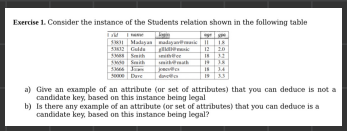
Homework 2.

[redacted]

03/07/2022

1)



Exercise 1. Consider the instance of the Students relation shown in the following table

r.cid	name	login	age	gpa
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53666	Smith	smith@cs	18	3.2
53650	Smith	smith@math	19	3.8
53666	Jones	jones@cs	18	3.4
53688	Dave	dave@cs	19	3.5

a) Give an example of an attribute (or set of attributes) that you can deduce is not a candidate key, based on this instance being legal

b) Is there any example of an attribute (or set of attributes) that you can deduce is a candidate key, based on this instance being legal?

1a) Non-candidate keys can be name or age. Based on the data that is given, we cannot tell if gpa can be a non-candidate key even though technically more than one student could have the same gpa.

1b) We cannot make predictions on the instances based only on what we have been given. This is because the instance that has been shown for this problem is just one possibility of the relation because at some other time there could be an instance that contains different tuples completely.

Exercise 2. Consider the following relations:

Students(*sid:* string, *name:* string, *login:* string, *age:* integer, *gpa:* real)
Faculty(*fid:* string, *fname:* string, *sal:* real)
Courses(*cid:* string, *cname:* string, *credits:* integer)
Rooms(*rno:* integer, *address:* string, *capacity:* integer)
Enrolled(*sid:* string, *cid:* string, *grade:* string)
Teaches(*fid:* string, *cid:* string)
Meets_In(*cid:* string, rno: integer, *time:* string)

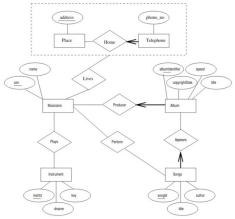
a) List all the foreign key constraints among these relations.
b) Give an example of a (plausible) constraint involving one or more of these relations that is not a primary key or foreign key constraint.

2a) Courses - cid
Faculty - fid
Rooms - rno
Students - sid

2b) Domain constraints : example → cid because it has a string domain constraint.

3)

Exercise 3. Given the following ER diagram (see exercise 1 of homework 1)



Show the SQL statements for creating relations corresponding to the entity sets and relationships. Identify any constraints in the ER diagram that you are unable to capture in the SQL statements and briefly explain why you could not express them.

constraints:

CREATE TABLE Musicians (ssn CHAR(10), name CHAR(30), PRIMARY KEY(ssn));

CREATE TABLE Instruments (instrId CHAR(10), dname CHAR(30), key CHAR(5), PRIMARY KEY (instrId));

CREATE TABLE Plays (ssn CHAR(10), instrId INTEGER, PRIMARY KEY (ssn, instrId), FOREIGN KEY(ssn) REFERENCES Musicians, FOREIGN KEY (instrId) REFERENCES Instrument);

CREATE TABLE Songs-Appears (songID INTEGER, author CHAR(30), title CHAR(30), albumIdentifier INTEGER NOT NULL, PRIMARY KEY (phone), FOREIGN KEY(address) REFERENCES Place);

CREATE TABLE Lives (ssn CHAR(10), phone CHAR(11), address CHAR(30), PRIMARY KEY(ssn, address), FOREIGN KEY(phone, address) References TelephoneHome, FOREIGN KEY (ssn) REFERENCES Musicians);

CREATE TABLE Place (address CHAR(30), PRIMARY KEY(address));

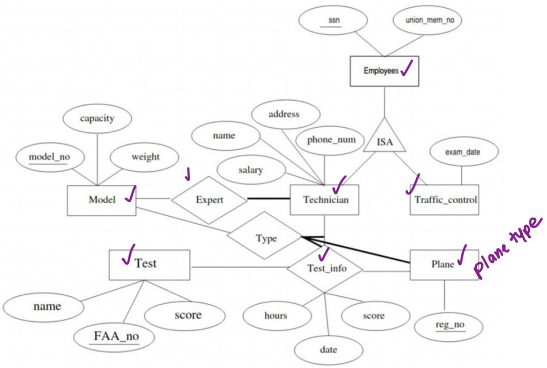
CREATE TABLE Perform (songId INTEGER, ssn (CHAR(10), PRIMARY KEY (ssn, songId), FOREIGN KEY (songId) REFERENCES Songs, FOREIGN KEY (ssn) REFERENCES Musicians);

CREATE TABLE Album Producer (albumIdentifier INTEGER, ssn CHAR(10), copyrightDate DATE, speed INTEGER, title CHAR(30), PRIMARY KEY (albumIdentifier), FOREIGN KEY (ssn) REFERENCES Musicians);

not NULL!

4)

Exercise 4. Given the following ER diagram (see exercise 2 of homework 1)



Translate it into a relational schema. If there are constraints that cannot be captured by your translation, explain why.

→ CREATE TABLE Expert(ssn CHAR(11), model_no INTEGER, PRIMARYKEY(ssn, model_no), FOREIGN KEY(ssn) REFERENCES Technician, FOREIGN KEY (model_no) REFERENCES Models);

→ CREATE TABLE Model (model_no INTEGER, capacity INTEGER, weight INTEGER, PRIMARY KEY (model_no));

→ CREATE TABLE Employees (ssn CHAR(11), union_mem_no INTEGER, PRIMARY KEY(ssn));

→ CREATE TABLE Technician (ssn CHAR(11), name CHAR(20), address CHAR(20), phone_no CHAR(14), PRIMARY KEY (ssn), FOREIGN KEY (ssn) REFERENCES Employees ON DELETE CASCADE);

→ CREATE TABLE traffic_control (ssn CHAR(11), exam_date DATE, PRIMARY KEY (ssn), FOREIGN KEY (ssn) REFERENCES Employees ON DELETE CASCADE);

→ CREATE TABLE Plane_Type (Reg_no INTEGER, Model_no INTEGER, Primary KEY (reg_no), FOREIGN KEY(model_no) REFERENCES Models);

→ CREATE TABLE Test_info (FAA_no INTEGER, ssn CHAR(11), reg_no INTEGER, hours INTEGER, date DATE, score INTEGER, PRIMARY KEY (ssn, reg_no, FAA_no), FOREIGN KEY(reg_no) REFERENCES Plane_Type, FOREIGN KEY(FAA_no) REFERENCES Tesr, FOREIGN KEY (ssn) REFERENCES Employees);

→ CREATE TABLE Test (FAA_no INTEGER, name CHAR(10), max_score INTEGER, hours INTEGER, date DATE, score INTEGER, PRIMARY KEY(FAA_no);

For this, I accidentally did it using CREATE TABLE like in problem 3, but I redid it in terms of a relational schema and have attached it below.

Question 4: ACTUAL ANSWER:

Expert (ssn CHAR(11),
model no INTEGER,
PRIMARY KEY (SSN, model_ no)
FOREIGN KEY (SSN) REFERENCES Technician,
FOREIGN KEY (model_no) REFERENCES Models)

Models (model_no INTEGER,
capacity INTEGER, Weight INTEGER,
PRIMARY KEY (model_no))

Employees (ssn CHAR (11),
union_mem_no INTEGER,
PRIMARY KEY (ssn))

Technician_Employees (ssn CHAR (11),
name CHAR(20),
address CHAR (20),
phone_no CHAR (14),
PRIMARY KEY (ssn),
FOREIGN KEY (ssn) REFERENCES Employees)

Traffic_control_Employees (ssn CHAR (11),
exam_date DATE,
PRIMARY KEY (ssn),
FOREIGN KEY (ssn) REFERENCES Employees)

Plane_Type (reg_no INTEGER,
model_no INTEGER,
PRIMARY KEY (reg_no),
FOREIGN KEY (model_no) REFERENCES Models)

Test_info(FAA_ no INTEGER,
ssn CHAR (I1),
reg_no INTEGER,
hours INTEGER,
date DATE,
Score INTEGER,
PRIMARY KEY (ssn, reg_no, FFA_no),
FOREIGN KEY (reg_no) REFERENCES Plane_Type,
FOREIGN KEY (FAA_no) REFERENCES Test,
FOREIGN KEY (ssn) REFERENCES Employees)