

Kaijun He
CSC 553
Assignment 1

/*

Part 1, Question A

*/

select s.*

From Student s, enrolled e, course c

Where s.SID = e.StudentID AND e.CourseID = c.CID AND s.Career = 'UGRD' AND c.CourseNr > 400;

/*

Part 1, Question B

*/

select s.LastName, s.SID, c.Department

From Student s, enrolled e, course c

Where s.SID = e.StudentID AND e.CourseID = c.CID AND s.SSN IS NULL;

--Part 1, Question C

Select CID, CourseName, Department

From course

Where CourseName LIKE 'Theory%' OR CourseName LIKE '%Analysis';

--Part 1, Question D

--way one

Select S.LASTNAME, S.FIRSTNAME, S.SID

From Student s

Where s.SID NOT IN

(select studentID

From Enrolled

)

Kaijun He
CSC 553
Assignment 1

-- way two

Select S.LASTNAME, S.FIRSTNAME, S.SID

From Student s

LEFT JOIN Enrolled e ON e.studentID = s.SID

where e.StudentID IS NULL

-- Part 2

-- par2 A

Select model, year

From Cars

Where make = 'Honda' OR price >= 22000;

--Part 2 B

Select *

From Cars

Where make != 'Kia' OR price >=12000;

--part 2 C

Insert into Cars values (50,'Kia', 12000, '01/20/2019');

--part 2 D

Select model, year

From Cars

natural join Dealer on cid = did

where drevenue < 1000000 AND dstate = 'MA';

--part 2 E

select model

Right outer Join Warranty on cid = wid

Where cost <5500 AND Worigin IS NULL;

Part 4:

- A. Assume a table that uses 4096-byte blocks. If a relation has 120 rows and each row takes 50 bytes, how many blocks of storage does this table require?

ANS: number of rows fit in one block = $4096 / 50 = 81$ rows;

Number of blocks the table required = $120 \text{ rows} / 81 \text{ rows} = 2$ block;

- B. If we were to index a key of 16 bytes and assuming that each pointer requires 4 bytes (also using 4096-byte blocks), how many values and pointers can fit on a single block?

ANS: size for key + pointer = $4 + 16 = 20$ bytes;

Number of values and pointers = $4096 / 20 = 204$;

- C. If you had exactly 1 block for an index, how many values can it index? (i.e., how many pointers can it hold?)

ANS: we know that each pointer has 4 bytes from B, so number of values it can index = $4096 / 4 = 1024$;

- D. If you had a four-level index (root, two intermediate levels and a leaf level), how many values can it index? (at most)

P1 V1 P2 V2 P3 V3 P4

HINT: For your reference, picture above shows 3 values and 4 pointers. Assuming this many fit on a single DB page, a single-page index can index 3 values. With a two level index, it could index 12 values (4 pointers at top level, each pointing to a 3-pointer page for a total of 12 indexed values).

Note that a single page index in the example above cannot index 4 values and only indexes 3 values because each indexed entry at the leaf level is a (value, pointer) pair.

Kaijun He

CSC 553

Assignment 1

Ans: since we know the leaf level only index 3 values, so for four-level index at most numbers of values = $4*4*4*3 = 192$;

E. Explain, in your own terms, the difference between a dense and a sparse index.

ANS: Dense index: each data record has entry index, sparse index, for data file, only each page or data block has entry index, for examples, for a set of data in database, id from 1- 20, for dense index, each rows has an entry index, for sparse index, we divide the data into 4 block, 5 rows per block, then sparse index only have entry index for each block, index will be 1, 6, 11, 16. So dense index will be faster in general without optimizing data and sparse index will use less space, when doing search, insert or deletion events, it will be also faster than dense index.