Santiago Bermudez

ID: 301118090

CSC 134-02

Assignment 5

Due: 5/5/22

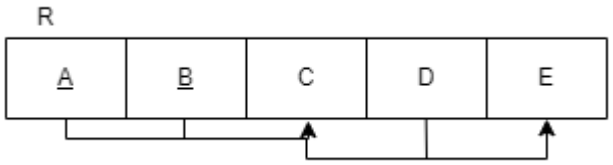
Question 1 (10 pts): Please define 1NF, 2NF, 3NF and BCNF, respectively.

1NF (*First Normal Form) is a database normalization form that simply requires that all attribute values be single atomic values, or that there are no multi-valued attributes. 2NF first requires that the tables be in 1NF and that there is no non-prime attribute partially dependent on any key. In other words, data in a relation should be based on the whole key, not part of the key for 2NF. 3NF requires that the tables be in 2NF and that there is NO non-prime attribute transitively dependent on any key. Basically, for 3NF, the data in a relation should be based on nothing but the key. BCNF requires that a relation be in 3NF and that the left side of every functional dependency in a relation must be a superkey.

Question 2 (10 pts): Please explain what the DB normalization process is.

Database normalization can be thought of as a "purification" process where we gradually improve the database design by making it achieve a higher normal form. To do this, we break down a relation into a number of smaller relations step by step. We do so with the following principles in mind: That any common attributes shared by two relations must be either foreign key or primary key, that there is to be no loss of attributes, and that we must preserve as many functional dependencies as possible. With database normalization, we want to have relations such that it is easy to explain its meaning and that redundancy is at the minimum.

Question 3 (10 pts): Consider a relation *R(A, B, C, D, E)* with the following functional dependencies:
{A, B} -> {C}
{C, D} -> {E}
Identify the key and explain your answer.



Key: {A, B}

I determined the key to be {A,B} because every other attribute in the relation either is directly determined by attributes A and/or B or is transitively determined by A and/or B through another attribute.

Question 4 (10 pts): Compare the techniques of data stripping and data mirroring used in RAID and their respective goals.

Data striping distributes data over multiple disks and allows multiple Read/Write heads to work in parallel. Data mirroring involves copying data onto more than one disk.
Data striping emphasizes speed when it comes to accessing data, whereas data mirroring focuses on backing up data and keeping it secure in the event that a disk fails or is damaged.

Question 5 (10 pts): Compare the following RAID levels: Level 0, Level 1, Level 5, and Level 10.

Level 0 just uses data striping (*explained above) and level 1 just uses data mirroring (*also explained above). Level 5 requires 3 or more disks and uses data striping and parity blocks across all disks. In level 5, data is striped across multiple disks with parity, so as to allow the rebuilding of data, should one of the disks fail. Level 10 requires a minimum of 4 disks and uses the principles of level 0 and level 1. With level 10, a set of 2 disks are mirrored with level 1, and both sets of disks would be striped with level 0. Basically, level 10 benefits from the speed of level 0 and fault tolerance of level 1, but with a significant storage cost.

Question 6 (50 pts): Consider a disk drive device with block size 512 bytes and data transfer rate is 512 bytes/msec. The disk drive rotates at a speed of 2400 RPM (rotations per minute). The average seek time is 30 msec.

(a) How much time does it take (on average) in msec to transfer a single block given its block address?

s (*seek time) = 30 msecs

rd (*rotational delay) = (1/2) * (1/p) min = 0.5 * (60/2400rpm) = 0.0125 or 12.5 msecs!

btt (*block transfer time) = B (*block size) / tr (*transfer rate) msec = 512bytes/(512 bytes/msec) = 1ms.

= s + rd + btt

= 30 + 12.5 + 1 = 43.5 msecs!

(b) Calculate the average time it would take to transfer 10 random blocks on disk and compare it with the time it would take to transfer 10 consecutive blocks.

Time to transfer 10 random blocks = 10 * (s + rd + btt) = 10 * 43.5 = 435 msec

Time to transfer 10 consecutive blocks = s + rd + (10 * btt) = 30 + 12.5 + (10*1) = 52.5 msec

(c) What is the time on average to search for a record in a file containing 100 blocks if the file blocks are stored on consecutive disk blocks? Suppose the file is unordered.

*For linear search, we search on average half the blocks.

s + rd + (50 * btt) = 30 + 12.5 + (50*1) = 92.5, but…

*If unordered or scattered. Then

50 * (s + rd + btt) = 50 * (30 + 12.5 + 1) = 2175 msec or 2.175 sec

(d) What is the time to read/write an entire file containing 1000 blocks if the file blocks are stored on consecutive disk blocks?

s + rd + (1000 * btt) = 30 + 12.5 + (1000*1) = 1042.5 msec

(e) What is the time to read/write an entire file containing 1000 blocks if we distribute the file blocks evenly over 10 disks using data stripping?

= 30 + 12.5 + ((1000/10) * btt) = 30 + 12.5 + (100*1) = 142.5 msec