DBMS END SEMESTER

VARUN.B.G
19BCS111

1. Using empname as a clustered index is possible only when every employee will have a unique name. If this is checked, the employees will be arranged ~~automatically~~ accordingly in alphabetical order

Using empid as a clustered index is definitely possible because they would already have unique-id alloted to each employed. They would be arranged according to emp-id.

Both emp_name and emp-id. We can have one as clustered index the other as 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 update and access data. It is not important for representing the data

VARUN.B.G
19BCS111

3. Yes, the above mentioned statement is true. DBMS interleaves the actions of different transactions to improve the execution time of user's queries. By interleaving users don't have to wait for other user's transaction to complete fully before their own transaction begins. Without interleaving if user 1 starts his transaction that takes 5 seconds to complete. Then user 2 has to wait for 5 seconds for his transactions to begin

4. (6 marks)

a) A user must ensure that his or her transaction should not damage the database. It should not corrupt the database. Taking the example of Banking system mentioned above in the question. A user must guarantee that a cash withdraw transaction accurately models the amount a person removes from his or her account. It would be pointless if a person removes ₹1000 but his balance is reduced by ₹2000.

b) A DBMS must guarantee that transaction 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

the only transaction running. The transaction should either complete fully or will be aborted and the database should be returned to it's initial state. This ensures that the database remains consistent.

25.

a) DDL is important in representing information in DBMS because it is used to describe external and logical schemas

b) DML is used to update and access data. It is not important for representing the data.

5. Yes, we can determine the key of relation with the help of instance. Eg:- In a one to many relation we can consider the column / attribute with unique values as a primary key.

VARUN.B.G
19BCS111

6.

(1) * Create a clustered Index studentName_index on student (studentName ASC)

=> it will create Index on student Name where student is table name.

* Select Email from student

Output :-

Email
Vn@ abc.com
bg@ abc.com
varun@ abc.com

VARUN·B·G
19BCS111

7.

P(R1, Catalog)
P(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	₹100
2	1	₹ 90
2	3	₹ 340
3	1	₹ 110

R1 X R2 gives us

SID	PID	Cost	SID	PID	Cost
1	1	₹100	1	1	₹100
1	1	₹100	2	1	₹90
1	1	₹100	2	3	₹340
1	1	₹100	3	1	₹110
2	1	₹90	1	1	₹100
2	1	₹90	2	1	₹90
2	1	₹90	2	3	₹340
2	1	₹90	3	1	₹110
2	3	₹340	1	1	₹100
2	3	₹340	2	1	₹90
2	3	₹340	2	3	₹340
2	3	₹340	3	1	₹110

VARUN.B.G
19BCS111

3	1	₹110	1	1	₹ 100
3	1	₹110	2	1	₹ 90
3	1	₹110	2	3	₹ 340
3	1	₹110	3	1	₹ 110

σ R1.pid=R2.pid gives us.

SID	PID	Cost	SID	PID	Cost
1	1	₹100	1	1	₹100
1	1	₹100	2	1	₹90
1	1	₹100	3	1	₹110
2	1	₹90	1	1	₹100
2	1	₹90	2	1	₹90
2	1	₹90	3	1	₹110
2	3	₹340	2	3	₹340
3	1	₹110	1	1	₹100
3	1	₹110	2	1	₹90
3	1	₹110	3	1	₹110

σ R1.pid=R2.pid ∧ R1.sid!=R2.sid gives us

SID	PID	Cost	SID	PID	Cost
1	1	₹100	2	1	₹90
1	1	₹100	3	1	₹110
2	1	₹90	1	1	₹100
2	1	₹90	3	1	₹110
3	1	₹110	1	1	₹100
3	1	₹110	2	1	₹90

Projecting on PID gives us a single part number - 1

SQL :-

VARUN.B.G

19BCS111

```
SELECT R1.sid
FROM Catalog R1
WHERE EXISTS (SELECT R1.sid
              FROM catalog R1
              WHERE R1.pid = R2.pid AND R1.sid != R2.sid)
```

8) It is a invalid query.

⇒ 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. Emp (eid : int, ename : string, age : int, salary : real)

Here, the new view must be updatable

```
CREATE OR REPLACE VIEW
        [Emp-test] AS

SELECT *
FROM Emp
```

The view can be updated using these commands

```
UPDATE  column-name
SET     new-value
```

By using, CREATE or REPLACE and UPDATE SET the views are updated.