

Quiz 9: Physical DB Design and NoSQL

Initial Score (out of 10)**Taken by:** _____**Name**

Student ID

- We will discuss the answers right after the quiz.
 - You are to self-grade and record your actual *initial* score (above) as we do so.
 - You should also record the correct answers for any problems that you miss.
 - A full 10 points for taking this quiz will be included as part of your quiz grade.
-

1. (5 points) Consider the following schema for an e-commerce application:

Suppliers(sid, sname, rating, city)

Inventory(sid, pid, quantity)

Parts(pid, pname, color)

The DBA in charge of the database system for this new e-commerce application never bothered to take CS122a, and so far they've actually declared no keys and have created no indexes. (Sheesh! Good help is so hard to find these days...) Come up with a good physical database design to support a workload that consists of the following expected mix of queries:

Q0: SELECT * FROM Parts WHERE pid = value1;

Q1: SELECT * FROM Parts WHERE pname LIKE '%value1%';

Q2: SELECT color, COUNT(*) FROM Parts GROUP BY color;

**Q3: SELECT P.pname, P.color FROM Parts P, Inventory I, Suppliers S
WHERE S.city = value1 AND S.sid = I.sid AND I.pid = P.pid;**

Q4: SELECT * FROM Suppliers WHERE sname LIKE 'value1%'

Q5: SELECT city, AVG(rating) FROM Suppliers GROUP BY city;

Suppose this data will be residing in a DBMS that only offers B+ Trees as its choice of index type. Come up with a good set of indexes for the data and the workload above – indicating, for each recommended index, the indexed columns and what reason, query, or queries above, has led you to recommend that index. Indicate your recommendations in the table below. Use the first column of the table to number the indexes from 1 to N for quick reference in subsequent parts of this problem. (We've entered the first row of the table for you. Not all rows need to be filled in.)

Index Number	Indexed Table	Indexed Column(s)	Motivating Query(s)
1	Parts	pid	Q0, Q3
2	Parts	color	Q2
3	Suppliers	city	Q3, Q5
4	Inventory	(sid, pid) or sid	Q3
5	Suppliers	sname	Q4

(2) (3 points) Your design above should have at least one entry for each of the database's tables. Indicate below (by index number) which of your indexes should be clustered, if any, and briefly say why.

(i) Clustered index(es) for Suppliers: 3 (better?) or 5: either will work.

(ii) Clustered index(es) for Parts: no clustered indexes on this table - not 1 (unique entry - primary key), not 2 (index-only plan query)

(iii) Clustered index(es) for Inventory: 4 - (sid, pid): un-clustered, sid: clustered

3. (2 points) One way to map the relational schema above into the NoSQL world would be to have one dataset corresponding to each of the tables above. There are at least two other ways that one might ponder representing this data given the flexibility of NoSQL; identify them both and briefly discuss their pros and cons relative to the 1NF approach.

(i) Suppliers contains a bag of (pid, quantity) pairs.
 pros: this can save join cost
 cons: objects gets big.

(ii) Parts contains a bag of (sid, quantity) pairs.

Overall, the best strategy for these tables is leave it as it is.