UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2020 – 3rd Year Examination – Semester 6

IT6405: Database Systems II

(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 in English.
- This paper has **4 questions** and **20 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.
- Non-Programmable calculators are **allowed.**

Questions Answered

Indicate by a cross (×), (e.g. [1X]) on 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) Consider the following SQL script.

```
CREATE TABLE Employee(EmpID CHAR(5),
              Fname VARCHAR(25) NOT NULL,
              Lname VARCHAR(25) NOT NULL,
              NIC VARCHAR(10),
              EPF VARCHAR(5),
              Salary FLOAT,
              ProID CHAR(4),
              CONSTRAINT emp_pk PRIMARY KEY (EmpID));

CREATE TABLE Project(pID CHAR(4) PRIMARY KEY,
              Name VARCHAR(25),
              Client VARCHAR(25) );


CREATE TABLE Performance(EmpID CHAR(5) PRIMARY KEY,
              projID CHAR(4),
              Score INT(2),
              EvaluatorID CHAR(5));
```

(i). Change the SQL script to enforce an additional constraint called **ck_salary** that would be applied to check whether the Salary of all the employees are between 15,000 and 150,000.

[2 marks]

ANSWER IN THIS BOX

ALTER TABLE Employee ADD CONSTRAINT ck_salary CHECK(Salary<150000.00 and Salary>15000.00);

(ii). Change the SQL script to enforce an additional constraint called **uc_employee** that would ensure NIC and EPF columns will not have duplicate values.

[2 Marks]

ANSWER IN THIS BOX

ALTER TABLE Employee ADD CONSTRAINT uc_employee UNIQUE (NIC, EPF);

(iii). Change the SQL script to enforce an additional constraint called **fk_project** that would ensure data integrity between both the Employee and Project Tables.

[2 marks]

ANSWER IN THIS BOX

```
ALTER TABLE Employee ADD CONSTRAINT fk_project FOREIGN KEY
(ProID) REFERENCES Project(pID);
```

(b) Write a trigger named **best_employees** that accepts insertions and updates into the Performance table, and checks the Score of each employee. If the Score is greater than or equal to 10, that employee will be automatically inserted into a new table called "EmployeeOfTheYear" which consists of the two attributes "EmpID" and "projID".
Note: Assume that the "Score" is a value between 0 and 20 and Updates are done only to increase the score value.

Write down any constraint that should be enforced to ensure the consistency of data.

[6 Marks]

ANSWER IN THIS BOX

[5 Marks]

```
CREATE OR REPLACE TRIGGER best_employees
AFTER INSERT OR UPDATE ON Performance
FOR EACH ROW
WHEN (NEW.Score >=10)
BEGIN
Insert into EmployeeOfTheYear values(NEW.EmpID,
NEW.projID)
END
```

[1 Mark for any correct constrain identification]

ex:

Referential integrity constraints between EmpID, EvaluatorID in Performance table and EmpID in Employee table

projID in Performance table should be equal to the proID in Employee table for the relevant employee.

(c) Create a row-level trigger named ***display_salary_difference*** that would fire for INSERT, UPDATE or DELETE operations performed on the Employee table.
This trigger should display the previous salary, new salary and the difference between the old and new salary values.

[6 marks]

ANSWER IN THIS BOX

```
CREATE OR REPLACE TRIGGER display_salary_difference
BEFORE INSERT OR UPDATE OR DELETE ON Employee
FOR EACH ROW
WHEN (NEW.empID >0)
DECLARE sal_diff NUMBER;
```

```
BEGIN

    Sal_diff := :NEW.Salary  -  :OLD.Salary;
    DBMS_OUTPUT.PUT_LINE('Previous Salary: ' || :OLD.Salary);
    DBMS_OUTPUT.PUT_LINE('New Salary: ' || :NEW.Salary);
    DBMS_OUTPUT.PUT_LINE('Salary Difference: ' || sal_diff);

END;
```

(d) If the following SQL statements were executed, what would be the results of the trigger in part (c) above?

```
INSERT INTO Employee (EmpID, Fname, Lname, NIC, EPF, Salary, ProID)
VALUES ('E1001', 'Ranidu', 'Premachndra', '842365899V', '231',
25000.00, '4');
```

[2 Marks]

ANSWER IN THIS BOX

Previous Salary:

New Salary: 25000.00

Salary Difference: 25000.00

(e) Write a procedure named ***update_salary (eid, amount)*** to update the salary by adding the given amount, for any given employee.

[5 marks]

ANSWER IN THIS BOX

```
CREATE OR REPLACE Procedure update_salary
(eid IN Char, amount IN Float)  AS
Declare Wage Float;
BEGIN
Select Salary into Wage from Employee where EmpID = eid;
Wage := Wage + amount;
UPDATE Employee SET Salary = Wage
WHERE EmpID = eid;
COMMIT;
END;
```

2) (a) Briefly describe the following three indexing types.

(i) Primary Index
(ii) Secondary Index
(iii) Clustering Index

[6 Marks]

ANSWER IN THIS BOX

(i) Primary index is defined on an ordered data file. The data file is ordered on a key field. The key field is generally the primary key of the relation. [2 Marks]

(ii) Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values. [2 Marks]

(iii) Clustering index is defined on an ordered data file. The data file is ordered on a non-key field. [2 Marks]

(b) Consider the following relational schema taken from a University database.

Staff (EmpNo, Name, Department, Designation, DoB, Salary)

Given below are two SQL queries executed using two different query plans. Suggest the most suitable index file organization for each query. Justify your answer.

(i) SELECT Name, Salary FROM Staff WHERE Department =" CS ";

(ii) SELECT Name, Designation FROM Staff WHERE Salary > 100000 and Salary < 200000;

[2 X 2 = 4 Marks]

ANSWER IN THIS BOX

(i) . Clustering index is most suitable when records are required on a non-key field.

In clustering index records are ordered on a non-key field. Therefore, clustering index on "Department" will improve the query performance.

OR

Hash index is suitable for querying individual records.

(ii) B+ tree is the most suitable index file for range level access.

A B+ tree represents sorted data and allows for efficient retrieval of records, each of which is identified by a key.
It is a dynamic, multilevel index, with maximum and minimum bounds on the number of keys in each index segment. In a B+ tree, all records are stored at the leaf level of the tree.

(c) Given below are four relational schemas taken from the same University database.

Student(<u>Std_Id</u>, S_name, Address, Gender, Dept_Id, GPA)
Department(<u>Dept_Id</u>, D_name, Location)
Enroll(<u>Std_Id, Course_Id</u>,Marks)
Course(<u>Course_Id</u>, C_name, Year, Lecturer)

Student file has 30,000 student records and each record is 68 bytes long. These students belong to 15 different departments. Department file has 15 records and each record is 30 bytes long. Each student must enroll to three (03) courses and all that information is available in the Enroll file. Each record in the Enroll file is 35 bytes long. Assume that the Block size is 1024 bytes.

(i) Calculate the number of blocks required to store the file with students' enrollment details.

[2 Marks]

<u>ANSWER IN THIS BOX</u>

Number of records in Enroll file = 3*30,000 (0.5 marks)

Blocking factor for enrol record = 1024/35 = 30 (0.5 marks)

Number of blocks required = 90,000 / 30 = 3,000 blocks (1 Mark)

(ii) If each index entry is 16 bytes long, calculate the total number of block accesses required to get a student record of who belongs to a given department if a secondary multi-level index on the non-key attribute "Dept_Id" on the Student file is to be used. (Assume that the students are evenly distributed among Departments)

[5 marks]

ANSWER IN THIS BOX

Blocking factor for index = 1024/16 = 64 (0.5 marks)

Number of index entries is equal to the number of records in the data file since secondary index is used , therefore → 30,000
(0.5 marks)

The number of first level blocks

L1 = [30000 / 64] = 469 blocks (0.5 marks)

L2 = [469 / 64] = 8 blocks (0.5 marks)

L3 = [8 / 64] = 1 block (0.5 marks)

Number of levels = 3 (1 marks)

Number of students in a department is 30000/15=2000 (0.5 marks)

Of block accesses when secondary index = 3+2000 = 2003 blocks (1 marks)

(d) Consider the following query:

```
SELECT S.S_name, S.Department,
   FROM Student S, Enrol E, Course C
        WHERE S.Std_Id = E.Std_Id AND
        E.Course_Id = C.Course_Id AND S.GPA > 3.5 AND
        C.C_name='% Engineering' AND C.Lecturer='Dr.Alwis'
```

Assume that a lecturer is allowed to teach a maximum of 5 courses and about 0.1% of the students obtain GPA > 3.5.

(i) What would the above query retrieve?

[2 Marks]

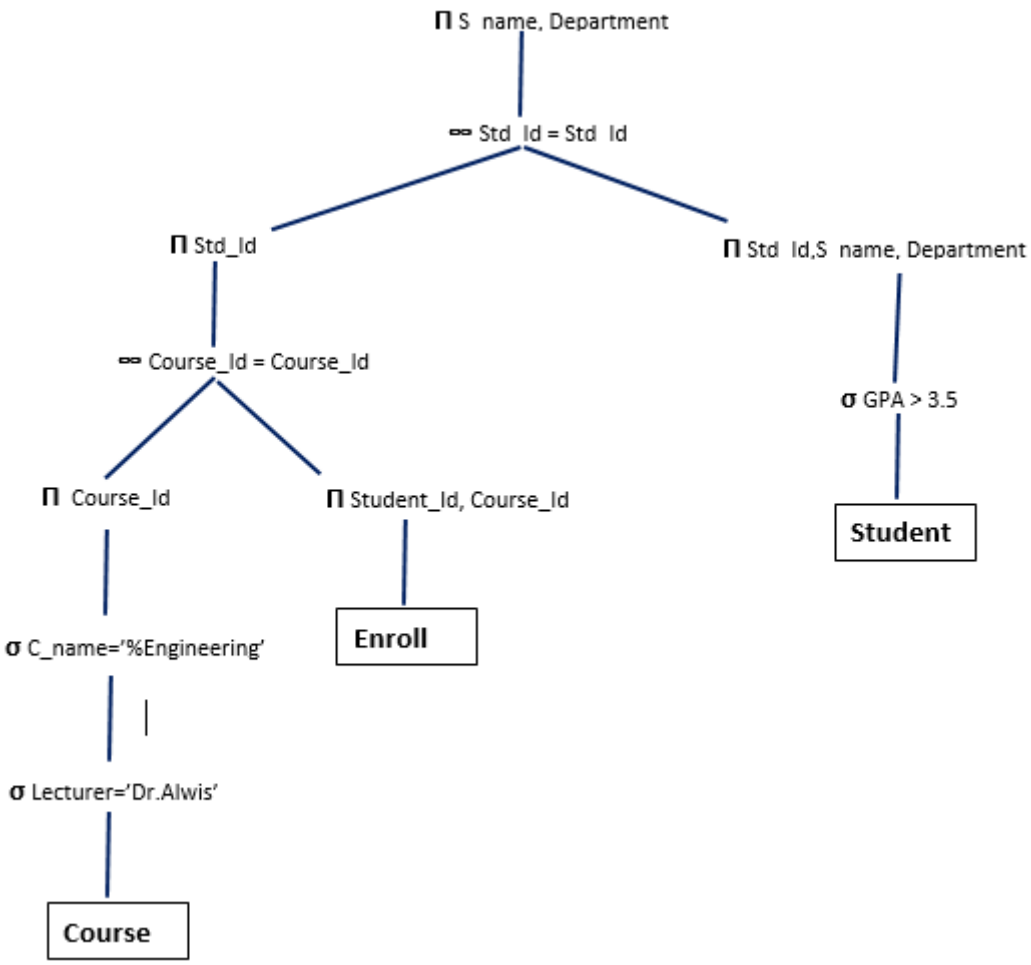
ANSWER IN THIS BOX

Student names and their departments who have GPA > 3.5 and follow Courses where the name ending with Engineering and delivered by the lecturer Dr.Alwis.

(ii) Draw the optimized query tree for the query given above.

[6 marks]

ANSWER IN THIS BOX



3) (a) Briefly explain the properties of a transaction.

[4 Marks]

ANSWER IN THIS BOX

Atomicity: A transaction is an atomic unit of processing. It is either performed in its entirety or not performed at all.

Consistency preservation: A correct execution of the transaction must take the database from one consistent state to another.

Isolation: A transaction should not make its updates visible to other transactions until it is committed.

Durability (Permanency): Once a transaction changes the database and the changes are committed, these changes must never be lost because of subsequent failures.

(b) Transactions T1, T2 and T3 are associated with an airline reservation system.

Note: The number of seats in flights UL203 and UL405 are 125 and 300 respectively and, initially there are no bookings for these flights.

T1	T2	T3
		Read seats from UL203
		Total = UL203 seats
Read seats from UL203		
Book 4 seats		
Write seats		
	Read seats from UL405	
	Book 15 seats	
	Write seats	
		Read seats from UL405
		Total = Total + UL405 seats

If the operations are interleaved as given above, identify the final result of T3 and the problem associated with this schedule. Briefly explain the problem and suggest an appropriate solution.

[4 Marks]

ANSWER IN THIS BOX

Total = 410 (Total should be 406) [1 Mark]

Due to inconsistent retrieval of data Incorrect Summary problem occurs [1 mark]

This problem occurs when one transaction is calculating summary functions on particular data items while other transactions are updating those data items. The transaction performing the calculations may read some data items before they are updated and others after. [1 Marks]

Introduce isolation to transactions – repeatable read/ serializability [1 mark]

(c) Consider the following two schedules, S1 and S2, and determine if they are conflict-serializable or not. If the schedule is serializable, determine an equivalent serial schedule. If it is not, briefly explain the reason. Clearly identify the data items that causes the conflicts.

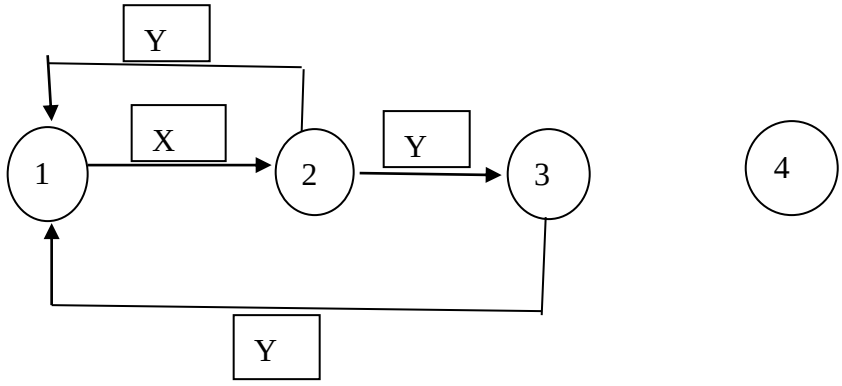
Note: $R_1(X)$ denotes Transaction 1 Read X value.
$W_1(X)$ denotes Transaction 1 Write X value.

(i) **S1:** $R_2(Y)$, $W_2(Y)$, $R_3(Y)$, $R_1(X)$, $W_4(Z)$, $W_1(X)$, $W_3(Y)$, $R_2(X)$, $R_1(Y)$, $W_1(Y)$

[5 Marks]

ANSWER IN THIS BOX

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



Correct Precedence graph (3 marks)

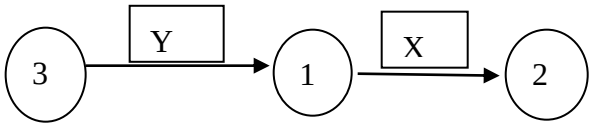
(ii) **S2:** $R_3(Y)$, $R_3(Z)$, $R_1(X)$, $W_1(X)$, $W_3(Y)$, $R_2(Z)$, $R_1(Y)$, $R_2(X)$, $W_1(Y)$, $W_2(X)$

[3 Marks]

ANSWER IN THIS BOX

The precedence graph contains no cycles; therefore, the schedule is conflict serializable. (1 marks)

Correct Precedence graph (1 marks)



Correct serial schedule→ T3, T1, T2 (1 mark)

(d) Consider the following two schedules S3 and S4

S3: $R_1(X)$, $R_2(Y)$, $W_2(Y)$, $R_1(Y)$;
S4: $W_1(X)$, $W_2(X)$, $W_3(X)$, $W_1(Y)$, $W_2(Y)$;

(i) Identify which schedule is allowed by Two-Phase Locking (2PL) and not by Time Stamp Ordering protocol
(ii) Identify which schedule is allowed by Time Stamp Ordering and not by 2PL.

Note: $R_1(X)$ denotes Transaction 1 Read X value.
$W_2(X)$ denotes Transaction 2 Write X value.

[2x2 =4 Marks]

ANSWER IN THIS BOX

S3 is allowed by locks and not by timestamps.
Read X is allowed only if $t \geq$ w-max(X). Since t1-read(B) < w-max(B), this condition is violated.
2PL can be applied

S4 is allowed by timestamps and not by locks.
Here it is not possible to acquire locks based on 2PL protocol.

Can draw TO tables and justify

(e) Consider the following transaction log from a database system that is using undo/redo logging with checkpointing (CKPT) for crash recovery. The log entries for database updates are in the format: **<Transaction id, Variable, New value, Old value>**

1	<START T1>
2	<T1, A, 50, 30>
3	<T1, B, 40, 0>
4	<START T2>
5	<T2, C, 60, 20>
6	<T1, A, 80, 50>
7	<COMMIT T1>
8	<START T3>
9	<T3, D, 70, 20>
10	<T2, E, 60, 40>
11	<CHECKPOINT>
12	<T2, C, 90, 60>
13	<COMMIT T2>
14	<START T4>
15	<T4, F, 30, 10>
16	<COMMIT T3>
17	<T4, F, 100, 30>
18	<COMMIT T4>

(i) What are the values of the data items A, B, C, D and E on disk after recovery if the system crashes just before line 11 is written to disk?

[2 Marks]

ANSWER IN THIS BOX

A= 80

B=40

C=20

D=20

E=40

(ii) What are the values of the data items A, B, C, D, E, and F on disk after recovery if the system crashes just before line 18 is written to disk?

[3 Marks]

ANSWER IN THIS BOX

A= 80

B=40

C=90

D=70

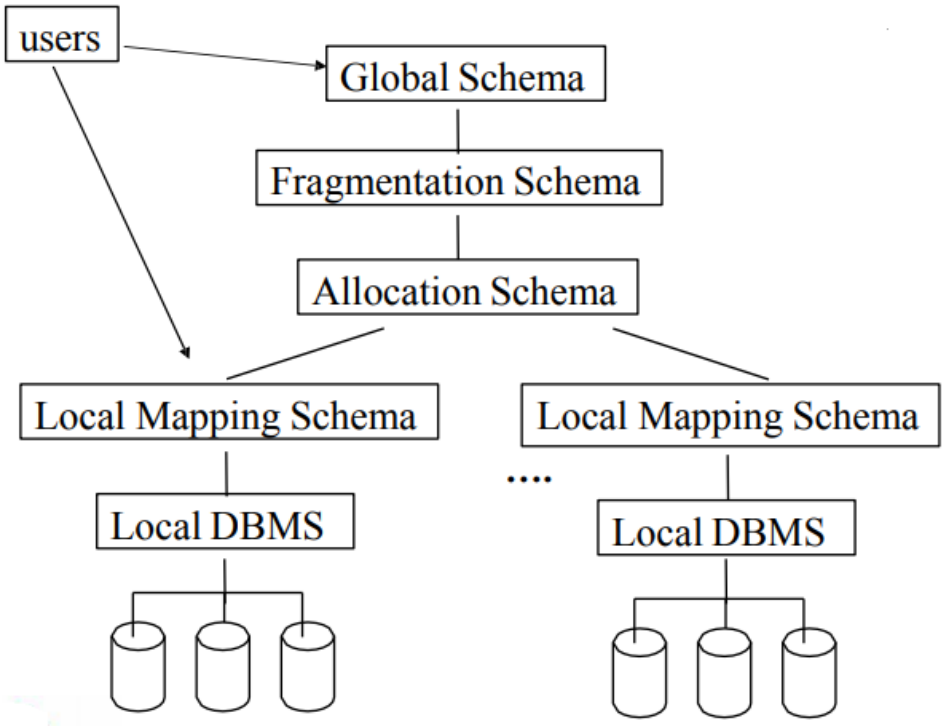
E=60

F=10

4 (a) Draw the architectural diagram for a Distributed Data Base Management System and identify its components.

[5 Marks]

ANSWER IN THIS BOX



(b) Given below is a set of relations available in a database of a Research Institute. This institute has four (04) main departments namely HR, Finance, Research and Development and Media. Research and Development has three divisions for Low- budget, Medium-budget and High-budget research projects and handled by 3 different labs. All the departments and divisions have their own databases and maintain their data as a synchronized distributed database system. Especially due to privacy concerns, each lab has its own database to manage its data.

Employee (<u>Emp_ID</u>, E_Name, Designation, Joined_Date, Dept_Name, Salary)
Project (<u>P_ID</u>, P_Name, Budget, Duration, Principle_Investigator, Funding_Org)
Works_On (<u>Emp_ID, P_ID</u>, Role, Hours)

The following rules are applicable when sharing data with the departments.

- Each department has the rights to maintain data about its employees by replicating data in its site except the salary information, which is handled only by the HR department.
- The Research and Development department has only information about the Employees who work in it. Project details, employees who work on each project, their role and number of hours worked is available with each lab.
- The Low-budget research projects are allocated to Lab A. It handles projects with a budget less than or equal to 1 million, Medium-budget research projects are allocated to Lab B. It handles projects with a budget higher than 1 million but less than or equal to 5 million. High-budget research projects are allocated to Lab C. It handles projects with a budget higher than 5 million.

Assume that based on the above rules, a distributed database has to be designed.

(i) Give fragmented relations for each department and labs.

(ii) Express the fragmentation conditions using relational algebra for each fragment given in (i).

(iii) How would you reconstruct the original relations?

[15 Marks]

<u>ANSWER IN THIS BOX</u>

(1 mark each for the correct fragmentation written using relational algebra –> 5 marks)

HR Department

Emp1 = π Emp_ID, Salary (Employee)

Emp2 = π Emp_ID, E_Name, Designation, Joined_Date, Dept_Name (Employee)

Finance Department

Emp3 = σ Dept_Name = ”Finance” (Emp2)

Media Department

Emp4 = σ Dept_Name = ”Media” (Emp2)

Research and Development Department

Emp5 = σ Dept_Name = ”Research and Development” (Emp2)

(Correct fragment identification using relational algebra for any Lab – 4 Marks and correct update for the same expressions in other two Lab fragments will get 2 marks each)

Lab A

Proj1 = σ budget ≤ 1000000 (Project)

P1 = π_{P_ID} (Proj1)

R1= P1 $\infty_{P\_ID = P\_ID}$ (Works_On)

W1 = $\pi_{Emp\_ID,Role,Hours}$ (R1)

Emp6= Emp5 $\infty_{Emp\_ID = Emp\_ID}$ W1

Lab B

Proj2 = σ budget > 1000000 AND budget ≤ 5000000 (Project)

P2 = π_{P_ID} (Proj2)

R2= P2 $\infty_{P\_ID = P\_ID}$ (Works_On)

W2 = $\pi_{Emp\_ID,Role,Hours}$ (R2)

Emp7= Emp5 $\infty_{Emp\_ID = Emp\_ID}$ W2

Lab C

Proj3 = σ budget > 5000000 (Project)

P3 = π_{P_ID} (Proj3)

R3= P3 $\infty_{P\_ID = P\_ID}$ (Works_On)

W3 = $\pi_{Emp\_ID, Role, Hours}$ (R3)

Emp8= Emp5 $\infty_{Emp\_ID = Emp\_ID}$ W3

(1 mark each for identifying correct reconstruction method - 2 marks)

Horizontally fragmented relations should reconstruct using Union and vertical fragments should Join.

(c) Assume that the Research Institute has stored their data in a distributed database system

created according to the design you have given above, in part (b). Additionally, they decided to keep a copy of High-budget projects' details in Research and Development department database as well.

Suppose the Chief Research Officer of the Institute wants to list all the projects funded by "ABCD" company.

(i) Write down the SQL statement to retrieve the above information under fragmentation transparency.

[2 Marks]

ANSWER IN THIS BOX

```
Select * from Project where Funding_Org=”ABCD”;
```

(ii) Write down the SQL statement to retrieve the above information under location transparency only.

[3 Marks]

ANSWER IN THIS BOX

```
Select * from Proj1 where Funding_Org =”ABCD”;
UNION
Select * from Proj2 Site where Funding_Org =”ABCD”;
UNION
Select * from Proj3 Site where Funding_Org =”ABCD”;
```

****