

Three check CONSTRAINT:

1. Employees' hire date - employees' DOB is bigger than 18 years
2. Comments in the training table should be either pass or good or excellent
3. Total cost of a project should be larger than or equal to 0

```
ALTER TABLE department
```

```
DROP CONSTRAINT department_manager_ID_FK;
```

```
DROP TABLE Assignment;
```

```
DROP TABLE Project;
```

```
DROP TABLE Client;
```

```
DROP TABLE training;
```

```
DROP TABLE employee;
```

```
DROP TABLE skill;
```

```
DROP TABLE department;
```

```
---create table skill
```

```
CREATE TABLE Skill
```

```
(Code NUMBER(4),
```

```
  Name VARCHAR2(20),
```

```
  Category VARCHAR2(20),
```

```
  CONSTRAINT skill_code_PK PRIMARY KEY(code));
```

```
---create table department
```

```
CREATE TABLE Department
(Dept_Code NUMBER(4),
Name VARCHAR2(50),
Location VARCHAR(30),
Phone NUMBER(10),
Manager_ID NUMBER(6),
CONSTRAINT department_dept_code_PK PRIMARY KEY(dept_code));
```

```
---create table employee
```

```
---check constraint: employees' hire date - employees' DOB is bigger than 18 years
```

```
CREATE TABLE Employee
(Emp_Num NUMBER(6),
LName VARCHAR(10),
FName VARCHAR(10),
DOB          DATE DEFAULT CURRENT_DATE,
Hire_Date DATE,
Super_ID NUMBER(6)
CONSTRAINT employee_super_id_FK
REFERENCES Employee(Emp_Num),
Dept_Code NUMBER(4)
CONSTRAINT employee_dept_code_FK
REFERENCES Department (dept_code),
```

```
CONSTRAINT employee_emp_num_PK PRIMARY KEY(emp_num),  
CONSTRAINT employee_DOB_Hire_CK  
    CHECK (MONTHS_BETWEEN(Hire_date,DOB)>18*12));
```

```
---create table training
```

```
---check constraint:comments in the training table should be either pass or good or excellent
```

```
CREATE TABLE Training
```

```
(Train_Num NUMBER(4),
```

```
Code NUMBER(4)
```

```
CONSTRAINT training_code_FK
```

```
    REFERENCES Skill(code),
```

```
Emp_Num NUMBER(6)
```

```
CONSTRAINT training_emp_num_FK
```

```
    REFERENCES Employee(Emp_Num),
```

```
Name VARCHAR2(40),
```

```
Date_Acquired DATE,
```

```
Comments VARCHAR2(9),
```

```
CONSTRAINT train_num_PK PRIMARY KEY(train_num),
```

```
CONSTRAINT training_Comments_CK
```

```
    Check (comments IN('Pass','Good','Excellent')));
```

---create table client

CREATE TABLE Client

(Client\_ID NUMBER(5)

CONSTRAINT client\_ID\_PK PRIMARY KEY,

Name VARCHAR2(15),

Street VARCHAR2(18),

City VARCHAR2(15),

State VARCHAR2(20),

Zip\_Code VARCHAR2(5),

Industry VARCHAR2(10),

Web\_Address VARCHAR2(50),

Phone VARCHAR2(12),

Contact\_Lname VARCHAR2(15),

Contact\_Fname VARCHAR2(10));

---create table project

---check constraint: total cost should be larger than or equal to 0

CREATE TABLE Project

(Proj\_NUMBER NUMBER(5)

CONSTRAINT Project\_Number\_PK PRIMARY KEY,

Name VARCHAR2(15),

Start\_Date DATE DEFAULT SYSDATE,

```

Total_Cost NUMBER(10) DEFAULT 0
CONSTRAINT Project_Total_Cost_ck CHECK (Total_Cost>=0) ,
Dept_Code NUMBER(6) CONSTRAINT Project_Dept_Dode_FK
    REFERENCES Department (Dept_Code),
Client_ID NUMBER(5) CONSTRAINT Project_Client_ID_FK
    REFERENCES Client (Client_ID) ,
Code NUMBER(4) CONSTRAINT Project_Code_FK
    REFERENCES Skill (Code));

```

---create table assignment

```

CREATE TABLE Assignment
(
    Assign_Num NUMBER(5)
    CONSTRAINT Assign_Num_PK PRIMARY KEY,
    Proj_number number(4) CONSTRAINT Assignment_projnum_FK
        REFERENCES Project(Proj_number),
    Emp_Num NUMBER(6) CONSTRAINT Assignment_Emp_Num_FK
        REFERENCES Employee(Emp_Num),
    Date_Assigned DATE DEFAULT SYSDATE,
    Date_Ended DATE,
    Hours_Used NUMBER(5),
    CONSTRAINT Assignment_DateAssignEnd_ck CHECK (Date_Assigned <= Date_Ended));

```

```
---add fk constraint to table department
ALTER TABLE department
ADD CONSTRAINT department_manager_ID_FK
    FOREIGN KEY (Manager_ID)
    REFERENCES Employee (Emp_Num);
```