

SQL stored procedure

Server query optimizer

About project

Instead of sending multiple SQL statements from the client to the server, I encapsulate them in a stored procedure on the server and send one statement from the client end to execute them.

Benefits: Stored procedures can be useful if you have an SQL query that you write and execute over and over again. You can save it as a stored procedure, and then just call it to execute it directly in the database server. In stored procedures, you can also pass parameters so that a stored procedure can act based on the passed parameter values.

Project case study: I business dealing on pet sales wants to automate the price update of pets on based on the present health status of the pet(good, bad, worse), they could not keep determining this price manually as that would take a lot of time with their database hosted on the cloud(IBM Db2). The task is-How can this business automate the price update of a particular pet given its health status? That is the question this project aims to answer. I created a stored procedure routine named **UPDATE_SALEPRICE** with parameters **Animal_ID** and **Animal_Health**.

- This **UPDATE_SALEPRICE** routine contains SQL queries to update the sale price of the animals in the PETSALE table depending on their health conditions, **BAD** or **WORSE**.

- This procedure routine takes animal ID and health condition as parameters which will be used to update the sale price of animal in the PETSALE table by an amount depending on their health condition. Suppose:

- For animal with ID XX having BAD health condition, the sale price will be reduced further by 25%.
- For animal with ID YY having WORSE health condition, the sale price will be reduced further by 50%.
- For animal with ID ZZ having other health condition, the sale price won't change.

Skills utilised



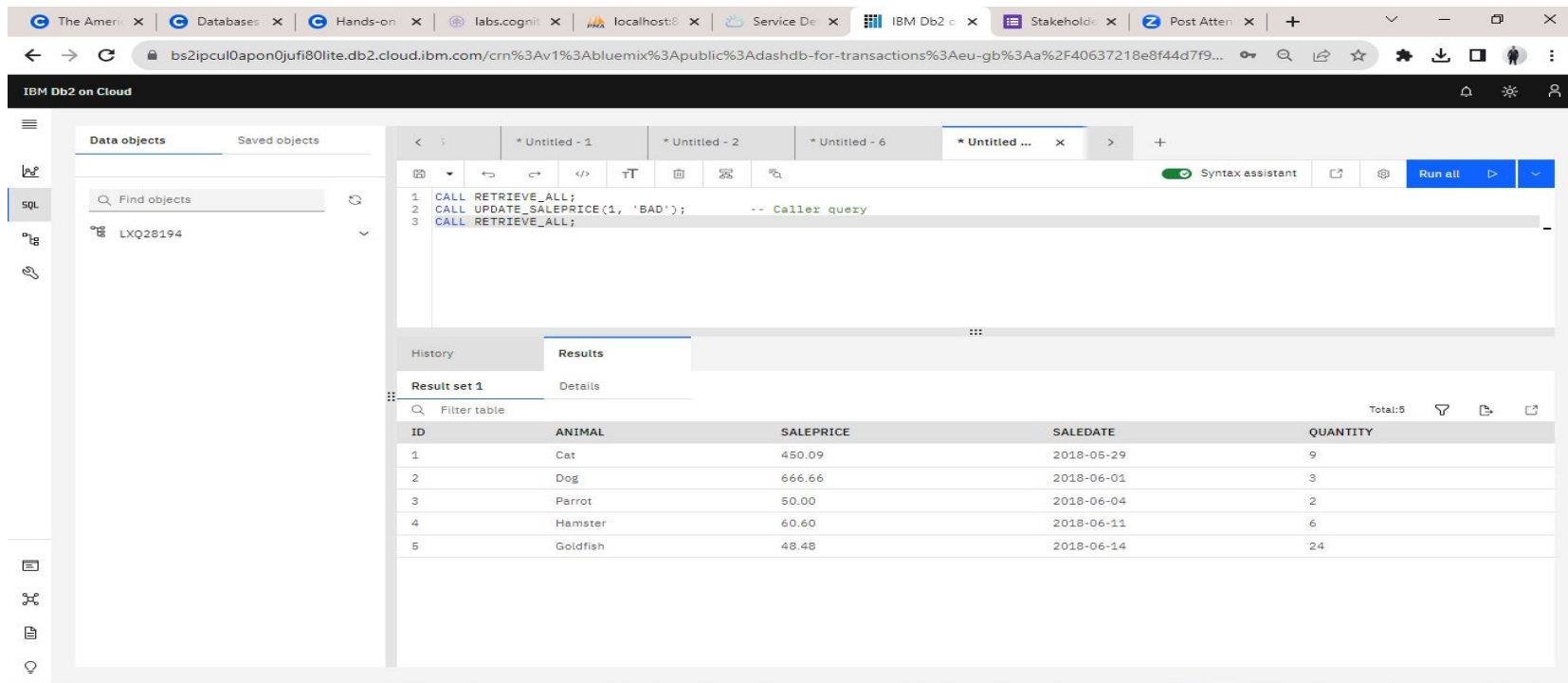
Software used: IBM Db2 cloud database

Project screenshots and explanations

Next five slides

Create database

This was what the database looks like in IBM Db2 database instance. The table shown contains list of different pets with their prices and quantity in stock. I retrieved the table with a VIEW call that retrieves all the the data in the original table.



The screenshot displays the IBM Db2 on Cloud web interface. The browser's address bar shows the URL: `bs2ipcul0apon0jufi80lite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2F40637218e8f44d7f9...`. The interface includes a sidebar with navigation options like 'Data objects', 'Saved objects', and 'SQL'. The main area shows a SQL editor with the following code:

```
1 CALL RETRIEVE_ALL;  
2 CALL UPDATE_SALEPRICE(1, 'BAD'); -- Caller query  
3 CALL RETRIEVE_ALL;
```

Below the editor, the 'Results' tab is active, displaying 'Result set 1' as a table. The table has five columns: ID, ANIMAL, SALEPRICE, SALEDATE, and QUANTITY. It contains five rows of data. The total number of rows is 5.

ID	ANIMAL	SALEPRICE	SALEDATE	QUANTITY
1	Cat	450.09	2018-05-29	9
2	Dog	666.66	2018-06-01	3
3	Parrot	50.00	2018-06-04	2
4	Hamster	60.60	2018-06-11	6
5	Goldfish	48.48	2018-06-14	24

Create Stored Procedure and save in the server

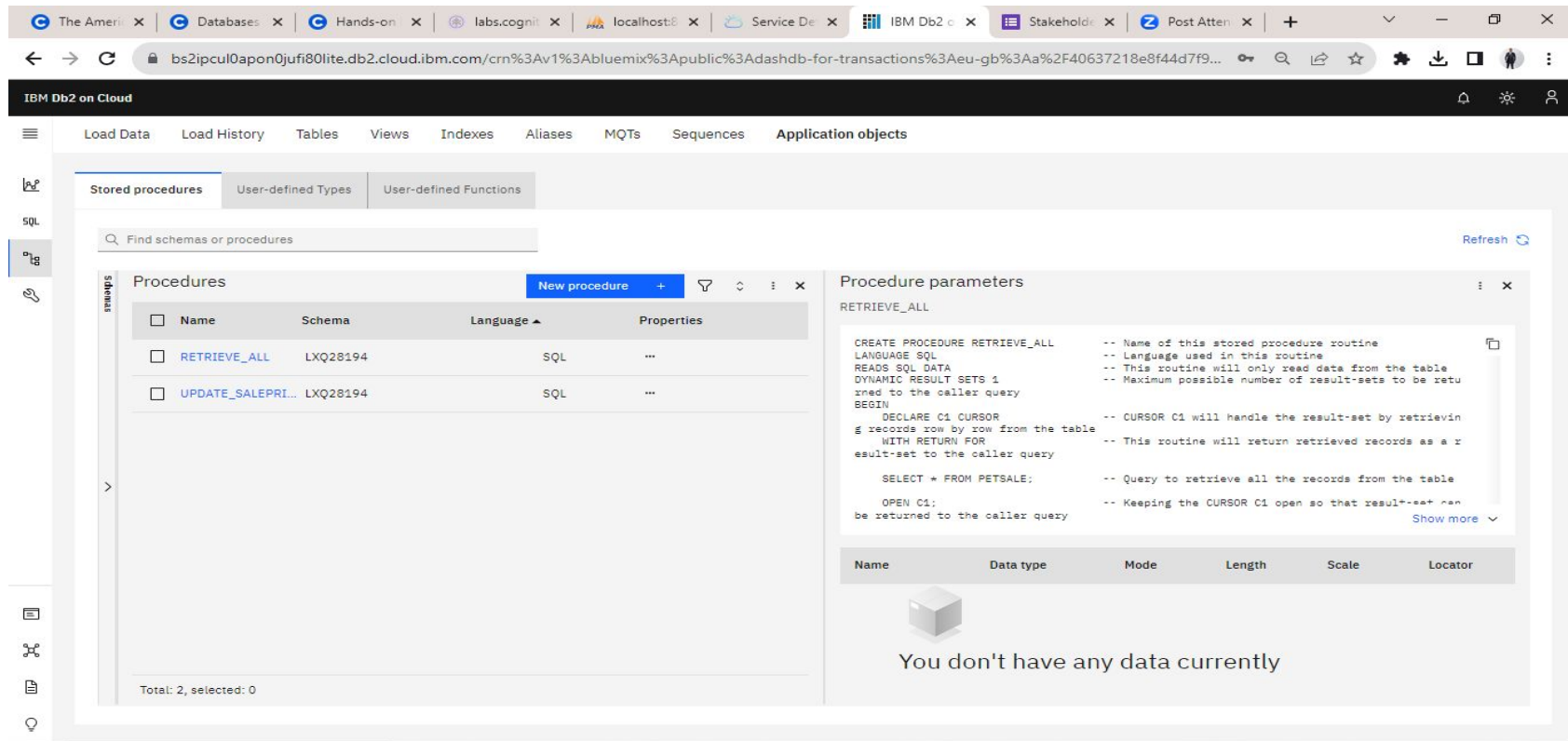
This is screenshot of the stored procedure and the SQL successfully created, ran and stored in the server

The screenshot displays the IBM Db2 on Cloud interface. On the left, the 'Data objects' tab is active, showing a search bar and a list of objects including 'LXQ28194'. The main area is a SQL editor with a script named 'Untitled - 5' containing a PL/SQL procedure 'UPDATE_SALEPRICE'. The script includes comments and logic to update the 'SALEPRICE' based on 'Animal_Health' status. Below the editor, the 'History' tab is selected, showing a table of executed statements with columns for Script, Date, Status, and Runtime.

Script	Date	Status	Runtime
Untitled - 5	Aug 18, 2023 2:03:40 PM	✓ 1	0.071 s
CREATE PROCEDURE UPDATE_SALEPRICE (IN Animal_ID INTEGER, IN Animal_Health VARCHAR(5)) ...		✓	0.071 s
Untitled - 1	Aug 18, 2023 1:32:32 PM	✓ 1	0.189 s
CALL RETRIEVE_ALL		✓	0.189 s
Untitled - 5	Aug 18, 2023 1:30:26 PM	✓ 1	0.050 s

Confirm the stored procedure is saved on the server

This screenshot shows on the left hand side that the UPDATE_SALESPRICE Procedure now exists in the database server and ready to run when called



The screenshot displays the IBM Db2 on Cloud interface. The top navigation bar includes tabs for Load Data, Load History, Tables, Views, Indexes, Aliases, MQTs, Sequences, and Application objects. The 'Stored procedures' tab is selected, showing a list of procedures in the LXQ28194 schema. The 'Procedures' table lists two procedures: RETRIEVE_ALL and UPDATE_SALESPRICE, both in the LXQ28194 schema and using SQL. The 'Procedure parameters' panel on the right shows the SQL code for RETRIEVE_ALL, which is a stored procedure that reads data from the PETSale table and returns it as a result set.

Name	Schema	Language	Properties
RETRIEVE_ALL	LXQ28194	SQL	...
UPDATE_SALESPRICE	LXQ28194	SQL	...

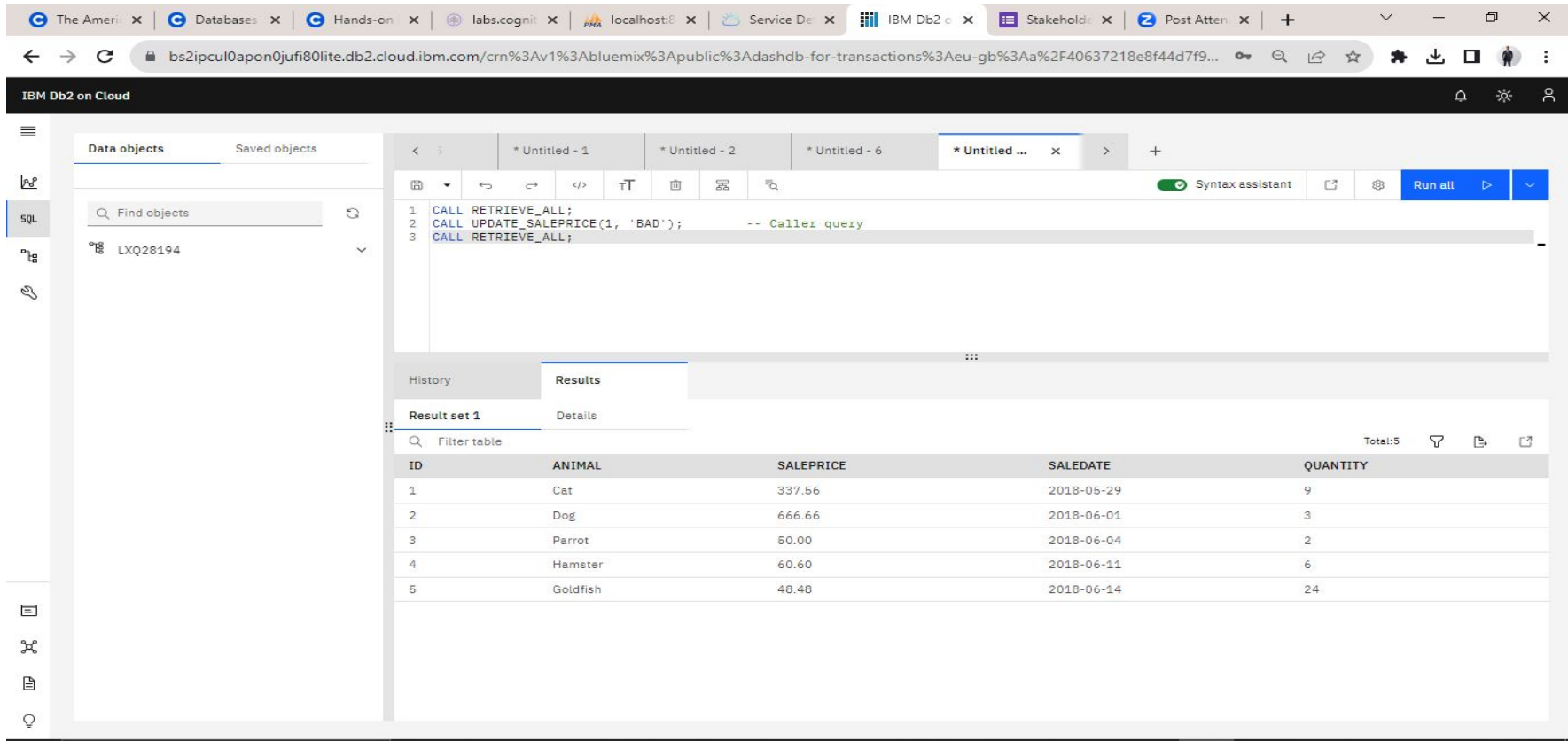
Procedure parameters: RETRIEVE_ALL

```
CREATE PROCEDURE RETRIEVE_ALL
LANGUAGE SQL
READS SQL DATA
DYNAMIC RESULT SETS 1
BEGIN
  DECLARE C1 CURSOR
  FOR SELECT * FROM PETSale;
  OPEN C1;
  SELECT * FROM PETSale;
```

You don't have any data currently

Call stored Procedure for PET ID '1' and health condition 'BAD'

I called the stored procedure with input **parameters 1 and BAD** and the price of **Cat** changed from 450 to 337, returning a 25% decrease in price as stipulated earlier



The screenshot shows the IBM Db2 on Cloud web interface. The browser address bar displays the URL: `bs2ipcul0apon0jufi80lite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2F40637218e8f44d7f9...`. The interface includes a sidebar with navigation icons and a main workspace. The workspace has a top bar with tabs for 'Data objects' and 'Saved objects'. Below this is a search bar labeled 'Find objects' and a list of objects, including 'LXQ28194'. The main area contains a SQL editor with the following code:

```
1 CALL RETRIEVE_ALL;  
2 CALL UPDATE_SALEPRICE(1, 'BAD'); -- Caller query  
3 CALL RETRIEVE_ALL;
```

Below the editor, there are tabs for 'History' and 'Results'. The 'Results' tab is active, showing 'Result set 1' with a 'Details' sub-tab. The results are displayed in a table with the following columns: ID, ANIMAL, SALEPRICE, SALEDATE, and QUANTITY. The table contains 5 rows of data. A 'Filter table' search bar is located above the table. The total number of rows is indicated as 'Total: 5'.

ID	ANIMAL	SALEPRICE	SALEDATE	QUANTITY
1	Cat	337.56	2018-05-29	9
2	Dog	666.66	2018-06-01	3
3	Parrot	50.00	2018-06-04	2
4	Hamster	60.60	2018-06-11	6
5	Goldfish	48.48	2018-06-14	24

Call stored Procedure for PET ID '3' and health condition 'WORSE'

I called the stored procedure with input **parameters 3 and WORSE** and the price of **Parrot** changed from 50 to 25, returning a 50% decrease in price as stipulated.

The screenshot shows the IBM Db2 on Cloud web interface. The browser address bar displays the URL: `bs2ipcul0apon0jufi80lite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2F40637218e8f44d7f9...`. The interface includes a sidebar with navigation icons for Data objects, Saved objects, SQL, and other database functions. The main area shows a SQL editor with the following code:

```
1 CALL RETRIEVE_ALL;  
2 CALL UPDATE_SALEPRICE(3, 'WORSE'); -- Caller query  
3 CALL RETRIEVE_ALL;
```

Below the editor, the 'Results' tab is active, displaying 'Result set 1' with a table of 5 rows. The table has columns: ID, ANIMAL, SALEPRICE, SALEDATE, and QUANTITY. The data shows a price update for the Parrot (ID 3) from 50.00 to 25.00.

ID	ANIMAL	SALEPRICE	SALEDATE	QUANTITY
1	Cat	337.56	2018-05-29	9
2	Dog	666.66	2018-06-01	3
3	Parrot	25.00	2018-06-04	2
4	Hamster	60.60	2018-06-11	6
5	Goldfish	48.48	2018-06-14	24

Conclusion

Stored procedure is a key server query optimisation method for increased transaction speed and database performance, enabling a high degree of security and automation and reducing query redundancy. With this, businesses can automate inventory updates faster.

— — —

ngwuogbonnaprince@gmail.com