Assume we have two relations R(A,B) and S(B,C). All three attributes (A, B, and C) are integer attributes. Assume that Relation R contains the following tuples: (1,2), (2,3), and (3,4). Assume that Relation S contains the following tuples (2,2), (2,3) and (5,1).

a) Is A a key for R. Answer (yes/no):
b) Is (B, C) a superkey for S. Answer (yes/no):
c) How many tuples are in the result of the cross-product between R and S? Answer (a number):

Consider the following relational schema (keys are underlined):

Product(pid, name, price, mfgr), Buys(cid, pid), Customer(cid, cname, age)

Write the following query **in relational algebra without using the division operator**: “Find the names of all customers who have purchased all products that are **not** manufactured by Sears.”

Why does the database system usually implement its own buffer manager (instead of relying on the operating system’s file system cache)? Give two different reasons.

What are the two main ideas in RAID? Describe for each of them **in one sentence** why they are used.

Draw a linear hashing index such that insertion of an arbitrary data entry will cause the Next Pointer to move to the first bucket. That is, in your index the Next pointer should **not** point to the first bucket, but insertion of an arbitrary data entry should cause movement to the first bucket.

Consider a B+-tree index of order d=3. The B+-tree does currently not contain any data entries (i.e., the index is completely empty). What is the minimum number of data entries that I have to insert into the tree in order to get a B+-tree of height 4?