UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2017 – 3rd Year Examination – Semester 6

IT6405: Database Systems II

26th November 2017
(TWO HOURS)

To be completed by the candidate

BIT Examination Index No: .....

Important Instructions:

- The duration of the paper is **2 (two) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and 19 **pages**.
- **Answer all questions.** All questions **carry** equal marks.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.
- Calculators are **not** allowed.

Questions Answered

Indicate by a cross (×), (e.g. [X]) the numbers of the questions answered.

	Question numbers				
To be completed by the candidate by marking a cross (×).	1	2	3	4	
To be completed by the examiners:					

1) (a) Write down three differences between stored procedures and triggers. **[6 marks]**

ANSWER IN THIS BOX

Any three of the followings:

- Trigger requires to identify event and action whilst a stored procedure does not have that requirement.
- Trigger gets executed automatically if the event is occurred but the stored procedure does not run automatically and it should be executed manually.
- A stored procedure could be called inside a trigger but not vice versa.
- Stored procedure may take input parameters but input parameters cannot be passed to a rigger.
- Stored procedure could be called from front end but a trigger cannot be called from front end.

(b) Consider the tables given below which keep track of employees and projects. An employee is allowed to work on only one project and there are many employees working on one project. Each employee receives a payment for working on a project. Total_payment and Total_emp attributes in the Project table depict the total payment and total number of employees working on a particular project.

Employee (Empno, Ename, Job, Payment, Projno)
Project (Projno, Pname, Total_payment, Total_emp)

Given an ***Empno***, write down a stored procedure ***Pay_Rise*** to update the current payment of that employee with an increment as given below:

- 10% salary rise if salary is < 50,000.
- 15% salary rise if salary is ≥ 50,000.

The procedure should also print a message with the old payment and new payment values against the employee id. For example the message should be displayed as *' The payment of **emp_id** was increased from **prev_payment** to **new_payment**'.* **[7 marks]**

ANSWER IN THIS BOX

```
-- The procedure to update payment
CREATE PROCEDURE Pay_Rise (empno# INTEGER)
AS
old_payment# NUMBER(7,2) := 0;
new_payment# NUMBER(7,2) := 0;
BEGIN
 SELECT payment INTO old_payment#
      FROM Employee WHERE Empno = empno#;
 IF old_payment# < 50000 THEN
      new_payment# := old_payment# * 1.1;
 ELSE
      new_payment# := old_payment# * 1.15;
 END IF;
 UPDATE Employee
      SET payment = new_payment#
   WHERE Empno = empno#;
   DBMS_OUTPUT.PUT_LINE('The payment of '||empno#||' was
     increased from '||old_payment#||' to
     '||new_payment#);
  COMMIT;
 END;
  /
```

(c) Write down a trigger named **Proj_Sum1** to maintain the consistency of the database in the scenario given in (b) above.

[6 marks]

ANSWER IN THIS BOX

```
CREATE OR REPLACE TRIGGER Proj_Sum1
AFTER UPDATE OF Payment ON Employee
FOR EACH ROW
WHEN (NEW.Projno IS NOT NULL)
BEGIN
    UPDATE Project
    SET Total_pay = total_pay + (:NEW.Payment - :OLD.Payment)
    WHERE Projno = :OLD.Projno;
END;
/
```

(d) Write down three other events with respect to the Employee table (in addition to the one given in part c) that may cause inconsistency in the database, and specify the corresponding actions to be taken in order to maintain its consistency.

[6 marks]

ANSWER IN THIS BOX

The following events to the Employee table will *cause* an inconsistency in the database as they will in turn change the values of Total_sal and Total_emp in the Project table.

i. Inserting (one or more) new employee tuples.

- Increment the Total_emp in the Project table.
- Add the new salary to the current Total_sal in the Project table.

ii. Changing the assignment of existing employees from one department to another

- Decrement the corresponding number of employees from the Total_emp value of the previous project and increment the the Total_emp value of the new project.
- Deduct the corresponding salary values from the Total_sal value of the previous project and add them to the Total_sal value of the new project.

iii. Deleting (one or more) employee tuples

- Decrement the Total_emp in the Project table.
- Deduct the salary value from the Total_sal in the Project table.

2) (a) Name two mechanisms other than indexing that can be used to improve the performance of queries. Under what circumstances would you suggest these mechanisms? **[3 marks]**

ANSWER IN THIS BOX

Explain any of the two mechanisms given below:

Denormalization

Vertical partitioning of table

Horizontal partitioning of table

Denormalisation is recommended to avoid join operation which hinders the query performance.

Moreover, indexing is not suitable for situations where there are frequent updates to the indexed attributes causing index maintenance expensive. Then alternative mechanisms as mentioned above to improve query performance should be used.

(b) A company is maintaining the following schema with respect to its Employees and Departments. Assume that Mgrid represents the employee identity (Eid) of the employee who manages the department and that an employee manages only one department.

Department(Deptid, Dname, Location, Budget, Mgrid)
Employee(Eid, Ename, Salary, Designation, Deptid)

You are informed that the following queries are extremely important:

i. Given a Deptid, find the average salary of employees who are working for that department.
ii. List the id, name, and address of employees who work in the department with a user-specified department name.
iii. Retrieve the number of employees who are working for a given department.

Explain the decisions that you make with respect to the file structures, indexes (B+ tree/ Hashing, clustered / unclustered, dense/sparse), index only plans to improve the efficiency of each query given in (i) – (iii) above. **[6 marks]**

ANSWER IN THIS BOX

Create a dense unclustered B+ tree index on <Deptid, Salary> of the **Employee** table.

So it is possible to do an index-only scan on all of a department's employees.

Create a dense unclustered B+Tree index on Deptid of the **Employee** table and another unclustered index on <Dname, Deptid> in the **Deptartment** table. Then, it is possible to do an index only search on Department and then get the corresponding Employee records through the Deptid index on Employee.

A dense unclustered B+Tree index on Deptid of the **Employee** table and this is an index only plan.

c) Consider the schema given below and the following statistics:

Employee (<u>Empid</u>, Name, Designation, Salary, Address)
Works_on (<u>Empid</u>, <u>Projid</u>, Hours)

The Works_on table records employees with respect to the projects that they are working on. It is possible for an employee to work on many projects and a project to have many employees.

Consider that Employee is an unordered file with 40,000 records stored on a disk with block size B = 1024 bytes. File records are of fixed size and are unspanned, with record length R = 100 bytes.

Consider that the Works_on file has 100,000 fixed-length records of size R = 50 bytes stored on a disk with block size B = 1024 bytes.

(i) Consider the following query

```
SELECT  E.Name, E.Salary
FROM  Employee E
WHERE E.Empid = '1243';
```

If the Empid attribute is 9 bytes and the block pointer is 6 bytes long, compute the number of block accesses required for the query given above with

I. Secondary indexing
II. Multi-level indexing **[4 marks]**

<u>ANSWER IN THIS BOX</u> **Bfr – Blocking factor**

$Bfr_e = [\frac{1024}{100}] = 10$ No. of blocks in employee file $= \frac{40\,000}{10} = 4000$

$Bfr_w = [\frac{1024}{50}] = 20$ No. of blocks in works on file $= \frac{100,000}{20}$

Index record size → 9 + 6 = 15 bytes $= 5000$

$Bfr_I = [\frac{1024}{15}] = 68$

I) Secondary index :

No. of bocks needed for the index $= [\frac{40000}{68}] = 589$ blocks

Binary search on secondary index $= [\log_2 589] = 10$ blocks accesses

To search for a record using index requires additional block access to the data file $= 10 + 1 = 11$ bock accesses

II) Mulilevel index :

No of first level blocks b_1 = 589 blocks

No. of second level blocks b_2 $= [\frac{589}{68}] = 9$ blocks

No. of Third level blocks b_3 $= [\frac{9}{68}] = 1$ block

No. of block access through multilevel index = 3 + 1 = 4 block accesses

(ii) Consider the following query.

```
SELECT  E.Name, E.Designation
FROM  Works_on W, Employee E
WHERE  W.Empid = E.Empid AND
   W.Projid = '50' AND E.Salary > 80000;
```

Suppose that there are about 50 different projects and that there is uniform distribution of projects in the Works_on table. It is also known that about 25% of the employees are earning more than Rs.80,000.

I) Express the above query using relational algebra. First apply more restrictive operations and finally eliminate the unwanted attributes.

[3 marks]

ANSWER IN THIS BOX

$R_1 = \sigma_{Projid=05}$ (Project)

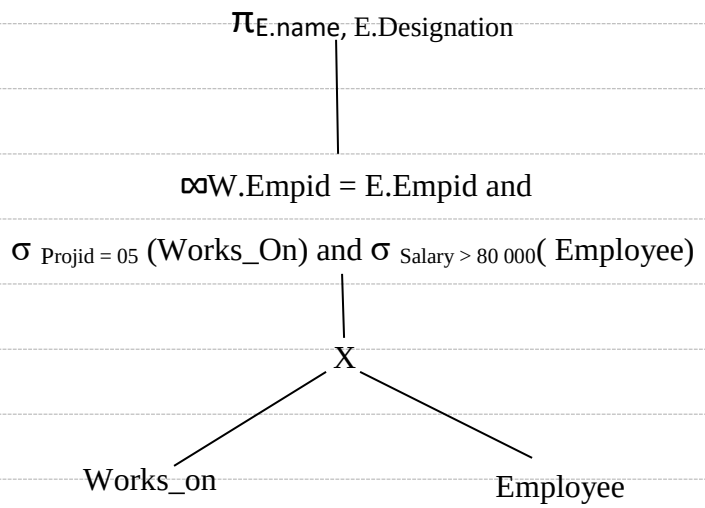
$R_2 = \sigma_{Salary>80,000}$ (Employee)

$R_3 = R_1 \bowtie_{Empid = Empid} R_2$

Result = $\pi_{Name, Salary}$ (R_3)

II) Draw an initial query tree (in canonical form) for the above SQL-query and estimate the cost based on the initial query tree in terms of the number of I/O pages. **[4 marks]**

ANSWER IN THIS BOX



Works_On is more restrictive than Employee and if it is considered as the outer relation then for the initial query tree the cost would be

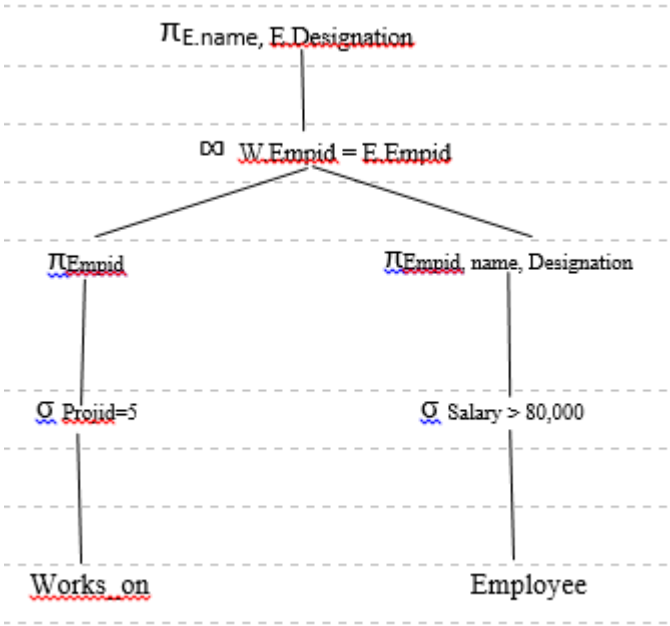
Scan Works_On (5000) + 5000 * 4000

= 20005000 I/O

III) Draw the optimized query tree. Estimate the number of blocks selected from each of the two tables satisfying the given conditions: *Projid = '05'* and *Salary > 80000*.

[5 marks]

ANSWER IN THIS BOX



No. of tuples in works_on =100,000

No. of tuples for *projid = 5* $\rightarrow \frac{100,000}{50}$

= 2000 tuples

Bfr_w (Blocking factor for Works on) = 20

No of blocks accessed in Works on satisfying the given condition $= \frac{2000}{20}$

= 100 blocks

No of tuples in employee = 40, 000

No. of tuples for *salary > 80, 000* = 10, 000 x $\frac{1}{4}$

= 10, 000 tuples

Bfr_e = 10

No of blocks accessed in Employee satisfying the given condition $= [\frac{10,000}{10}]$

= 1000 blocks

3 (a) Consider the following two transactions T1 and T2 executed concurrently on the relation Employee (emp_ID,name).

T1	T2
INSERT INTO Employee VALUES (1,'Kamal'); INSERT INTO Employee VALUES (2,'Nimal');	
	INSERT INTO Employee VALUES (3,'Sunil'); DELETE FROM Employee WHERE name LIKE 'Ka%'; INSERT INTO Employee VALUES (4,'Saman');
DELETE FROM Employee WHERE name LIKE 'Sam%'; COMMIT;	
	DELETE FROM Employee WHERE name LIKE 'Ni%';
	COMMIT;

Assume that the table is initially empty, the transactions are run in isolation level READ COMMITTED, and that the commands are issued in the order indicated above. What is the content of Employee table after the execution of the transactions? Justify your answer.

[5 Marks]

ANSWER IN THIS BOX

(3 marks)

emp_ID	name
1	Kamal
3	Sunil
4	Saman

Only the committed data will be visible when DELETE command is issued. Therefore only Nimal's data will be deleted. **(2 marks)**

(b) Consider the following three transactions and the given schedule assuming that time increases from top to bottom. Is this schedule conflict-serializable? Explain why or why not.

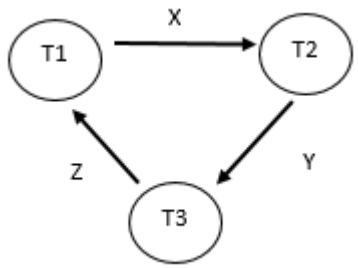
[5 Marks]

T1	T2	T3
Read(X)		
Z=X-10		
	Read(X)	
		Read(Y)
		Y=Y+30
	Read(Y)	
	X=X+20	
		Write(Y)
Read(Z)		
	Write(X)	
		Read(Z)
Z=Z+15		
Write(Z)		

ANSWER IN THIS BOX

The precedence graph contains a cycle, therefore the schedule is not conflict serializable. **(2 marks)**

Correct precedence graph **(3 marks)**



(c) Consider the three schedules S1,S2 and S3 given below.

Note: R1(X) denotes Transaction 1 Read X value.
W2(X) denotes Transaction 2 Write X value.

S1: R1(X),W2(X),R1(Y)
S2: R1(Y),R1(X),W2(X)
S3: R1(Y), W2(X),R1(X)

State whether the following schedules are conflict-equivalent or not with reasons.

i. S1 & S2
ii. S2 & S3
iii. S1 & S3

[5 marks]

ANSWER IN THIS BOX

S1,S2 are conflict equivalent because the conflicting operation order is R1,W2 in both.

(2 marks)

S2,S3 are not conflict equivalent because the conflicting operation order of S2 is R1,W2 and W2,R1 in S3 **(2 marks)**

S1, S3 are not conflict equivalent because the conflicting operation order of S1 is R1,W2 and W2,R1 in S3 **(1 marks)**

(d)

Given below is the database log as it appears at three scenarios.

(a)	(b)	(c)
<T_0 start>	<T_0 start>	<T_0 start>
<T_0, A, 950>	<T_0, A, 950>	<T_0, A, 950>
<T_0, B, 2050>	<T_0, B, 2050>	<T_0, B, 2050>
	<T_0 commit>	<T_0 commit>
	<T_1 start>	<T_1 start>
	<T_1, C, 600>	<T_1, C, 600>
		<T_1 commit>

If the log on stable storage at the time of a crash is as given above, and the differed update protocol has been used as the log based recovery technique, explain what are the redo actions needed to be taken at each scenario a, b and c.

[6 marks]

ANSWER IN THIS BOX

(2 marks for each)

(a) No redo actions need to be taken

(b) redo(T0) must be performed since <T0 commit> is present

(c) redo(T0) must be performed followed by redo(T1) since <T0 commit> and <T1 commit> are present

e) Consider concurrency control by time stamp protocol. Given below are several sequences of events. These sequences represent real time transactions, and the timestamp-based scheduler will allocate timestamps to transactions in the order of their start times. In each case below, identify what happens with the last request.

[4 marks]

You have to choose between one of the following three possible answers:
(I) the request is accepted,
(II) the transaction is delayed,
(III) the transaction is rolled back.

i)

T1	**T2**
Start	
	Start
R(X)	
	R(X)
W(Y)	
	W(Y)

ii)

T1	**T2**
Start	
	Start
	R(X)
	Commit
R(X)	
W(Y)	

ANSWER IN THIS BOX

Answer: **(2 marks for each)**

(i) The request is accepted

(ii) The request is rolled back

4 (a) Distribution transparency is a property of distributed databases because of which the internal details of a distribution are hidden from the users. Briefly explain the three dimensions of distribution transparency. **[6 Marks]**

ANSWER IN THIS BOX

(2 marks for each)

Location Transparency - DBMSs presented globally to user as though a single centralized DBMS. The fact that the table or its fragments are stored at remote site in the distributed database system, should be completely oblivious to the end user.

Fragmentation Transparency- enables users to query upon any table as if it were unfragmented. Thus, it hides the fact that the table the user is querying on is actually a fragment or union of some fragments.

Replication transparency - ensures that replication of databases are hidden from the users. It enables users to query upon a table as if only a single copy of the table exists.

(b) Consider the following Employee relation with data.

emp_ID	name	salary	branch
EMP 1	Amal	100,000	Colombo
EMP 2	Kamal	50,000	Galle
EMP 3	Sunil	175,000	Colombo
EMP 4	Nimal	80,000	Kandy
EMP 5	Gamini	150,000	Kurunegala

(i). Give horizontally fragmented relations (with data) for the Employee relation so that employees with salary less than 150,000 are separated from employees with salary greater than 150,000. Express the fragmentation conditions using relational algebra for each fragment (with data). Indicate how the original relation could be reconstructed.

[10 Marks]

ANSWER IN THIS BOX

Emp1 ($\sigma_{salary<150000}$ Employee) **(1 mark)**

emp_ID	name	salary	branch
EMP 1	Amal	100,000	Colombo
EMP 2	Kamal	50,000	Galle
EMP 4	Nimal	80,000	Kandy

(3 mark)

Emp2 ($\sigma_{salary>150000}$ Employee) **(1 mark)**

emp_ID	name	salary	branch
EMP 3	Sunil	175,000	Colombo

(1 mark)

Emp3 ($\sigma_{salary=150000}$ Employee) **(1 mark)**

emp_ID	name	salary	branch
EMP 5	Gamini	150,000	Kurunegala

(1 mark)

Reconstruction is done through Union of the three relations. Emp1 ∪ Emp2 ∪ Emp3. **(2 mark)**

(ii) Give a vertical fragmentation of the above Employee relation into two sub-relations (with data), so that one contains only the information about branch, whereas the other contains employee name and salary. Indicate how the original relation could be reconstructed.

[5 Marks]

ANSWER IN THIS BOX

Emp4 **(2 marks)**

emp_ID	**branch**
EMP 1	Colombo
EMP 2	Galle
EMP 3	Colombo
EMP 4	Kandy
EMP 5	Kurunegala

Emp5 **(2 marks)**

emp_ID	**name**	**salary**
EMP 1	Amal	100,000
EMP 2	Kamal	50,000
EMP 3	Sunil	175,000
EMP 4	Nimal	80,000
EMP 5	Gamini	150,000

Reconstruction is by joining using the primary key to the relation (emp_ID) available in both fragments **(1 mark)**

(c) Briefly describe any two data allocation techniques and one advantage and one disadvantage of each. **[4 marks]**

ANSWER IN THIS BOX

(**For any 2 ; 2 marks for each)**

- **Centralized** : all the data is located at a single site
- **Partitioned**: database is partitioned into disjoint fragments and each fragment is assigned to a particular site
- **Replicated**: allocate a full copy of the data base to each site
- **Selective Replication**: partitioned data into critical and non-critical fragments and replicate the critical fragments to achieve the required level of availability and performance.

****