

Divyansh Tripathi

19BCS039

DBMS - Endsem

1. > Using empname as a ~~clustered~~ 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 has a unique id assigned to them. The tuples will be organized according to empid.
- Using both empid & empname as a clustered Indexes may not be possible ~~it~~ but it is possible to have one clustered Index and one non-clustered Index.

2. > DDL is important in representing information in DBMS

because it is used to describe external and logical schemas.

- DML is used to access and update data. it is not important for representing the data.

3. > True, DBMS interleave the actions of different transactions instead of executing transactions one after the other. Because 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. 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 others transactions to complete fully before their own transaction begins.

4. > a. > A user must guarantee that his or her transaction does not corrupt data or insert nonsense in the database. For example, in a 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 some amount from an ATM but the transaction set their balance to 0.

b. > A DBMS must guarantee that transactions are executed fully and independently of other transactions. An essential

Divyansh Tripathi
19B1S039

property of DBMS is that a transaction should execute atomically i.e. as if it is the only transaction running.

Also, transactions will either complete fully, or will be aborted and the database returned to its initial state. This ensures that database remains consistent.

2.) The following view on Emp can be updated automatically by updating Emp:

```
CREATE VIEW
```

5.) Yes we can determine the key of relation with the help of instance. e.g. In a one to many relation we can consider the column/attribute with unique values as a primary key.

```
6.) CREATE CLUSTERED INDEX Ix_empname_index ON  
STUDENTTable (StudentName DESC)  
SELECT Email FROM STUDENTTable
```

This query displays all the Emails in descending order of the Student name. First the ~~table~~ table gets sorted based on Student name in descending order then select query displays the email in that order.

b.)

Student ID	Student Name	Email	Age	Age
1005	Krishna	Krishna@xyz.com		22
1030	John	NULL		23
1020	John	Jh@xyz.com		22

7) Let the two suppliers be R_1 & R_2 .

$P(R_1, \text{catalog})$

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

$$\text{Join } \pi_{R_1.\text{pid}} \sigma_{R_1.\text{pid} = R_2.\text{pid} \wedge R_1.\text{sid} \neq R_2.\text{sid}} (R_1 \times R_2)$$

using the following

SID	PID	Cost
1	1	1000
2	1	900
2	3	3400
3	1	1100

$R_1 \times R_2$ gives us :-

SID	PID	Cost	SID	PID	Cost
1	1	1000	1	1	1000
1	1	1000	2	1	900
1	1	1000	2	3	3400
1	1	1000	3	1	1100
2	1	900	1	1	1000
2	1	900	2	1	900
2	1	900	2	3	3400
2	1	900	3	1	1100
2	3	3400	1	1	1000
2	3	3400	2	1	900
2	3	3400	2	3	3400
2	3	3400	3	1	1100
3	1	1100	1	1	1000
3	1	1100	2	1	900
3	1	1100	2	3	3400
3	1	1100	3	1	1100

$\nabla R_1.pid = R_2.pid$ gives us:-

SID	PID	Cost	SID	PID	Cost
1	1	1000	1	1	1000
1	1	1000	2	1	900
1	1	1000	3	1	1100
2	1	900	1	1	1000
2	1	900	2	1	900
2	1	900	3	1	1100
2	3	3400	2	3	3400
3	1	1100	1	1	1000
3	1	1100	2	1	900
3	1	1100	3	1	1100

$\nabla R_1.pid = R_2.pid \wedge R_1.sid \neq R_2.sid$ gives us

SID	PID	Cost	SID	PID	Cost
1	1	1000	2	1	900
1	1	1000	3	1	1100
2	1	900	1	1	1000
2	1	900	3	1	1100
3	1	1100	1	1	1000
3	1	1100	2	1	900

SOL:-

```

SELECT C.sid
FROM Catalog C
WHERE EXISTS (SELECT C1.sid
FROM Catalog C1

```


where θ

WHERE $C1.PID = C.PID$ AND $C1.SID \neq C.SID$

8.7 Invalid query:-

Explanation:- This relational algebra statement does not return anything because of the sequence of projection operators. Once the SID is projected, it is the only field in the set. Therefore, projecting on same will not return anything.

9.7 The following view on Emp can be updated ~~can~~ automatically by updating Emp:-

```
CREATE VIEW Senior Emp (eid, name, age, salary)
AS SELECT E.eid, E.ename, E.age, E.salary
FROM Emp E
WHERE E.age > 50.
```