Homework 3

[xix90@pitt.edu](mailto:xix90@pitt.edu)

**problem1:**

图示

描述已自动生成

**Problem2:**

**CREATE SCHEMA ‘online\_conference\_review\_system’;**

**CREATE TABLE authors (**

**email VARCHAR(20) NOT NULL,**

**name VARCHAR(20) DEFAULT NULL,**

**phone\_number VARCHAR(20) DEFAULT NULL,**

**website VARCHAR(20) DEFAULT NULL**

**PRIMARY KEY(email)**

**)**

**CREATE TABLE organizers(**

**email VARCHAR(20) NOT NULL,**

**name VARCHAR(20) DEFAULT NULL,**

**phone\_number VARCHAR(20) DEFAULT NULL,**

**website VARCHAR(20) DEFAULT NULL**

**PRIMARY KEY(email)**

**)**

**CREATE TABLE Reviewers(**

**email VARCHAR(20) NOT NULL,**

**name VARCHAR(20) DEFAULT NULL,**

**PRIMARY KEY(email)**

**)**

**CREATE TABLE conference(**

**id INT NOT NULL,**

**name VARCHAR(20),**

**organizers VARCHAR(50),**

**submission\_page INT,**

**PRIMARY KEY(id),**

**Foreign key (id) references paper (conference\_id),**

**Foreign key (organizers) references organizer (email)**

**)**

**CREATE TABLE Paper(**

**id INT NOT NULL,**

**title VARCHAR(50) DEFAULT NULL,**

**authors VARCHAR(50) DEFAULT NULL,**

**key\_word VARCHAR(50) DEFAULT NULL,**

**submitted\_time DATE,**

**conference\_id VARCHAR(50) DEFAULT NULL,**

**reviewers VARCHAR(50) DEFAULT NULL,**

**download\_link VARCHAR(50) DEALLOCATE NULL,**

**PRIMARY KEY(id),**

**Foreign key (authors) references authors (email),**

**Foreign key (reviewers) references Reviewers (email),**

**Foreign key (conference\_id) references conference (id)**

**)**

**Problem3:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FD** | **Satisfied by R** | **Hold on R** | **Trivial** |
| A🡪B | No | No | No |
| B🡪A | No | No | No |
| AC🡪D | Yes | Unknown | No |
| ABD🡪B | Yes | Yes | Yes |
| AC🡪B | No | No | No |
| AD🡪B | No | No | No |
| C🡪ABC | No | No | No |
| BC🡪D | Yes | Unknown | No |
| BD🡪D | Yes | Yes | Yes |
| BD🡪A | No | No | No |

**Problem4:**

|  |  |  |
| --- | --- | --- |
| **FD** | **YES/NO** | **Proof if yes** |
| **ABC🡪G** | YES | B🡪DE, AC🡪E, ABC🡪DE, (1&2) ABC🡪D (Decomp of 3)  CD🡪GF(Decomp), ABC🡪CD(Decomp)  ABC🡪G(Trans) |
| **AC🡪F** | NO |  |
| **BF🡪G** | YES | BF🡪DEF, F🡪C, CD🡪GF,  BF🡪CDEFG, BF🡪CDEF  BF🡪G(Decomp) |
| **BCD🡪F** | YES | BCD🡪DECD, BCD🡪DECDFG  BCD🡪F |
| **ABC🡪DEF** | YES | ABCF🡪EBF, B🡪DE, ABCF 🡪BEDF  ABCF🡪DEF |

**Problem5:**

|  |  |  |
| --- | --- | --- |
| **Decomposition** | **Lossless(yes/no)** | **why** |
| **R1(ABF) and R2(CDE)** | **NO** |  |
| **R1(ABCEF) and R2(CDE)** | **NO** | **CE isn’t a key in these two sets** |
| **R1(ABDE) and R2(ACDF)** | **NO** | **AD isn’t a key I R1** |
| **R1(ACDF) and R2(BCDE)** | **YES** | **CD is a key in R1 and R2**  **CD🡪CD, D🡪A, C🡪BF**  **Thus CD🡪ACDF**  **C🡪F**  **so Lossless** |
| **R1(ABEF) and R2(BCDF)** | **NO** | **BF isn’t a key in these two sets** |

**Problem6:**

|  |  |  |
| --- | --- | --- |
| **Relation, FD** | **Answer(a, b or c)** | **Solution** |
| **R1(A, B, C, D)**  **{AB🡪C, C🡪D}** | **c** | **A🡪ABC, C🡪D, AB🡪ABCD**  **AB is a primary key, and C is not a key.**  **R1 is neither 3NF nor BCNF.** |
| **R2(A, B, C, D)**  **{AC🡪BD}** | **a, b** | **AC🡪BD, AC🡪ABCD,**  **AC is the primary key.**  **R2 is both in 3NF and BCNF.** |
| **R3(A, B, C, D)**  **{AB🡪CD, D🡪A}** | **b** | **AB🡪CD, D🡪A, AB🡪ABCD**  **AB is primary key. D isn’t a key.**  **A depends on D.**  **R3 is 3NF.** |
| **R4(A, B, C, D, E)**  **{AC🡪D, D🡪B}** | **c** | **AC🡪D, D🡪B**  **both not primary key.**  **R4 is neither 3NF nor BCNF.** |
| **R5(A, B, C, D, E)**  **{A🡪CE, D🡪CE}** | **c** | **A🡪CE, D🡪CE, D🡪CDE, A🡪CE**  **A and D both are not primary key.**  **R5 is neither 3NF nor BCNF.** |

**Problem7:**

**A:5 kBytes.**

**Running table T needs to take 1 block size in the disk, so it will take 5 kBytes.**

**Problem8:**

**A:300 Mbytes.**

**According to the equation: n\*size\_of\_tuple = block\_size.**

**Dbsm has to read the whole table, so there are 300 Mbytes for this query to be read.**

**Problem9:**

**A:61440 blocks, 300 Mbytes(314572800bytes).**

**Firstly, it will ready 300 Mbytes= 30010241024bytes=314572800bytes from the disk.**

**So, it will be read 61440 blocks from the disk.**

**problem10:**

**the clustered index will be included in the table T, so there need 8 Kbytes equals to 2 blocks.**

**Best case, there will have 2 blocks + n tuples = 2 + 61440 = 61442 blocks**

**If all the num != 500, so there will be no data need to be read, then the blocks are 2.**

**problem11:**

**图片包含 日程表

描述已自动生成**