**Sample questions for Storage and Indexing**

**These are some sample questions. Solving only these problems are not sufficient. You have to go through the book.**

**Storage Management and Indexing**

Q.1: (a) Construct a 2-level (first level (inner level) dense and the second level (outer level) sparse) index structure on customer id (c-id) for the relation given below:

c-1

c-6

c-10

c-1

c-2

c-3

c-4

c-5

c-6

c-7

c-8

c-9

c-10

c-11

|  |  |  |
| --- | --- | --- |
| **c-id** | **Name** | **City** |
| C-1 | Name 1 | City 1 |
| C-2 | Name 2 | City 1 |
| C-3 | Name 3 | City 1 |
| C-4 | Name 4 | City 1 |
| C-5 | Name 5 | City 1 |
| C-6 | Name 6 | City 1 |
| C-7 | Name 7 | City 1 |
| C-8 | Name 8 | City 1 |
| C-9 | Name 9 | City 1 |
| C-10 | Name 10 | City 1 |
| C-11 | Name 11 | City 1 |
| C-12 | Name 12 | City 1 |

(b) Do you need to create index for city? Why?

(c ) create a sparse index on c-id and delete one record that affect the index structure. Delete another record that does not affect the index structure. With this example, explain the deletion performance in sparse index.

Answer: a

Similar problem has been solved in class. See the class note.

Answer b

No need to create index for city because only one search key and any query for city shall return the entire relation. So no need to create index on city.

Answer c

Discussed in class.

Q. 2: (a)

Given the relational schema as follows:

Student (id, NID, name, f-name, f-NID, m-name, m-NID, DOB, cgpa, tot-cred, uni-id, uni-name, uni-street, uni-city, house-no, street, city, d-no, d-name, building)

Takes (id, course-no, semester, year, grade)

Course (course-no, title, credit, pre-req)

The record size for student, takes and course are 400, 100 and 80 bytes respectively. The block size is 4 KB. Show the slotted page structure after storage of one tuple (record) from each relation as per the above mentioned order. Show the steps.

Answer:

You have to store 400 byte student tuple first, then takes tuple and then course tuple as follows:

Step 1: Storage of one student tuple of 400byte

1 400

3696 4096

Step 2: Storage of one takes tuple of 100byte

2 400 100

3596 4096 3696

Step 3: Similar as step 2 (Do yourself)

(b) Show the file structure with free list after deletion of C3, C5 and C6 from the customer relation of question 1.

Q. 3: There are six disks D1, D2, D3, D4, D5 and D6. You have to store data blocks B1, B2, B3, B4, B5. B6 and B7 into the disks. Show the storage of the blocks into disks using RAID level 1 and RAID level 5.

See the class note. Similar problem has been solved.

Q.4: Given the following relation schema for student. 3

Student (id, NID, name, f-name, f-NID, m-name, m-NID, DOB, cgpa, tot-cred, uni-id, uni-name, uni-street, uni-city, house-no, street, city, d-no, d-name, building)

ID is the primary key and the relation has been stored in the disk in sorted order of ID. You have created three indices on three attributes ID, uni-id and city. Identify which one is primary index and which one is secondary index? Which index can be sparse or dense and which index must be dense? Now fill up the following table.

|  |  |  |
| --- | --- | --- |
| Index Name | (Primary/Secondary) | Possible index (Dense/Sparse) |
| ID |  |  |
| Uni-id |  |  |
| City |  |  |

Q.5: Why is the disk access time is the main concern in developing algorithms for DBMS?

Answer: Disk access is much slower (in the range of millisecond) than the memory access (in the range of nanoseconds). That is why the disk access time is the main concern in developing algorithms for DBMS.

Normalization

Q. 1: Given the relation

Book\_Pub\_Author(Title, PubId, AuId, Price, AuAddress)

The functional dependencies are given below.

* 1. Key 🡪 {Title, PubId, AuId}
  2. {Title, PubId, AuID} 🡪 {Price}
  3. {AuID} 🡪 {AuAddress}

Which normal form is the relation? Normalize it up to 3NF.

Q. 2: Given the relation

**City\_Population(City, Street, HouseNumber, HouseColor, CityPopulation)**

* 1. **key** 🡪 **{City, Street, HouseNumber}**
  2. **{City, Street, HouseNumber}** 🡪 **{HouseColor}**
  3. **{City}** 🡪 **{CityPopulation}**

Which normal form is the relation? Normalize it up to 3NF.

Ans: Solved in the slide

Q.3. Given the relation R(A, B, C, D, E, F, G, H)

The functional dependencies are given as follows:

ABC 🡪 DEFGH

So ABC is the primary key. The other functional dependencies are given as follows:

BC 🡪D

ABC 🡪 E

E 🡪 F

E 🡪 G

Q. Which normal form R is?

Ans: The relation is in 1NF.

Q. Is the relation is 2NF? Why?

The relation is not 2NF because BC 🡪D is a partial dependency of non-key attribute. No other partial dependency.

Q. Transform it into 2NF.

Removing the partial dependency BC 🡪D we get

R1(B, C, D)

The primary key and other attributes except D forms

R2(A, B, C, E, F, G, H)

Q. Is R1 and R2 are in 3NF? Why?

Ans. R1 is in 3NF because there is no transitive dependency.

R2 is not in 3NF because there are transitive dependency (non-key attribute have dependency to one another) for the following:

ABC 🡪 E, E 🡪 F So ABC 🡪 E (transitive dependency)

ABC 🡪 E, E 🡪 G, So ABC 🡪 G (transitive dependency)

Q. Transform R2 into 3NF.

Ans. Removing the attributes related to transitive dependencies (non-key attribute have dependency to one another) i.e., E and G and adding primary key for them as E, we get

R2-1(E, F, G)

With the remaining attributes of R2, we get

R2-2(A, B, C, E, H)

These two relations are in 3NF.

Q. is R1, R2-1, R2-2 in BCNF? Why?

Ans: Yes, all three relations are in BCNF because there are one key in each relation and no functional dependencies exists within the attributes of the key.

**Practice problem:** The relation schema and the functional dependencies are given as follows:

R(A, B, C, D, E, F, G, H, I)

ABC → DEFGHI

B → G, B → H, D → E, D → I

Normalize up to 3NF.

|  |
| --- |
| Transaction Management |
| Q. 1: **:** The concurrent schedule of transactions T1 and T2 are given below:   |  |  | | --- | --- | | T1 | T2 | | Read (A) |  | |  | Write (A) | | Write (B) |  | |  | Write (B) | | Write (C) |  | |  | Read (C ) |   Find equivalent serial schedule by swapping non-conflicting instructions step by step. |

Q. 2**:** The concurrent schedule of transactions T1 and T2 are given below:

|  |  |
| --- | --- |
| T1 | T2 |
| Read (A) |  |
|  | Write (A) |
| Write (B) |  |
|  | Write (B) |
| Write (C) |  |
|  | Write (C ) |
| Read (C) |  |

Draw the precedence graph for the above schedule and find whether the schedule is serializable or not.

Ans:

As there is a cycle, the schedule is not serializable.

Sample Short questions

|  |
| --- |
| Q. 1:   1. In a hard disk, there are 10 platters with 20 surface, 10000 tracks per surface, 1000 sectors per track and 1 kilo bytes per sector. What is the disk capacity? 2. The time for the read-write head to traverse from outer most track to inner most track is 5ms time of the spindle for a complete rotation is 2ms. What is the worst case and the best case access time? |
| Q. 2:  There are disks of 1TB each. You need to set up a database storage system that requires 4TB effective storage. Find the minimum number of disks required in i) RAID 0, ii) RAID 1 and iii)RAID 5. Which RAID level will you suggest for a system with less cost and also no data loss? |
| Q. 3:  Give the following relation:  Person (NID, name, DOB, street, city)  Person relation is stored physically sorted order of NID. Answer the following questions.   1. Which type of indexing is created for DOB (primary/secondary)? 2. Which type of indexing is created for ID (primary/secondary)? 3. Why secondary index must be dense index? |
| Q. 4:  Write the abbreviation of ACID in transaction. Show the transaction states in a diagram. |