```
----- Create Database ------

CREATE DATABASE Olympic_DB;
USE Olympic_DB;

------ Create Tables ------

CREATE TABLE Country (
   CountryCode CHAR(3) NOT NULL,
   CountryName VARCHAR(40) NOT NULL,
   Continent VARCHAR(20) NOT NULL,
   GDP NUMERIC(6,1),
   Population NUMERIC(10,0),
   CONSTRAINT Country_PK PRIMARY KEY (CountryCode));

CREATE NONCLUSTERED INDEX IX_Country_Continent ON Country(Continent ASC)

CREATE TABLE Olympic (
   OlympicID INTEGER IDENTITY(1,1) NOT NULL,
   City VARCHAR(25) NOT NULL,
   [Year] DATE NOT NULL,
   Season VARCHAR(10),
   CountryCode CHAR(3) NOT NULL,
   CHECK (Season in ('Summer', 'Winter')),
   CONSTRAINT Olympic_FK FOREIGN KEY(CountryCode) REFERENCES Country (CountryCode),
   CONSTRAINT Olympic_PK PRIMARY KEY (OlympicID));

CREATE NONCLUSTERED INDEX IX_Olympic_Season ON Olympic(Season ASC)

CREATE TABLE Athlete (
   AthleteID INTEGER IDENTITY(1,1) NOT NULL,
   AthleteName VARCHAR(50) NOT NULL,
   Gender VARCHAR(6) NOT NULL,
   DOB DATETIME,
   CountryCode CHAR(3) NOT NULL,
   CHECK (Gender IN ('Male', 'Female')),
   CONSTRAINT Athlete_FK FOREIGN KEY(CountryCode) REFERENCES Country (CountryCode),
   CONSTRAINT Athlete_PK PRIMARY KEY (AthleteID));

CREATE NONCLUSTERED INDEX IX_Athlete_Gender ON Athlete(Gender ASC)

CREATE NONCLUSTERED INDEX IX_Athlete_Country ON Athlete(CountryCode ASC)

CREATE TABLE [Group] (
   GroupID INTEGER IDENTITY(1,1) NOT NULL,
   GroupType VARCHAR(10) NOT NULL,
   CountryCode CHAR(3) NOT NULL,
   CHECK (GroupType in ('Individual', 'Team')),
   CONSTRAINT GROUP_FK FOREIGN KEY(CountryCode) REFERENCES Country (CountryCode),
   CONSTRAINT GROUP_PK PRIMARY KEY (GroupID));

CREATE NONCLUSTERED INDEX IX_Group_CountryCode ON [Group](CountryCode ASC)

CREATE NONCLUSTERED INDEX IX_Group_GroupType ON [Group](GroupType ASC)

CREATE TABLE Sport (
   SportID INTEGER IDENTITY(1,1) NOT NULL,
   SportName VARCHAR(25) NOT NULL,
   CONSTRAINT Sport_PK PRIMARY KEY (SportID));

CREATE TABLE [Event] (
   EventID INTEGER IDENTITY(1,1) NOT NULL,
   EventName VARCHAR(50) NOT NULL,
   EventType VARCHAR(10) NOT NULL,
   Discipline VARCHAR(25) NOT NULL,
   EventSex VARCHAR(6) NOT NULL,
   SportID INTEGER NOT NULL,
   CHECK (EventSex in ('Male', 'Female', 'Mixed')),
   CHECK (EventType in ('Individual', 'Team')),
   CONSTRAINT Event_FK FOREIGN KEY(SportID) REFERENCES Sport (SportID),
   CONSTRAINT Event_PK PRIMARY KEY (EventID));

CREATE NONCLUSTERED INDEX IX_Event_EventType ON Event(EventType ASC)

CREATE NONCLUSTERED INDEX IX_Event_EventSex ON Event(EventSex ASC)

CREATE TABLE Venue (
   VenueID INTEGER IDENTITY(1,1) NOT NULL,
   VenueName VARCHAR(40) NOT NULL,
   Capacity INTEGER,
   CONSTRAINT Venue_PK PRIMARY KEY (VenueID));

CREATE TABLE EventMatch (
   MatchID INTEGER IDENTITY(1,1) NOT NULL,
   EventPhase VARCHAR(30) NOT NULL,
   MatchDate DATE NOT NULL,
   OlympicDay VARCHAR(10),
   OlympicID INTEGER NOT NULL,
   VenueID INTEGER NOT NULL,
   EventID INTEGER NOT NULL,
   CONSTRAINT EventMatch_FK1 FOREIGN KEY(OlympicID) REFERENCES Olympic (OlympicID),
   CONSTRAINT EventMatch_FK2 FOREIGN KEY(VenueID) REFERENCES Venue (VenueID),
   CONSTRAINT EventMatch_FK3 FOREIGN KEY(EventID) REFERENCES Event (EventID),
   CONSTRAINT EventMatch_PK PRIMARY KEY (MatchID));

CREATE NONCLUSTERED INDEX IX_EventMatch_EventID ON EventMatch(EventID ASC)

CREATE NONCLUSTERED INDEX IX_EventMatch_VenueID ON EventMatch(VenueID ASC)

CREATE NONCLUSTERED INDEX IX_EventMatch_Olympic ON EventMatch(OlympicID ASC,
OlympicDay ASC)

CREATE TABLE Record (
   RecordID INTEGER IDENTITY(1,1) NOT NULL,
   RecordScore VARCHAR(30) NOT NULL,
   RecordType VARCHAR(20) NOT NULL,
   ScoreType VARCHAR(15) NOT NULL,
   MatchID INTEGER NOT NULL,
   GroupID INTEGER NOT NULL,
   Description VARCHAR(50),
   CONSTRAINT Record_FK1 FOREIGN KEY(MatchID) REFERENCES EventMatch (MatchID),
   CONSTRAINT Record_FK2 FOREIGN KEY(GroupID) REFERENCES [Group] (GroupID),
   CONSTRAINT Record_PK PRIMARY KEY (RecordID));

CREATE NONCLUSTERED INDEX IX_Record_RecordType ON Record(RecordType ASC)

CREATE TABLE AthleteOlympic (
   Height DECIMAL(3,2) NULL,
   [Weight] NUMERIC(3,0) NULL,
   AthleteID INTEGER NOT NULL,
   OlympicID INTEGER NOT NULL,
   CONSTRAINT AthleteOlympic_FK1 FOREIGN KEY(AthleteID) REFERENCES Athlete (AthleteID),
   CONSTRAINT AthleteOlympic_FK2 FOREIGN KEY(OlympicID) REFERENCES Olympic
(OlympicID),
   CONSTRAINT AthleteOlympic_PK PRIMARY KEY (OlympicID, AthleteID));

CREATE TABLE MatchResult (
   Score VARCHAR(30) NOT NULL,
   Position VARCHAR(15) NOT NULL,
   ScoreType VARCHAR(15),
   MatchID INTEGER NOT NULL,
   GroupID INTEGER NOT NULL,
   CONSTRAINT MatchResult_FK1 FOREIGN KEY(MatchID) REFERENCES EventMatch (MatchID),
   CONSTRAINT MatchResult_FK2 FOREIGN KEY(GroupID) REFERENCES [Group] (GroupID),
   CONSTRAINT MatchResult_PK PRIMARY KEY (GroupID, MatchID));

CREATE NONCLUSTERED INDEX IX_MatchResult_MatchID_Postion ON MatchResult(MatchID
ASC, Position ASC)

CREATE TABLE Medal (
   MedalID Integer IDENTITY(1,1) NOT NULL,
   MedalType VARCHAR(10) NOT NULL,
   MatchID INTEGER NOT NULL,
   GroupID INTEGER NOT NULL,
   CHECK (MedalType in ('Gold', 'Silver', 'Bronze')),
   CONSTRAINT Medal_FK1 FOREIGN KEY(MatchID) REFERENCES EventMatch (MatchID),
   CONSTRAINT Medal_FK2 FOREIGN KEY(GroupID) REFERENCES [Group] (GroupID),
   CONSTRAINT Medal_PK PRIMARY KEY (MedalID));

CREATE NONCLUSTERED INDEX IX_Medal_GroupID ON Medal(GroupID ASC)

CREATE NONCLUSTERED INDEX IX_Medal_MedalType ON Medal(MedalType ASC)

CREATE TABLE GroupAthlete (
   AthleteID INTEGER NOT NULL,
   GroupID INTEGER NOT NULL,
   CONSTRAINT GroupAthlete_FK1 FOREIGN KEY(AthleteID) REFERENCES Athlete (AthleteID),
   CONSTRAINT GroupAthlete_FK2 FOREIGN KEY(GroupID) REFERENCES [Group] (GroupID),
   CONSTRAINT GroupAthlete_PK PRIMARY KEY (AthleteID, GroupID));

-- User Defined Function to calculate Age of Athlete
GO
CREATE OR ALTER FUNCTION AGE(@DateOfBirth AS DATETIME, @TodayDate AS DATETIME)
RETURNS INT
AS

BEGIN

   DECLARE @Age  INT

   IF @DateOfBirth >= @TodayDate
      RETURN 0

   SET @Age = DATEDIFF(YEAR, @DateOfBirth, @TodayDate)

   IF MONTH(@DateOfBirth) > MONTH(@TodayDate) OR
   (MONTH(@DateOfBirth) = MONTH(@TodayDate) AND
    DAY(@DateOfBirth)   > DAY(@TodayDate))
     SET @Age = @Age - 1

 RETURN @Age

END
GO

DECLARE @ans INT
SET @ans = dbo.AGE('1998-05-21', '2016-04-21')
PRINT @ans
```