UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2018 – 3rd Year Examination – Semester 6

IT6405: Database Systems II

07th October, 2018
(TWO HOUR)

To be completed by the candidate

BIT Examination Index No: .....

Important Instructions:

- The duration of the paper is **2 (two) hour**.
- The medium of instruction and questions is English.
- This paper has **4 questions** and **18 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:					

Index No …….

1) (a) Triggers can be classified as DDL triggers and DML triggers. Briefly explain these two trigger categories along with the events relevant to these triggers. [6 marks]

ANSWER IN THIS BOX

DDL triggers

DDL triggers fire whenever any of the DDL triggering events occur. The triggering events could be ALTER, AUDIT, CREATE, DROP, GRANT, RENAME, REVOKE, or TRUNCATE.

A DDL trigger can be either on the entire database or on an individual schema.

BEFORE triggers are fired before changes to the system tables. AFTER triggers are fired after system table changes. When a DDL statement fires a trigger that raises an exception (BEFORE or AFTER) the statement will not be committed.

DML triggers

DML triggers fire when a data manipulation language (DML) event takes place that affects the table or view defined in the trigger. DML events include INSERT, UPDATE, or DELETE statements. DML triggers can be used to enforce business rules, data integrity and query other tables.

(b) Consider the tables given below with respect to an inventory system which keeps track of equipment parts and their reorder levels. For each part a reorder level and stock in hand are maintained. The table `Reorder` keeps track of the reorders placed for each part with the date that the reorder is placed. The reorders can be placed before or after a part reaches its reorder level. The attribute `Received` in the `Reorder` table is a Boolean type ('Y' or 'N') to denote whether the order is received or not. Once the reorder is received this value is updated to 'Y' and the record will subsequently be deleted. Consequently, it can be assumed that only one record per `Partid` is in the `Reorder` table at a time.

Part (Partid Char(05), Part_name Varchar(10), Stock_inhand Integer, Reorder_level Integer)

Reorder (Partid, Reorder_qty, Sysdate, Received)

You are required to create a trigger named ***Place_reorder*** to place a reorder in the event that the `Stock_inhand` is updated in such a way that it reaches below the reorder level. Placing a reorder is not required if a reorder has already been placed for that part.

[7 marks]

ANSWER IN THIS BOX

```
CREATE TRIGGER Place_reorder
AFTER UPDATE OF Stock_inhand ON inventory
FOR EACH ROW
WHEN(new. Stock_inhand < new.Reorder_level)
DECLARE
  x NUMBER;
BEGIN
  SELECT COUNT(*) INTO x  /* query find out if a reorder has already been placed */
    FROM Reorder                /* for that part. x= 1 if that is so and x = 0 otherwise */
    WHERE Partid = :new. Partid;
  IF x = 0  THEN
    INSERT INTO Reorder
      VALUES (:new. Partid, :new.Reorder_qty, sysdate, 'N');
  END IF;
END;
```

(c) Write down the code to implement the stored procedures given below:

i) Procedure `Del_reorder` to delete the record of a given `Partid` [`Partid` is of the type char(05)] when the Received status is 'Y'.

ii) Procedure `Update_stock` to update the `Stock_inhand` with respect to a given quantity (which is the quantity sold) for a given `Partid`.

[7 marks]

ANSWER IN THIS BOX

```
CREATE OR REPLACE Procedure Del_reorder
(pid IN  char(05))       /* or Part.Partid%TYPE */
IS
BEGIN
DELETE FROM Reorder WHERE Partid = pid AND Received = 'Y';
END;  /
CREATE OR REPLACE Procedure Update_stock
(pid IN  char(05), qty IN integer)       /* or Part.Partid%TYPE
*/
IS
Stock integer;
BEGIN
Select Stock_inhand into Stock from Part where Partid = pid;
Stock := Stock - qty;
UPDATE Part SET Stock_inhand = Stock
WHERE Partid = pid;
COMMIT;
END;
/
```

(d) Triggers can be used to enforce referential integrity constraints. Consider the Employee table given below which refers to the `Deptid` in `Department` table. Write down a trigger to enforce *On Delete Cascade* which is to delete the corresponding employees on deletion of a department.

Employee (Empid, Ename, Designation, Salary, Deptid)

[5 marks]

ANSWER IN THIS BOX

```
CREATE OR REPLACE TRIGGER Del_cascade
  AFTER DELETE ON Department
    FOR EACH ROW
  BEGIN
  BEGIN
      DELETE FROM Employee
          WHERE Employee.Deptid = :OLD.Deptid;
  END;
/
```

2) (a) Explain the concept *Denormalization* and how it can be used as a design decision for improving query performance.

[3 marks]

ANSWER IN THIS BOX

The process of storing the logical database design which may be in BCNF or 3NF in a weaker normal form, for example 2NF or 1NF is known as denormalization.

Under denormalisation concept the designer would add attributes to table irrespective of associated redundancy problems in order to avoid join operation which hinders the query performance.

(b) The following schema is maintained to keep track of football players and their clubs. Assume that a player is attached to only one football club and that each player is paid a monthly salary from the club.

Club(<u>Clubid</u>, Club_name, City, Stadium)
Player(<u>Pid</u>, Pname, Salary, Position, Clubid)

You are informed that the following queries are extremely important:

i. Given a `Clubid`, find the average salary of the players who are playing for that club.
ii. List the player id, name, and position (such as left wing or goalie) of players who play for the user-specified club name.
iii. Retrieve the number of players attached to a given club represented by a `Clubid`.

Explain the decisions that you would 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]

<u>ANSWER IN THIS BOX</u> (An alternative answer is given with Or and it is sufficient to give one answer)

i) Create a dense unclustered B+ tree index on <Clubid, Salary> of the **Player** table.

This enables to perform an **index-only scan** on all players who are playing for that club.

Or Clustered index on Clubid of the Player table.(this is a sparse index)

- ii) Create a dense unclustered B+Tree index on Clubid (or alternatively clustered B+Tree index on Clubid) of the Player table and another unclustered index on <Club_name, Clubtid> in the **Club** table. Then, it is possible to do an index only search on Club and then get the corresponding player records through the Clubid index on Player table. **Or** unclustered index on Club_name of the Club table.

iii) A dense unclustered B+Tree index on Clubid of the **Player** table and this is an index only plan **Or** a sparse clustered B+Tree index on Clubid of the **Player** table

(c) Consider the schema given below with the following statistics:

Student (<u>Studid</u>, SName, Address, ContactNo, Degree, GPA)
Grades (<u>Studid</u>, <u>Courseid</u>, Grade, Marks)

The `Grades` table records grades and marks of the students with respect to the courses of their degree programme. Consider `Student` is an unordered file with 50,000 records stored on a disk with block size B = 1024 bytes. File records are of fixed size and are unspanned, with record length R = 128 bytes.

Consider that the `Student` file has 400,000 fixed-length records of size R = 64 bytes stored on a disk with block size B = 1024 bytes.

i) Consider the following query

SELECT S.SName, S.GPA
FROM Student S
WHERE S.Studid = 'CS342';

If the `Studid` 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 on `Studid`
II. Multi-level indexing on `Studid`

[4 marks]

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

Bfr_S= [1024/128] = 8 No. of blocks in Student file = 50000/8 = 6250

Index record size → 9 + 6 = 15 bytes

Bfr_I= [1024/15] = 68

I) Secondary index :

No. of bocks needed for the index = [50000/68] = 735 blocks

Binary search on secondary index = [$\log_2$735] = 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 = 735 blocks

No. of second level blocks b_2 = [735/68] = 11 blocks

No. of Third level blocks b_3 = [11/68] = 1 block

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

Alternatively due to misprint in the question the following answer is also considered.

Bfrs= [1024/64] = 16 No. of blocks in Student file = 400000/16 = 25000

I) Secondary index :

No. of bocks needed for the index = [400,000/68] = 5883 blocks

Binary search on secondary index = [$\log_2$5883] = 13 blocks accesses

To search for a record using index requires additional block access to the data file = 13 + 1 = 14 bock accesses

I) Mulilevel index :

No of first level blocks b1 = 5883 blocks

No. of second level blocks b2 = [5883/68] = 87 blocks

No. of Third level blocks b3 = [87/68] = 2 block

No. of Third level blocks b4 = [2/68] = 1 block

No. of block accesses through multilevel index = 4 + 1 = 5 block accesses

ii) Consider the following query.

```
SELECT  S.SName, S.Degree
FROM  Grades G, Student S
WHERE  G.Studid = S.Studid  AND
   G.Courseid = 'IT234' AND S.GPA > 3.0;
```

Suppose that there are about 40 different courses and that there is a uniform distribution of courses in the `Grades` table. It is known that about 25% of the students have scored more than 3.0 GPA.

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

$R_1 = \sigma_{Courseid = 'IT234'}$ (Grades)

$R_2 = \sigma_{GPA>2.5}$ (Student)

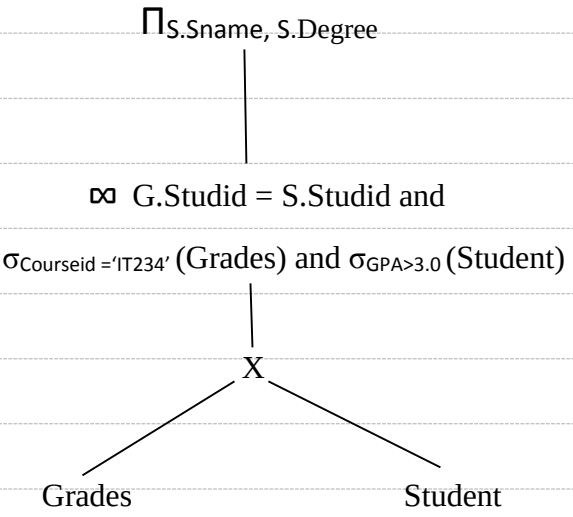
$R_3 = R_1 \bowtie_{Studid=Studid} R_2$

Result = $\pi_{SName,Degree}$ (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



For those who have understood the question in spite of the misprint one of the following answers were considered. Marks are allocated for other genuine efforts due to misprint.

If Grades is considered as the outer relation
then for the initial query tree the cost would be

Scan Grades (25000) + 25000 * 6250

= 156275000 I/O

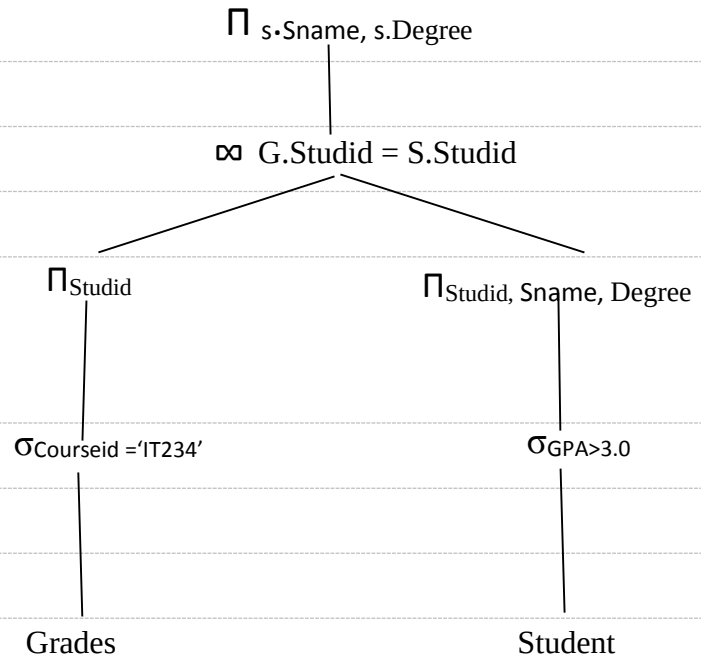
If query tree is produced considering Student as the outer relation
then for the initial query tree the cost would be

Scan Student (6250) + 25000 * 6250

= 156256250 I/O

III. Draw the optimized query tree. Estimate the number of tuples and then the number of blocks selected from each of the two tables satisfying the conditions *Courseid = 'IT234'* and *GPA > 3.0*. [5 marks]

ANSWER IN THIS BOX



For those who have understood the question in spite of the misprint one of the following answers were considered.

No. of tuples in Grades = 400,000

No. of tuples for *Courseid = 'IT234'* → 400,000/40

= 10,000 tuples

Bfr_w (Blocking factor for Grades) = 16

No of blocks accessed in Grades on satisfying the given condition = 10,000/16

= 625 blocks

No of tuples in Student = 50, 000

No. of tuples for *GPA > 3.0* = 50, 000 x $\frac{1}{4}$

= 12, 500 tuples

Bfr_s = 08

No of blocks accessed in Student satisfying the given condition = 12500/8

= 1563 blocks

Or due to misprint if the other file is considered as Student (However with some attempt to compute no. of blocks accessed from Grades file)

No of tuples in Student = 400, 000
No. of tuples for GPA > 3.0 = 400, 000 x 1/4

= 100,000 tuples
Bfrs = 16
No of blocks accessed in Employee satisfying the given condition = 100,000/16
= 6250 blocks

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) Consider the following schedule created using five transactions. Is this schedule conflict-serializable? Give reasons.

S1: $R_2(X)$, $R_1(Y)$, $R_1(Z)$, $R_5(V)$, $R_5(W)$, $W_5(W)$, $R_2(Y)$, $W_2(Y)$, $W_3(Z)$, $R_1(V)$, $R_4(Y)$, $W_4(Y)$, $R_4(Z)$, $W_4(Z)$, $R_1(U)$, $W_1(U)$

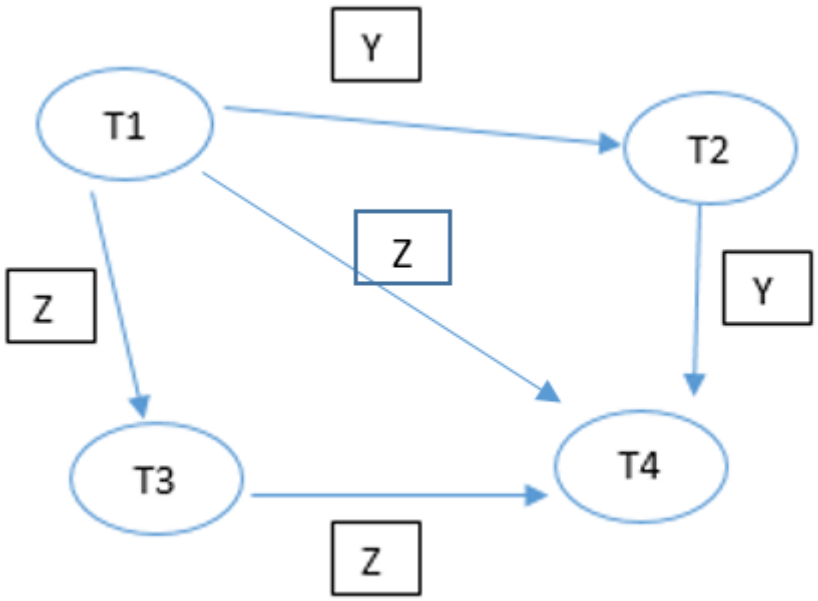
Note: R1(X) denotes Transaction 1 Read X value.
W2(X) denotes Transaction 2 Write X value.

[7 Marks]

ANSWER IN THIS BOX

The precedence graph contains no cycle, therefore the schedule is conflict serializable. (2 marks)

Correct Precedence graph (5 marks)



(c) Consider the following schedule S involving transactions T1 and T2.

T1	T2
Read(X)	
X=X-1000	
Write(X)	
	Read(X)
	A=X-50
	X= X+A
	Write (X)
Read(Y)	
Y=Y+1000	
Write(Y)	
Commit	
	Read(Y)
	Y=Y*0.1
	Write(Y)
	Commit

If the above schedule follows a two-phase locking protocol, identify all the locking activities.

[8 marks]

ANSWER IN THIS BOX

T1	T2
S-Lock(X)	
Read(X)	
X=X-1000	
X-Lock(X)	
Write(X)	
	Waiting for lock X
	Read(X)
	A=X-50
	X= X+A
	Write (X)
s-Lock(Y)	
Read(Y)	
Y=Y+1000	
X-Lock(Y)	
Write(Y)	
Commit	
Release X-Locks for X,Y	
	S-Lock(X)
	X-Lock(X)
	S-Lock(Y)
	Read(Y)

	Y=Y*0.1
	X-Lock(Y)
	Write(Y)
	Commit
	Release X-Locks for X,Y

(d) Consider the database log given below at the time of a system crash.

LSN 1	<START T1>
LSN 2	<T1 X 5>
LSN 3	<START T2>
LSN 4	<T1 Y 7>
LSN 5	<T2 X 9>
LSN 6	<START T3>
LSN 7	<T3 Z 11>
LSN 8	<COMMIT T1>
LSN 9	<START CKPT(T2,T3)>
LSN 10	<T2 X 13>
LSN 11	<T3 Y 15>
	SYSTEM CRASH

i) What is the earliest possible LSN read by the recovery manager?
ii) Identify the actions and how the values change during the recovery process.
iii) What is the value of X at the end of the recovery?

[6 marks]

ANSWER IN THIS BOX

i. LSN 3

ii. Y=15

X=13

Z=11

X=9

iii. X=9

4 (a) Briefly explain the two terms "distributed database (DDB)" and "distributed database management system (DDBMS)".

[4 Marks]

ANSWER IN THIS BOX

Distributed Database : a collection of multiple logically interrelated databases distributed over a computer network

Distributed Database Management System: a software system that manages a distributed database while making the distribution transparent to the user.

(b) Given below are a set of relations taken from a database of a company. The company has branches located in Kurunegala, Jaffna and Matara. The main branch is located in Colombo and has full access to the database. Each branch is given the rights to maintain data about their employees and project details through replicating data in their sites. Assume all the branches have their own database and maintain their data as a synchronized distributed database system.

Employee

e_ID	**name**	**address**	**salary**	**branch**	**proj_id**
EMP 1	Nimal	Galle	100,000	Colombo	10
EMP 2	Janaka	Kegalle	50,000	Matara	19
EMP 3	Saumya	Kurunegala	175,000	Colombo	10
EMP 4	Mihira	Colombo	80,000	Colombo	18
EMP 5	Piyal	Anuradhapura	150,000	Kurunegala	16
EMP 6	Janaka	Matara	60,000	Matara	19
EMP 7	Samudra	Kekirawa	90,000	Jaffna	18
EMP 8	Sunil	Dambulla	130,000	Kurunegala	12

Project

proj_id	proj_name	duration	budget
10	ABC-Hospitals	1 year	500,000
12	Smart Home	2 years	1,000,000
16	Nimal Construction	6 months	300,000
18	XY Holdings	2 years	800,000
19	Mac Systems	1 year	600,000

Give fragmented relations (with data) for each branch. Express the fragmentation conditions using relational algebra for each fragment. Indicate how the original relation would be reconstructed.

[16 Marks]

ANSWER IN THIS BOX

Colombo Branch

Emp1($\sigma_{branch='Colombo'}$ Employee)

e_ID	name	address	salary	branch	proj_id
EMP 1	Nimal	Galle	100,000	Colombo	10
EMP 3	Saumya	Kurunegala	175,000	Colombo	10
EMP 4	Mihira	Colombo	80,000	Colombo	18

Proj1($\sigma_{proj\_id=10\ OR\ proj\_id=18}$ Project)

proj_id	proj_name	duration	budget
10	ABC-Hospitals	1 year	500,000
18	XY Holdings	2 years	800,000

Kurunegala Branch

Emp2($\sigma_{branch='Kurunegala'}$ Employee)

e_ID	name	address	salary	branch	proj_id
EMP 5	Piyal	Anuradhapura	150,000	Kurunegala	16
EMP 10	Sunil	Dambulla	130,000	Kurunegala	16

Proj2($\sigma_{proj\_id=16}$ Project)

proj_id	proj_name	duration	budget
16	Nimal Construction	6 months	300,000

Jaffna Branch

Emp3($\sigma_{branch='Jaffna'}$ Employee)

e_ID	name	address	salary	branch	proj_id
EMP 7	Samudra	Kekirawa	90,000	Jaffna	18

Proj3($\sigma_{proj\_id=18}$ Project)

proj_id	proj_name	duration	budget
18	XY Holdings	2 years	800,000

Matara Branch

Emp4($\sigma_{branch='Matara'}$ Employee)

e_ID	name	address	salary	branch	proj_id
EMP 2	Janaka	Kegalle	50,000	Matara	19
EMP 6	Janaka	Matara	60,000	Matara	19

Proj4($\sigma_{proj\_id=19}$ Project)

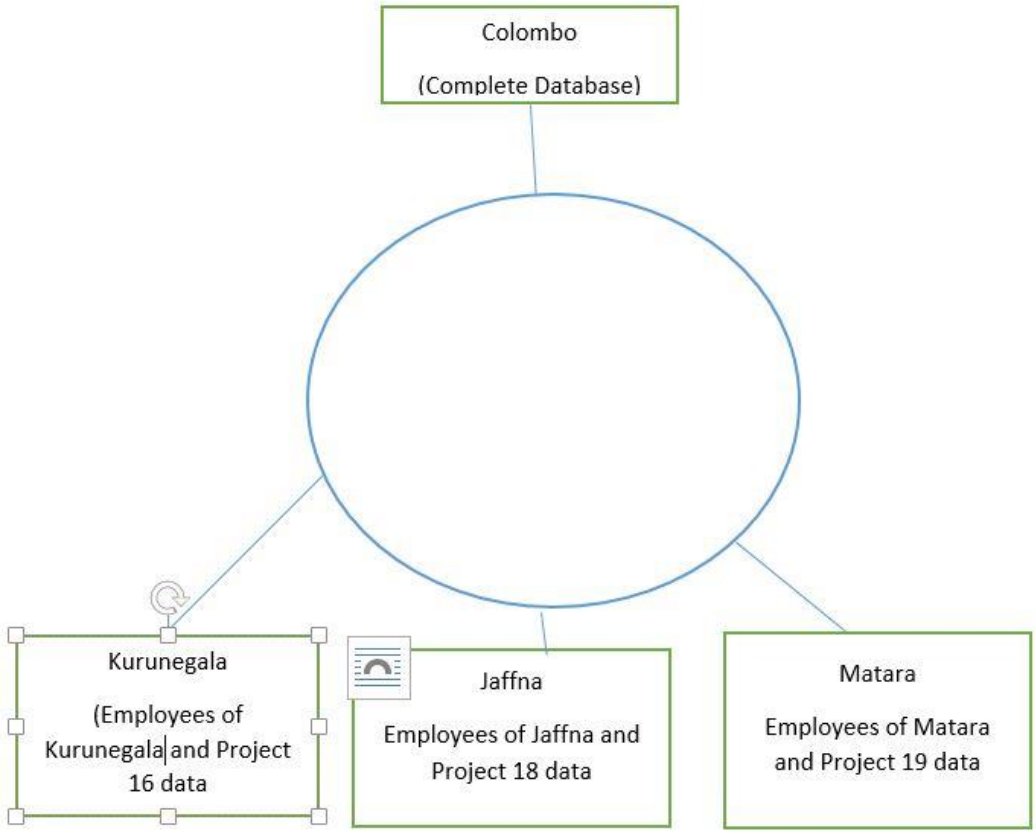
proj_id	proj_name	duration	budget
19	Mac Systems	1 year	600,000

Reconstruction can be done using **Union**

(c) Draw a rough diagram to show the different sites and how these sites are connected in this distributed system. Indicate the type of data which would be stored at respective sites by identifying all fragmentations and replications applicable to this system.

[5 marks]

ANSWER IN THIS BOX



****