CS 4347 — Database Systems, Homework 5

https://iniyanjoseph.com | iniyanijoseph@gmail.com

What is the difference between extendible and dynamic hashing? Provide an example to support your answer

In extendible hashing, the most significant d bits of the hashed data are used as the index for storing the data. In this model, overflows of the hashed data are dealt with by splitting the bucket into two. In contrast, in dynamic hashing, the entirety of the hash value is used to position it within a tree structure d. This means that there are no splitting or merging operations.
For example, imagine hash values 11 and 10. In dynamic hashing, there is never a hash collision. In contrast, with extendible hashing, if the local depth starts at 2, after the insertion of 10, the index containing 11 must be split before 10 is inserted. This guarantees no space is wasted.

What are the differences among primary, secondary, and clustering indexes? Which of the indexes are dense, and which are not?

A primary index is a file consisting of two attributes: A primary key (of the same data type as the ordering field) and a pointer to a disk block. Primary indices are sparse. A clustering index is used for speeding up access to records for a certain field. This is a dense index, since all possible values for the field are reserved in memory. A secondary key is used as a secondary means of accessing a record. This is similar, but distinct from a primary key, since it is based on a nonordering field of the records. This is also a unique key (and thus a candidate key). This index is dense.

Consider the following relation instance

	x	y	z
t1	1	4	2
t2	1	5	3
t3	1	5	3
t4	3	2	2
t5	1	5	6

Which of the following functional dependencies may exist in the above relation instance? If the functional dependency cannot hold, specify the tuples that cause the violation.

$XY \to Z$

The functional dependency cannot hold, becuase t_2 and t_3 conflict with t_5

$YZ \to X$

This functional dependency may hold

$Y \to Z$

This functional dependency may hold

$X \to Z$

This functional dependcy many not hold $t_1, t_2, \wedge t_5$ conflict with each other. Although t_2, t_3 do not conflict

$XZ \to Y$

This funcitonal dependency may hold

Given the relation schema $R(A, B, C, D, E, F)$ with key (A, B) and the following set of functional dependencies, $F = \{\{A, B\} \to \{C\}, \{B\} \to \{D\}, \{D\} \to \{E, F\}\}$. Is R in the second normal form? Explain your answer, if yes explain way, if not explain why and normalize it to the second normal form.

No, because $B \to D$ is a partial dependency. So we must now split this into two relations to normalize it

$$R_1(B, D) = \{B\} \to \{D\}$$

$R_2(A, B, C, E, F) = \{\{A, B\} \to \{C\}, \{D\} \to \{E, F\}\}$

Given the relation schema $R(A, B, C, D)$ with key (A) and the following functional dependencies, $F = \{\{A\} \to \{B, C\}, \{B, C\} \to \{D\}, \{D\} \to \{C\}\}$ Is R in the third normal form? Explain your answer. If not, explain why and normalize it to the second normal form.

This is not in the third normal form because $\{B, C\} \to \{D\}, \{D\} \to \{C\}$ are both transitive dependencies. To fix this, we must make new relations

$$R_1(A, B, C) = \{A\} \to \{B, C\}$$
$$R_2(B, C, D) = \{B, C\} \to \{D\}$$
$$R_3(D, C) = \{D\} \to \{C\}$$