DBMS End Sem

Name : Pranav K M

Roll no: 19BCS088

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 withdraw 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 transaction running. Also ~~transactions~~ 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.
* ~~It's~~ DML is used to access and update data it not important for representing the data.

7) ρ (R1, Catalog)

ρ (R2, Catalog)

$\pi_{R1.pid}\ \sigma_{R1.pid = R2.pid \wedge R1.sid != R2.sid}\ (R1 \times R2)$

Using the following.

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

R1 x R2 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	₹ 9.00	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   C.pid
FROM     Catalog C
WHERE    EXISTS (SELECT   C1.sid
                 FROM     Catalog C1
                 WHERE    C1.pid = C.pid AND ~~C1.sid~~
                          C1.sid ≠ 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 to empname alphabetically.

Using empid as a clustered index is definitely possible considering everyone already has a uniqueid designed 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
```

8) All supplers selling parts of red color which costs less than 100.

5) Examples of non-candidate keys include the following {name}, {age}, {Note that {gpa} can not be declared a non candidate key from this evidence (even though common sense tells us that clearly more than 1 student have the same grade point average.)

You cannot determine a key of a relation given only one instance of the relation. The fact that the instance is "legal" is immaterial. A candidate key, as defined here, is a key, not something that only might be a key. The instance is just one possible snapshot of the relation. At other times the same relation may have an instance that contains a totally different set of tuples and we cannot predict.