Step 1: Create Database

We need to create a new database for our operations. A database is a container that stores tables, procedures, views, and other database objects.

Query to Create a New Database

```
CREATE DATABASE sample2;
```

Explanation

The command creates a database named `sample2`. After creating it, we can work within this database by switching context.

Step 2: Switch Context to the New Database

After creating a database, you must use it for subsequent operations.

Query to Switch Database Context

```
USE sample2;
```

Explanation

The `USE` command tells the SQL server to execute all commands within the context of the database `sample2`. Every table, insert, or query will now take place in this database.

Step 3: Create Table

We'll now create a `customer` table in the database with specified fields, constraints, and defaults.

Query to Create Table

```
CREATE TABLE customer (
    customerid INT IDENTITY(1,1) PRIMARY KEY,
    customernumber INT NOT NULL UNIQUE CHECK (customernumber > 0),
    lastname VARCHAR(30) NOT NULL,
    firstname VARCHAR(30) NOT NULL,
    areacode INT DEFAULT 71000,
    address VARCHAR(50),
    country VARCHAR(50) DEFAULT 'Malaysia'
);
```

Explanation

1. **customerid INT IDENTITY(1,1) PRIMARY KEY**:
   - Creates an auto-incrementing integer as a unique identifier for each customer.
2. **customernumber INT NOT NULL UNIQUE CHECK (customernumber > 0)**:
   - Ensures `customernumber` is a unique number and must always be greater than `0`.
3. **lastname & firstname VARCHAR(30) NOT NULL**:
   - These columns must always have values; they cannot be `NULL`.
4. **areacode INT DEFAULT 71000**:
   - If no value is provided during insertion, `areacode` will default to `71000`.
5. **address VARCHAR(50)**:
   - This column can accept customer address as a string up to 50 characters.
6. **country VARCHAR(50) DEFAULT 'Malaysia'**:
   - The default country value is 'Malaysia' unless explicitly specified.

Step 4: Insert Data into the Table

We will insert some initial records into the `customer` table.

Query to Insert Data

```
INSERT INTO customer (customernumber, lastname, firstname, areacode, address, country)

VALUES

(100, 'Fang Ying', 'Sham', 418999, 'sdadasfdfd', DEFAULT),

(200, 'Mei Mei', 'Tan', DEFAULT, 'adssdsadsd', 'Thailand'),

(300, 'Albert', 'John', DEFAULT, 'dfdsfsdf', DEFAULT);
```

Explanation

1. **First Record**:
   - `customernumber = 100`, name `Fang Ying Sham`, area code `418999`, address `'sdadasfdfd'`, and the default value for `country`.
2. **Second Record**:
   - `customernumber = 200`, name `Mei Mei Tan`, default area code, address `'adssdsadsd'`, and `country = 'Thailand'`.
3. **Third Record**:
   - `customernumber = 300`, name `Albert John`, default area code, address `'dfdsfsdf'`, and the default `country`.

Step 5: Display Records

We can view the records in the table.

Query to Display All Records

```
SELECT * FROM customer;
```

Explanation

The `SELECT *` retrieves all rows and columns from the table.

Query to Display Specific Columns

```
SELECT customerid, customernumber, lastname, firstname
FROM customer;
```

Explanation

Here we select only the `customerid`, `customernumber`, `lastname`, and `firstname`. This shows how you can query specific columns rather than fetching all data.

Step 6: Add New Column

We can alter the table structure by adding a new column.

Query to Add a New Column

```
ALTER TABLE customer

ADD phonenumber VARCHAR(20);
```

Explanation

1. The `ALTER TABLE` command modifies the structure of an existing table.
2. We added the column `phonenumber` to the `customer` table with datatype `VARCHAR(20)`.

Step 7: Update the New Column with Data

Once the column is added, we can insert values into it.

Query to Update Records

```
UPDATE customer

SET phonenumber = '1234545346'

WHERE customerid = 1;


UPDATE customer

SET phonenumber = '45554654'

WHERE customerid = 2;
```

Explanation

1. The first query updates the `phonenumber` for `customerid = 1`.
2. The second query updates the `phonenumber` for `customerid = 2`.

Step 8: Delete a Column

To remove a column from a table, use the `ALTER TABLE DROP COLUMN`.

Query to Remove Column

```
ALTER TABLE customer
DROP COLUMN phonenumber;
```

Explanation

The `DROP COLUMN` command removes the column `phonenumber` entirely from the table.

Step 9: Delete Records

Deleting specific rows from a table based on conditions.

Query to Delete Specific Records

```
DELETE FROM customer

WHERE country = 'Thailand';
```

Explanation

This deletes only the rows where the `country = 'Thailand'`. **Note:** If you omit the `WHERE` clause, all rows would be deleted.

Step 10: Delete the Entire Table

If the table is no longer needed, you can drop it entirely.

Query to Drop Table

```
DROP TABLE customer;
```

Explanation

The DROP TABLE removes the entire table and all of its data from the database.

Step 11: Change Data Type

You can alter column data types to meet evolving requirements.

Query to Change Data Type

```
ALTER TABLE customer

ALTER COLUMN phonenumber VARCHAR(10);
```

Explanation

The ALTER COLUMN changes the datatype of the phonenumber column to VARCHAR(10). This is useful for schema evolution when requirements change.