SQL DDL and DML

- SQL Data Definition Language (DDL)
  - CREATE TABLE
  - ALTER TABLE
  - DROP TABLE
- SQL Data Manipulation Language (DML)
  - INSERT
  - UPDATE
  - DELETE

Constraints

- Constraints can be defined within the CREATE TABLE statement, or they can be added to the table after it is created using the ALTER table statement

- Five types of constraints:
  - PRIMARY KEY may not have null values
  - UNIQUE may have null values
  - NULL/NOT NULL
  - FOREIGN KEY
  - CHECK

Creating Relationships

```
CREATE TABLE ARTIST(
    ArtistID            int                 NOT NULL IDENTITY (1, 1),
    Name                char (25)           NOT NULL,
    Nationality         char (30)           NULL,
    Birthdate           numeric (4, 0)      NULL,
    DeceasedDate        numeric (4, 0)      NULL,

    CONSTRAINT  ArtistPK  PRIMARY  KEY (ArtistID) ,
    CONSTRAINT  ArtistAK1  UNIQUE  (Name)

) ;

CREATE TABLE WORK   (
    WorkID              int                 NOT NULL IDENTITY (500, 1),
    Title               char (25)           NOT NULL,
    Copy                char (8)            NOT NULL,
    Description         varchar (1000)      NULL,
    ArtistID            int                 NOT NULL,

    CONSTRAINT  WorkPK  PRIMARY  KEY (WorkID) ,
    CONSTRAINT  WorkAK1  UNIQUE  (Title, Copy) ,
    CONSTRAINT  ArtistFK  FOREIGN  KEY (ArtistID) REFERENCES  ARTIST (ArtistID)
            ON  DELETE  NO  ACTION
            ON  UPDATE  NO  ACTION
) ;
```

SQL for Constraints

```
CREATE TABLE ARTIST(
        ArtistID              int                   NOT NULL IDENTITY (1, 1),
        Name                  char (25)             NOT NULL,
        Nationality           char (30)             NULL,
        Birthdate             numeric (4, 0)        NULL,
        DeceasedDate          numeric (4, 0)        NULL,

        CONSTRAINT ArtistPK PRIMARY KEY (ArtistID) ,
        CONSTRAINT ArtistAK1 UNIQUE (Name) ,
        CONSTRAINT NationalityValues CHECK
                (Nationality IN ( 'Canadian' , 'English' , 'French' , 'German',
                'Mexican' , 'Russian' , 'Spanish' , 'US' ) ) ,
        CONSTRAINT BirthValuesCheck CHECK (Birthdate < DeceasedDate) ,
        CONSTRAINT ValidBirthYear CHECK (Birthdate LIKE ' [1 - 2] [0 - 9] [0 - 9] [0 - 9]' ) ,
        CONSTRAINT ValidDeathYear CHECK (DeceasedDate LIKE ' [1 - 2] [0 - 9] [0 - 9] [0 - 9]' )

) ;

CREATE TABLE WORK   (
        WorkID                int                   NOT NULL IDENTITY (500, 1),
        Title                 char (25)             NOT NULL,
        Copy                  char (8)              NOT NULL,
        Description           varchar (1000)        NULL DEFAULT 'Unknown provenance' ,
        ArtistID              int                   NOT NULL,

        CONSTRAINT WorkPK PRIMARY KEY (WorkID) ,
        CONSTRAINT WorkAK1 UNIQUE (Title, Copy) ,
        CONSTRAINT ArtistFK FOREIGN KEY (ArtistID) REFERENCES ARTIST (ArtistID)
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
) ;
```

ALTER Statement

- ALTER statement changes table structure, properties, or constraints after it has been created
- Example:

```
ALTER TABLE ASSIGNMENT
    ADD CONSTRAINT EmployeeFK FOREIGN KEY (EmployeeNum)
        REFERENCES EMPLOYEE (EmployeeNumber)
            ON UPDATE CASCADE
            ON DELETE NO ACTION;
```

Adding and Dropping Columns

- The following statement will add a column named MyColumn to the CUSTOMER table:

  ```
  ALTER TABLE CUSTOMER ADD MyColumn Char(5) NULL;
  ```

- You can drop an existing column with the statement:

  ```
  ALTER TABLE CUSTOMER DROP COLUMN MyColumn;
  ```

Adding and Dropping Constraints

- ALTER can be used to add a constraint as follows:

  ```
  ALTER TABLE CUSTOMER ADD CONSTRAINT MyConstraint CHECK
      ([Name] NOT IN ('Robert No Pay'));
  ```

- ALTER can be used to drop a constraint:

  ```
  ALTER TABLE CUSTOMER DROP CONSTRAINT MyConstraint;
  ```

Removing Tables

- ## SQL DROP TABLE:

  ```
  DROP TABLE [TRANSACTION];
  ```

- ## If there are constraints:

  ```
  ALTER TABLE CUSTOMER_ARTIST_INT
      DROP CONSTRAINT Customer_Artist_Int_CustomerFK;
  ALTER TABLE [TRANSACTION]
      DROP CONSTRAINT  TransactionCustomerFK;
  DROP TABLE CUSTOMER;
  ```

SQL DML - INSERT

- ## INSERT command:

    INSERT INTO ARTIST ([Name], Nationality, Birthdate,
                                DeceasedDate)
    VALUES ('Tamayo', 'Mexican', 1927, 1998);

- ## Bulk INSERT:

    INSERT INTO ARTIST ([Name], Nationality, Birthdate)
        SELECT [Name], Nationality, Birthdate
        FROM IMPORTED_ARTIST;

SQL DML: UPDATE

- ## UPDATE command:

```
UPDATE     CUSTOMER
    SET     City = 'New York City'
    WHERE   CustomerID = 1000;
```

- ## Bulk UPDATE:

```
UPDATE     CUSTOMER
    SET     AreaCode = '333'
    WHERE   City = 'Denver';
```

SQL DML: DELETE

- DELETE command:

  ```
  DELETE  FROM CUSTOMER
  WHERE   CustomerID = 1000;
  ```

- If you omit the WHERE clause, you will delete every row in the table!