




Homework 1

DATABASE MANAGEMENT

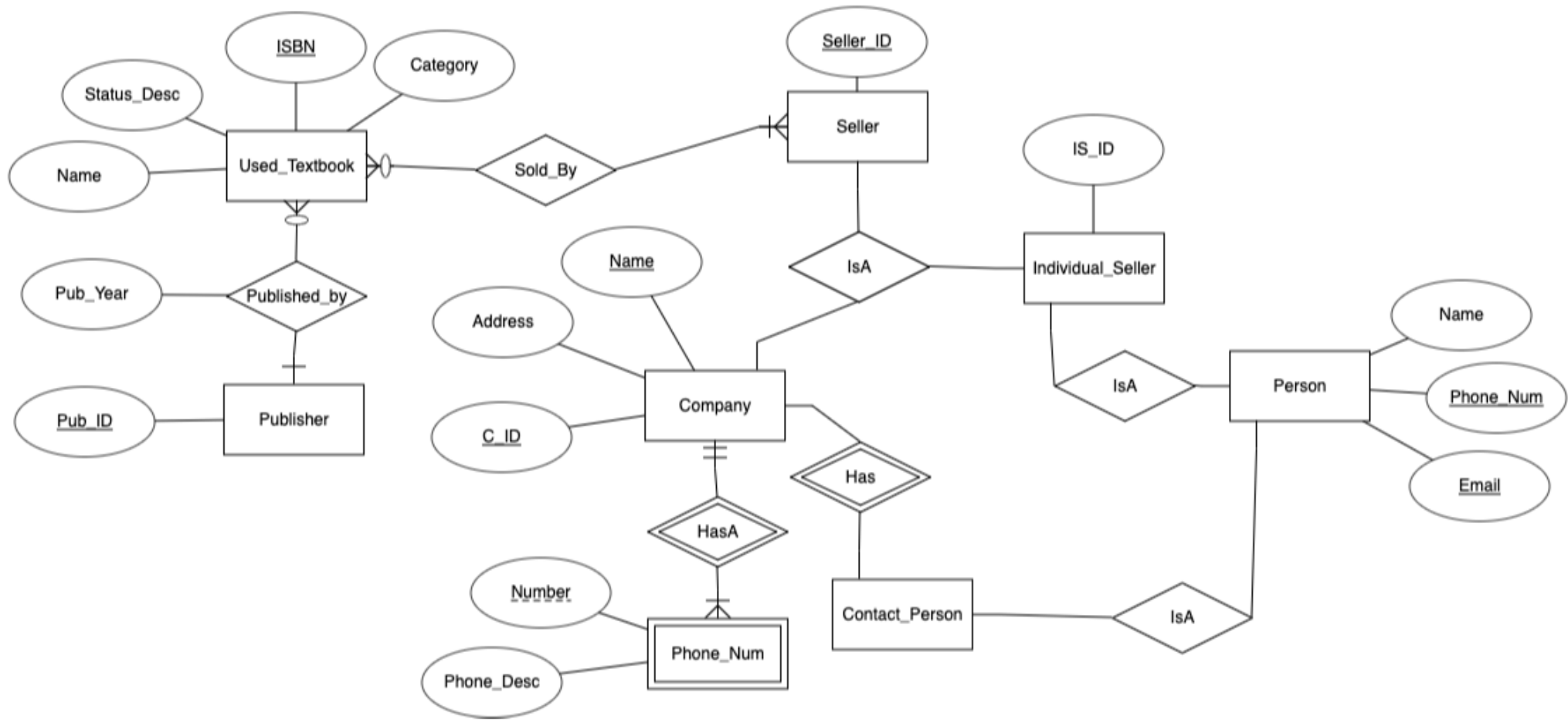
Fatima Cheema

CS550-DL1 (Fall 2022)



Problem 1

In this exercise, you will design a small database for used textbooks, as might be used by an online trading site like Amazon. The database will store textbooks for sale. Each textbook has an ISBN, a category (e.g., Math), and a name. In addition, status descriptions about the books are maintained. Each book is required to have one such description so that the system can rate it fairly. Each textbook is published by a company in the publishing year. Each publisher has to have published at least some textbooks. Each textbook has one or more sellers, which may be either companies (corporation sellers) or individuals (individual sellers). For each company, the database maintains a name of the company, its address, its phone numbers (could be more than one phone number, each with a number and a description), and its contact person (who is an individual with all the related information for individuals, see next sentence). For each individual, the database keeps a name, a phone number and an email address. A contact person whose company sells a book cannot be selling the same book as an “individual seller” at the same time (he/she may sell other books as an individual seller).



The ER- diagram doesn't reflect "contact person whose company sells a book can't be selling the same book as an individual seller at the same time"

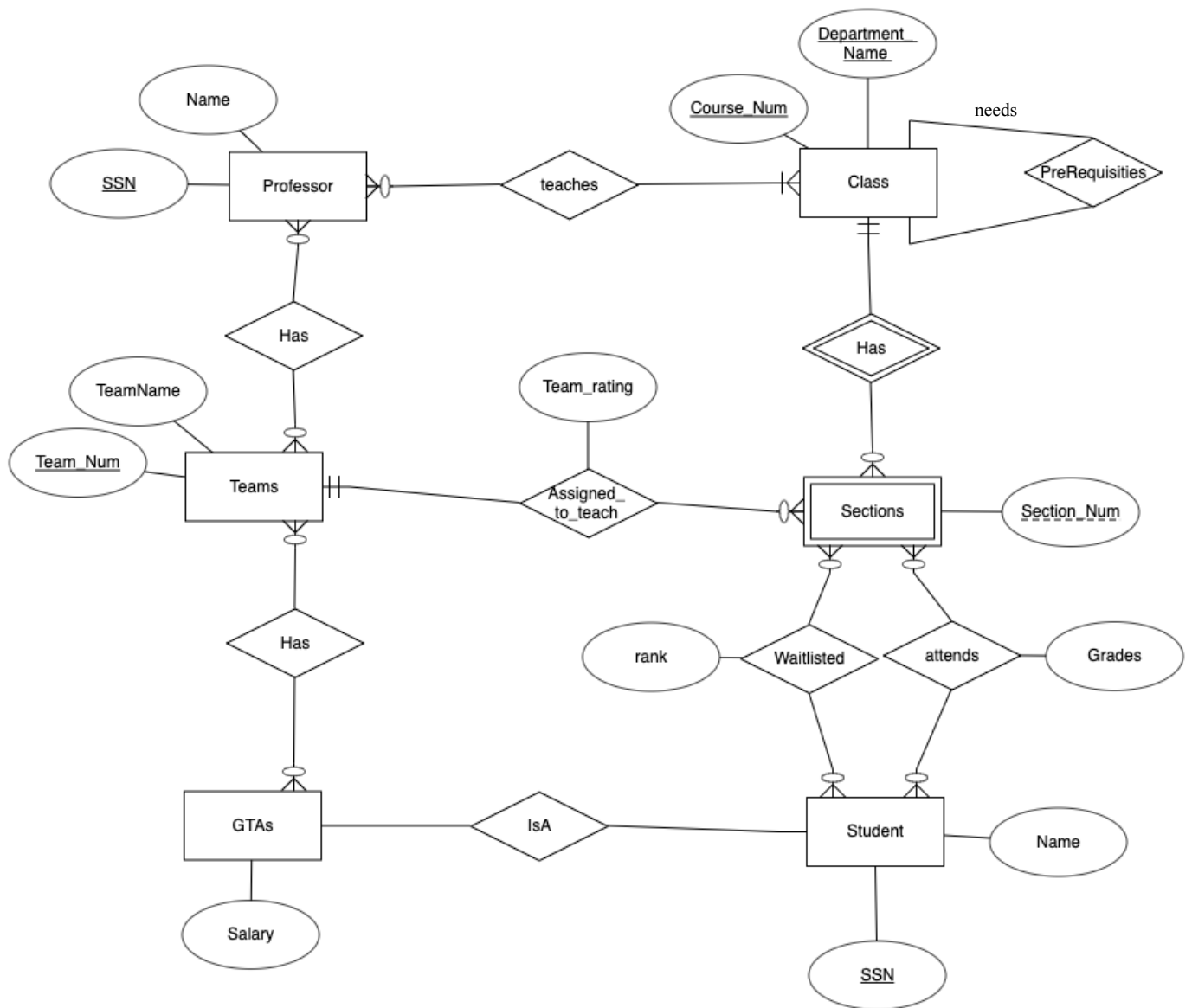
Problem 1

The ER- diagram identifies the following: all the entity sets; all the relationship sets and their multiplicities; the primary key for each entity set and each relationship set.

The property which was not reflected in the design is “a contact person whose company sells a book can’t be selling the same book as an individual seller at the same time,” because according to the diagram a company’s contact person can also be an individual seller selling the same book.

Problem 2

George Mason University (GMU) has decided to consolidate the functionality of three small overlapping database systems, which support applications for 1) teaching (e.g., instructor assignment and evaluation), for 2) registration (e.g. online course status, waiting lists), and for 3) student records (e.g. transcript generation). The resulting new system will support the following enterprise description: Professors and GTAs are assigned to teach the sections of each class being offered in a semester. At the end of the semester, they get a "team rating" (professors and GTAs together get one rating per section, rating is not done on individual). To support the assignment of professors to sections, a record is kept of which class each professor can teach. Classes can have one or more prerequisite classes. Students can take several sections each semester and receive a grade for taking each section. Students may end up waiting for some sections and receive a "rank" (determining the order they will be admitted if other students drop). However, no more than 10 students can wait on a class at the same time. Note that GTAs are students, however they differ in that they have a salary. All people (e.g., students, professors) are uniquely identified by their social security number. All classes are identified by department name (e.g., "INFS") and course number (e.g., "614"). Sections of classes are distinguished by their section number (e.g., "02").



The ER- diagram doesn't reflect that 'no more than 10 students can wait on a class at the same time.'

P2.Q2. Indicate feature(s)/property(ies) in the above description that are NOT captured by your ER-diagram

The ER- diagram doesn't reflect that 'no more than 10 students can wait on a class at the same time,' because the numerical limitation/constraint can't be depicted in the diagram.

P2.Q3. Give 2 examples of the types of reports that can be obtained from the database, and state the involved entity sets and/or relationship sets.

Example 1

List of all students assigned to a professor for course 550.

Example 2

List of GTAs and Professor working in a team to teach course 550.

ER-diagram for the database, identifies the following:

- (i) all the entity sets.
- (ii) all the relationship sets and their multiplicity.
- (iii) the primary key for each entity set (and weak entity set, if any) and each relationship set.

/* THE COMMANDS/QUERIES GIVEN IN HW1.SQL FILE */

```
CREATE TABLE PROFESSOR( SSN INT PRIMARY KEY NOT NULL, NAME varchar(50));
CREATE TABLE STUDENT( SSN INT PRIMARY KEY NOT NULL, NAME varchar(50));
CREATE TABLE GTA( SSN INT PRIMARY KEY NOT NULL, NAME varchar(50), SALARY FLOAT(10), FOREIGN
KEY(SSN) REFERENCES STUDENT(SSN) ON DELETE CASCADE);
create TABLE TEAM( TEAM_NUMBER INT, PROFESSOR_ID INT, GTA_ID INT, FOREIGN KEY(PROFESSOR_ID)
REFERENCES PROFESSOR(SSN), FOREIGN KEY(GTA_ID) REFERENCES GTA(SSN));
CREATE TABLE CLASS(DEPARTMENT_NAME VARCHAR(50), COURSE_NUMBER INT, PREREQ_COURSE_NUM INT, PRIMARY
KEY(DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM));
CREATE TABLE SECTION( SEC_NUM INT NOT NULL, DEPARTMENT_NAME VARCHAR(50), COURSE_NUMBER INT,
PREREQ_COURSE_NUM INT, PRIMARY KEY( SEC_NUM),FOREIGN KEY(DEPARTMENT_NAME,COURSE_NUMBER,
PREREQ_COURSE_NUM ) REFERENCES CLASS(DEPARTMENT_NAME,COURSE_NUMBER, PREREQ_COURSE_NUM) ON DELETE
CASCADE);
CREATE TABLE CLASSES_TAUGHT_BY_PROF (SSN INT, DEPARTMENT_NAME VARCHAR(50), COURSE_NUMBER INT,
PREREQ_COURSE_NUM INT, SEC_NUM INT, COURSE_EVALUATION VARCHAR(70), FOREIGN KEY(SSN) REFERENCES
PROFESSOR(SSN), FOREIGN KEY(DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM) REFERENCES
CLASS(DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM), FOREIGN KEY(SEC_NUM) REFERENCES
SECTION(SEC_NUM));
CREATE TABLE CLASSES_TAKEN_BY_STUDENT (SSN INT, PROFESSOR_ID INT, SEC_NUM INT, DEPARTMENT_NAME
VARCHAR(50), COURSE_NUMBER INT, PREREQ_COURSE_NUM INT, GRADE VARCHAR(2), FOREIGN KEY(ssn)
REFERENCES STUDENT(SSN), FOREIGN KEY(PROFESSOR_ID) REFERENCES PROFESSOR(SSN), FOREIGN
KEY(SEC_NUM) REFERENCES SECTION(SEC_NUM), FOREIGN KEY(DEPARTMENT_NAME,COURSE_NUMBER,
PREREQ_COURSE_NUM) REFERENCES CLASS(DEPARTMENT_NAME,COURSE_NUMBER,PREREQ_COURSE_NUM));
CREATE TABLE PREREQ_FOR_CLASS (REQUIRED_COURSE_DEPARTMENT VARCHAR(50), REQUIRED_COURSE_NUMBER
INT, PREREQ_COURSE_NUM INT, FOREIGN KEY (REQUIRED_COURSE_DEPARTMENT, REQUIRED_COURSE_NUMBER,
PREREQ_COURSE_NUM) REFERENCES CLASS(DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM));
CREATE TABLE STUDENT_ENROLLMENT (E_ID INT PRIMARY KEY, GNUMBER INT NOT NULL, SEC_NUM INT NOT
NULL, WAIT_LIST VARCHAR(5), WL_RANK INT NOT NULL, CHECK(WL_RANK >=1 AND WL_RANK<=10),
PROFESSOR_ID INT, GTA_ID INT, FOREIGN KEY(GNUMBER) REFERENCES STUDENT(SSN) ON DELETE CASCADE,
FOREIGN KEY(SEC_NUM) REFERENCES SECTION(SEC_NUM), FOREIGN KEY(PROFESSOR_ID) REFERENCES
PROFESSOR(SSN), FOREIGN KEY (GTA_ID) REFERENCES GTA(SSN));

/***** INSERT VALUES *****/
INSERT INTO PROFESSOR(SSN, NAME) VALUES (123,'JOHN DOE');
INSERT INTO PROFESSOR(SSN,NAME) VALUES (456,'JAMES CHARLES');

INSERT INTO STUDENT(SSN, NAME) VALUES (789,'MARY SMITH');
INSERT INTO STUDENT(SSN,NAME) VALUES (101,'HARRY SMITH');
INSERT INTO STUDENT(SSN,NAME) VALUES (103,'LARRY G. ');

INSERT INTO GTA(SSN,NAME,SALARY) VALUES (789,'MARY SMITH',1000);
INSERT INTO GTA(SSN,NAME,SALARY) VALUES (101,'HARRY SMITH',2000);

INSERT INTO TEAM(TEAM_NUMBER, PROFESSOR_ID, GTA_ID) VALUES (1, '456', '789');
INSERT INTO TEAM(TEAM_NUMBER, PROFESSOR_ID, GTA_ID) VALUES (2, '123', '101');
```



```
INSERT INTO CLASS( department_name, course_number, prereq_course_num) VALUES ('COMP SCI', 451, 450);
```

```
INSERT INTO CLASS( department_name, course_number, prereq_course_num) VALUES ('COMP SCI', 550, 451);
```

```
INSERT INTO SECTION( sec_num, department_name, course_number, PREREQ_COURSE_NUM) VALUES (001, 'COMP SCI', 451, 450);
```

```
INSERT INTO SECTION( sec_num, department_name, course_number, PREREQ_COURSE_NUM) VALUES (002, 'COMP SCI', 550, 451);
```

```
INSERT INTO CLASSES_TAUGHT_BY_PROF( SSN, DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM, SEC_NUM, COURSE_EVALUATION) VALUES (123, 'COMP SCI', 451, 450, 001, 'NA');
```

```
INSERT INTO CLASSES_TAUGHT_BY_PROF( SSN, DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM, SEC_NUM, COURSE_EVALUATION) VALUES (456, 'COMP SCI', 550, 451, 002, 'NA');
```

```
INSERT INTO CLASSES_TAKEN_BY_STUDENT( SSN, PROFESSOR_ID, SEC_NUM, DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM, GRADE) VALUES (789, 123, 001, 'COMP SCI', 451, 450, 'A');
```

```
INSERT INTO CLASSES_TAKEN_BY_STUDENT( SSN, PROFESSOR_ID, SEC_NUM, DEPARTMENT_NAME, COURSE_NUMBER, PREREQ_COURSE_NUM, GRADE) VALUES (101, 456, 002, 'COMP SCI', 550, 451, 'B');
```

```
INSERT INTO PREREQ_FOR_CLASS (REQUIRED_COURSE_DEPARTMENT, REQUIRED_COURSE_NUMBER, PREREQ_COURSE_NUM) VALUES ('COMP SCI', 451, 450);
```

```
INSERT INTO PREREQ_FOR_CLASS (REQUIRED_COURSE_DEPARTMENT, REQUIRED_COURSE_NUMBER, PREREQ_COURSE_NUM) VALUES ('COMP SCI', 550, 451);
```

```
INSERT INTO STUDENT_ENROLLMENT(E_ID, GNUMBER, SEC_NUM, WAIT_LIST, WL_RANK, PROFESSOR_ID, GTA_ID) VALUES (1, 101, 001, 'YES', 2, '456', '789');
```

```
INSERT INTO STUDENT_ENROLLMENT(E_ID, GNUMBER, SEC_NUM, WAIT_LIST, WL_RANK, PROFESSOR_ID, GTA_ID) VALUES (2, 103, 001, 'YES', 9, '123', '101');
```