Phase 3: Development Part

e-Commerce platform on IBM Cloud Foundry

How to begin building your artisanal e-commerce platform and setting up a database on IBM Cloud Foundry.

1. IBM Cloud Foundry Setup:

- First, ensure you have an IBM Cloud account and have logged in.

- Access IBM Cloud Foundry and create a new Cloud Foundry application to host your platform.

2. Platform Layout Design:

- Sketch out the layout of your e-commerce platform, including the homepage, product pages, and checkout process. Consider user experience and intuitive navigation.

3. Database Setup:

- Choose a suitable database service from IBM Cloud, such as IBM Db2 or IBM Cloudant, based on your project requirements.

- Create a new database instance and configure it to store product information. Define the necessary fields like product name, description, price, and images.

4. Connect Platform and Database:

- Integrate your platform with the database by configuring database connection parameters in your application code.

- Implement functions to read, write, update, and delete product information from the database.

5. Testing:

- Test your platform thoroughly to ensure that it can retrieve and display product information correctly from the database.

- Verify that the layout and design are user-friendly and responsive on different devices.

6. Deployment:

- Once testing is successful, deploy your artisanal e-commerce platform on IBM Cloud Foundry. Ensure it's accessible and functional in the live environment.

7. Submission:

- Document your design choices, database schema, and the steps you took to set up the platform and database.

- Prepare any necessary documentation or reports required for your assignment submission.

To create a database in IBM Cloud Foundry, you typically use a database service provided by IBM Cloud. Here's an example of creating a database using IBM Db2 on Cloud, which is one of IBM's database offerings. Please note that the exact steps and commands might vary based on the specific service you are using and the command-line interface you prefer (Cloud Foundry CLI or IBM Cloud CLI).

Using IBM Cloud Foundry CLI:

1. Log in to IBM Cloud:

cf login -a https://api.ng.bluemix.net

2. Create a Db2 Service Instance:

cf create-service dashDB Entry my-db-instance

Here, `dashDB` is the service name, `Entry` is the plan, and `my-db-instance` is the name you are giving to your service instance.

3. Bind the Service to Your Application:

cf bind-service YOUR-APP-NAME my-db-instance

Replace `YOUR-APP-NAME` with the name of your Cloud Foundry application.

4. Restage Your Application:

cf restage YOUR-APP-NAME

This step is necessary to ensure your application recognizes the newly bound database service.

5. Access Database Credentials:

cf service-key my-db-instance my-db-credentials

Replace `my-db-instance` with your service instance name and `my-db-credentials` with the name you want to give to your credentials.

This command will provide you with the necessary connection credentials (such as hostname, port, database name, username, and password) that your application needs to connect to the Db2 database.

Please adapt these commands based on the specific database service and plan you are using, and make sure to refer to the official IBM Cloud documentation for the most accurate and up-to-date instructions. The combination of the code set and territory values must be valid.

Not all collating sequences are valid with every code set and territory combination.

The table space definitions that are specified on CREATE DATABASE apply to all database partitions on which the database is being created. They cannot be specified separately for each database partition. If the table space definitions are to be created differently on particular database partitions, the CREATE TABLESPACE statement must be used.

The combination of the code set and territory values must be valid.

Not all collating sequences are valid with every code set and territory combination.

The table space definitions that are specified on CREATE DATABASE apply to all database partitions on which the database is being created. They cannot be specified separately for each database partition. If the table space definitions are to be created differently on particular database partitions, the CREATE TABLESPACE statement must be used.

When defining containers for table spaces, $N can be used. $N will be replaced by the database partition number when the container is actually created. This is required if the user wants to specify containers in a multiple logical partition database.