EE4791 Database Systems -Tutorial 8 and Sample Answer

1. Consider the following three relations for a College:

   STUDENT (StudentID, StudentName, Major, Age, MaritalStatus, PostalCode)
   REGISTRATION (StudentID, CourseID, Mark)
   COURSE (CourseID, CourseName)

   a) Choose Oracle data types for the attributes in table STUDENT based on their common characteristics.
   b) Suppose you are designing a default value for the age field in the database. What possible values you will consider and why? How might the default value vary by other characteristics about the student, such as school within the university or degree sought?
   c) When a student has not chosen a major at a university, the university often enters a value of “Undecided” for the major field. What is the most suitable default value for major? What is the difference between “Undecided” and “null” as default value?
   d) In addition to the information shown in the relations and discussed in b) and c), we further assume that StudentID and CourseID are foreign keys of the table REGISTRATION. Using SQL to define the tables for the three relations. Specify primary key and referential integrity constraints.

Answer

a)

StudentID CHAR(10) NOT NULL
StudentName CHAR(25)
Major CHAR(15)
MaritalStatus CHAR(8)
Age NUMBER(3)
PostalCode CHAR(8)

b)

(1) Since most students are accepted in the first year after high school graduation, the average age of students in the first year of college would be a good choice for a default value during record creation. In most countries, we may use 18. For Singapore case, we may use 18 for female and 20 for male, since boys need to attend NS first.

(2) Degree seeking students are generally younger than non-degree seeking (mature) students, and the default value might be a higher age for non-degree seeking students.

c)

Since every student who hasn't explicitly declared a major of his or her choice is in the status of "undecided" and he/she will have to decide a major later, therefore, "undecided" is the most appropriate default value for major. The null value is an empty value, thus assigning a default value of "Undecided" is not the same as setting the value to null.

d)

```
CREATE TABLE STUDENT
        (StudentID         CHAR(10)  NOT NULL,
         StudentName      CHAR(25) NOT NULL,
         Major             CHAR(15) DEFAULT “Undecided”,
        MaritalStatus  CHAR(8),
        Age                NUMBER(3) DEFAULT 18,
        PostalCode     CHAR(8),
CONSTRAINT STUDENT_PK PRIMARY KEY
(StudentID));

CREATE TABLE COURSE
        (CourseID         CHAR(12)  NOT NULL,
         CourseName         CHAR(30) NOT NULL,
CONSTRAINT COURSE_PK PRIMARY KEY (CourseID));

CREATE TABLE REGISTRATION
        (Student_ID    CHAR(10)  NOT NULL,
         Course_ID     CHAR(12) NOT NULL,
         Mark        NUMBER(3),
CONSTRAINT REGISTRATION_PK  PRIMARY KEY
         (Student_ID, Course_ID),
CONSTRAINT REGISTRATION _FK1 FOREIGN KEY
         (Student_ID) REFERENCES STUDENT(StudentID),
CONSTRAINT REGISTRATION_FK2 FOREIGN KEY
         (Course_ID) REFERENCES COURSE(CourseID));
```

2. Consider the EER diagram shown in Figure 1. The estimates on the usage for this EER diagram are as follows:

- 5,000 parts in which 3% are manufactured parts and 100% are purchased parts
- 2000 suppliers
- an average of 4 supplies per supplier
- 50,000 direct accesses of part per hour
- 20,000 direct accesses of purchased part per hour
- 50% of accesses to purchased part will lead to the accesses of all the related supplies and their suppliers.
- an average of 3 supplies associated with each purchased part
- 8,000 direct accesses to supplier
- 1% of accesses to supplier will lead to the access of all the associated supplies and purchased part associated

Based on these estimates, draw a usage map for this EER diagram. If all these accesses are carried out interactively, identify two most useful ways to support the performance of these assesses without using keys or indexes.

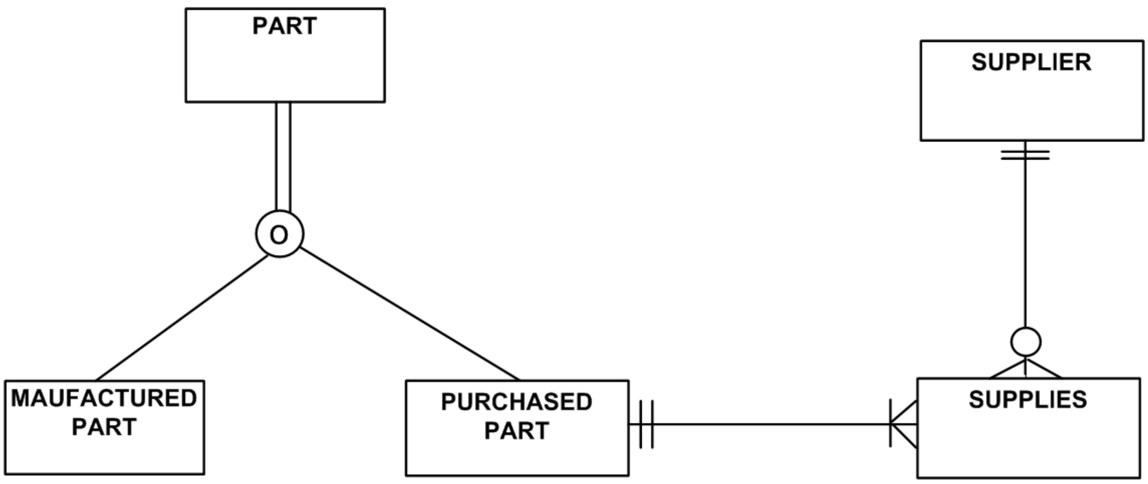
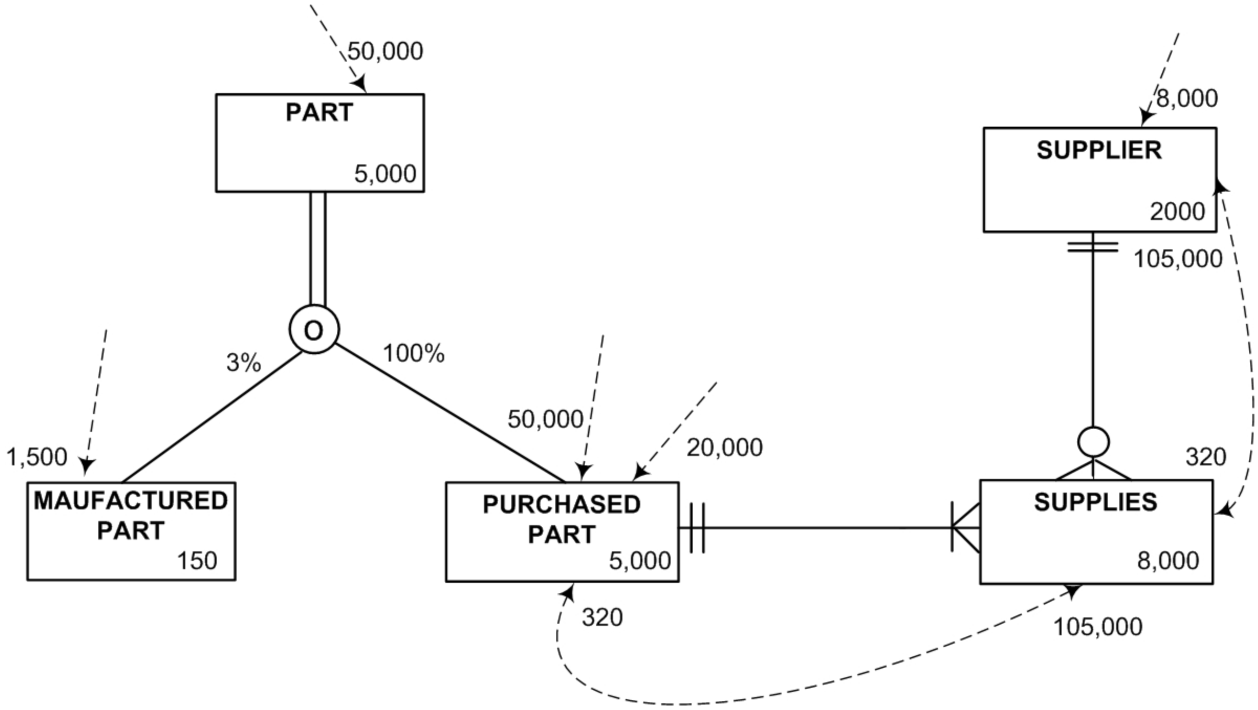


Figure 1. An EER Diagram

Answer

Usage map:



Two most useful ways to support the performance of the accesses:

(1) Since there are a total of 70,000 accesses of purchased part but there is only 1,500 accesses of manufactured part per hour and there is only 3% overlap between them, we should store MANUFACATURED PART and PURCHASED PART as two separate tables and organize them differently due to the significantly different access volume.

(2) Since for each of the 105,000 accesses of supplies, the associated supplier is also accessed, the information of the associated supplier should be added to each supply due to the high volume of these co-accesses.

3. Consider the EER diagram shown in Figure 2. Based on the following estimates on the average usage of the system:

- There are 50,000 customers, and of these, 80 percent represent regular accounts and 30 percent represent national accounts.
- Currently, the system stores 800,000 orders, although this number is constantly changing.
- There are 3,000 products.
- Approximately, 500 orders are placed per hour and each of them has an average of 20 products. When an order is placed, a customer record is accessed directly and all the product records involved are also accessed directly.

Based on these assumptions, draw a usage map for this EER diagram.

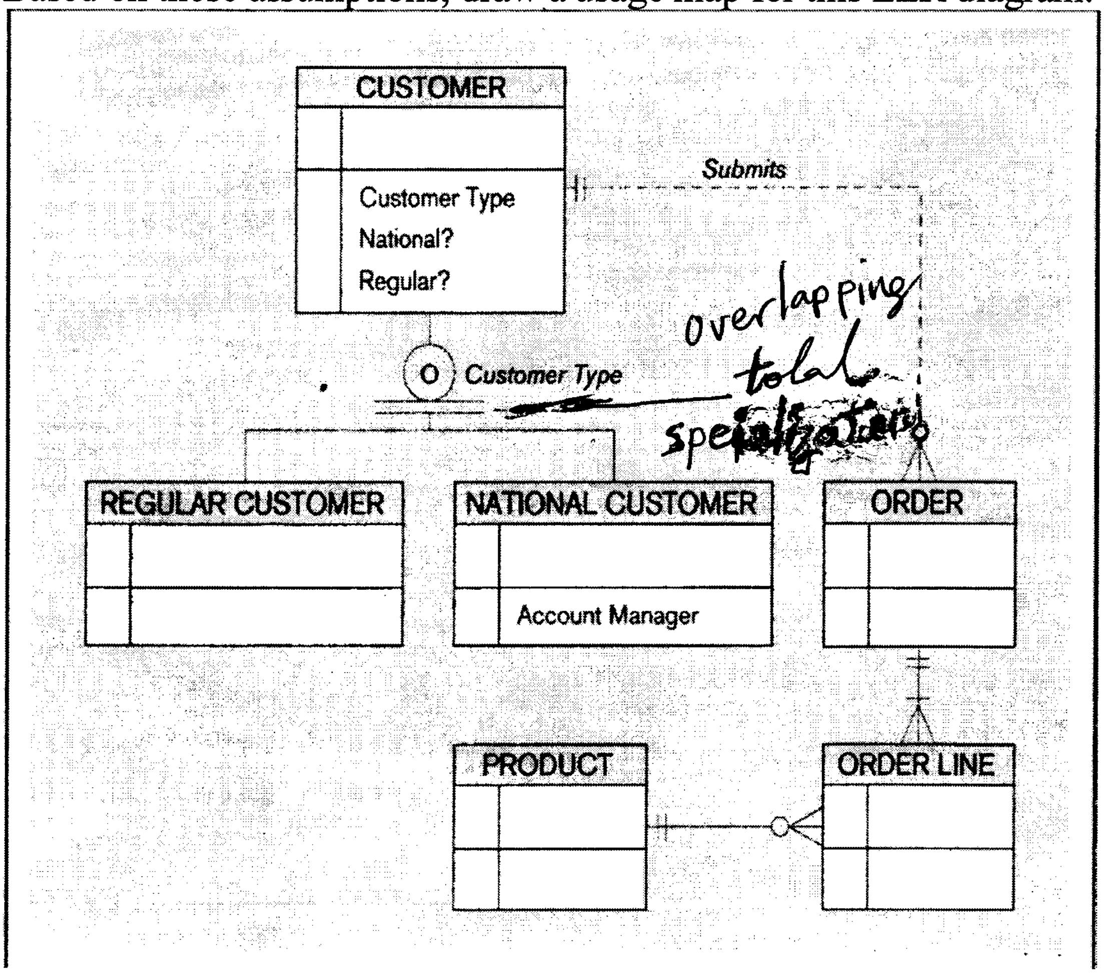
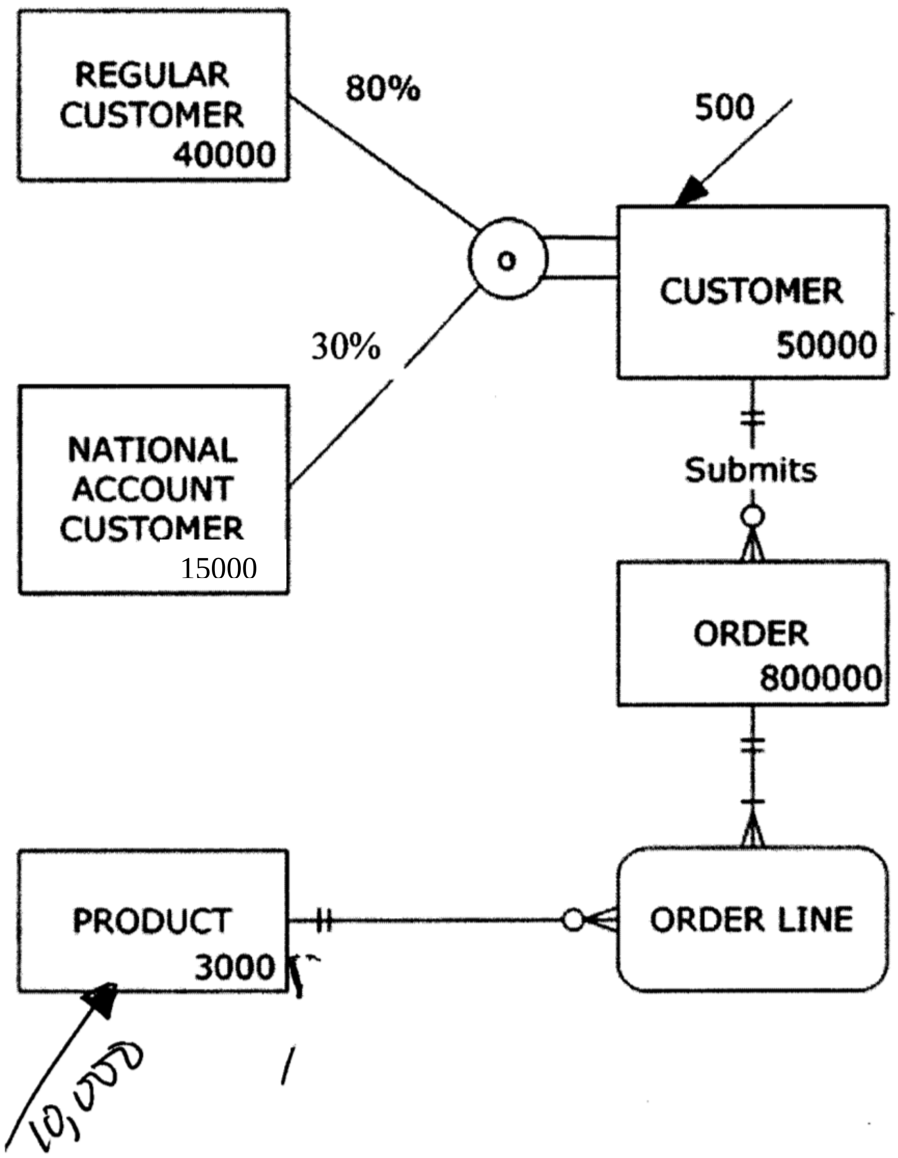


Figure 2. An EER Diagram

EER drawn using Microsoft Vision (notations slightly different)

Answer


REGULAR CUSTOMER 40000
80%
500
CUSTOMER 50000
30%
NATIONAL ACCOUNT CUSTOMER 15000
Submits
ORDER 800000
PRODUCT 3000
ORDER LINE
10,000

4. Consider the following normalized relations from a database in a large retail chain:

   EMPLOYEE (<u>EmployeeID</u>, Employee_Name, EmployeeAddress, PhoneM)
   SCHEDULE (<u>DepartmentID</u>, <u>EmployeeID</u>, Date)

   Primary keys are underlined.

   a) Explain why normalized relations may not be efficient physical records. What opportunities might exist for denormalizing these relations when defining the physical records for this database? Under what circumstances would you consider creating such denormalized records here?
   b) Denormalize the tables assuming employee data is only accessed via schedule.
   c) List drawbacks of denormalization.

Answer

a)

Often all the attributes in a relation are not used together, but data from different relations are needed together to satisfy a query or report.

- Example: Order processing likely requires both order details and customer shipping data from Customer and Order tables.

Thus, although normalized relations solve data maintenance anomalies, normalized relations, if implemented one for one as physical records, may not yield efficient data processing.

Common denormalization opportunities:

- One-to-one relationship (Fig. 5.3 in Unit 2):
  - ✖ Even if one relation is optional in the relationship, in this case, fields without values will be set to null values.
  - ✖ Does not introduce complication to updating
- Many-to-many relationship with non-key attributes (associative entity, Fig. 5-4 in Unit 2):
  - ✖ Rather than joining three tables to extract data from two basic entities with m:m relationship, attributes from one of the entities are combined into the record representing m:m relationship
  - ✖ this will introduce excessive updating if duplicated data changed
- Reference data (Fig. 5-5 in Unit 2):
  - ✖ one-to-many relationship where 1-side has data not used in any other relationship,
  - ✖ introduce additional updating if duplicated data changed

Circumstances will consider denornalization:

It MUST serve good purpose – only when it enhances performance for an application or user group (which/who need to search frequently for some reference data).

(b)

EMPLOYEE_SCHEDULE (<u>Department ID, EmployeeID</u>,
EmployeeName, EmployeeAddress, PhoneM, Date)

(c) Drawbacks of denormalization:

- Increase chance of errors and inconsistencies: due to the omission in updating of redundant copies of the same data in a synchronized way
- Reintroduce anomalies
- Force reprogramming when business rules change