**ComS 363 Spring 2022**

**Class Participation on Week 3**

**Date: Feb. 1, 2022**

**Gerald Edeh and** [Chimzim Ogbondah](mailto:ogbondah@iastate.edu)

**Topic: Relational data model**

**Learning objective:**

1. To gain a deeper understanding about the relational data model.

2. Learn some Data Definition Language (DDL) specified by the Structured Query Language (SQL).

**Instruction:**

**As a team, answer the following questions.**

**Questions**

1. The schema of the **emp\_work\_dept** relation is below. This relation keeps information about employees, departments, and relationships about which departments an employee works for. Furthermore, it also keeps information about department managers.

Keep in mind that each attribute for each row stores at most one value (null or a valid value in the domain of that attribute). Recall that null is a reserved value, meaning empty or not-applicable in the relational data model. From your client, you also know that the attribute *eid* uniquely identifies each employee, and the attribute *did* uniquely identifies each department.

**schema**

**emp\_work\_dep**t(eid int, ename varchar(50) **not null**, salary decimal(10,0), did int, dname varchar(40) **not null**, budget decimal(10,0), managerid int, **primary key (eid,did)**);

The corresponding SQL statement for creating the above schema is below.

create table emp\_work\_dept

(eid int,

ename varchar(50) not null,

salary decimal(10,0),

did int,

dname varchar(40) not null,

budget decimal(10,0),

managerid int,

primary key (eid,did));

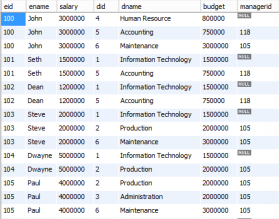


Fig. 1: An instance of the emp\_work\_dept relation

Answer the following questions. Your answer should come from looking at the schema, not just the given instance.

a. Can *eid* store null as its value? **No, because it's a Primary Key**

b. Can *did* store null as its value? **No, because it’s a Primary Key**

c. Can there be more than one row with the same values of both *eid* and *did* (e.g., two rows with eid=100 and did=4)? **No, because eid and did are primary keys**

d. Can *ename* store null as its value? **No because has Not Null constraint**

e. Can there be multiple rows with the same *ename* value? **Yes because there is no “Not Null” constraint**

f. Can an employee work for more than one department? **Yes because eid & did together form the primary key**

g. Can a department have more than one employee working for it? **Yes because edi & did together form the primary key**

h. Must each department have a manager? **No because the managerid can be Null**

i. Can a department have more than one manager? **Yes, see class example (Section 2)**

j. Can you change the managerid value of any row to 200 given that 200 is not one of the existing values of the eid attribute? **Yes because there are no other constraints**

k. Can you store an employee without knowing which department the employee works for? **No because did is part of the primary key**

l. Can you add a new department with the *did* value of 7 without having any employee to work for this new department? **No because eid is part of the primary key**

m. If you change the *dname* attribute value in the row where did = 4 and eid=100 to “Human” instead of “Human Resources,” what happens to your database? **It would add inconsistency since did=4 now has department name “Human” and “Human Resources”**

2. Suppose you execute the following SQL statement to make managerid a foreign key to the emp\_work\_dept(eid).

alter table emp\_work\_dept

add foreign key (managerid) references emp\_work\_dept(eid);

Can you change the value of the *managerid* attribute of any row to 200 given that 200 is not one of the existing eid values in the current instance of emp\_work\_dept? **No, because the foreign key(managerid) now references an existing emp\_work\_dept(eid) and so if the employee id doesn’t exist it can’t be created.**

**3.** Given the ER diagram in Figure 2, design your relation schemas to avoid redundancy as much as possible and use the least number of relations as possible (for increasing the speed of the execution of queries)? Be sure that no attributes are lost. Try to capture as many constraints as possible. You can omit data types of attributes from your answers. Indicate which constraints cannot be captured.

**Employees(**SSN, BDATE, supervisorSSN, gender, dNumber NOT NULL , Salary, Fname, Address, Mname, Lname, Picture, PK(SSN), foreign key(supervisorsSSN) references **Employees(SSN**), foreign key(dNumber) references **Department(dnumber)**

**Works\_on(**eSSN, pid, hours, PK(eSSN, pid) foreign key(eSSN) references **Employees(SSN)**, foreign key(pid) references **Projects(pNumber)**,

**Works\_for(**eSSN, dNumber, PK(eSSN, dNumber) foreign key(eSSN) references **Employees(SSN)**, foreign key(dNumber) references **Department(dnumber)**,

**Department**(dname, dnumber NOT NULL UNIQUE, startDate PK(dnumber),

**location\_department**(locid, dNumber, PK(locid, dNumber), foreign key(locid) references **Locations(LOCID)** foreign key(dNumber) references **Department(dnumber),**

**Locations(**LOCID, Address, PK(LOCID))

**Projects(**pname NOT NULL UNIQUE, pNumber, site, PK(pNumber))

Example: R1(A, B, C NOT NULL UNIQUE, D), primary key(A), foreign key(C) references R2(E).

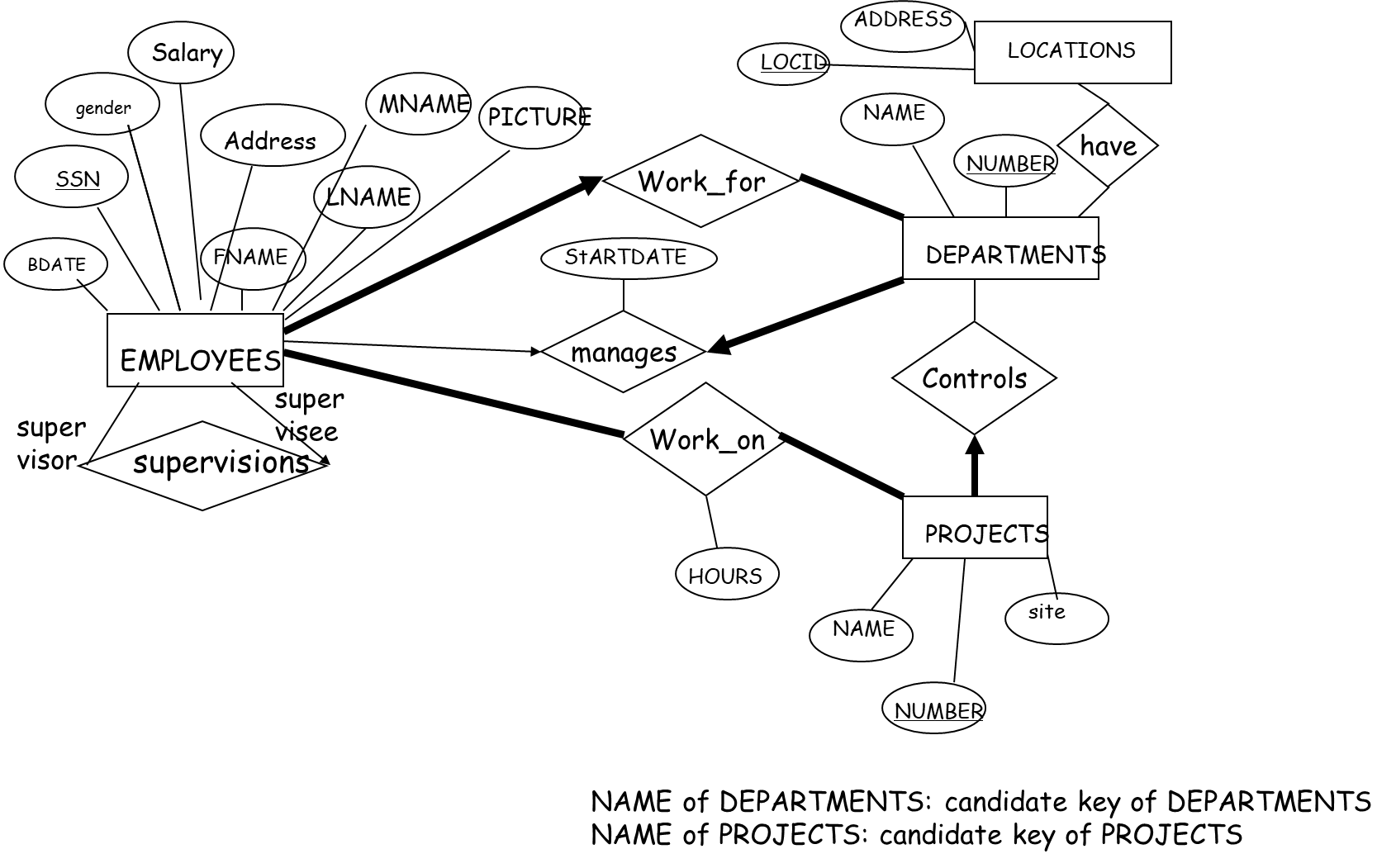


Fig. 2: ER diagram