1. What is denormalization?

**Denormalization** is not normalizing a database to a fully normalized status. One or more tables may be left in a state that we know is not fully normalized.

1. When is denormalization justified?

Denormalization is justified when usability or performance considerations indicate that a fully normalized set of tables is not optimum.

1. Define the terms parent and child as they apply to tables in a database design and give an example of each.

A parent is a row on the one side of a 1:N relationship, while a child is a row on the many side of a 1:N relationship. For example, if DEPARTMENT and EMPLOYEE have a 1:N relationship, a row of DEPARTMENT is a *parent* and the rows of EMPLOYEE that pertain to that department are the *children*.

1. Show how to represent the 1:N relationship in your answer to question 5.9. Use an IE Crow’s Foot E-R diagram.

BOAT to RENTAL\_FEE is 1:N.

Assume relations:

**BOAT (LicenseNumber, Type, Length)**

**RENTAL\_FEE (ChargeNumber, ChargeDate, ChargeAmount).**

Place LicenseNumber in RENTAL-CHARGE as a foreign key.

**BOAT (LicenseNumber, Type, Length)**

**RENTAL\_FEE (ChargeNumber, ChargeDate, ChargeAmount, *LicenseNumber*)**

Where

**LicenseNumber in RENTAL\_FEE must exist in LicenseNumber in BOAT**



1. For your answer to question 5.15, describe a method for obtaining data for all the children, given the key of the parent. Describe a method for obtaining data for the parent, given a key of the child.

Given a value of LicenseNumber in BOAT [Parent], look up all rows in RENTAL\_FEE [Child] having that value for LicenseNumber.

Given a ChargeNumber, look up the row in RENTAL\_FEE [Child] having that number, and then use that value to look up BOAT [Parent] data to obtain the value of the LicenseNumber for the parent.

1. For your answer to question 5.15, code a SQL statement that creates a table that has all data from both tables.

SELECT \*

FROM BOAT, RENTAL\_FEE

WHERE BOAT.LicenseNumber = RENTAL\_FEE.LicenseNumber;

1. For a 1:N relationship, explain why you must place the key of the parent table in the child table, rather than place the key of the child table in the parent table.

There is literally not enough room in the parent, since we can only have one value per cell in the relational model.

1. Why is it not possible to represent N:M relationships with the same strategy used to represent 1:N relationships?

Because there is only room for one value per cell in a relation, there is literally not enough room. Thus neither of the two relations in relationship can hold foreign key values referencing the other relation.

1. What is an associative entity? What is an association relationship? Give an example of an association relationship other than one shown in this text. Illustrate your answer using an IE Crow’s Foot E-R diagram.

An **associative entity** is the equivalent of an intersection table with additional columns beyond the composite primary key. Although an “intersection entity” is not included in a data model, an associative entity can be, and indeed must be. The pattern of two (or more) strong entities related to each other via an associative entity is called an **association relationship**. We can turn the N:M BOAT and OWNER example into an association relationship by simply adding a non-key column to BOAT\_OWNER. For example, we can add a column named PercentOfOwnership.