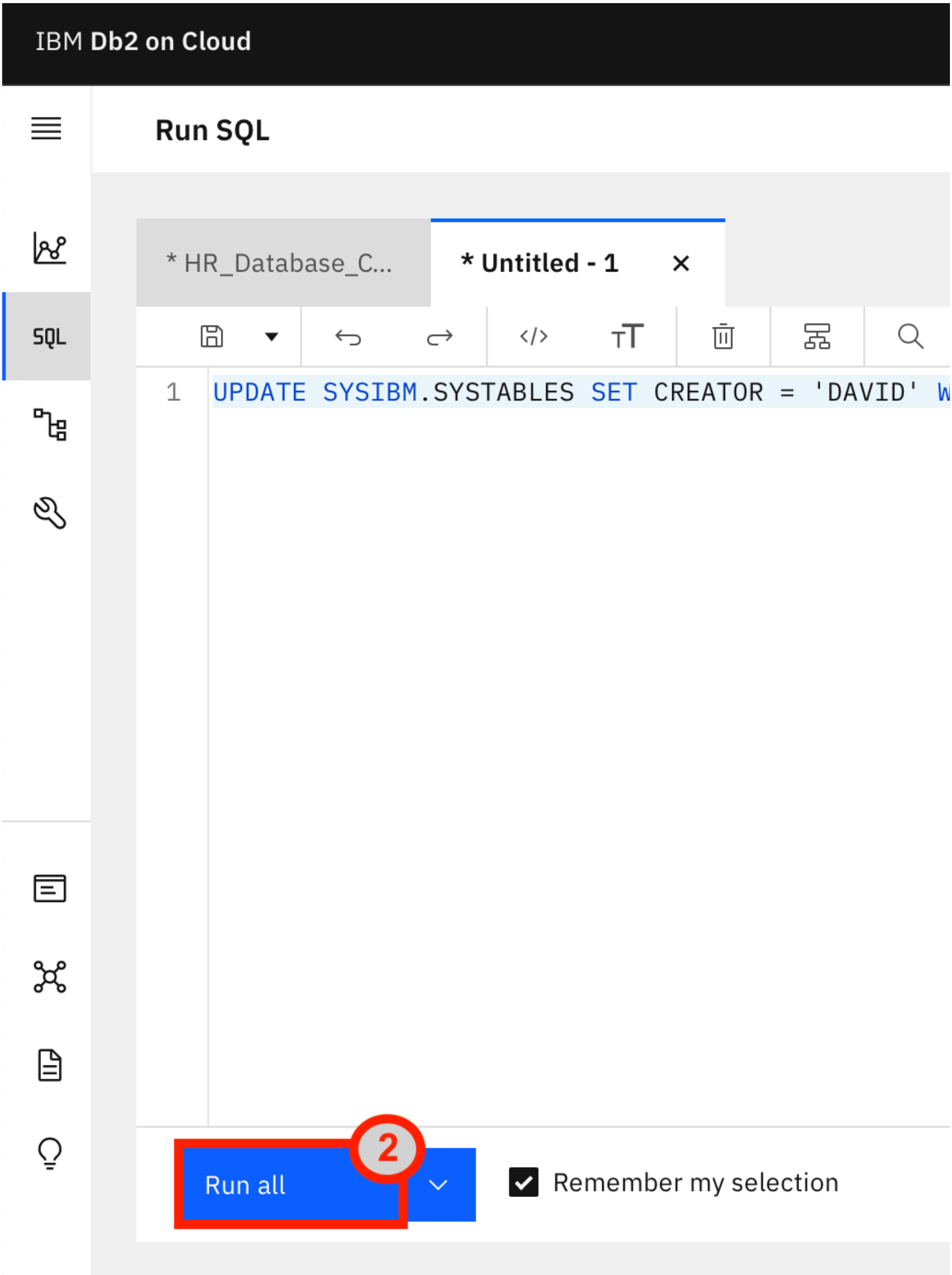
In a Db2 database, the system tables cannot be edited directly with `INSERT`, `UPDATE`, or `DELETE` statements. When you create, modify, or delete other objects in the database, the system tables are automatically updated to reflect the changes. Therefore, system tables are only for *retrieving* metadata about other objects in your database, you do not modify them directly.

The current name for the “CREATOR” of the five tables that belong to the sample HR database appears to be some random combination of letters and numbers, but suppose you wanted to modify the entries in “SYSTABLES” to have a different schema name in the “CREATOR” column for those tables.

Let’s see what happens if you try to modify the “CREATOR” column in “SYSTABLES”.

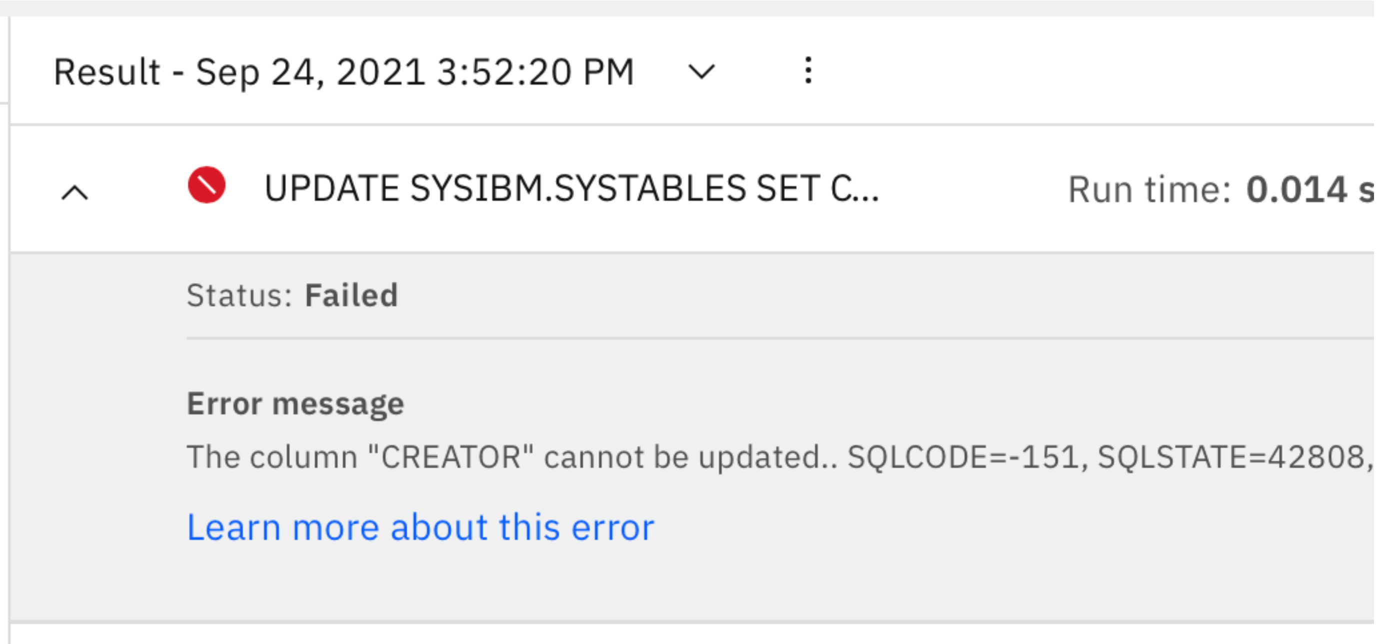
1. In the SQL editor, enter the following command and replace `<current creator name>` with the schema name you copied earlier in Step 3 of Task B. Replace `<new creator name>` with a new name of your choice, then click the “Run all” button.

```
UPDATE SYSIBM.SYSTABLES SET CREATOR = '<new creator name>' WHERE CREATOR ='<current creator name>';
```



```
UPDATE SYSIBM.SYSTABLES SET CREATOR = 'DAVID' W
```

Running this code results in the following error in the output:


Result - Sep 24, 2021 3:52:20 PM

UPDATE SYSIBM.SYSTABLES SET C... Run time: **0.014 s**

Status: **Failed**

Error message

The column "CREATOR" cannot be updated.. SQLCODE=-151, SQLSTATE=42808,

Learn more about this error

As you can see, attempting to execute an `UPDATE` command directly on SYSIBM.SYSTABLES, which is a system table, results in an error.

Exercise 3: Try it yourself!

As mentioned in the previous exercise, Db2 system tables cannot be modified directly. Instead, they store metadata about all objects in the database and are automatically updated to reflect any changes made to them, including the creation or deletion of database objects.

You will now get an opportunity to apply what you learned and see this in action for yourself. In this practice exercise, use the SQL editor along with the steps given below.

1. **Try it yourself:** Query the `SYSTABLES` table by the schema name to display the five tables that belong to the sample HR database.

   ▶ **Hint** (Click Here)
   ▶ **Solution** (Click Here)

2. **Try it yourself:** Try directly deleting the row corresponding to the EMPLOYEES table from the SYSIBM.SYSTABLES system table.

   ▶ **Hint** (Click Here)
   ▶ **Solution** (Click Here)

3. **Try it yourself:** Drop the EMPLOYEES table from the database.

   ▶ **Hint** (Click Here)
   ▶ **Solution** (Click Here)

4. **Try it yourself:** Once again, query the `SYSTABLES` table by schema name to confirm that the EMPLOYEES table was successfully dropped from the database.

   ▶ **Hint** (Click Here)
   ▶ **Solution** (Click Here)

Conclusion

Congratulations on completing this hands-on lab! You now have some understanding and experience of working with the system tables of a Db2 server and are ready for the next step in developing your skill set as a database administrator.

Author

David Pasternak

Other Contributors

Rav Ahuja, Sandip Saha Joy

© IBM Corporation 2023. All rights reserved.