Dov Kassai dk778

3.2) There would be 22 tuples. If the cardinality is 22 that means there must be 22 distinct tuples.

3.4) A primary key is something that is selected from the candidate keys to uniquely identify the tuple. A super key is where a key is contained in a set of attributes.

3.5) 1: An example of a non-candidate key is {name}.

2: You would need to have more than one instance for it define if it is a candidate key. If that is the only instance though you would be able to define it as the candidate key.

3.6) A foreign key constraint is something that would prevent actions that would ruin the data between the two tables. It makes sure that the data stays consistent. They are important because they provide a level of security for keeping the integrity of the data. It lets the user know when they are doing something that is not allowed, or makes sense. It also helps minimize the error across the application.

Referential integrity means that all foreign key constraints are enforced.

3.7) 1: You would not need to have a foreign key constraint on Students, Faculty, Courses, or Room relations. They are basic relationships and have to be free standing. When entering the data you just need to be careful.

2: In the enrollment relation, SID and CID both have foreign key constraints on them. This means that there must be students in the courses. Also FID and CID must also have foreign key constraints because there needs to be a teacher for the course being taught. For a relationship Meets (for where the class is) should have a foreign key constraint of the CID and the ROOM#.

3.12) CREATE TABLE Teaches (ssn CHAR(10), CourseID INTEGER, Semester Char(10), Primary Key (ssn, CourseID, semester), Foreign Key(ssn) REFERENCES Professor, Foreign Key (CourseID) REFERENCES Course, Foreign Key (Semester) REFERENCES Semester)

The table corresponding to Course is Created by:

CREATE TABLE Course (CourseID INTEGER, Primary Key (CourseID))

CREATE TABLE Teaches (ssn CHAR(10), CourseID INTEGER, Semester CHAR(10), PRIMARY KEY (ssn, CourseID), FOREIGN KEY (ssn) REFERENCES Professor, FOREIGN KEY (CourseID) REFERENCES Course))

Table for professor and semester is the same as course from before

The tables created in the first part of this questions are the best we can do without using check constraints. The participation constraint cannot be captured using only primary and foreign key constraints because we can make sure that every entry in professor has an entry in teaches.

CREATE TABLE Professor_teaches (ssn CHAR(10), CourseID INTEGER, Semester CHAR(10), PRIMARY KEY (ssn), FOREIGN KEY (CourseID) REFERENCES Course))

CREATE TABLE Course (CourseID INTEGER, PRIMARY KEY (CourseID))

we don't need a separate table for professor

We also do not need a table for course because A) every course must be taught B) the only attribute for course is CourseID, which is held in the Professor_teaches table. If Course ha other attributes we might need a separate table for course, and would not be able to enforce the constraint that every course should be taught by some professor.

CREATE TABLE Teaches (gid INTEGER, CourseID INTEGER, semester CHAR(10), PRIMARY KEY(gid, CourseID) , FOREIGN KEY (gid) REFERENCES Group, FOREIGN KEY (CourseID) REFERENCES Course))

CREATE TABLE Memberof(ssn CHAR(10), gid INTEGER, PRIMARY KEY (ssn, gid), FOREIGN KEY (ssn) REFERENCES Professor, FOREIGN KEY(gid) REFERENCES Group)

CREATE TABLE Course (CourseID INTEGER, PRIMARY KEY (CourseID))

CREATE TABLE Group (gid INTEGER, PRIMARY KEY (gid))

CREATE TABLE Professor (ssn CHAR(10), PRIMARY KEY (ssn))

3.15) 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 Instruments)

CREATE TABLE Songs Appears (songId INTEGER, author CHAR(30), title CHAR(30), albumIdentifier INTEGER NOT NULL, PRIMARY KEY (songId), FOREIGN KEY (albumIdentifier) REFERENCES Album Producer)

CREATE TABLE Telephone Home (phone CHAR(11), address CHAR(30), 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 Telephone Home, FOREIGN KEY (ssn) REFERENCES Musicians)

CREATE TABLE Place (address CHAR(30))

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)

3.16) CREATE TABLE Expert (ssn CHAR(11), model_no INTEGER, PRIMARY KEY (ssn, model_no), FOREIGN KEY (ssn) REFERENCES Technician, FOREIGN KEY (model_no) REFERENCES Models)

The participation constraint cannot be captured in the table.

CREATE TABLE Models (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_emp (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_emp (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 Test, 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))

The constraint that tests on a plane must be conducted by a techinicain who is an expert on that specific model and can be expressed as follows in sql:

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 Test, FOREIGN KEY (ssn) REFERENCES Technician_emp) CONSTRAINT MODEL CHECK (SELECT * FROM Expert, Type WHERE Expert.ssn = ssn AND Expert.model_no = Type.model_no AND Type.reg_no = reg_no)