Homework 3

Lior Trachtman ID:203835806

1. **Table Normalization**

**1NF:**Separate the days to atomic parts, and set ClassID and Day together to be the primary keys cause can't be duplicate values that have the same ClassID and Day

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ClassID | SID | Day | ProfID | Phone | Room | Grade | Grader | Topic |
| 10127 | 145 | T | DA001 | x201 | 202 | 89 | 34459 | DBMS |
| 10127 | 145 | Th | DA001 | x201 | 202 | 89 | 34459 | DBMS |
| 10121 | 145 | M | KM002 | x320 | 201 | 75 | 04366 | Algorithms |
| 10121 | 145 | W | KM002 | x320 | 201 | 75 | 04366 | Algorithms |
| 10127 | 287 | T | DA001 | x201 | 202 | 93 | 34459 | DBMS |
| 10127 | 287 | Th | DA001 | x201 | 202 | 93 | 34459 | DBMS |
| 10303 | 396 | F | KM002 | x320 | 301 | 82 | 87254 | OS |
| 10303 | 396 | M | KM002 | x320 | 301 | 82 | 87254 | OS |
| 10303 | 396 | W | KM002 | x320 | 301 | 82 | 87254 | OS |
| 10334 | 396 | M | DS003 | X518 | 113 | 100 | 34459 | Compilers |
| 10334 | 396 | T | DS003 | X518 | 113 | 100 | 34459 | Compilers |
| 10334 | 396 | W | DS003 | X518 | 113 | 100 | 34459 | Compilers |
| 10334 | 396 | Th | DS003 | X518 | 113 | 100 | 34459 | Compilers |
| 10334 | 396 | F | DS003 | X518 | 113 | 100 | 34459 | Compilers |

**2NF:**

Every non-key attribute is **functionally dependent** on the entire key of the table

Room topic and grader **functionally dependent** on ClassID and Days

Phone is **functionally dependent** on ProfID

Professor can be teach many Topics and topic can be teach be many professors.

And grade is **functionally dependent** on SID and ClassID.

|  |  |  |  |
| --- | --- | --- | --- |
| ClassID | Days | Room | Topic |

|  |  |
| --- | --- |
| ProfID | Phone |

|  |  |
| --- | --- |
| ClassID | ProfID |

|  |  |  |  |
| --- | --- | --- | --- |
| SID | ClassID\* | Grade | Grader |

**3NF:**  
Non-key attributes are not dependent on attributes that are not keys

Well, in the 2nf we already did this,

|  |  |  |  |
| --- | --- | --- | --- |
| ClassID | Days | Room | Topic |

|  |  |
| --- | --- |
| ProfID | Phone |

|  |  |
| --- | --- |
| ClassID | ProfID |

|  |  |  |  |
| --- | --- | --- | --- |
| SID | ClassID\* | Grade | Grader |

As we can see, phone is a non-key attribute, that not dependent on attributes that are not keys, and the same for grade, room, topic and grader

1. **Relational Algebra**

SUPPLIERS (Sid, Name, City)

PARTS (Pid, Pname, Color)

CATALOG (Sid, Pid, Cost)

1. πName((σColor='red'(PARTS)⋈( CATALOG) ⋈ SUPPLIERS)
2. πName((σColor='red'(PARTS)⋈ σCost<100(CATALOG)) ⋈ SUPPLIERS)
3. πSid(σColor='red' AND σCost<100(SUPPLIERS ⋈ CATALOG ⋈ PARTS) ) ∩

πSid(σColor='green' AND σCost<100(SUPPLIERS ⋈ CATALOG ⋈ PARTS) )

1. Way one:

πName(σColor='red' AND σCost<100(SUPPLIERS ⋈ CATALOG ⋈ PARTS) ) ∩

πName(σColor='green' AND σCost<100(SUPPLIERS ⋈ CATALOG ⋈ PARTS) Way two:

πName(SUPPLIERS ⋈ (σc1.Color='red' AND σc2.Color='green' AND σc1.Cost<100 AND σc2.Cost<100(ρc1(CATALOG ⋈ PARTS) ⋈ ρc2(CATALOG ⋈ PARTS) ⋈ σc1.Sid= σc2.Sid)))

1. Cid1🡨 CATALOG

Cid2🡨 CATALOG

πCid1.Sid,Cid2,Sid(Cid1.Pid=Cid2.Pid AND Cid1.Cost<Cid2.Cost)

1. **Storage and indexing**
2. A table has 300,000 rows of 100 bytes each, it mean 1 row is 100 bytes and 10 rows is 1000 bytes and 10 rows is fit on a page.
3. To store the table we will need 30,000 pages.
4. To read a page it takes 15\*10^-3. We need to read 30,000 pages so it will take 30,000\*15\*10^-3 = 450 sec.
5. The worst case time for reading all the rows in some random order is

10 \*30,000\*15\*10^-3 = 4,500 sec.