End-Semester Examination (Autumn'2018) IT 214 Database Management Systems	B

Student ID: 201601049 Name: MONIL SONI

Time: 90 minutes Max Points: 115

IMPORTANT NOTE:
1. There are six pages make sure that you have them all in your set.
2. You need to answer all questions in question paper itself.
3. Write answers neat and clean. Over-writing is not allowed.

Marks

1	2	3	4	5	6	7	8	9	~~10~~	
7½	19.5	5	10.5	5	5	10	7.5	20		90

1. Tick the correct option (there can be multiple options be correct; check them all) – [10]

i. Unbound cursor can be bound with dynamic query <Yes/No>

ii. TP_OP is a <implicit local variable/implicit parameter variable/global variable/ no such variable>

iii. Embedded SQL provides an environment in which SQL statements are sent to DBMS for execution <Yes/No/both-depends on configuration>

iv. Using "CREATE ASSERTION", we can create constraints that span to multiple tables <True/False/No such command>

v. In case of B+-tree based index, index scan mean "sequential scan of leaf nodes" <True/False/No such operation>

vi. Heap files are roughly good for operations <INSERT/DELETE/SEACRH>

vii. Secondary index can be <dense/clustered/b+-tree/sparse>

viii. Participation of a weak entity with identifying relationship is always total [True/False]

ix. Bloated index refers to index having <too much void spaces in blocks/ too much overflow/index corrupted /all of these>

x. Materialized view hold data <true/false> _____

If yes, what data? Data obtained by querying the 'view', query statement

2. Fill in the blanks- [30]

i. Which property of schedule ensures **Isolation** Serializability

ii. Which property of schedule ensures **Atomicity** Recoverability

iii. Which property of schedule ensures **Durability** Recoverability

iv. Weak entity is the one that needs attribute from some other table to uniquely identify tuples

v. Which SQL Isolation level may cause Dirty Read Read uncommitted

vi. Advantage of strict 2PL over standard 2PL doesn't let you release lock without committing or aborting ⇒ no dirty reads

vii. Referential Integrity constraint requires that referred tuple must exist in the referenced table

viii. Prepared statement helps in executing same query with diff parameters

ix. Entity Integrity constraint requires that primary key must uniquely identify tuple.

x. `CallableStatement` object in JDBC is used for stored procedure execution

xi. System logs helps in recovering from crash

xii. Write Ahead Logging Protocol is used for recoverability

xiii. Noted problem in basic Snapshot Isolation is concurrent starting may lead to unwanted results.

xiv. Main problem in 2PL protocol is deadlock

xv. Name one of the procedural data manipulation language
PostgreSQL

xvi. Can you determine Normal form of relation R(AB), in absence of FD information.
Note that, no information does not mean NO FD; if yes what is NF? NO

xvii. Main problem with basic time stamp ordering techniques is ~~sucy~~ cannot support multiple transactions at once. & read gets blocked sometimes

xviii. What is the name of technique that is used for avoiding cascaded rollbacks
Locking

xix. What is highest SQL isolation level that has Phantom row problem Nonrepeatable reads

xx. One of the most important advantage of sql views is
abstraction over classified data.

Student ID: 201601049 Name: MONIL SONI

3. Consider following keys; Primary Key(PK), Key(K), Candidate Key(CK), Super Key (SK), and choose which symbols is most appropriate to be placed in the blank space ($\subseteq$ or $\supseteq$ or $=$) [5]

i. CK $=$ K ✓

ii. CK $\subseteq$ SK ✓

iii. K $\subseteq$ SK ✓

iv. PK $=$ K ✓

v. PK $=$ CK ✓

4. Give short answers -

i. Translate following SQL query in terms of relational algebra: [3]

```
SELECT * FROM EMPLOYEE WHERE SSN IN
                (SELECT DISTINCT ESSN FROM DEPENDENT)
```

3

result $\leftarrow \Pi_{e.*}$ (EMPLOYEE e SEMI-JOIN DEPENDENT d)
$e.ssn = d.essn$

ii. Consider relation R(A, B, C) and set of FDs { {A → B}, {B → C}}; can you find out a join dependency here. [3]

If (a, b) exists and (b, c) exists in the decomposed relations, then ~~(a, c) must~~ (a, b, c) must exist in original rel^n.

iii. Given, A → B and XB → C, prove that XA → C. [3]

$A \rightarrow B \models AX \rightarrow \cancel{XB}\ XB$

~~AX → AB~~

$\{ XA \rightarrow XB,\ XB \rightarrow C \} \models XA \rightarrow C$ ✓ 3

iv. Suppose Employee relation has B+-tree index on DNO. Compute approximately execution cost of query "SELECT * FROM EMPLOYEE WHERE DNO=5"? May express in terms number of block read/writes, and assume other parameters! [3]

$H + \lceil B/2 \rceil$

3

H: height of index tree
B: number of blocks in the file

v. Suppose you have following query to be executed
$\sigma_{SALARY>30000 \wedge DNO=5}(EMPLOYEE)$
What could be best strategy to execute this query. [3]

0 min { $\sigma_{SALARY>30000}$ ($\sigma_{DNO=5}$ (EMPLOYEE)), #
$\sigma_{DNO=5}$ ($\sigma_{SALARY>30000}$ (EMPLOYEE)) }

vi. Compute Join selectivity of join EMPLOYEE JOIN DEPARTMENT ON DNO. What is "Join Selectivity" of the JOIN? May assume other parameters [3]

1 1/2

Join selectivity = $\frac{NE}{NE \times ND}$ NE : no of records in employee
ND : no of records in dept.

5. Suppose following schedule is executed by interleaving operations from transactions T1 and T2. What kind of concurrency problem do you see in following schedule? Give short reason. [5]

```
T1: Read X
T1: X = X + 50
T1: Write X
T2: Read X  // T2 reading uncommitted data.
T2: X = X + 100
T2: Write X
T1: Abort
T2: Commit
```

5 ✓ Dirty Read.
T2 reads data(X) which was written by T1. Since T1 then aborts its updates were supposed to be not taken into consideration, which doesn't happen here since T2 reads from those same updates & commits.

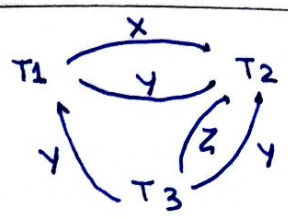
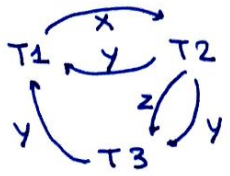
6. Is the schedule given in previous question is recoverable? Give short reason. [5]

No. For a schedule to be recover T2 should not be able to read/write data until T1 commits. This is the recoverability principle.

✓ 5

7. Consider the three transactions T1, T2, and T3, and the schedules S1 and S2 given below. Draw the serializability (precedence) graphs for S1 and S2, and state whether each schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s).

[10]

S1: r3(Y); r3(Z); r1(X); w1(X); w3(Y); w3(Z); r2(Z); r1(Y); w1(Y); r2(Y); w2(Y); r2(X); w2(X);	S2: r2(Z); r2(Y); w2(Y); r3(Y); r3(Z); r1(X); w1(X); w3(Y); w3(Z); r2(X); r1(Y); w1(Y); w2(X);
 T3; T1; T2 (8)	 Not serializable (5)

10

8. Consider following schedule executed on PostgreSQL. Assume that initial salary for employee with SSN 123 is 40000. [Here labels T1 and T2 against statements indicate Transaction that is issuing the statement.] What will be shown by SELECT statements at line numbers 3, 5, 7, 10 if transaction T2 is specified to execute at READ COMMITTED isolation level and SERIALIZABLE isolation level?

[6+6]

```
1  T1: begin;
2  T1: update employee set salary = salary+3000 where ssn = 123;
3  T1: select salary from employee where ssn = 123;
4  T2: begin;
5  T2: select salary from employee where ssn = 123;
6  T2: update employee set salary=salary+5000 where ssn = 123;
7  T2: select salary from employee where ssn = 123;
8  T1: commit;
9  T2: commit;
10 T1: select salary from employee where ssn = 123;
```

43000

7 1/2

READ COMMITTED level	SERIALIZABLE level
Line 3: 43000 ✓	Line 3: 43000 ✓
Line 5: 40000 ✓	Line 5: 40000 ✓
Line 7: 45000 ✗	Line 7: 45000 ✗
Line 10: 48000 ✓	Line 10: Error ✗

Student ID: 201601049 Name: MONIL SONI

9. Suppose following attributes are drawn from a sales/purchase system of trading company.

`Sales_bill_no, sales_bill_date, customer_no, customer_name, item_no, item_name, quantity_in_stock, quantity_in_bill, item_rate, item_rate_in_bill, item_category, item_category_sales_tax_rate, supplier_id, supplier_name, purchase_bill_no, purchase_bill_date, quantity_in_purchase_bill, item_rate_in_purchase_bill, average_purchase_price`

Assume that: (1) there is only one supplier for each ~~item~~ purchase bill, (2) an item comes only once in a bill (3) there is only one customer for a sales bill (4) sales tax rate in a bill depends on item category (5) a sales bill contains all items of same category.

Your tasks here are following - [10+10]

i. Identify Minimal FD Set

ii. Give normalized relations that are in BCNF. Specify Keys and FKs also.

FDs

sales_bill_no → {sales_bill_date, customer_no, ~~customer_name~~}

customer_no → customer_name

average_purchase_price

item_no → {item_name, quantity_in_stock, item_rate, item_category}

{sales_bill_no, item_no} → {quantity_in_bill, item_rate_in_bill}

supplier_id → supplier_name

item_category → item_category_sales_tax_rate

purchase_bill_no → {supplier_id, purchase_bill_date, quantity_in_purchase_bill}

{purchase_bill, item_no} → item_rate_in_purchase_bill

10

Relations

R_1(sales_bill_no, sales_bill_date, customer_no) FK: customer_no

R_2(customer_no, customer_~~name~~ name)

R_3(item_no, item_name, quantity_in_stock, item_rate, item_category, average_purchase_price) FK: item_category

R_4(sales_bill_no, item_no, quantity_in_bill, item_rate_in_bill) FK: sales_bill_no FK: item_no

R_5(supplier_id, supplier_name)

R_6(item_category, item_category_sales_tax_rate)

R_7(purchase_bill_no, supplier_id, purchase_bill_date, quantity_in_purchase_bill) FK: supplier_id

R_8(purchase_bill, item_no, item_rate_in_purchase_bill) FK: purchase_bill FK: item_no.

10