

DBMS End Sem

Name : Pranav K M

Roll no: 19BCS088

Page : 1

3) A DBMS is typically shared among many users. Transactions from these users can be interleaved to improve the execution time of users' queries. By interleaving queries, users do not have to wait for other user's transactions to complete fully before their own transaction begins. Without interleaving, if user A begins a transaction that will take 10 seconds to complete, and user B wants to begin a transaction, user B would have to wait an additional 10 seconds for user A's transaction to complete before the database would begin processing user B's request.

4) (a) A user must guarantee that his or her transaction does not corrupt data or insert nonsense in the database. For example, in banking database, a user must guarantee that a cash withdrawal transaction accurately models the amount a person removes from his or her account. A database application would be worthless if a person removed 20 rupees from ATM but transaction is set their balance to zero.

(b) A DBMS must guarantee that transactions are executed fully and independently of other transactions. An essential property of a DBMS is that a transaction should execute atomically, or as if it is the ~~only~~ only transaction running. Also ~~transac~~ transactions will either complete fully or will be aborted and the database returned to its initial state. This ensures that the database remains consistent.

2)

- * DDL is important in representing information in DBMS because it is used to describe external and logical schemas.
- * ~~The~~ DML is used to access and update data it not important for representing the data.

7) $P(R_1, \text{Catalog})$

$P(R_2, \text{Catalog})$

$$\pi_{R_1.pid} \circ \pi_{R_1.sid = R_2.pid} \wedge R_1.sid = R_2.sid (R_1 \times R_2)$$

Using the following.

SID	PID	Cost
1	1	₹ 10.00
1	1	₹ 9.00
2	3	₹ 34.00
3	1	₹ 11.00

$R_1 \times R_2$ gives us.

SID	PID	Cost	SID	PID	Cost
1	1	₹ 10.00	1	1	₹ 10.00
1	1	₹ 10.00	2	1	₹ 9.00
1	1	₹ 10.00	2	3	₹ 34.00
1	1	₹ 10.00	3	1	₹ 11.00
2	1	₹ 10.00	1	1	₹ 10.00
2	1	₹ 9.00	2	1	₹ 9.00
2	1	₹ 9.00	2	3	₹ 34.00
2	1	₹ 9.00	3	1	₹ 11.00
2	3	₹ 34.00	1	1	₹ 10.00
2	3	₹ 34.00	2	1	₹ 9.00
2	3	₹ 34.00	2	3	₹ 34.00
2	3	₹ 34.00	3	1	₹ 11.00
3	1	₹ 11.00	1	1	₹ 10.00
3	1	₹ 11.00	2	1	₹ 9.00
3	1	₹ 11.00	2	3	₹ 34.00
3	1	₹ 11.00	3	1	₹ 11.00

$\sigma_{R1.pid = R2.pid}$ gives us

SID	PID	Cost	SID	PID	Cost
1	1	₹ 10.00	1	1	₹ 10.00
1	1	₹ 10.00	2	1	₹ 9.00
1	1	₹ 10.00	3	1	₹ 11.00
2	1	₹ 9.00	1	1	₹ 10.00
2	1	₹ 9.00	2	1	₹ 9.00
2	1	₹ 9.00	3	1	₹ 11.00
2	3	₹ 34.00	2	3	₹ 34.00
3	1	₹ 11.00	1	1	₹ 10.00
3	1	₹ 11.00	2	1	₹ 9.00
3	1	₹ 11.00	3	1	₹ 11.00

$\sigma_{R1.pid = R2.pid \wedge R1.sid = R2.sid}$ gives us:

SID	PID	Cost	SID	PID	Cost
1	1	₹ 10.00	2	1	₹ 9.00
1	1	₹ 10.00	3	1	₹ 11.00
2	1	₹ 9.00	1	1	₹ 10.00
2	1	₹ 9.00	3	1	₹ 11.00
3	1	₹ 11.00	1	1	₹ 10.00
3	1	₹ 11.00	2	1	₹ 9.00

SQL

SELECT
FROM
WHERE

C.pid
Catalog C
EXISTS

(SELECT
FROM
WHERE

C1.sid
Catalog C1

C1.pid = C.pid AND ~~C1.sid~~
C1.sid \neq C.sid)

1) Using empname as a clustered index is possible only when every employee will have a unique name. If this is ensured, the tuples will be organized according ~~emp~~ to empname alphabetically.

2) Using empid as a clustered index is definitely possible considering everyone already has a uniqueid assigned to them. The tuples will be organized according to empid.

Using both empname & empid as a clustered indexes may not be possible but it is possible to have one clustered index and one non clustered index.

9) The following view on Emp can be updated automatically by updating

Emp:

```
CREATE VIEW Senior_Emp (eid, name, age, salary).  
CREATE VIEW Fresher (eid, name, age, salary)  
AS SELECT E.eid, E.ename, E.age, E.salary  
FROM EMP E  
WHERE E.age > 25
```