

These are the relational models of the 5 ER diagrams that were provided to you in the previous lesson.

When creating a table, the key cannot be null.

You do not need to mention here but you need to mention it in the CREATE TABLE COMMANDS

=====Ex 1:

Professors(prof_ssn, name, age, rank, specialty)

Depts (dno, dance, office, runs (not null))
runs FK referencing Professors

// Runs <-- should not be created

Work_Dept (dno, prof_ssn)
FK dno Referencing Depts
FK prof_ssn Referencing Professors

Project (pid, sponsor, start_date, end_date, budget, manages (not null))
manages FK Referencing Professor

Graduates (grad_ssn, age, name, deg_prog, major (not null), advisor (not null))
major FK to Depts
advisor FK to Graduates

Work_In (pid, prof)
pid FK to Projects
prof FK to Professor

// We cannot enforce the participation constraint ...

// but one thing we could do (although not the best option) is to add one extra attribute

// in Project that is called works_in and is not null and FK to Professor.

// Why? And why it is not perfect?

Work_proj (pid, ssn)
pid FK to Projects
ssn FK to Graduates

// We cannot enforce the participation constraint ...

// but one thing we could do (although not the best option) is to add one extra attribute

// in Project that is called works_proj and is not null and FK to Professor.

// Why? And why it is not perfect?

Normally we would have put
Supervises (pid, ssn, ssnProf)
pid FK to Project
ssn FK to Graduate
ssnProf FK to Professor

But since we have a many to one and total participation, we better simply create an

attribute in Work_proj that is not null and is called supervises and is FK to Professors

=====> Ex 2

Departments(dno, dname, budget)

Manages(employee, department)

employee FK to Employees

department FK to Departments

Employees(ssn, salary, phone)

Works_in(employee, department)

employee FK to Employees

department FK to Departments

Child(name, age, empl)

empl FK to Employees

Alternative design:

Instead of Manages and Departments we have:

Departments(dno, dname, budget, manages)

manages FK to Employees

=====> Ex 3

Employees (ssn, unionNameNo)

Plane(regno, type NOT NULL)

type is FK to Model

Tests(name, FAANo, score)

Model(modelNo, capacity, weight)

Traffic_control(ssn, examDate)

ssn FK to Employees

Technician (ssn, salary, name, address, phoneNum)

ssn FK to Employees

Expert(model, tech)

model FK to Model

tech FK to Technician

// you cannot model the total participation or at least you could do the technician as follows:

// Technician (ssn, salary, name, address, phoneNum, expert NOT NULL)
// expert FK to Model
// ssn FK to Employees

TestInfo(test, plane, tech, hours, date and score)
test FK to Test
plane FK to Plane
tech FK to Technician

===== Ex 4

Doctor(phy_ssn, specialty, name, expYears)
// Cannot implement the total participation but only if we make some trick,
// but even that is not perfect and is not advisable.

Patient(ssn, age, name, address, physician (not null))
physician FK to Doctor

Participation(ssn, physsn, pharmCo, drug)
pharmCo FK to Pharm_co
drug FK to Drug

Drug(trade_name, formula, PharmCo)
PharmCo FK to Pharm_Co

Pharmacy(name, address, phone_num)

Sell(pharmacy, trade_name, pharm)
trade_name FK to Drug
pharm FK to Pharm_co
pharmacy FK to Pharmacy

Contact(Pharmacy, comp, start_date, end_date, text, supervisor)
pharmacy FK to Pharmacy
comp FK to Pharm_co

===== Ex 5

Musicians (ssn, name)

Instrument (InstrId, name, key)

Plays (musician, instruments)
instruments FK to Instrument
musician FK Musicians

Songs(songId, title, author, appears(not null))
appears FK Albums

Perform(song, musician)
song FK to Songs
musician FK to Musicians

Albums(albumId, copyrightDate, speed, title, producer(not null))
producer FK to Musician

Telephone (phone_no)

Place (address, home(not null))
home FK to telephone

Lives (musician, place, telephone)
musician FK to Musicians
place FK to Place
telephone FK to Telephones