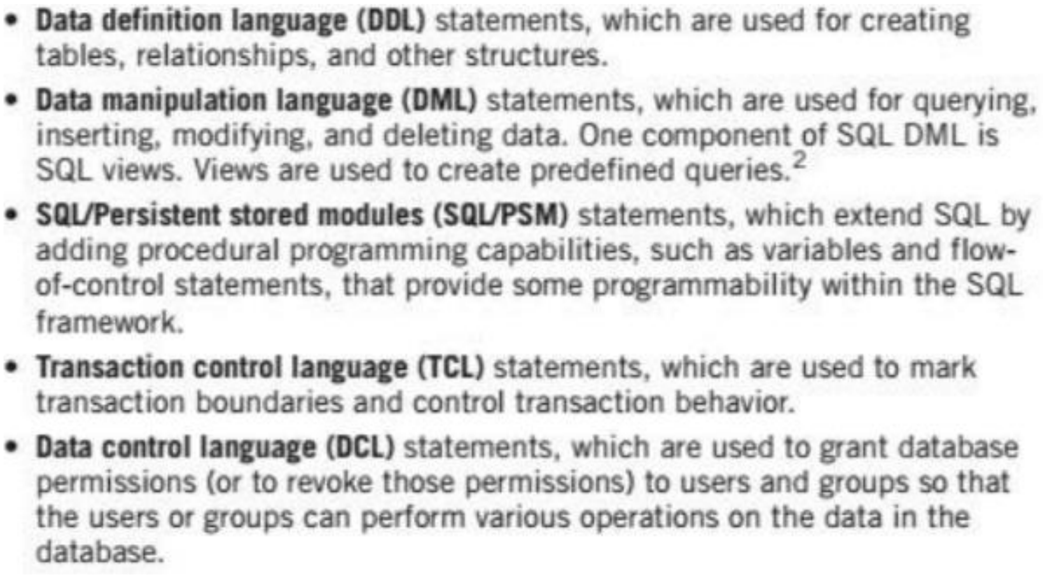
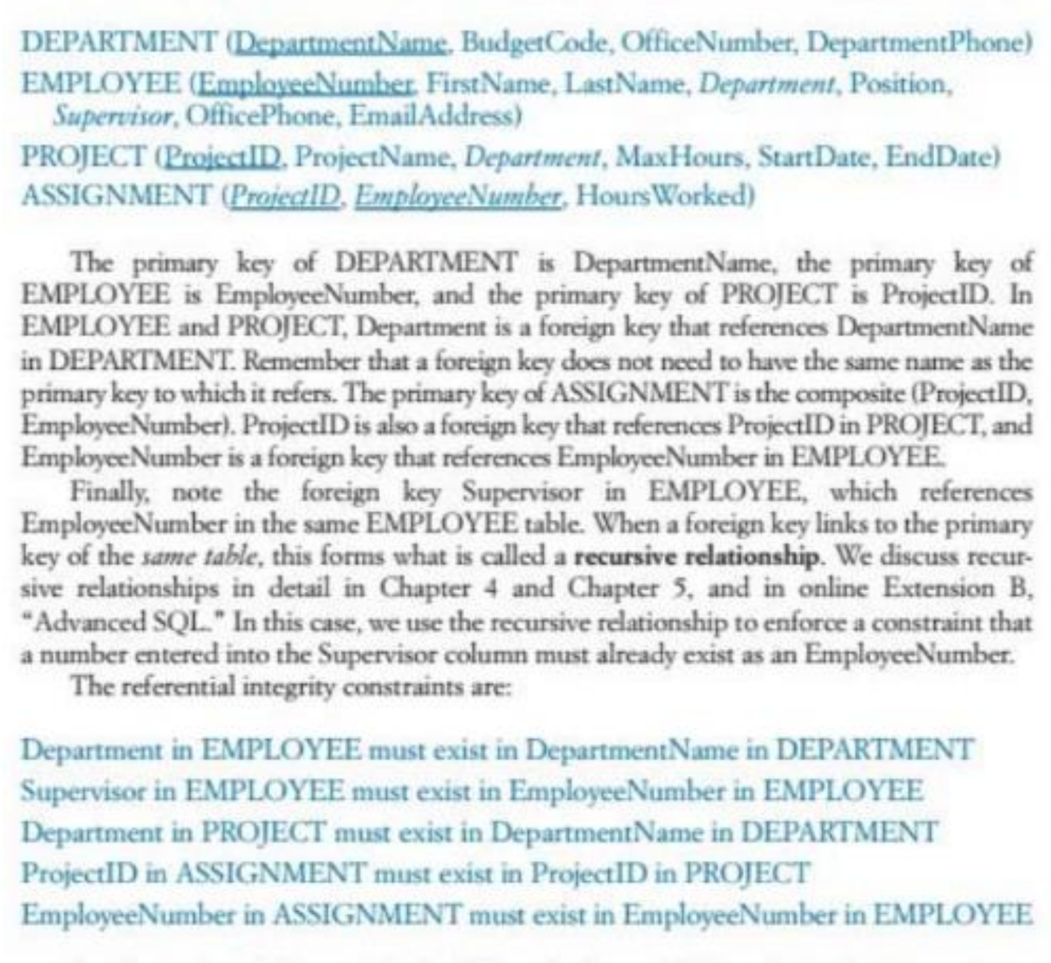
Database Concepts Notes



- **Data definition language (DDL)** statements, which are used for creating tables, relationships, and other structures.
- **Data manipulation language (DML)** statements, which are used for querying, inserting, modifying, and deleting data. One component of SQL DML is SQL views. Views are used to create predefined queries.[2]
- **SQL/Persistent stored modules (SQL/PSM)** statements, which extend SQL by adding procedural programming capabilities, such as variables and flow-of-control statements, that provide some programmability within the SQL framework.
- **Transaction control language (TCL)** statements, which are used to mark transaction boundaries and control transaction behavior.
- **Data control language (DCL)** statements, which are used to grant database permissions (or to revoke those permissions) to users and groups so that the users or groups can perform various operations on the data in the database.



DEPARTMENT (DepartmentName, BudgetCode, OfficeNumber, DepartmentPhone)
EMPLOYEE (EmployeeNumber, FirstName, LastName, *Department*, Position, *Supervisor*, OfficePhone, EmailAddress)
PROJECT (ProjectID, ProjectName, *Department*, MaxHours, StartDate, EndDate)
ASSIGNMENT (*ProjectID*, *EmployeeNumber*, HoursWorked)

The primary key of DEPARTMENT is DepartmentName, the primary key of EMPLOYEE is EmployeeNumber, and the primary key of PROJECT is ProjectID. In EMPLOYEE and PROJECT, Department is a foreign key that references DepartmentName in DEPARTMENT. Remember that a foreign key does not need to have the same name as the primary key to which it refers. The primary key of ASSIGNMENT is the composite (ProjectID, EmployeeNumber). ProjectID is also a foreign key that references ProjectID in PROJECT, and EmployeeNumber is a foreign key that references EmployeeNumber in EMPLOYEE.

Finally, note the foreign key Supervisor in EMPLOYEE, which references EmployeeNumber in the same EMPLOYEE table. When a foreign key links to the primary key of the *same table*, this forms what is called a **recursive relationship**. We discuss recursive relationships in detail in Chapter 4 and Chapter 5, and in online Extension B, "Advanced SQL." In this case, we use the recursive relationship to enforce a constraint that a number entered into the Supervisor column must already exist as an EmployeeNumber.

The referential integrity constraints are:

Department in EMPLOYEE must exist in DepartmentName in DEPARTMENT
Supervisor in EMPLOYEE must exist in EmployeeNumber in EMPLOYEE
Department in PROJECT must exist in DepartmentName in DEPARTMENT
ProjectID in ASSIGNMENT must exist in ProjectID in PROJECT
EmployeeNumber in ASSIGNMENT must exist in EmployeeNumber in EMPLOYEE

When a *foreign key (FK)* links to the primary key (PK) of the *same table* these forms what is called a **recursive relationship**. In ch3 we use the recursive relationship to enforce a constraint that a number entered the Supervisor column must already exist as an EmployeeNumber.

Business Rules:

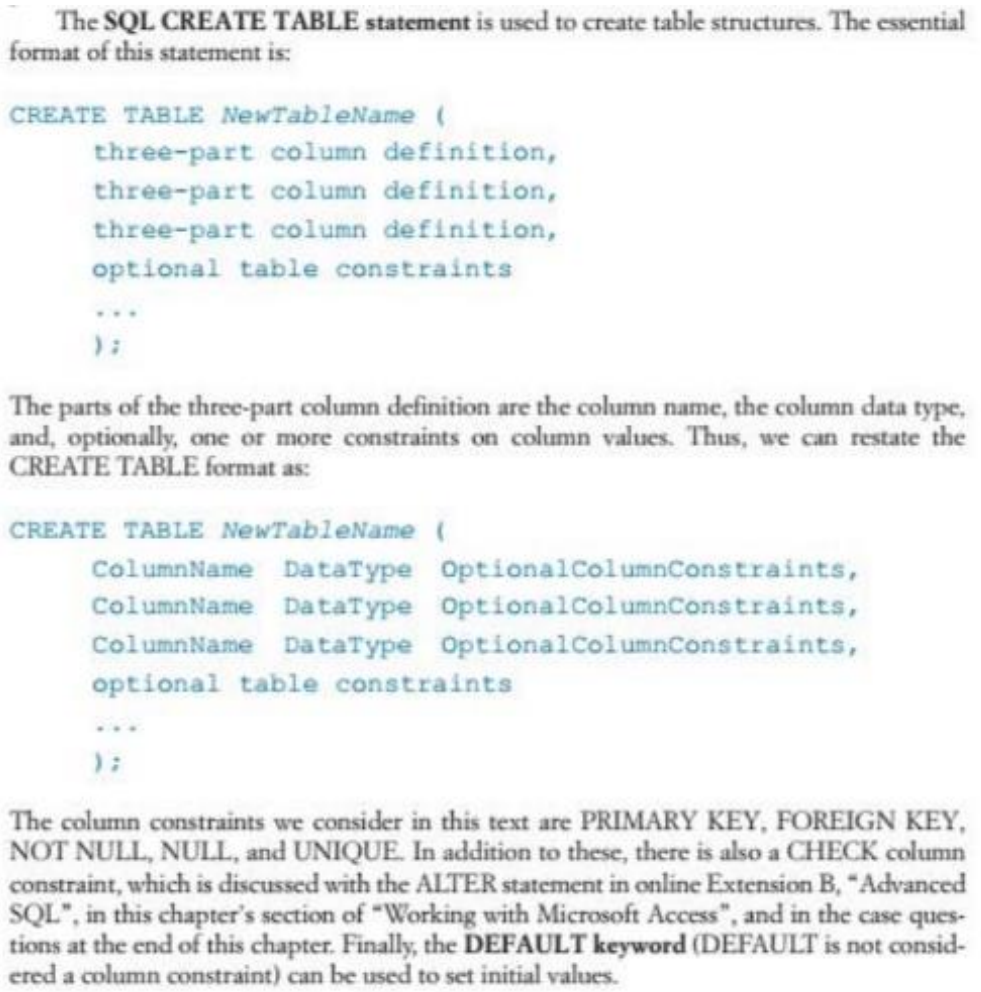
If an EMPLOYEE row is to be deleted and that row is connected to any ASSIGNMENT, the EMPLOYEE row deletion will be disallowed.

If a PROJECT row is deleted, then all the ASSIGNMENT rows that are connected to the deleted PROJECT row will also be deleted.

If an EMPLOYEE row is deleted (for ex if the employee is transferred), then someone must take over the employee's assignments, thus the application needs someone to reassign assignments before deleting the employee row.

If a PROJECT row is deleted, then the project has been canceled, and maintaining records of assignments to that project is unnecessary.

Database Concepts Notes



The **SQL CREATE TABLE statement** is used to create table structures. The essential format of this statement is:

```
CREATE TABLE NewTableName (
      three-part column definition,
      three-part column definition,
      three-part column definition,
      optional table constraints
      ...
      );
```

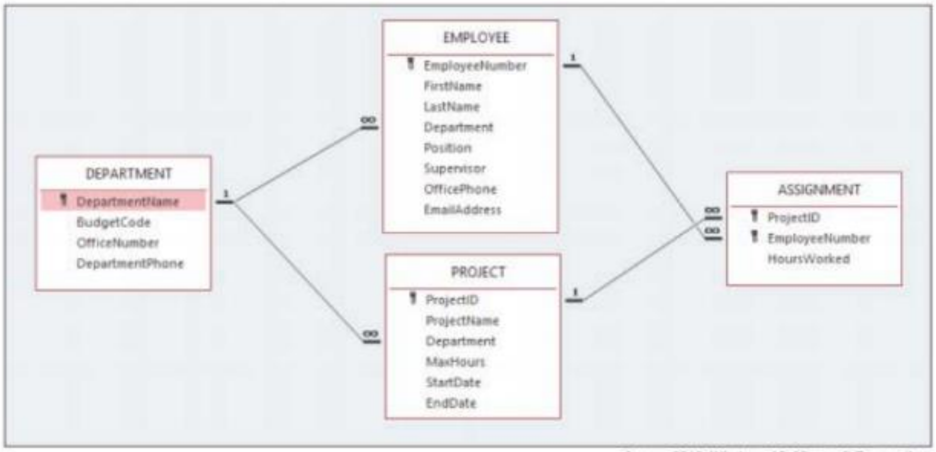
The parts of the three-part column definition are the column name, the column data type, and, optionally, one or more constraints on column values. Thus, we can restate the CREATE TABLE format as:

```
CREATE TABLE NewTableName (
      ColumnName  DataType  OptionalColumnConstraints,
      ColumnName  DataType  OptionalColumnConstraints,
      ColumnName  DataType  OptionalColumnConstraints,
      optional table constraints
      ...
      );
```

The column constraints we consider in this text are PRIMARY KEY, FOREIGN KEY, NOT NULL, NULL, and UNIQUE. In addition to these, there is also a CHECK column constraint, which is discussed with the ALTER statement in online Extension B, "Advanced SQL", in this chapter's section of "Working with Microsoft Access", and in the case questions at the end of this chapter. Finally, the **DEFAULT keyword** (DEFAULT is not considered a column constraint) can be used to set initial values.

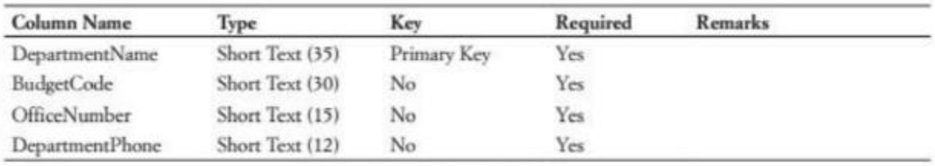
FIGURE 3-1

Database Column Characteristics for the WP Database



Access 2019, Windows 10, Microsoft Corporation.

(a) The WP Tables in Microsoft Access 2019



Column Name	Type	Key	Required	Remarks
DepartmentName	Short Text (35)	Primary Key	Yes	
BudgetCode	Short Text (30)	No	Yes	
OfficeNumber	Short Text (15)	No	Yes	
DepartmentPhone	Short Text (12)	No	Yes	

(b) DEPARTMENT Table

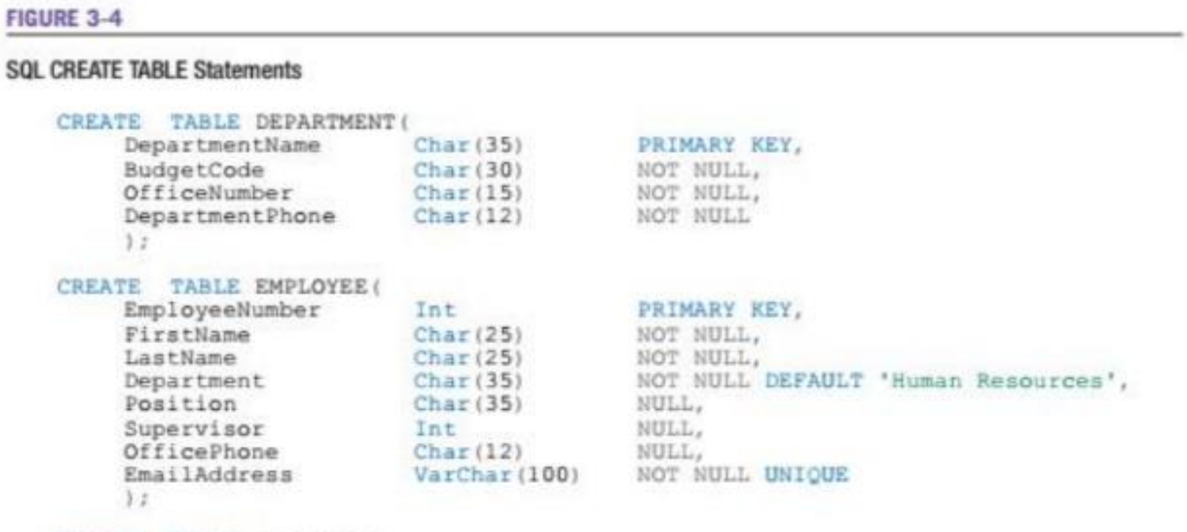


FIGURE 3-4

SQL CREATE TABLE Statements

```
CREATE  TABLE DEPARTMENT(
     DepartmentName       Char(35)        PRIMARY KEY,
     BudgetCode           Char(30)        NOT NULL,
     OfficeNumber         Char(15)        NOT NULL,
     DepartmentPhone      Char(12)        NOT NULL
     );

CREATE  TABLE EMPLOYEE(
     EmployeeNumber       Int             PRIMARY KEY,
     FirstName            Char(25)        NOT NULL,
     LastName             Char(25)        NOT NULL,
     Department           Char(35)        NOT NULL DEFAULT 'Human Resources',
     Position             Char(35)        NULL,
     Supervisor           Int             NULL,
     OfficePhone          Char(12)        NULL,
     EmailAddress         VarChar(100)    NOT NULL UNIQUE
     );
```

PROJECT table MaxHours column uses the numeric (8,2) data type. This means that the values consist of up to 8 decimal digits with 2 digits assumed to the right of the decimal point.

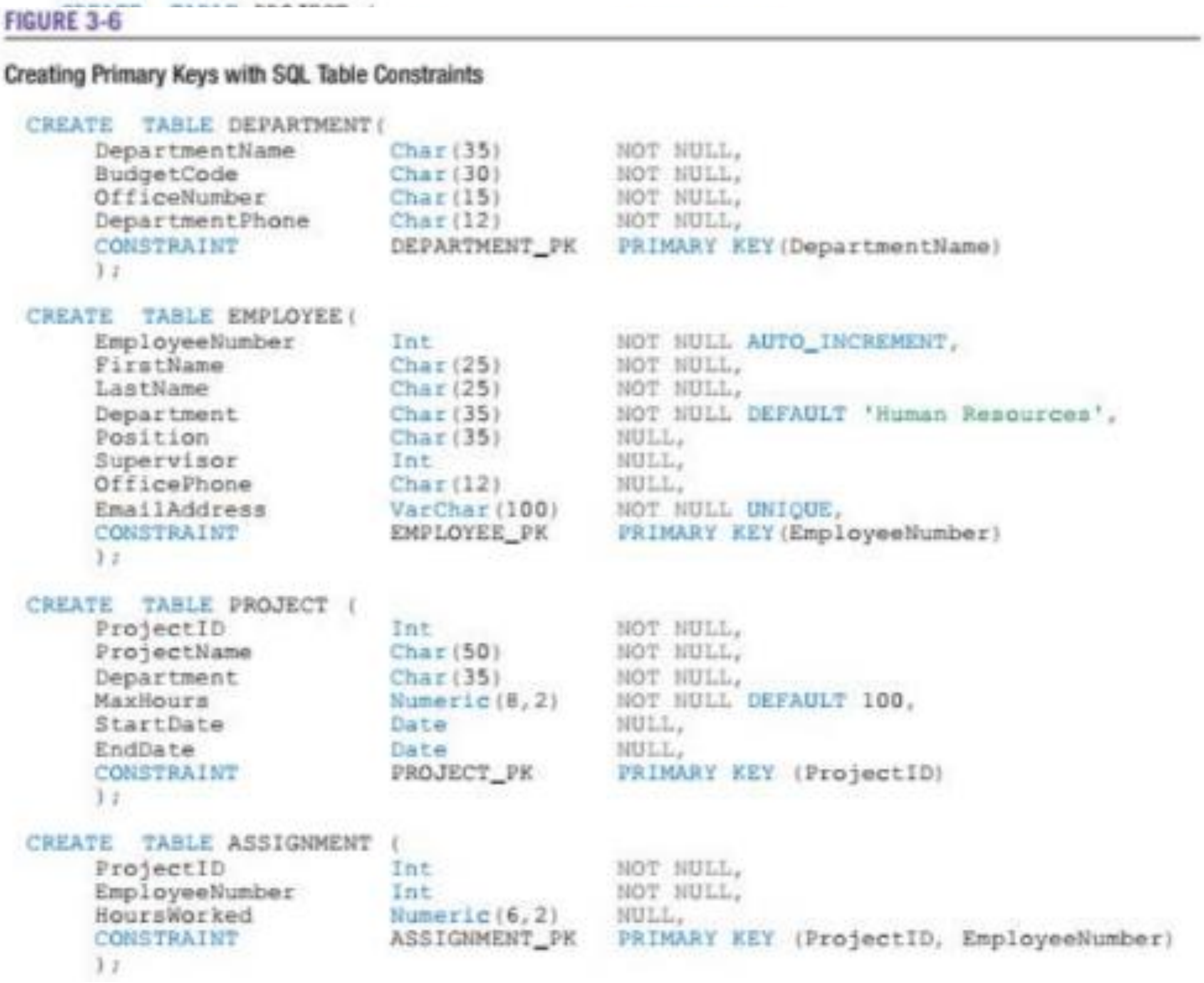


FIGURE 3-6

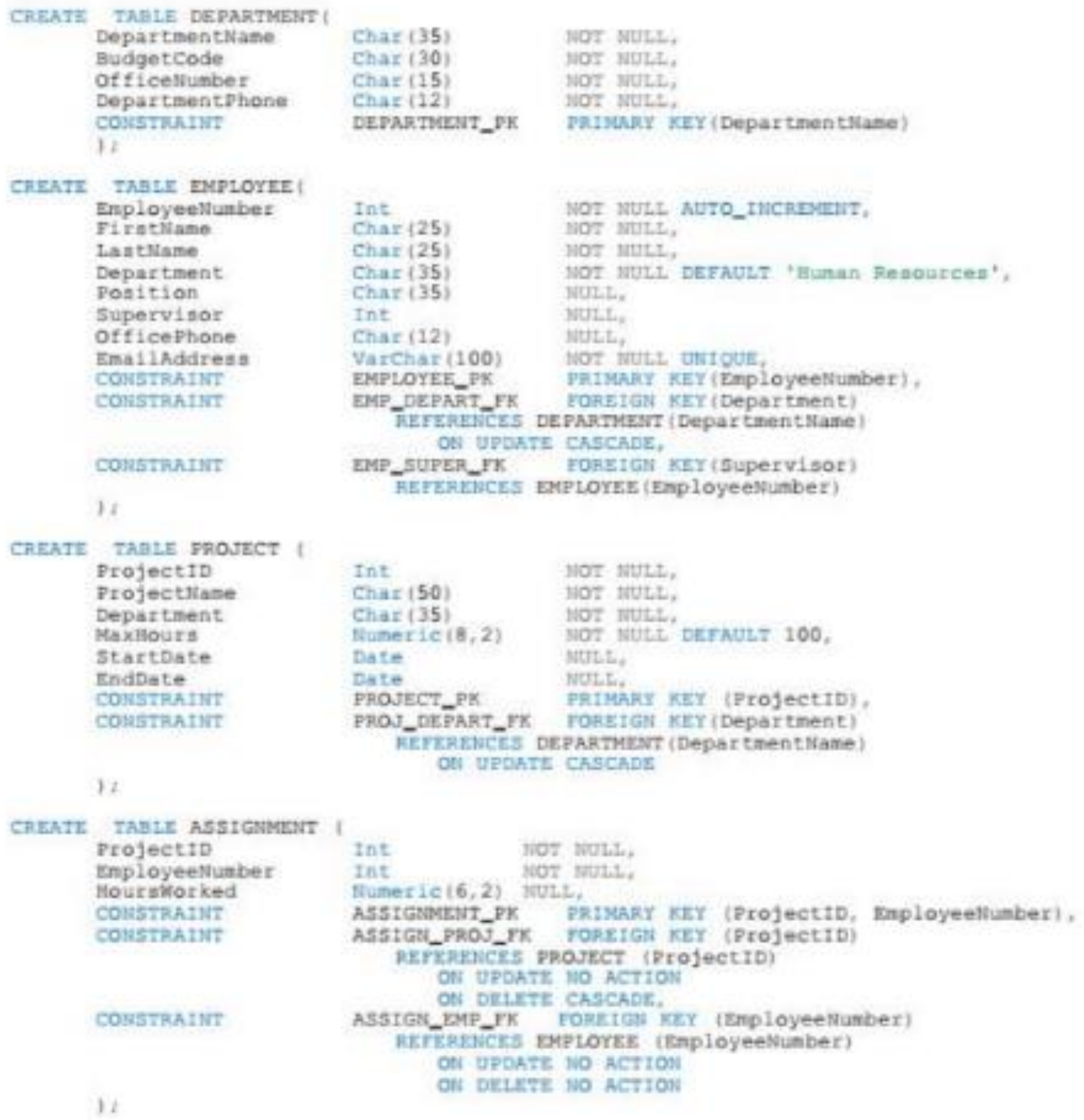
Creating Primary Keys with SQL Table Constraints

```
CREATE  TABLE DEPARTMENT(
     DepartmentName       Char(35)        NOT NULL,
     BudgetCode           Char(30)        NOT NULL,
     OfficeNumber         Char(15)        NOT NULL,
     DepartmentPhone      Char(12)        NOT NULL,
     CONSTRAINT           DEPARTMENT_PK   PRIMARY KEY(DepartmentName)
     );

CREATE  TABLE EMPLOYEE(
     EmployeeNumber       Int             NOT NULL AUTO_INCREMENT,
     FirstName            Char(25)        NOT NULL,
     LastName             Char(25)        NOT NULL,
     Department           Char(35)        NOT NULL DEFAULT 'Human Resources',
     Position             Char(35)        NULL,
     Supervisor           Int             NULL,
     OfficePhone          Char(12)        NULL,
     EmailAddress         VarChar(100)    NOT NULL UNIQUE,
     CONSTRAINT           EMPLOYEE_PK     PRIMARY KEY(EmployeeNumber)
     );

CREATE  TABLE PROJECT (
     ProjectID            Int             NOT NULL,
     ProjectName          Char(50)        NOT NULL,
     Department           Char(35)        NOT NULL,
     MaxHours             Numeric(8,2)    NOT NULL DEFAULT 100,
     StartDate            Date            NULL,
     EndDate              Date            NULL,
     CONSTRAINT           PROJECT_PK      PRIMARY KEY (ProjectID)
     );

CREATE  TABLE ASSIGNMENT (
     ProjectID            Int             NOT NULL,
     EmployeeNumber       Int             NOT NULL,
     HoursWorked          Numeric(6,2)    NULL,
     CONSTRAINT           ASSIGNMENT_PK   PRIMARY KEY (ProjectID, EmployeeNumber)
     );
```

Defining PK w/ Table Constraints – table constraints are identified by the CONSTRAINT keyword and can be used to implement various constraints.

FIGURE 3-7

Creating Foreign Keys with SQL Table Constraints



```
CREATE  TABLE DEPARTMENT(
        DepartmentName      Char(35)        NOT NULL,
        BudgetCode          Char(30)        NOT NULL,
        OfficeNumber        Char(15)        NOT NULL,
        DepartmentPhone     Char(12)        NOT NULL,
        CONSTRAINT          DEPARTMENT_PK   PRIMARY KEY(DepartmentName)
        );

CREATE  TABLE EMPLOYEE(
        EmployeeNumber      Int             NOT NULL AUTO_INCREMENT,
        FirstName           Char(25)        NOT NULL,
        LastName            Char(25)        NOT NULL,
        Department          Char(35)        NOT NULL DEFAULT 'Human Resources',
        Position            Char(35)        NULL,
        Supervisor          Int             NULL,
        OfficePhone         Char(12)        NULL,
        EmailAddress        VarChar(100)    NOT NULL UNIQUE,
        CONSTRAINT          EMPLOYEE_PK     PRIMARY KEY(EmployeeNumber),
        CONSTRAINT          EMP_DEPART_FK   FOREIGN KEY(Department)
                                REFERENCES DEPARTMENT(DepartmentName)
                                    ON UPDATE CASCADE,
        CONSTRAINT          EMP_SUPER_FK    FOREIGN KEY(Supervisor)
                                REFERENCES EMPLOYEE(EmployeeNumber)
        );

CREATE  TABLE PROJECT (
        ProjectID           Int             NOT NULL,
        ProjectName         Char(50)        NOT NULL,
        Department          Char(35)        NOT NULL,
        MaxHours            Numeric(8,2)    NOT NULL DEFAULT 100,
        StartDate           Date            NULL,
        EndDate             Date            NULL,
        CONSTRAINT          PROJECT_PK      PRIMARY KEY (ProjectID),
        CONSTRAINT          PROJ_DEPART_FK  FOREIGN KEY(Department)
                                REFERENCES DEPARTMENT(DepartmentName)
                                    ON UPDATE CASCADE
        );

CREATE  TABLE ASSIGNMENT (
        ProjectID           Int           NOT NULL,
        EmployeeNumber      Int           NOT NULL,
        HoursWorked         Numeric(6,2)  NULL,
        CONSTRAINT          ASSIGNMENT_PK   PRIMARY KEY (ProjectID, EmployeeNumber),
        CONSTRAINT          ASSIGN_PROJ_FK  FOREIGN KEY (ProjectID)
                                REFERENCES PROJECT (ProjectID)
                                    ON UPDATE NO ACTION
                                    ON DELETE CASCADE,
        CONSTRAINT          ASSIGN_EMP_FK  FOREIGN KEY (EmployeeNumber)
                                REFERENCES EMPLOYEE (EmployeeNumber)
                                    ON UPDATE NO ACTION
                                    ON DELETE NO ACTION
        );
```

Defining *FK* w/ table constraints: FOREIGN KEY constraints used to define *FK* and their associated referential integrity constraint.

ON UPDATE phrase shows what action should be taken if a value of the primary key

CASCADE means that the same changes should be made to the related Department column in EMPLOYEE.

NOT NULL – value must be supplied when creating new row/**NULL** – allows null value

DEFAULT – sets a default value if no other is given.

VarChar(**100**) -a variable length character data type like email address using characters (@), **UNIQUE constraint** for Email Address means that there cannot be any duplicated values in that column.