COSC 344
Assignment 2

Overview

In this assignment, you will turn your ER diagram into a set of Oracle tables, normalize your design, and populate your database.

Due date for assignment 2

Friday, 24 August 2018 at 4 PM

Description

1. Relational Schema

Revise your ER diagram based on the feedback from Assignment 1, and then use the 7-step algorithm introduced in Lecture 4 to convert the revised ER diagram to a relation schema. The result should be similar to the diagram discussed in Lecture 4. Underline the primary key, and show all attributes (no need to include the derived attributes). For referential integrity constraints, draw an arrow from the foreign key terminating on the primary key it refers to.

2. Normalization

Turn your schema diagram into a set of tables in BCNF. You should go to 1NF, 2NF, 3NF, then BCNF. It might happen that a relation may be already in a given normal form (e.g 3NF). In such a case, you just explain why it is in that normal form. Explain your steps in a manner similar to the example given in the attachment. If you cannot get to BCNF, explain why. Your write-up should be reasonable to follow and include no jumps in logic – it's possible that your BCNF schema may be a 'good representation' of your miniworld, but 'not be correct' based on your initial ERD.

3. Create Database

Create an SQL script which will create your tables in their BCNF form. The CREATE TABLE commands should be as complete as possible. This means where possible they should:

1. specify domains if some subset of normal data types is appropriate
2. specify the primary key
3. specify uniqueness, not null, and/or a default where appropriate
4. referential integrity constraints where appropriate

Depending upon your constraints, the order of table creation may be important.

Put a series of DROP TABLE commands ahead of the CREATE TABLE commands in your script so that the database can be reloaded without any error. Again the order may matter.

Populate your database with a series of INSERT commands. These should follow the CREATE TABLE commands. If at all possible, the INSERT commands for a given table should be grouped together and followed by a COMMIT.

The script described above must be called `load.sql` and is a deliverable. It should execute without errors. The allowed exception is a DROP TABLE error when the table does not exist.

Teamwork Model

The team leader coordinates the allocation of the tasks, and makes sure the task allocation is fair to every team member. For example, each member should map at least one entity type and one relationship type. Each member should normalize at least one relation and create at least one table.

At the end of the report there should be a summary of the teamwork, showing which part was done by which member, and which parts have been discussed by all group members. You can add any comment on the teamwork in your group.

Assignment submission

The assignment MUST be submitted electronically. Include the following into your cosc344_asgn2 folder:

1. A single document (prefer to be in Word format) that includes the following:

- Revised ERD (if there is no change, just put the one in your 1st assignment), and explain what has been changed.
- Relational schema diagram to show integrity constraints (refer to the example given in slide 23 of Lecture 3)
- Normalization description
- load.sql

2. the load.sql file

The team leader (or any team member) creates a folder named `asgn2_groupX`, where X is the group number. Make sure this folder contains all the files that need be submitted.

Change into the directory that **contains** this folder, and submit your assignment using the COSC344 submit script as follows:

```
$ submit344 asgn2_groupX
```

The `submit344` script is put at `/home/cshome/coursework/bin/.` If the above command does not work, please give the full pathname for the script, that is,

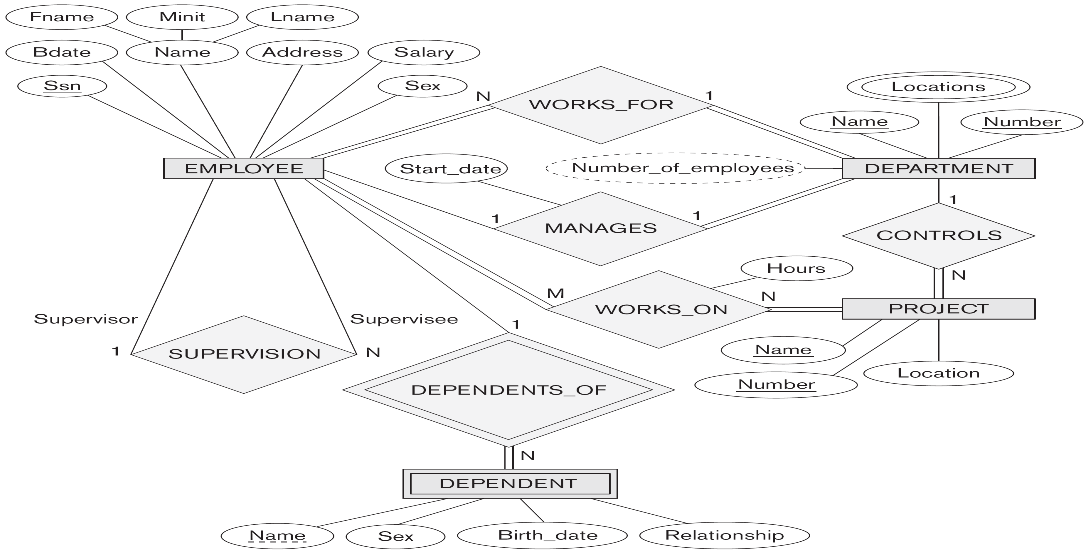
```
$ /home/cshome/coursework/bin/submit344 asgn2_groupX
```

The script displays its progress so you can see that it has worked. You can resubmit before the due date if you wish -- your last submission is the one that will be marked

COSC344 Assignment 2 (Sample)

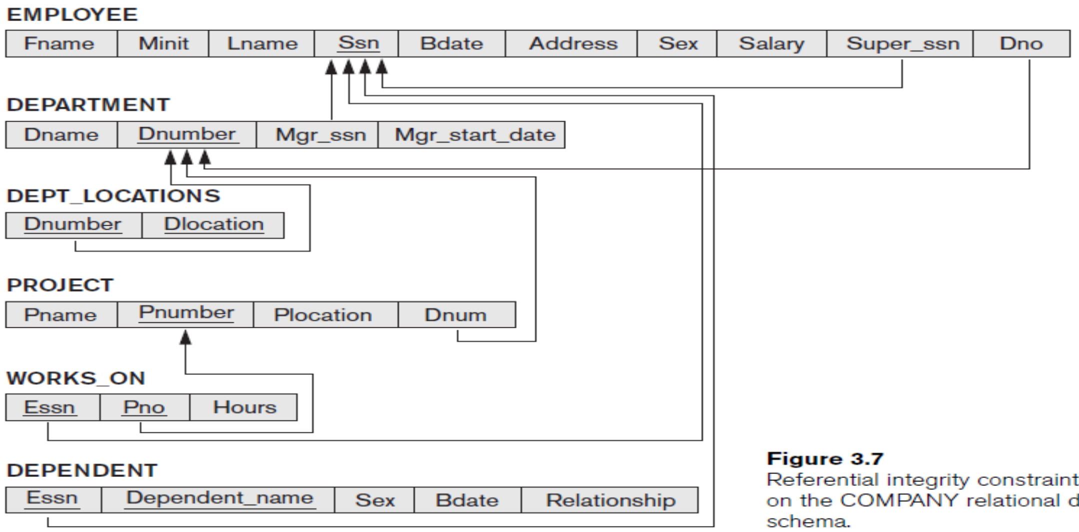
Group X
Leader: [put leader name here]
Members: [put names of all members here]

1. Revised ER Diagram



[Explain what has been changed]

2. Relational Schema



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

4. Normalization

Normalization should be done step by step from 1NF to BCNF.

Suppose your relation schema has the following relation where PHONES is a multivalue attribute and each office has a unique OFF#:

OFFICE

OFF#	AREA	DEPT#	PHONES

You should normalized it into BCNF as follows:
Step 1: 1NF

OFFICE is not in 1NF. To normalize it into 1NF, the multivalued attribute should be eliminated by dividing it into the following two relations:

```
OFFICE1( off#, area, dept# )
PHONE( phone#, off# )
```

Step 2: 2NF
PHONE is already in 2NF, but OFFICE1 is not in 2NF as area only depends on off#. To get it into 2NF, use only OFF# as the primary key, and we get the following relation in 2NF

```
OFFICE2( off#, area, dept# )
```

Step3: 3NF
Both relations are in 3NF. For `OFFICE2`, `area` is the only nonprime attribute, and it is not transitively dependent on the primary key. For `PHONE, off#` is the only nonprime attribute, and it is not transitively dependent on the primary key.

Step4: BCNF
Both relations are in BCNF.

5. Load.sql

```
DROP TABLE dependent;
DROP TABLE works_on;
DROP TABLE project;
DROP TABLE dept_locations;
DROP TABLE employee cascade constraints;
DROP TABLE department cascade constraints;

CREATE TABLE department
  (dname          VARCHAR2(15)     NOT NULL          UNIQUE,
   dnumber        INT               PRIMARY KEY,
   mgrssn         CHAR(9)           NOT NULL,
   mgrstartdate DATE);

INSERT INTO department VALUES
   ('Research', 5, '333445555', TO_DATE('22-05-1988','DD-MM-YYYY'));
INSERT INTO department VALUES
   ('Administration', 4, '987654321', TO_DATE('01-01-1995','DD-MM-YYYY'));
INSERT INTO department VALUES
   ('Headquarters', 1, '888665555', TO_DATE('19-06-1981','DD-MM-YYYY'));
INSERT INTO department VALUES
   ('Dummies', 0, '111100000', TO_DATE('31-12-2004','DD-MM-YYYY'));


CREATE TABLE employee
  (fname      VARCHAR2(15)       NOT NULL,
   minit      CHAR,
   lname      VARCHAR2(15)       NOT NULL,
   ssn        CHAR(9)            PRIMARY KEY,
   bdate      DATE,
   address    VARCHAR2(30),
   sex        CHAR,
   salary     NUMBER(6),
   superssn   CHAR(9)
     CONSTRAINT superssn_cnst REFERENCES employee(ssn) DISABLE,
   dno        INT                NOT NULL
     CONSTRAINT dno_cnst REFERENCES department(dnumber) DISABLE);

ALTER TABLE employee ENABLE CONSTRAINT dno_cnst;

INSERT INTO employee VALUES
  ('John','B','Smith','123456789',TO_DATE('09-01-1965','DD-MM-YYYY'),
   '731 Fondren, Houston, TX','M',30000,'333445555',5);
INSERT INTO employee VALUES
  ('Franklin','T','Wong','333445555',TO_DATE('08-12-1955','DD-MM-YYYY'),
   '638 Voss, Houston, TX','M',40000,'888665555',5);
INSERT INTO employee VALUES
  ('Alicia','J','Zelaya','999887777',TO_DATE('19-07-1968','DD-MM-YYYY'),
   '3321 Castle, Spring,TX','F',25000,'987654321',4);
INSERT INTO employee VALUES
  ('Jennifer','S','Wallace','987654321',TO_DATE('20-06-1941','DD-MM-YYYY'),
   '291 Berry, Bellaire, TX','F',43000,'888665555',4);
INSERT INTO employee VALUES
  ('Ramesh','K','Narayan','666884444',TO_DATE('15-09-1962','DD-MM-YYYY'),
   '975 Fire Oak, Humble, TX','M',38000,'333445555',5);
INSERT INTO employee VALUES
  ('Joyce','A','English','453453453',TO_DATE('31-07-1972','DD-MM-YYYY'),
   '5631 Rice, Houston, TX','F',25000,'333445555',5);

COMMIT;
```

6. Teamwork Summary

Put the summary of your team work here.