Name - Ayush Jain
SAP ID - 60004200132
Div - B / B1

15/01/2022

DBMS - Term Test 2

Solutions :

→ 3) 1) A transcation is a single logical unit of work which accesses and possibly modifies the contents of a database.
2) A transcation is a very small unit of program and it may contain several low level ~~keys~~ tasks. A transcation in a database system must contain Atomicity, Consistency, Isolation and durability - commonly known as ACID properties - in order to ensure accuracy, completeness and data integrity.
3) Example :

Let say your account in A and your friend's account is B, you are transferring 10000 from A to B, the steps of transcations are :

```
1   R(A);
2   A = A - 10000;
3   W(A);
4   R(B);
5   B = B + 10000;
6   W(B);
```

In the above transcation R ~~offers~~ refers to the read operation and W refers to write operation.
4) ACID transcations ensure the highest possible data reliability and integrity. They ensure that your data

never falls into an inconsistent state because of an operation that only partially completes.

5) For example, without ACID transcations, if you were writing some data to a database table, but the power went out unexpectedly, it's possible that only some of your data would have been saved, while some of it would not. Now your database is in an inconsistent state that is very difficult and time consuming to recover from.

1) $R_1(A, B, C, D, E, F, G, H, I, J, K, L)$

$F = \{ A \rightarrow BCDE$

$AFI \rightarrow JKL$

$F \rightarrow GH$

$\}$

$(AFI)^+ = \{ A\ B\ C\ D\ E\ F\ I\ J\ K\ L\ G\ H \}$

$\therefore (AFI)$ is candidate key.

In 3NF partial and transitive functional dependencies are not allowed.

$A \rightarrow BCDE$

$F \rightarrow GH$

are partial dependency so to convert the relation schema into 3NF

R_1 (A B C D E)

R_2 (F G H)

R_3 (A F I J K L)

∴ The decomposed table

R_1 (A B C D E), R_2 (F G H), R_3 (A F I J K L) are in second Normal form 2NF

Since there are no transitive functional dependencies,

R_1 (A B C D E), R_2 (F G H), R_3 (A F I J K L) are in 3NF.

2)

a)

```
CREATE TABLE Ticket (
    Ticketid INT,
    date DATE,
    source VARCHAR(30),
    destination VARCHAR(30))
    amount INT,
  Payment-method VARCHAR(20),
   Customer_id INT,
   PRIMARY KEY (Ticketid)
  );
```

```
CREATE TABLE Customer (
    customer_id   INT,
    customer_name VARCHAR(20),
     age          INT ,
    Address       VARCHAR (255),
     City         VARCHAR (50),
    PRIMARY KEY(Customer_id)
   );
```

b)
```
ALTER TABLE Ticket
ADD FOREIGN KEY (Customer_id)
REFERENCES Customer (customer_id) ;
```

c)
```
INSERT INTO Customer (Customer_id, Customer_name, age,
    Address, City )
VALUES ( 5, " John ', 19, " 60 feet Road', ' Thane' ) ;

INSERT INTO Customer
VALUES (6, ' Derer', 25, " Carter Road', ' Mumbai' );

INSERT INTO Ticket values
   (10, " 2020-10-05', ' Mumbai', ' Amemdabad', '20000, 'Net-banking',
                5 );
   (50, " 2020-10-08', ' Mumbai' ' UP', 25000, 'Debit-card', c);
```

d) Select * from Ticket where customer.city = 'Mumbai';

e) Alter table customer
Add contact_number BIGINT;

f) Select count(*) from ticket
where ticket.customer-id = customer.customer-id;