The A[illegible]nal University, School of Computing
[illegible]240 (Relational Databases)
[illegible]mester 2, 2022

[illegible]orkshop (Solutions)

Entity-Relationship Model

1 University Student Database

Consider the following data requirements for a university student database that is used to keep track of students' transcripts.

- The university keeps track of each student's name, student number, social security number, address, phone, and birthdate. Both social security number and student number have unique values for each student.

- Each student has exactly one major and may have a minor (if any) with departments.

- Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.

- Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.

- Each section of a course has an instructor, semester, year, and section number and the section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

- A grade record refers to each student and a particular section, consisting of a final mark and a letter grade from (F, D, C, B, A).

(1) *Design an EER diagram for this university student database. You can make more assumptions if necessary.*

Suggested solution:

Please refer to the provided ER diagram in Figure 1.

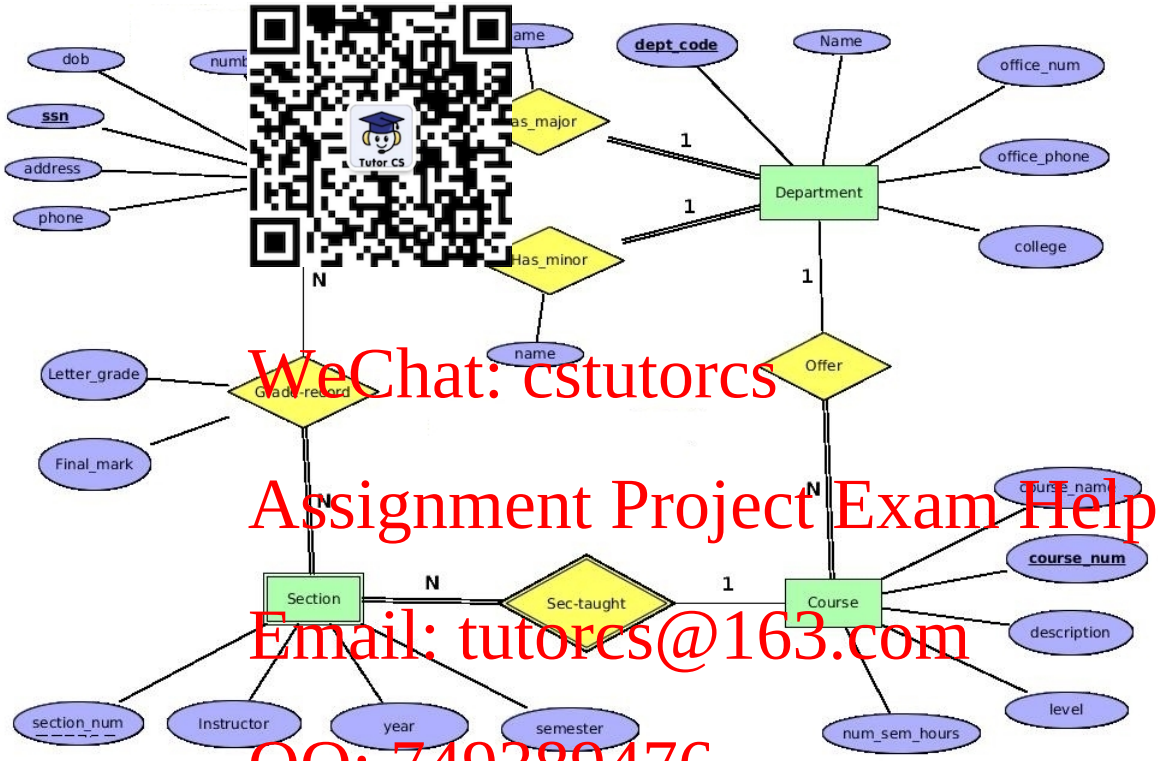

Figure 1: Sample ER diagram for Exercise (1)

- Identify cardinality ratios (add assumptions when they are not explicit from the description):
  - A student must have a major and a department may offer many majors: STUDENT - N – MAJOR - 1 - DEPARTMENT.
  - A student may have a minor and a department may offer many minors: STUDENT - N – MINOR - 1 - DEPARTMENT.
  - A course must be associated to a department and a department may offer many courses: DEPARTMENT – 1 - OFFER - N – COURSE
  - A section must be associated to a course and a course may have many sections: COURSE - 1 - SEC_TAUGHT - N – SECTION.

2 Retailer Company Database

A retailer company wants to build a database application for managing information about its sale process. The company sells products in both local shops and

webstores on the I shop has a name, contact details (e.g., phone number and email ation. The database application also needs to store the URL(last updated date of each webstore. Every product has a uni escription, an item price, and a quantity in stock. The databa d also record customers' details such as their name, address an omer is assigned a unique ID. A customer may place an orde t least one product and each order is from either a shop or a webstore. Customers have three payment options (i.e., cash, paypal, and credit card) but for each order only one payment option can be chosen. A delivery may be requested for each order. After full-payment is received, a delivery would be sent out subject to products' availability. Every delivery has a tracking number.

(2) *Design an EER diagram for this retailer company. You can make more assumptions if necessary.*

Please refer to the provided ER diagram in Figure 2.

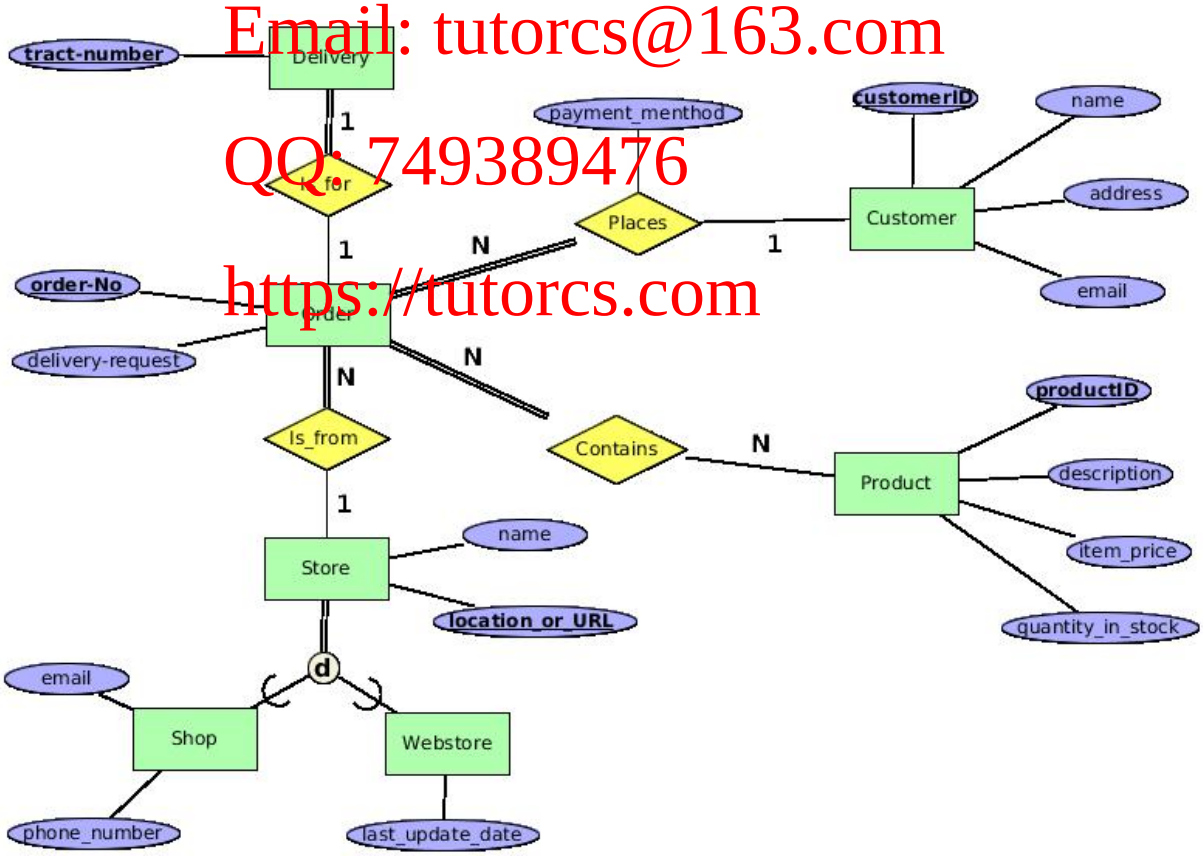

Figure 2: Sample ER diagram for Exercise (2)

(3) *Are there any ... rity constraints you are not able to represent in your EER diag... ive examples.*
For example, the f... or constraints cannot be represented in the EER diagram abo...

- Customers h... options (i.e., cash, paypal, and credit card).
- After full-payment is received, a delivery would be sent out subject to products' availability.

3 Translating ER to Relations

(4) *Translate the ER model in Exercise (1) into a relational database schema (be sure to indicate primary keys and foreign keys).*

Suggested solution:
For the university student database presented above, we may translate the given sample ER model depicted in Figure 1 into the following relational database schema:

- DEPARTMENT(dept_code, name, office_num, office_phone, college) with the primary key {dept_code} (following Step 1 in the lecture slides)
- STUDENT(ssn, number, name, address, phone, dob, major_dept, major_name, minor_dept, minor_name) with the primary key {ssn}, and the foreign keys [major_dept]⊆DEPARTMENT[dept_code] and [minor_dept]⊆DEPARTMENT[dept_code]. (following Steps 1&4 in the lecture slides)

- COURSE(course_num, course_name, description, num_sem_hours, level, dept_code) with the primary key {course_num}, and the foreign key [dept_code]⊆ DEPARTMENT[dept_code]. (following Steps 1&4 in the lecture slides)
- SECTION(section_num, instructor, semester, year, course_num) with the primary key {section_num, course_num}, and the foreign keys [course_num]⊆COURSE[course_num]. (following Step 2 in the lecture slides)
- GRADE_RECORD(ssn, section_num, course_num, letter_grade, final_mark) with the primary key {ssn, section_num, course_num}, and the foreign keys [ssn]⊆STUDENT[ssn] and [section_num, course_num]⊆SECTION[section_num, course_num]. (following Step 5 in the lecture slides)