Answers to Module 3 Problems

The CREATE TABLE statement solution uses the standard SQL data types. Your DBMS may provide a different collection of data types.

1.

Oracle

```
CREATE TABLE Customer
(CustNo VARCHAR2(8) CONSTRAINT CustNoNotNull NOT NULL,
 CustName VARCHAR2(30) CONSTRAINT CustNameNotNull NOT
NULL,
 Address VARCHAR2(50) CONSTRAINT AddressNotNull NOT
NULL,
 Internal CHAR(1) CONSTRAINT InternalNotNull NOT NULL,
 Contact VARCHAR2(35) CONSTRAINT ContractNotNull NOT
NULL,
 Phone VARCHAR2(11) CONSTRAINT CPhoneNotNull NOT NULL,
 City VARCHAR2(30) CONSTRAINT CityNotNull NOT NULL,
 State VARCHAR2(2) CONSTRAINT StateNotNull NOT NULL,
 Zip VARCHAR2(10) CONSTRAINT zipNotNull NOT NULL,
     CONSTRAINT PK_CUSTOMER PRIMARY KEY (CustNo) ) ;
```

MySQL

```
   CREATE TABLE Customer
    (CustNo VARCHAR(8) NOT NULL,
     CustName VARCHAR(30) NOT NULL,
     Address VARCHAR(50) NOT NULL,
     Internal CHAR(1) NOT NULL,
     Contact VARCHAR(35) NOT NULL,
     Phone VARCHAR(11) NOT NULL,
     City VARCHAR(30) NOT NULL,
     State VARCHAR(2) NOT NULL,
     Zip VARCHAR(10) NOT NULL,
     CONSTRAINT PK_CUSTOMER PRIMARY KEY (CustNo) ) ;
```

PostgreSQL

```
CREATE TABLE Customer
(CustNo VARCHAR(8) CONSTRAINT CustNoNotNull NOT NULL,
 CustName VARCHAR(30) CONSTRAINT CustNameNotNull NOT NULL,
 Address VARCHAR(50) CONSTRAINT AddressNotNull NOT NULL,
 Internal CHAR(1) CONSTRAINT InternalNotNull NOT NULL,
 Contact VARCHAR(35) CONSTRAINT ContractNotNull NOT NULL,
 Phone VARCHAR(11) CONSTRAINT CPhoneNotNull NOT NULL,
 City VARCHAR(30) CONSTRAINT CityNotNull NOT NULL,
```

```
 State VARCHAR(2) CONSTRAINT StateNotNull NOT NULL,
 Zip VARCHAR(10) CONSTRAINT zipNotNull NOT NULL,
     CONSTRAINT PK_CUSTOMER PRIMARY KEY (CustNo) ) ;
```

2. Oracle

```
CREATE TABLE Facility
(FacNo VARCHAR2(8) CONSTRAINT FacNoNotNull NOT NULL,
 FacName VARCHAR2(30) CONSTRAINT FacNameNotNull NOT NULL,
       CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo) );
```

MySQL

```
    CREATE TABLE Facility
    (FacNo VARCHAR(8) NOT NULL,
     FacName VARCHAR(30) NOT NULL,
     CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo) );
```

PostgreSQL

```
    CREATE TABLE Facility
    (FacNo VARCHAR(8) CONSTRAINT FacNoNotNull NOT NULL,
     FacName VARCHAR(30) CONSTRAINT FacNameNotNull NOT NULL,
     CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo) );
```

3. Oracle

```
   CREATE TABLE Location
(LocNo VARCHAR2(8) CONSTRAINT LocNoNotNull NOT NULL,
 FacNo VARCHAR2(8),
 LocName VARCHAR2(30) CONSTRAINT LocNameNotNull NOT NULL,
      CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo);
```

MySQL

```
    CREATE TABLE Location
    (LocNo VARCHAR(8) NOT NULL,
     FacNo VARCHAR(8),
     LocName VARCHAR(30) NOT NULL,
     CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
```

PostgreSQL

```
CREATE TABLE Location
(LocNo VARCHAR(8) CONSTRAINT LocNoNotNull NOT NULL,
 FacNo VARCHAR(8),
 LocName VARCHAR(30) CONSTRAINT LocNameNotNull NOT NULL,
 CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo) );
```

4.
There is one 1-M relationships: Facility (FacNo PK) – Location (FacNo FK).

5.
The CREATE TABLE statement has been extended with foreign keys for *FacNo*.

Oracle

```
   CREATE TABLE Location
(LocNo VARCHAR2(8) CONSTRAINT LocNoNotNull NOT NULL,
 FacNo VARCHAR2(8),
 LocName VARCHAR2(30) CONSTRAINT LocNameNotNull NOT NULL,
      CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
      CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
          REFERENCES FACILITY (FacNo) );
```

MySQL

```
CREATE TABLE Location
(LocNo VARCHAR(8) NOT NULL,
 FacNo VARCHAR(8),
 LocName VARCHAR(30) NOT NULL,
 CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
 CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
   REFERENCES FACILITY (FacNo) );
```

PostgreSQL

```
CREATE TABLE Location
(LocNo VARCHAR(8) CONSTRAINT LocNoNotNull NOT NULL,
 FacNo VARCHAR(8),
 LocName VARCHAR(30) CONSTRAINT LocNameNotNull NOT NULL,
 CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
 CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
   REFERENCES FACILITY (FacNo) );
```

6.

Null values are not allowed for *FacNo*. The sample data shows that each facility has a related location. In addition, common practice indicates that a location requires a facility. The CREAT TABLE statement has been extended with NOT NULL constraint.

Oracle

```
   CREATE TABLE Location
(LocNo VARCHAR2(8) CONSTRAINT LocNoNotNull NOT NULL,
 FacNo VARCHAR2(8) CONSTRAINT FacNoFKNotNull NOT NULL,
 LocName VARCHAR2(30) CONSTRAINT LocNameNotNull NOT NULL,
      CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
      CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
       REFERENCES FACILITY (FacNo) );
```

MySQL

```
   CREATE TABLE Location
   (LocNo VARCHAR(8) NOT NULL,
    FacNo VARCHAR(8) NOT NULL,
    LocName VARCHAR(30) NOT NULL,
    CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
    CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
      REFERENCES FACILITY (FacNo) );
```

PostgreSQL

```
   CREATE TABLE Location
   (LocNo VARCHAR(8) NOT NULL,
    FacNo VARCHAR(8) NOT NULL,
    LocName VARCHAR(30) NOT NULL,
    CONSTRAINT PK_LOCATION PRIMARY KEY (LocNo),
    CONSTRAINT FK_FACNO FOREIGN KEY (FacNo)
      REFERENCES FACILITY (FacNo) );
```

7. Oracle

```
CREATE TABLE Facility
(FacNo VARCHAR2(8) CONSTRAINT FacNoNotNull NOT NULL,
 FacName VARCHAR2(30) CONSTRAINT FacNameNotNull NOT NULL,
      CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo),
      CONSTRAINT Unique_FacName UNIQUE(FacName) );
```

MySQL

```
   CREATE TABLE Facility
   (FacNo VARCHAR(8) NOT NULL,
    FacName VARCHAR(30) NOT NULL,
    CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo),
```

```
    CONSTRAINT Unique_FacName UNIQUE(FacName));
```

PostgreSQL

```
CREATE TABLE Facility
(FacNo VARCHAR(8) CONSTRAINT FacNoNotNull NOT NULL,
 FacName VARCHAR(30) CONSTRAINT FacNameNotNull NOT NULL,
 CONSTRAINT PK_FACILITY PRIMARY KEY (FacNo),
 CONSTRAINT Unique_FacName UNIQUE(FacName));
```