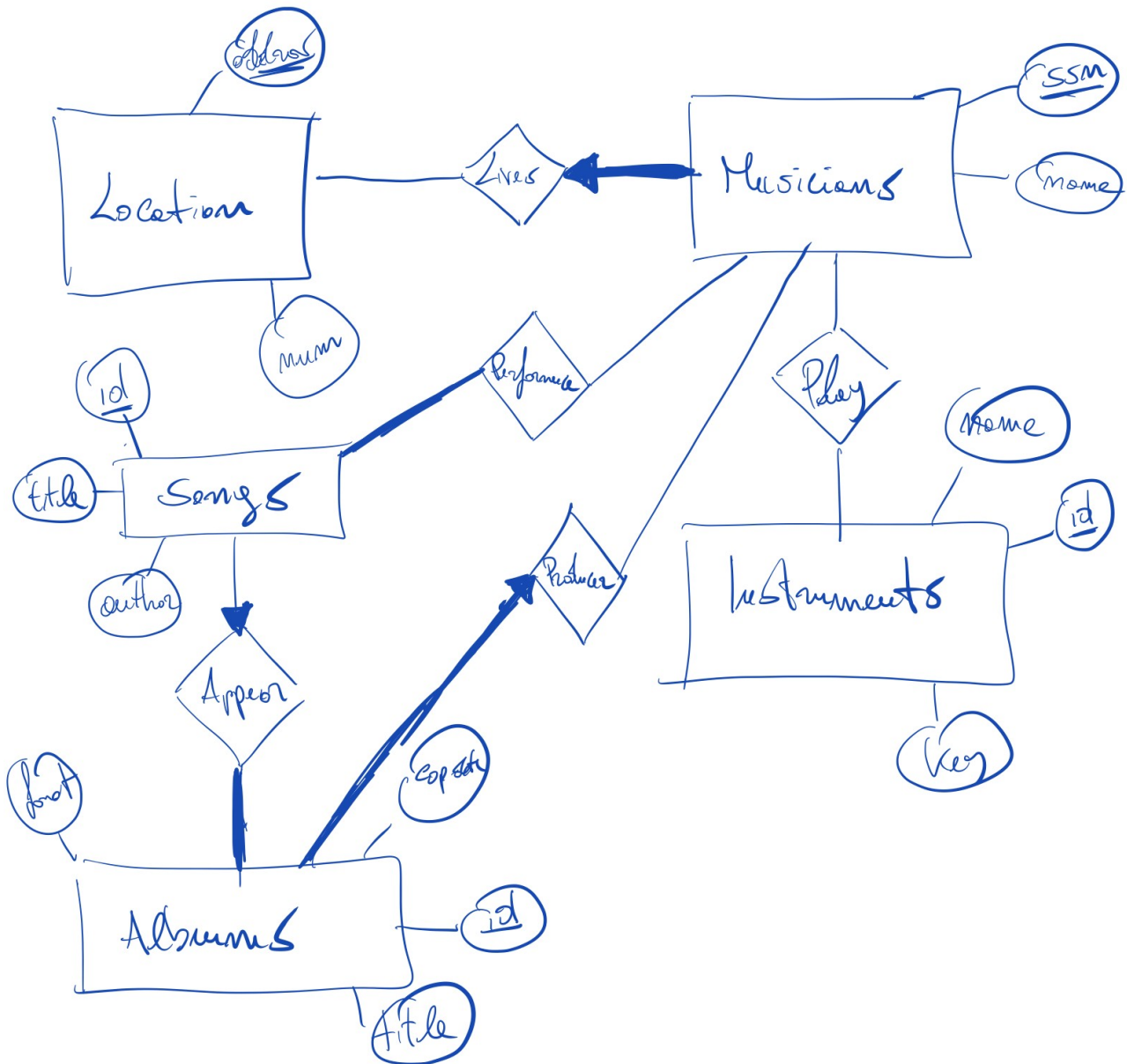# Esercizi sulla progettazione logica (ER to Relational)

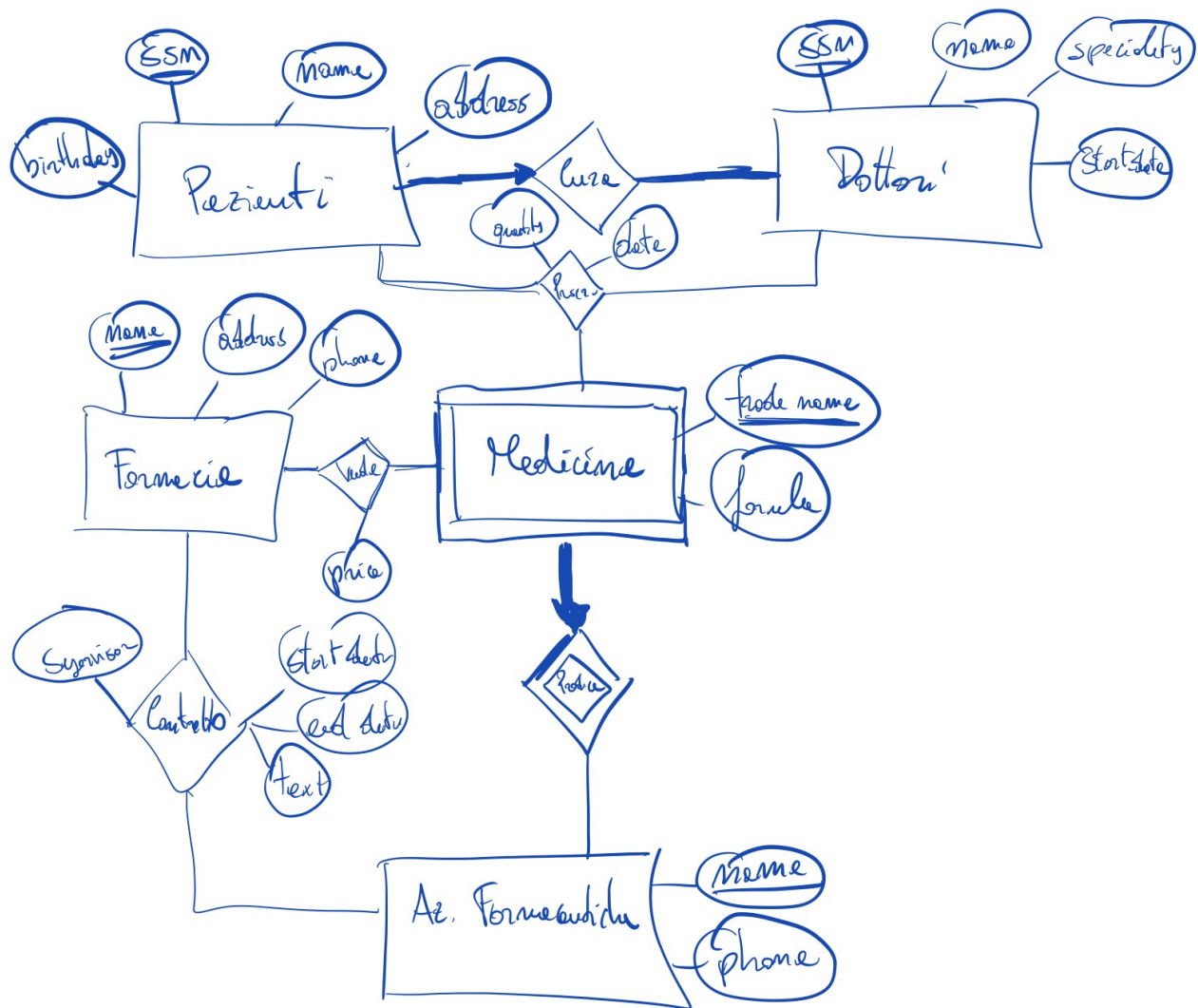Esercizio 1



| CREATE TABLE instruments (<br>    id INTEGER,<br>    name VARCHAR(20),<br>    musicalkey VARCHAR(10),<br>    PRIMARY KEY(id)); | CREATE TABLE location (<br>    address VARCHAR(50),<br>    num VARCHAR(20),<br>    PRIMARY KEY(address)); |
|---|---|
| CREATE TABLE musicians (<br>    ssn CHAR(11),<br>    name VARCHAR(20),<br>    locaddr VARCHAR(50) NOT NULL,<br>    PRIMARY KEY(ssn),<br>    FOREIGN KEY(locaddr)<br>    REFERENCES location); | CREATE TABLE play (<br>    iid INTEGER,<br>    ssn CHAR(11),<br>    PRIMARY KEY(iid,ssn),<br>    FOREIGN KEY(iid)<br>    REFERENCES instruments,<br>    FOREIGN KEY(ssn)<br>    REFERENCES musicians); |

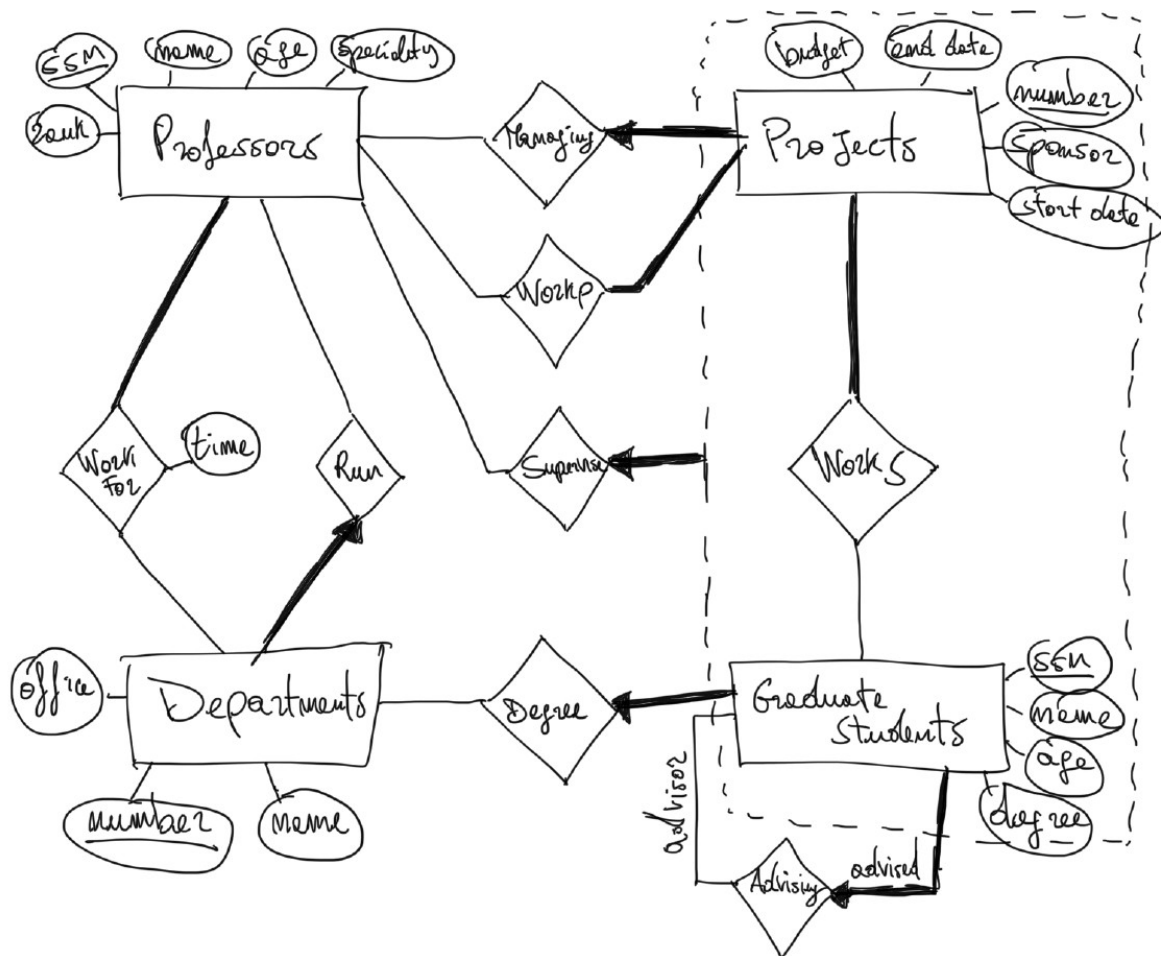| | |
|---|---|
| CREATE TABLE albums (<br>    id INTEGER,<br>    title VARCHAR(20),<br>    copyrightdate DATE,<br>    format AS ENUM ('Vinyl', 'CD', 'Digital'),<br>    producer_ssn CHAR(11) NOT NULL,<br>    PRIMARY KEY(id),<br>    FOREIGN KEY(producer_ssn)<br>    REFERENCES musicians); | CREATE TABLE performance (<br>    song_id INTEGER,<br>    mus_ssn CHAR(11),<br>    PRIMARY KEY(song_id, mus_ssn)); |
| CREATE TABLE songs (<br>    id INTEGER,<br>    title VARCHAR(20),<br>    author VARCHAR(20),<br>    album_id INTEGER,<br>    PRIMARY KEY(id),<br>    FOREIGN KEY(album_id)<br>    REFERENCES albums); | CREATE TABLE appear (<br>    song_id INTEGER,<br>    album_id INTEGER,<br>    PRIMARY KEY(song_id)); |

Esercizio 2



| | |
|---|---|
| CREATE TABLE az_farm (<br>    name VARCHAR(20),<br>    phone VARCHAR(20),<br>    PRIMARY KEY(name)); | CREATE TABLE medicine (<br>    tradename VARCHAR(20),<br>    formula VARCHAR(50),<br>    az_farm_name VARCHAR(20) NOT NULL,<br>    PRIMARY KEY(tradename,az_farm_name),<br>    FOREIGN KEY(az_farm_name)<br>    REFERENCES az_farm ON DELETE CASCADE); |
| CREATE TABLE farmacie (<br>    name VARCHAR(20),<br>    address VARCHAR(50),<br>    phone VARCHAR(2),<br>    PRIMARY KEY(name)); | CREATE TABLE contratti (<br>    farm_name VARCHAR(20),<br>    az_farm_name VARCHAR(20),<br>    supervisor VARCHAR(20),<br>    start_date DATE,<br>    end_date DATE,<br>    text TEXT,<br>    PRIMARY KEY(farm_name,az_farm_name),<br>    FOREIGN KEY(farm_name)<br>    REFERENCES farmacie,<br>    FOREIGN KEY(az_farm_name)<br>    REFERENCES az_farm); |

| | |
|---|---|
| CREATE TABLE vendite (<br>    farm_name VARCHAR(20),<br>    tradename VARCHAR(20),<br>    az_farm_name VARCHAR(20),<br>    price INTEGER,<br>    PRIMARY KEY(farm_name,<br>                tradename,<br>                az_farm_name),<br>    FOREIGN KEY(farm_name)<br>    REFERENCES farmacie,<br>    FOREIGN KEY(tradename, az_farm_name)<br>    REFERENCES medicine); | CREATE TABLE dottori (<br>    ssn CHAR(11),<br>    name VARCHAR(20),<br>    speciality VARCHAR(50),<br>    start_date DATE,<br>    PRIMARY KEY(ssn)); |
| CREATE TABLE pazienti (<br>    ssn CHAR(11),<br>    name VARCHAR(20),<br>    address VARCHAR(50),<br>    birthday DATE,<br>    doctor_ssn CHAR(11) NOT NULL,<br>    PRIMARY KEY(ssn),<br>    FOREIGN KEY(doctor_ssn)<br>    REFERENCES dottori); | CREATE TABLE prescrizioni (<br>    paz_ssn CHAR(11),<br>    doctor_ssn CHAR(11),<br>    tradename VARCHAR(20),<br>    az_farm_name VARCHAR(20),<br>    quantity INTEGER,<br>    prescr_date DATE,<br>    PRIMARY KEY(paz_ssn,doctor_ssn,tradename,az_farm_name),<br>    FOREIGN KEY(paz_ssn)<br>    REFERENCES pazienti,<br>    FOREIGN KEY(doctor_ssn)<br>    REFERENCES dottori,<br>    FOREIGN KEY(tradename,az_farm_name)<br>    REFERENCES medicine); |

## Esercizio 3



| CREATE TABLE professors ( | CREATE TABLE departments ( |
|---|---|
| ssn CHAR(11), <br> name VARCHAR(20), <br> rank VARCHAR(20), <br> age INTEGER, <br> speciality VARCHAR(50), <br> PRIMARY KEY(ssn) ); | number INTEGER, <br> name VARCHAR(20), <br> office VARCHAR(20), <br> prof_ssn CHAR(11) NOT NULL, <br> PRIMARY KEY(number), <br> FOREIGN KEY(prof_ssn) REFERENCES professors ); |
| CREATE TABLE projects ( <br> number INTEGER, <br> start_date DATE, <br> end_date DATE, <br> sponsor VARCHAR(20), <br> budget INTEGER, <br> prof_manager_ssn CHAR(11) NOT NULL, <br> PRIMARY KEY(number), <br> FOREIGN KEY(prof_manager_ssn) <br> REFERENCES professors ); | CREATE TABLE gradstudents ( <br> ssn CHAR(11), <br> name VARCHAR(20), <br> age INTEGER, <br> degree VARCHAR(50), <br> dep_num INTEGER NOT NULL, <br> advisor_ssn CHAR(11) NOT NULL, <br> PRIMARY KEY(ssn), <br> FOREIGN KEY(dep_num) REFERENCES departments, <br> FOREIGN KEY(advisor_ssn) REFERENCES gradstudents ); |
| CREATE TABLE work_for ( <br> prof_ssn CHAR(11), <br> dep_num INTEGER, | CREATE TABLE work_p ( <br> prof_ssn CHAR(11), <br> proj_num INTEGER, |

| | |
|---|---|
|    time INTEGER,<br>   PRIMARY KEY(prof_ssn,dep_num),<br>   FOREIGN KEY(prof_ssn) REFERENCES professors,<br>   FOREIGN KEY(dep_num) REFERENCES departments ); | PRIMARY KEY(prof_ssn, proj_num) ); |
| CREATE TABLE work_s (<br>   proj_num INTEGER,<br>   stud_ssn CHAR(11),<br>   prof_supervisor_ssn CHAR(11) NOT NULL,<br>   PRIMARY KEY(proj_num,stud_ssn),<br>   FOREIGN KEY(prof_supervisor_ssn)<br>   REFERENCES professors ); | |

Esercizio 4
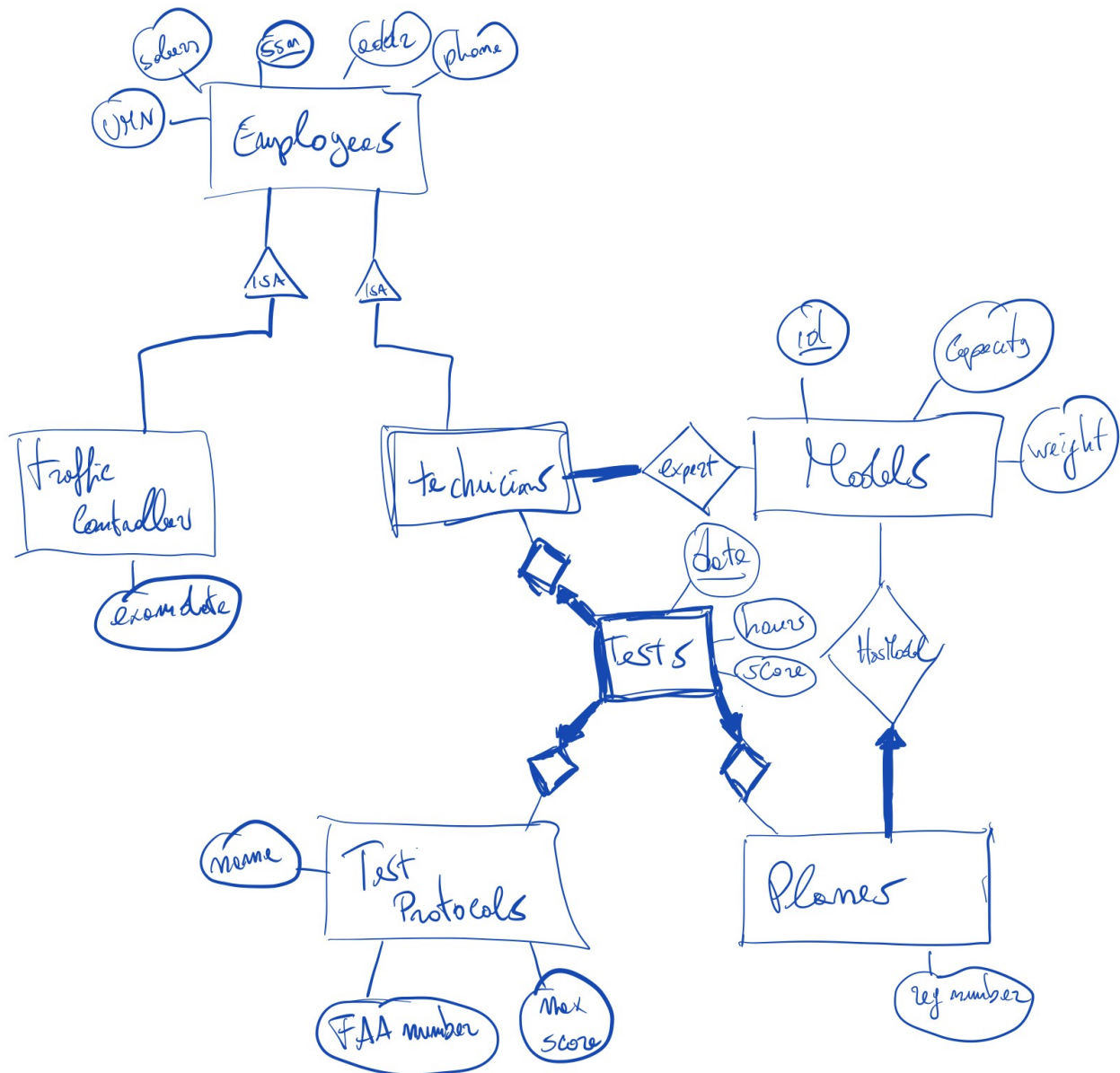


La IS-A usata qui, non avendo ulteriori chiarimenti, è senza COVER e senza OVERLAP (quindi possiamo avere dipendenti che non sono né traffic controller, né technicians, e in più non possiamo avere un dipendente che è sia traffic controller che technician. Quindi, si potrebbe adoperare la soluzione con una sola tabella che raccoglie tutti i dipendenti e con una enum distinguiamo la tipologia. Tuttavia, questa soluzione non ci permetterebbe di mettere in relazione i soli technicians con i Models o i Tests (visto che il diagramma ER lo richiede). Quindi, si opta per la soluzione base, utilizzando una tabella per entity set (in questo modo possiamo avere OVERLAP, ma meglio questo che non poter connettere technicians con Models e Tests).

| CREATE TABLE employees ( | CREATE TABLE traffic_controllers ( |
|---|---|
| ssn CHAR(11), | ssn CHAR(11), |
| address VARCHAR(30), | exam_date DATE, |
| phone VARCHAR(20), | PRIMARY KEY(ssn), |

| | |
|---|---|
| salary INTEGER,<br>umn VARCHAR(10),<br>PRIMARY KEY(ssn) ); | FOREIGN KEY(ssn)<br>REFERENCES employees<br>ON DELETE CASCADE ); |
| CREATE TABLE technicians (<br>  ssn CHAR(11),<br>  PRIMARY KEY(ssn),<br>  FOREIGN KEY(ssn)<br>  REFERENCES employees<br>  ON DELETE CASCADE ); | CREATE TABLE models (<br>  id INTEGER,<br>  capacity INTEGER,<br>  weight INTEGER,<br>  PRIMARY KEY(id) ); |
| CREATE TABLE planes (<br>  reg_number INTEGER,<br>  model_id INTEGER NOT NULL,<br>  PRIMARY KEY(reg_number),<br>  FOREIGN KEY(model_id)<br>  REFERENCES models ); | CREATE TABLE test_protocols (<br>  ffa_num INTEGER,<br>  name VARCHAR(50),<br>  max_score INTEGER,<br>  PRIMARY KEY(ffa_num) ); |
| CREATE TABLE tests (<br>  tech_ssn CHAR(11),<br>  test_prot_num INTEGER,<br>  plane_num INTEGER,<br>  date DATE,<br>  hours INTEGER,<br>  score INTEGER,<br>  PRIMARY KEY(tech_ssn,test_prot_num,plane_num,date),<br>  FOREIGN KEY(tech_ssn)<br>  REFERENCES technicians<br>  ON DELETE CASCADE,<br>  FOREIGN KEY(test_prot_num)<br>  REFERENCES test_protocols<br>  ON DELETE CASCADE,<br>  FOREIGN KEY(plane_num)<br>  REFERENCES planes<br>  ON DELETE CASCADE ); | |