SQL DATATYPES:

1. Numeric Data Types:

Numeric data types are used to store numeric values, such as integers and floating-point numbers.

Example 1: Integer
```sql
CREATE TABLE Employee (
    EmployeeID INT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Age INT
);
```

In this example, the `Age` column is defined as an `INT` (integer) data type, which stores whole numbers.

Example 2: Decimal
```sql
CREATE TABLE Product (
    ProductID INT,
    ProductName VARCHAR(100),
    Price DECIMAL(10, 2)
);
```

Here, the `Price` column is defined as a `DECIMAL(10, 2)`, meaning it can store numbers with up to 10 digits, 2 of which can be after the decimal point.

2. Character Data Types:

Character data types are used to store alphanumeric and text values.

Example 1: VARCHAR
```sql
CREATE TABLE Customer (
    CustomerID INT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100)
```

);
```

In this example, `FirstName`, `LastName`, and `Email` columns are defined as `VARCHAR`, which can store variable-length character strings.

#### Example 2: CHAR
```sql
CREATE TABLE Department (
    DeptID INT,
    DeptName CHAR(30),
    Location VARCHAR(100)
);
```

Here, `DeptName` is defined as `CHAR(30)`, which means it will always store 30 characters, padding with spaces if necessary.

3. Date and Time Data Types:

Date and time data types are used to store temporal values.

Example 1: DATE
```sql
CREATE TABLE Order (
    OrderID INT,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2)
);
```

In this example, `OrderDate` is defined as `DATE`, storing only the date without the time component.

Example 2: DATETIME
```sql
CREATE TABLE Event (
    EventID INT,
    EventName VARCHAR(100),
    StartTime DATETIME,
    Location VARCHAR(100)
);
```
```

Here, `StartTime` is defined as `DATETIME`, storing both date and time information.

Certainly! Let's continue with more examples for each category of SQL data types.

1. Numeric Data Types (Continued):

Example 3: FLOAT
```sql
CREATE TABLE TemperatureReading (
    ReadingID INT,
    SensorID INT,
    Temperature FLOAT
);
```

In this example, the `Temperature` column is defined as `FLOAT`, allowing the storage of floating-point numbers with a higher precision compared to `DECIMAL`.

Example 4: BIGINT
```sql
CREATE TABLE FinancialTransaction (
    TransactionID BIGINT,
    AccountID INT,
    Amount DECIMAL(18, 2),
    TransactionDate DATETIME
);
```

Here, `TransactionID` is defined as `BIGINT`, suitable for storing large integer values, often used as primary keys.

2. Character Data Types (Continued):

Example 3: TEXT
```sql
CREATE TABLE BlogPost (
    PostID INT,
    Title VARCHAR(255),
    Content TEXT,
    AuthorID INT
);
```

In this example, the `Content` column is defined as `TEXT`, suitable for storing large amounts of text data, such as blog post content.

Example 4: ENUM
```sql
CREATE TABLE Student (
    StudentID INT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Grade ENUM('A', 'B', 'C', 'D', 'F')
);
```

Here, the `Grade` column is defined as `ENUM`, allowing only specific values ('A', 'B', 'C', 'D', 'F').

3. Date and Time Data Types (Continued):

Example 3: TIME
```sql
CREATE TABLE Meeting (
    MeetingID INT,
    StartTime TIME,
    EndTime TIME,
    MeetingRoom VARCHAR(50)
);
```

In this example, `StartTime` and `EndTime` are defined as `TIME`, storing only the time component without the date.

Example 4: TIMESTAMP
```sql
CREATE TABLE LogEntry (
    LogID INT,
    LogMessage VARCHAR(255),
    LogTimestamp TIMESTAMP
);
```

Here, `LogTimestamp` is defined as `TIMESTAMP`, storing both date and time information, similar to `DATETIME`.