Test 2:

Question ONE

a) Security requirement for database management

- Multi-level access control
- Confidentiality
- Reliability
- Integrity
- Recovery

b) What happen in multilevel secure distributed database management system?

- User cleared at different level access and share a database at different security levels, (also called sensitive levels) without violating security. May also support users and data at different compartments and categories.

c) Modules of the SDP(similar to those of DDBMS)

- The secure distributed query processor(SDQP)
- The secure distributed transaction manager(SDTM)
- The secure distributed metadata manager(SDMM)
- The secure distributed security manager(SDSM)
- The secure distributed integrity manager(SDIM)

d) Two things/functions combined by parallel database

Database management and parallel processing to increase performance and availability.

e) Architecture of parallel distributed system

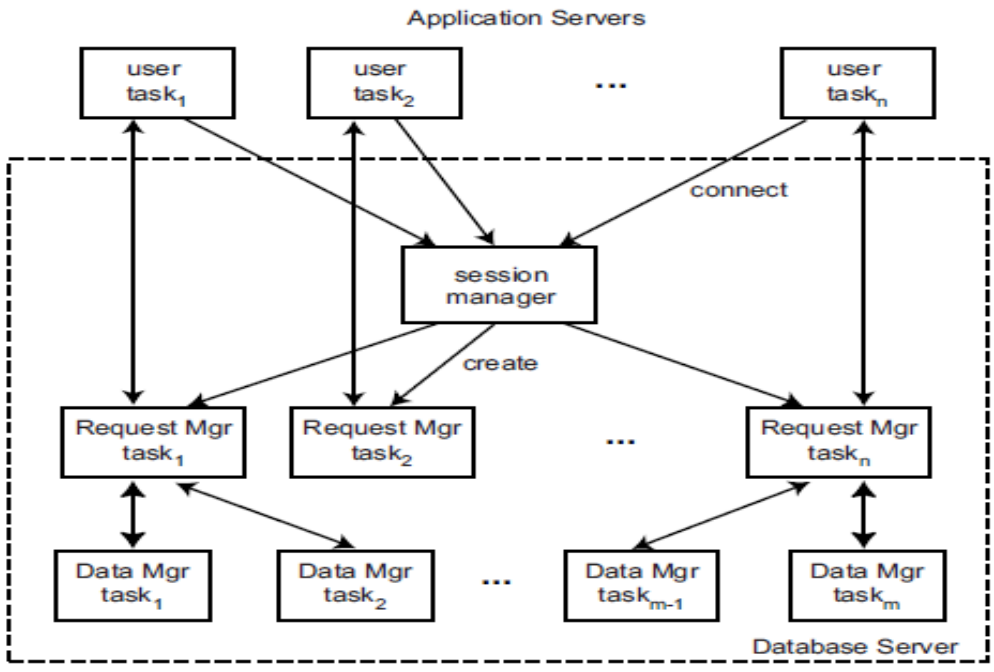
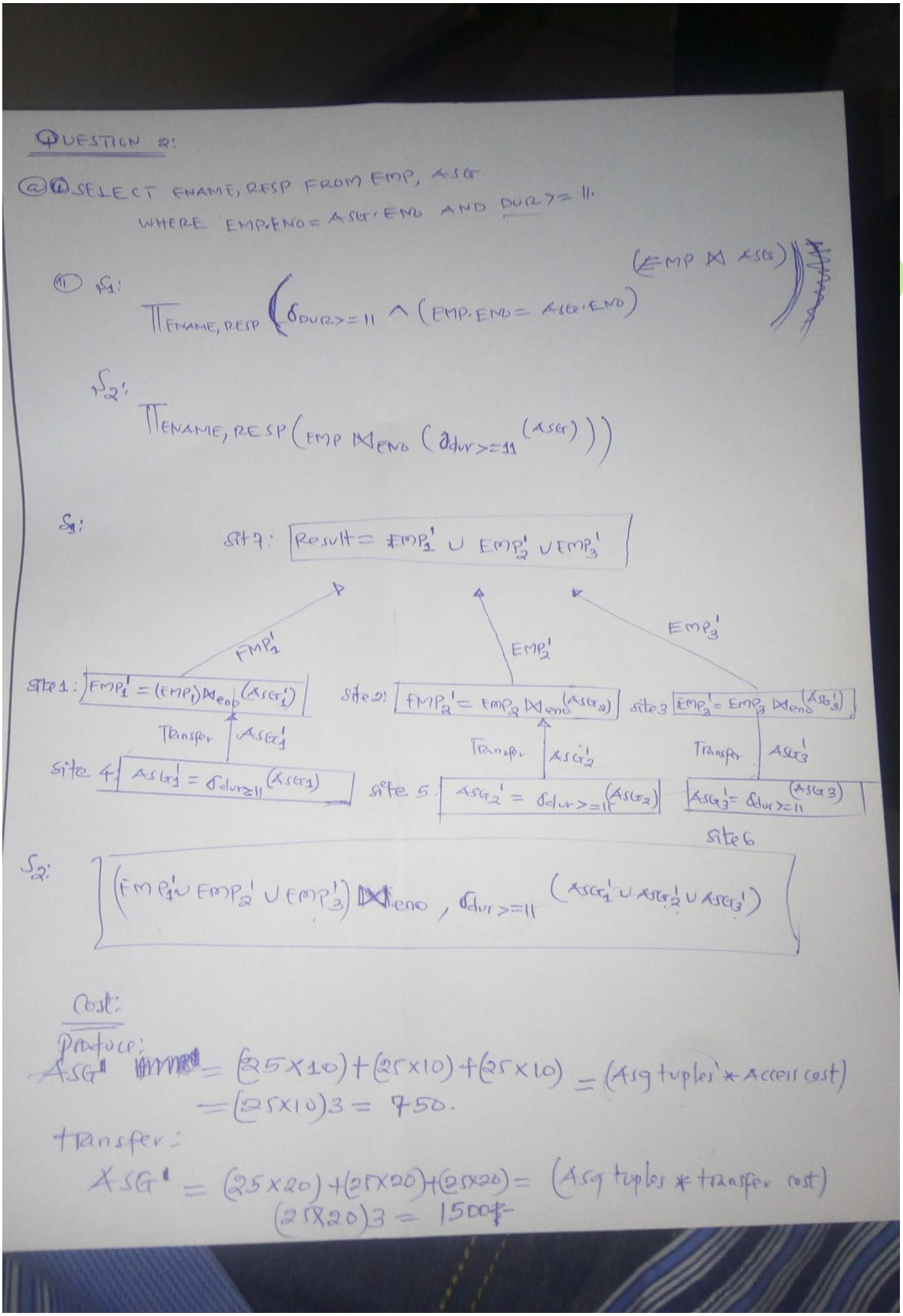
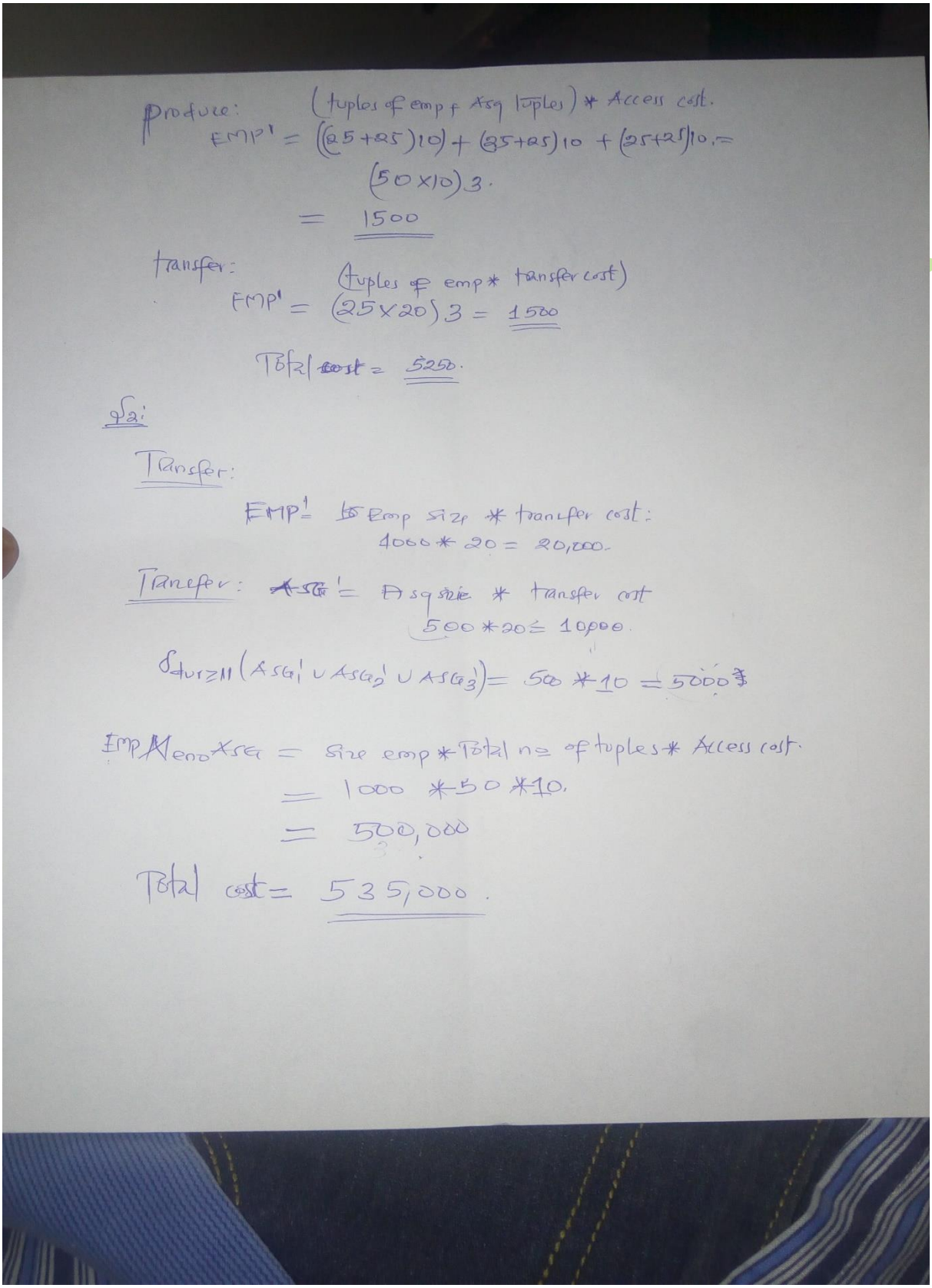


Fig. 14.2 General Architecture of a Parallel Database System

Question two



Question two



Produce: (tuples of emp + Asg tuples) * Access cost.

EMP' = ((25+25)10) + (25+25)10 + (25+25)10.=

(50×10)3.

= 1500

Transfer: (tuples of emp * transfer cost)

EMP' = (25×20)3 = 1500

Total cost = 5250.

q2:

Transfer:

EMP' = Emp size * transfer cost:

4000 * 20 = 20,000.

Transfer: ASG' = Asg size * transfer cost

500 * 20 = 10000.

Store all (ASG1' ∪ ASG2' ∪ ASG3') = 50 * 10 = 5000$

EmpMeroASG = Size emp * Total no of tuples * Access cost.

= 1000 * 50 * 10.

= 500,000

Total cost = 535,000.

Question three

a) **Query Optimization**: refers to the process by which the ***best or an optimal*** execution strategy for a given query is found from a set of alternatives.

b) Characteristics of distributed system R* query optimization algorithm.

- Only the whole relations can be distributed, i.e., fragmentation and replication is not considered
- Query compilation is a distributed task, coordinated by a master site, where the query is initiated
- Master site makes all inter-site decisions, e.g., selection of the execution sites, join ordering, method of data transfer,
- The local sites do the intra-site (local) optimizations, e.g., local joins, access paths

c) The hill-climbing algorithm proceeds as follows

1. Select initial feasible execution strategy ES0

i.e., a global execution schedule that includes all intersite communication

2. Split ES0 into two strategies: ES1 followed by ES2

3. Replace ES0 with the split schedule which gives

cost(ES1) + cost(local join) + cost(ES2) < cost(ES0)

4. Recursively apply steps 2 and 3 on ES1 and ES2 until no more benefit can be gained

5. Check for redundant transmissions in the final plan and eliminate them

d) Distribution system cost model

i) Two types reduce total time and reduce response time.

ii) **Reduce total time:**

–Reduce each cost component (in terms of time) individually, i.e., do as little for each cost component as possible.

–Optimize resource utilization (i.e., increase system throughput)

Reduce response time:

–Do as many things in parallel as possible

–May increase total time because of increased total activity

Question 4.

a) What do multi database systems represent

-represents the case where individual DBMSs (whether distributed or not) are fully autonomous and have no concept of cooperation. They may not even “know” of each other’s existence or how to talk to each other.

b) A mediator- is a software module that exploits encoded knowledge about certain sets or subset of data to create information for a higher layer of application.

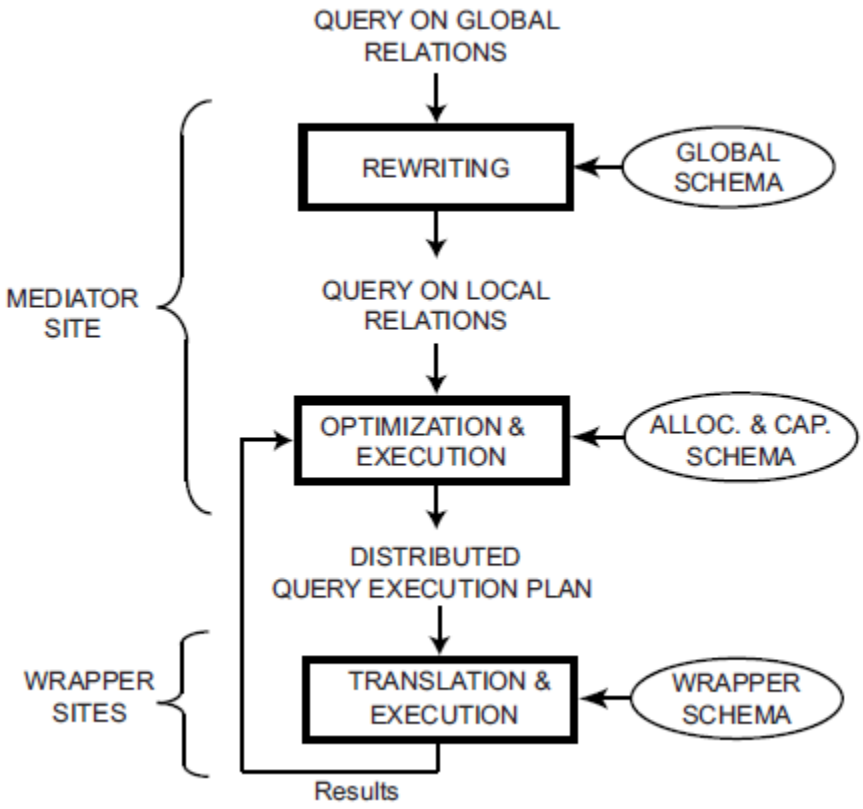
Wrapper- are implemented whose tasks is to provide between a source DBMS view and the mediators view.

c) Query processing in multi database is more complex than in distributed database system

1. The computing capabilities of the component DBMSs may be different, which prevents uniform treatment of queries across multiple DBMSs.
2. Similarly, the cost of processing queries may be different from different DBMSs, and the local optimization capability of each DBMS may be quite different.
3. The data models and languages of the component DBMSs may be quite different, for instance, relational, object-oriented, XML etc.
4. Since a multi database system enables access to very different DBMSs that may have different performance and behavior, distributed query processing techniques need to adapt to these variations.

-beside from being distributed the component database may be autonomous, have different database languages and query processing capabilities, and exhibit varying behavior.in particular component database may range from full-fledged SQL database to very simple data sources.

d) The general architecture of a multi-database query processing



Test 1 solution

Question one

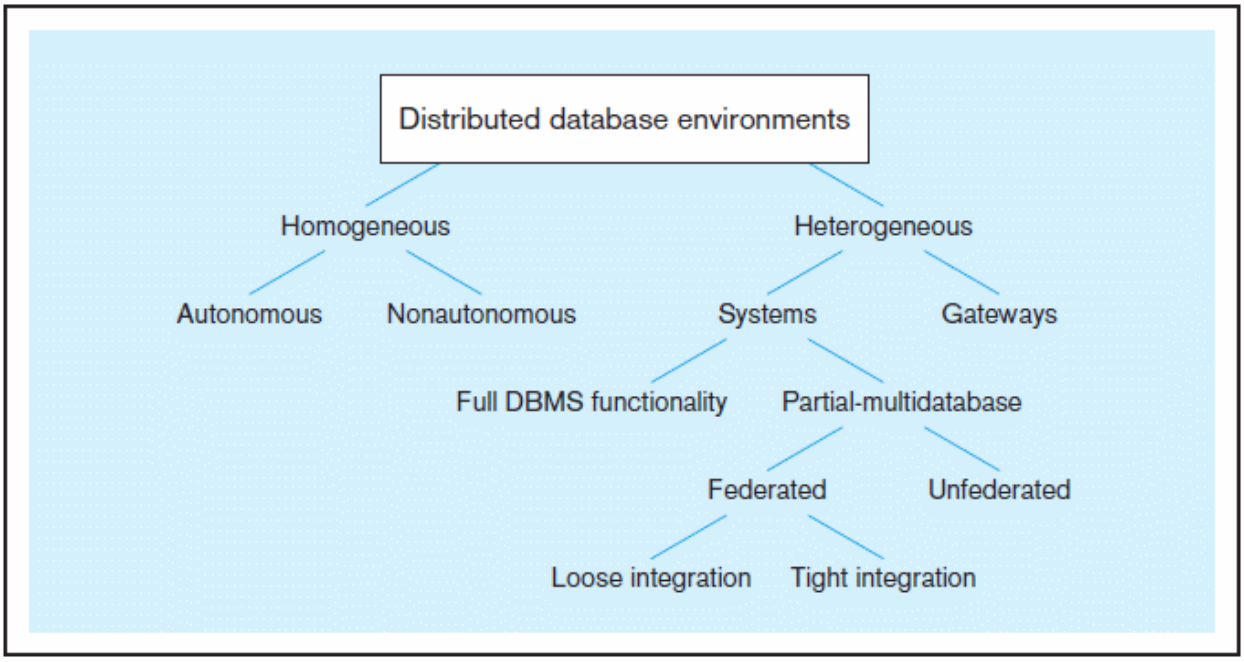
a) Difference between distributed database system and decentralized database system

- **Distributed Database:** A *single logical database* spread physically across computers in multiple locations that are connected by a data communications link
- **Decentralized Database:** A collection of *independent* databases on non-networked computers

b) Difference between intra query and inter query

- –**Inter-query parallelism:** *Ability to use multiple processors to execute several independent queries simultaneously.*
- **Intra-query parallelism:** **A***bility to break a single query into subtasks and to execute those subtasks in parallel using a different processor for each.*

c) Distributed database environments



Question two

a) What is a reasonable unit of distribution in distributed database system? Relation or fragment of relation?

Relations as unit of distribution:

– If the relation is not replicated, we get a high volume of remote data accesses.
However, if the relation is replicated, we get unnecessary replications, which cause problems in executing updates and waste disk space.

Fragments of relation as a unit of distribution:

– Application views are usually subsets of relations.
– Thus, locality of accesses of applications is defined on subsets of relations.
– Permits a number of transactions to execute concurrently, since they will access different portions of a relation.

Desa la kijasi

b) Table of distribution capabilities

TABLE 12-3 Distribution Capabilities in Key DBMSs

Vendor	Product	Important Features
IBM	• DB2 Data Propagator • Distributed Relational Database • Architecture (DRDA) • DB2 Information Integrator	• Works with DB2; replicates data to "regional transactional" databases • Primary site and asynchronous updates • Read-only sites subscribe to primary site • Supports both distribution and consolidation • Heterogeneous databases • Integrates data from multiple types of sources
Sybase	• Replication Server • SQL Anywhere Studio	• Primary site and distributed read-only sites • Update to read-only site as one transaction • Hierarchical replication • Data and stored procedures replicated • Databases located outside the traditional data center
Oracle	• Oracle Streams • Oracle Advanced Replication	• Sharing data with both Oracle and non-Oracle data stores • Hub-and-spoke replication • Synchronous capture for table replication • Multi-master replication supporting distributed applications
MySQL	• Built-in replication capabilities	• Scale-out solutions • Analytics • Long-distance data distribution
Microsoft	• SQL Server 2008	• Multiple types of replication: transactional, merge, and snapshot • Microsoft Sync Framework

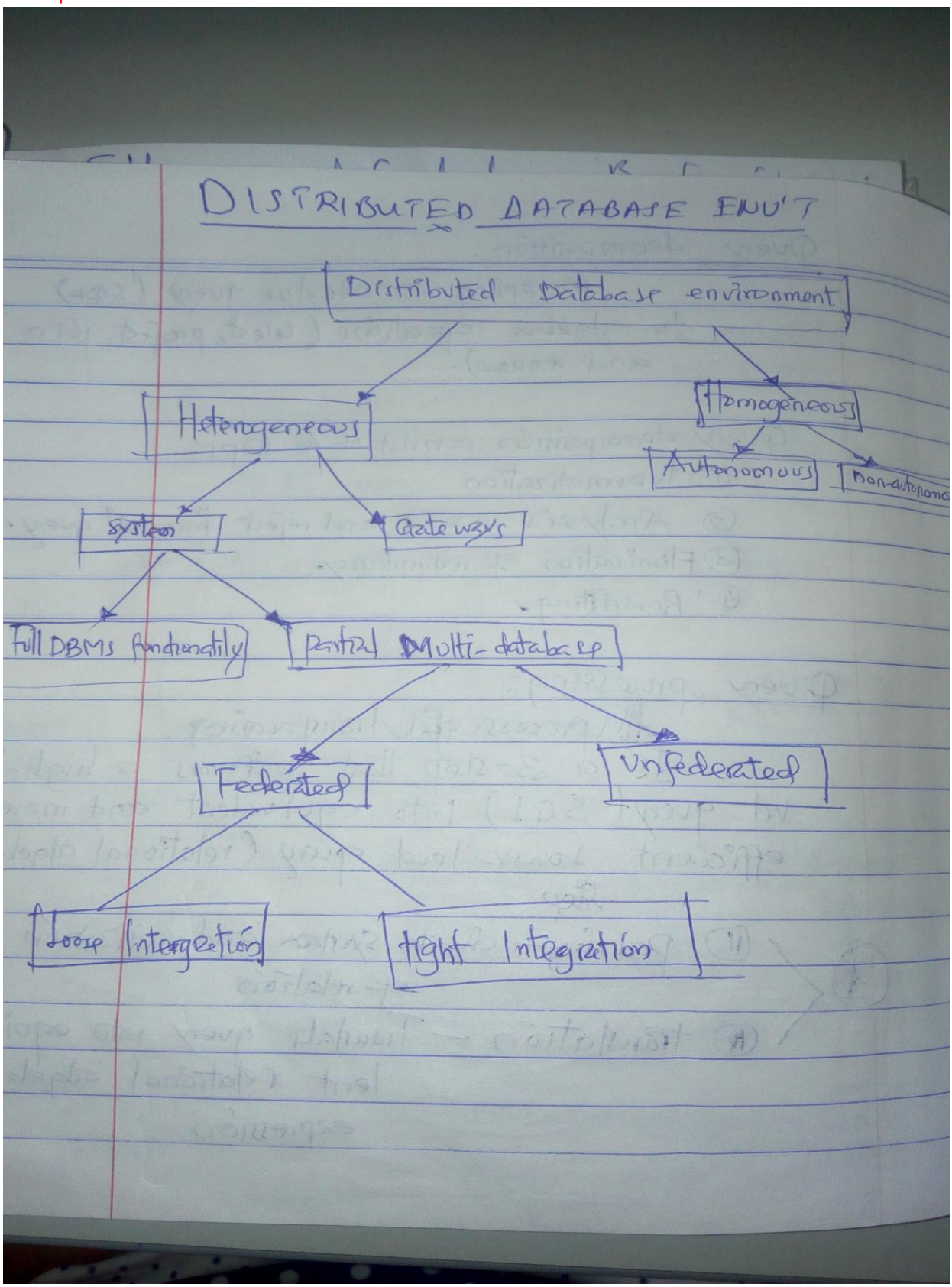
Question three

TABLE 12-1 Comparison of Distributed Database Design Strategies

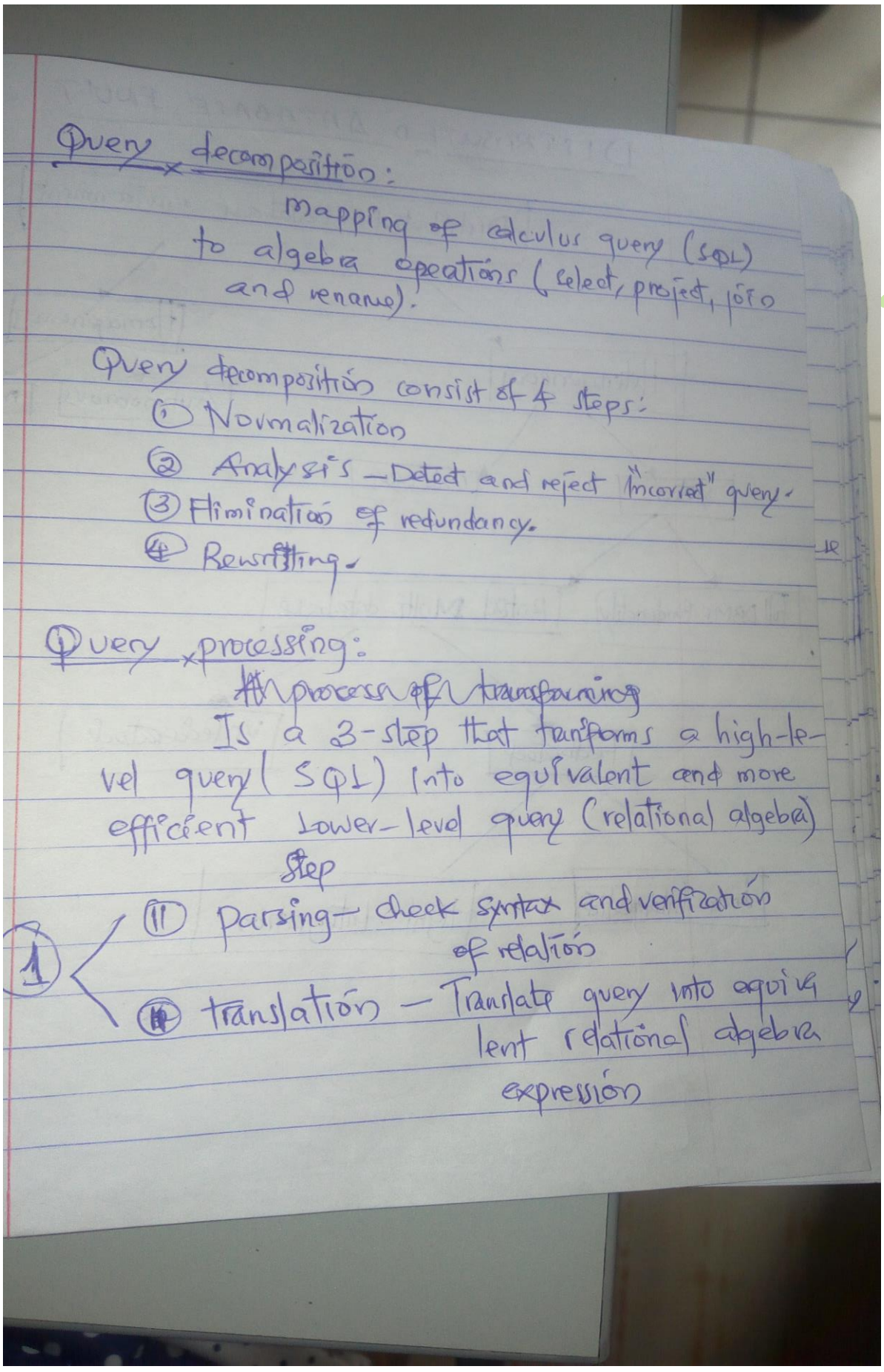
Strategy	Reliability	Expandability	Communications Overhead	Manageability	Data Consistency
Centralized	**POOR:** Highly dependent on central server	**POOR:** Limitations are barriers to performance	**VERY HIGH:** High traffic to one site	**VERY GOOD:** One monolithic site requires little coordination	**EXCELLENT:** All users always have the same data
Replicated with snapshots	**GOOD:** Redundancy and tolerated delays	**VERY GOOD:** Cost of additional copies may be less than linear	**LOW to MEDIUM:** Not constant, but periodic snapshots can cause bursts of network traffic	**VERY GOOD:** Each copy is like every other one	**MEDIUM:** Fine as long as delays are tolerated by business needs
Synchronized replication	**EXCELLENT:** Redundancy and minimal delays	**VERY GOOD:** Cost of additional copies may be low and synchronization work only linear	**MEDIUM:** Messages are constant, but some delays are tolerated	**MEDIUM:** Collisions add some complexity to manageability	**MEDIUM to VERY GOOD:** Close to precise consistency
Integrated partitions	**VERY GOOD:** Effective use of partitioning and redundancy	**VERY GOOD:** New nodes get only data they need without changes in overall database design	**LOW to MEDIUM:** Most queries are local, but queries that require data from multiple sites can cause a temporary load	**DIFFICULT:** Especially difficult for queries that need data from distributed tables, and updates must be tightly coordinated	**VERY POOR:** Considerable effort; and inconsistencies not tolerated
Decentralized with independent partitions	**GOOD:** Depends on only local database availability	**GOOD:** New sites independent of existing ones	**LOW:** Little if any need to pass data or queries across the network (if one exists)	**VERY GOOD:** Easy for each site, until there is a need to share data across sites	**LOW:** No guarantees of consistency; in fact, pretty sure of inconsistency

Desa la kijasi

Question four ??????????? go and google
More possible



Desa la kijasi



Query decomposition:

mapping of calculus query (SQL) to algebra operations (select, project, join and rename).

Query decomposition consist of 4 steps:

1. Normalization
2. Analysis – Detect and reject "incorrect" query.
3. Elimination of redundancy.
4. Rewriting.

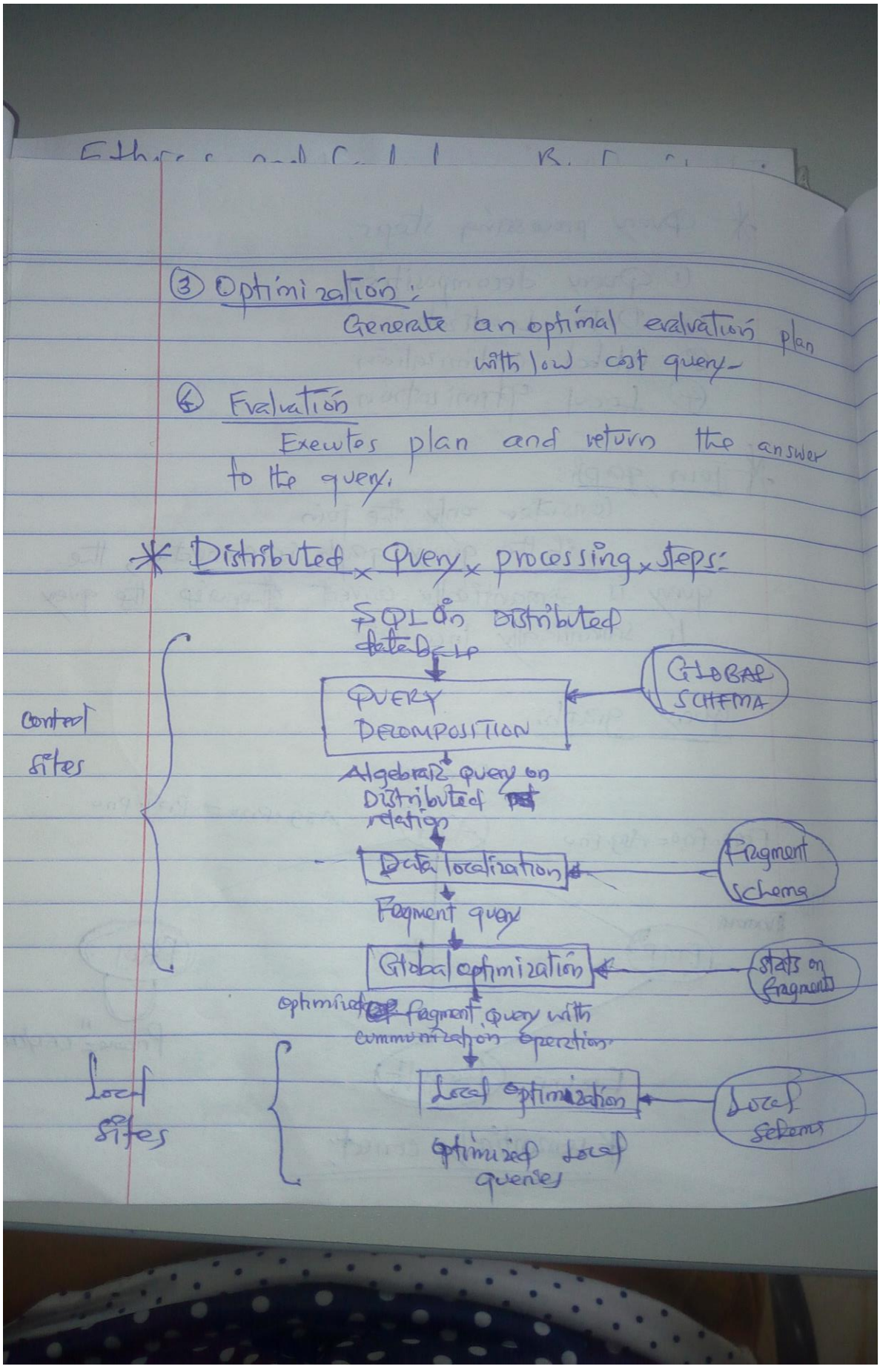
Query processing:

Is a 3-step that tranforms a high-level query (SQL) into equivalent and more efficient lower-level query (relational algebra)

1 Step

(i) Parsing – check syntax and verification of relation

(ii) translation – Translate query into equivalent relational algebra expression


③ Optimization:
Generate an optimal evaluation plan with low cost query.
④ Evaluation
Executes plan and return the answer to the query.
* Distributed Query processing steps:
SQL on distributed database
QUERY DECOMPOSITION
GLOBAL SCHEMA
Control sites
Algebraic query on Distributed relation
Data localization
Fragment schema
Fragment query
Global optimization
stats on fragments
optimized fragment query with communication operation
Local optimization
Local schema
Local sites
optimized local queries

* Query processing steps:

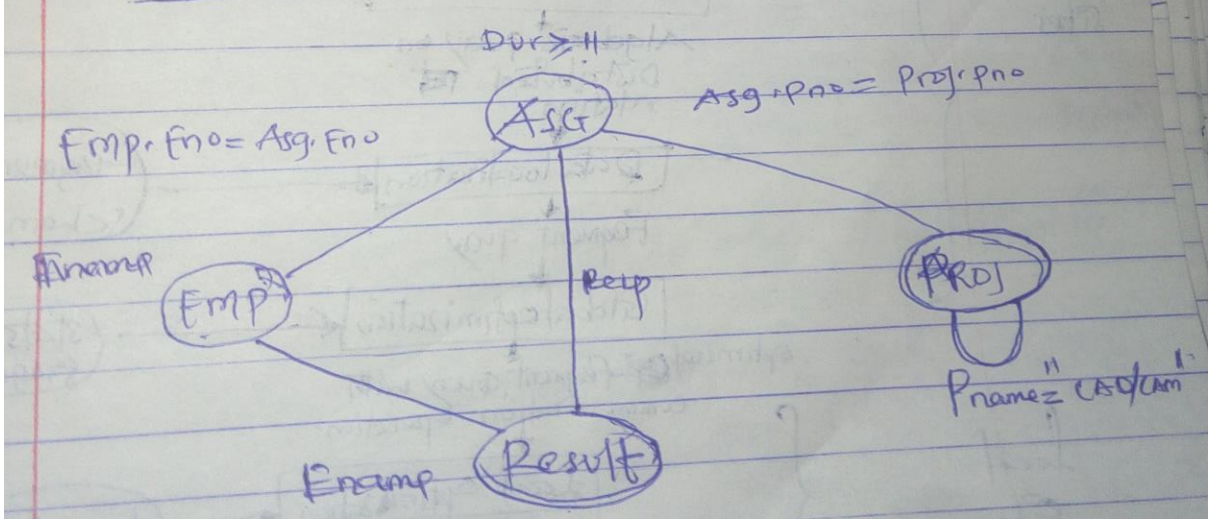
1. Query decomposition
2. Data localization
3. Global optimization
4. Local optimization

* Join graph:

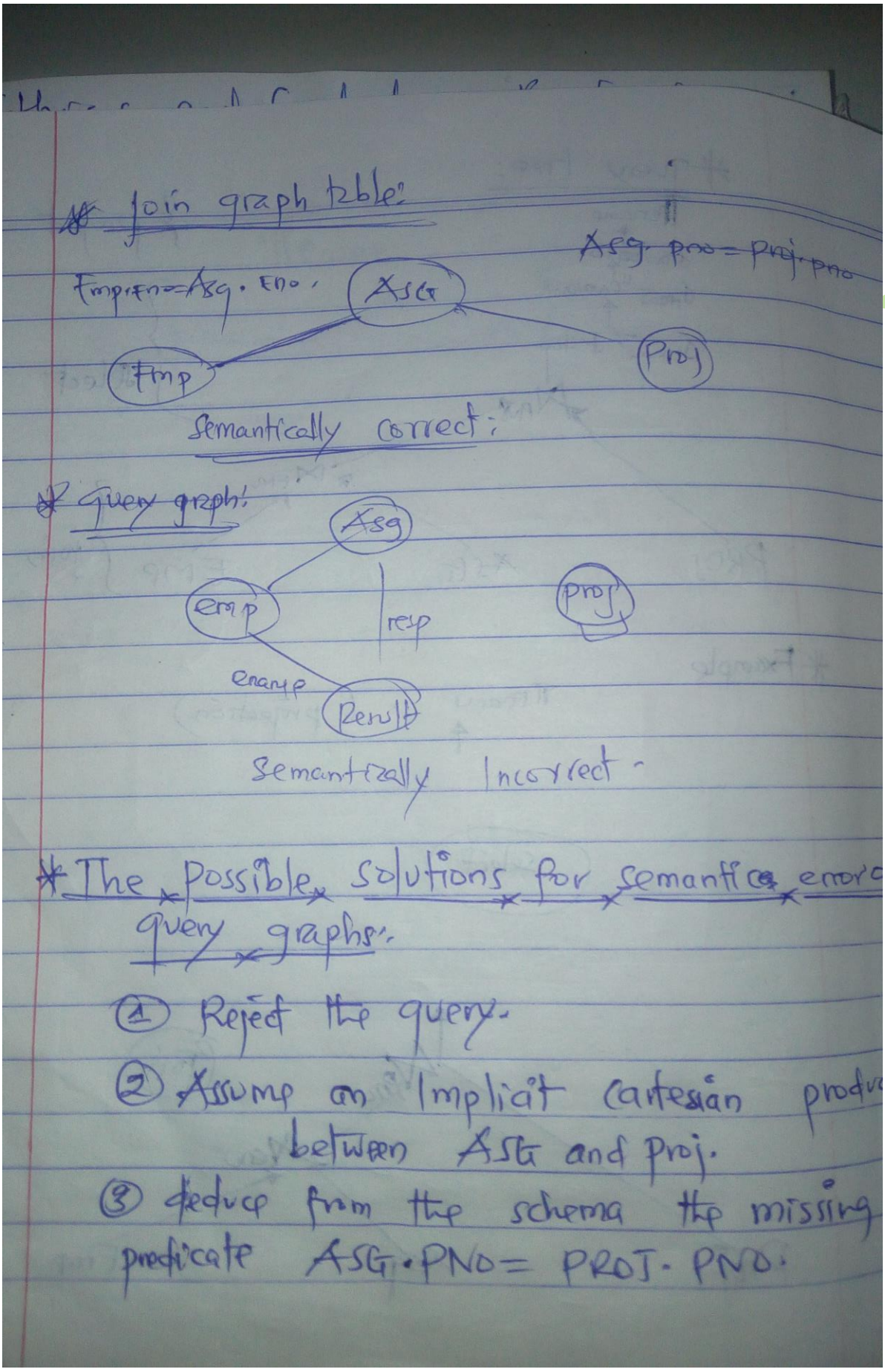
Consider only the join

If the query graph is connected, the query is semantically correct otherwise the query is semantically incorrect.

Query graph:



* semantically correct.



* Join graph tree:

Asg.pno = proj.pno

Emp.eno = Asg.eno

Semantically correct:

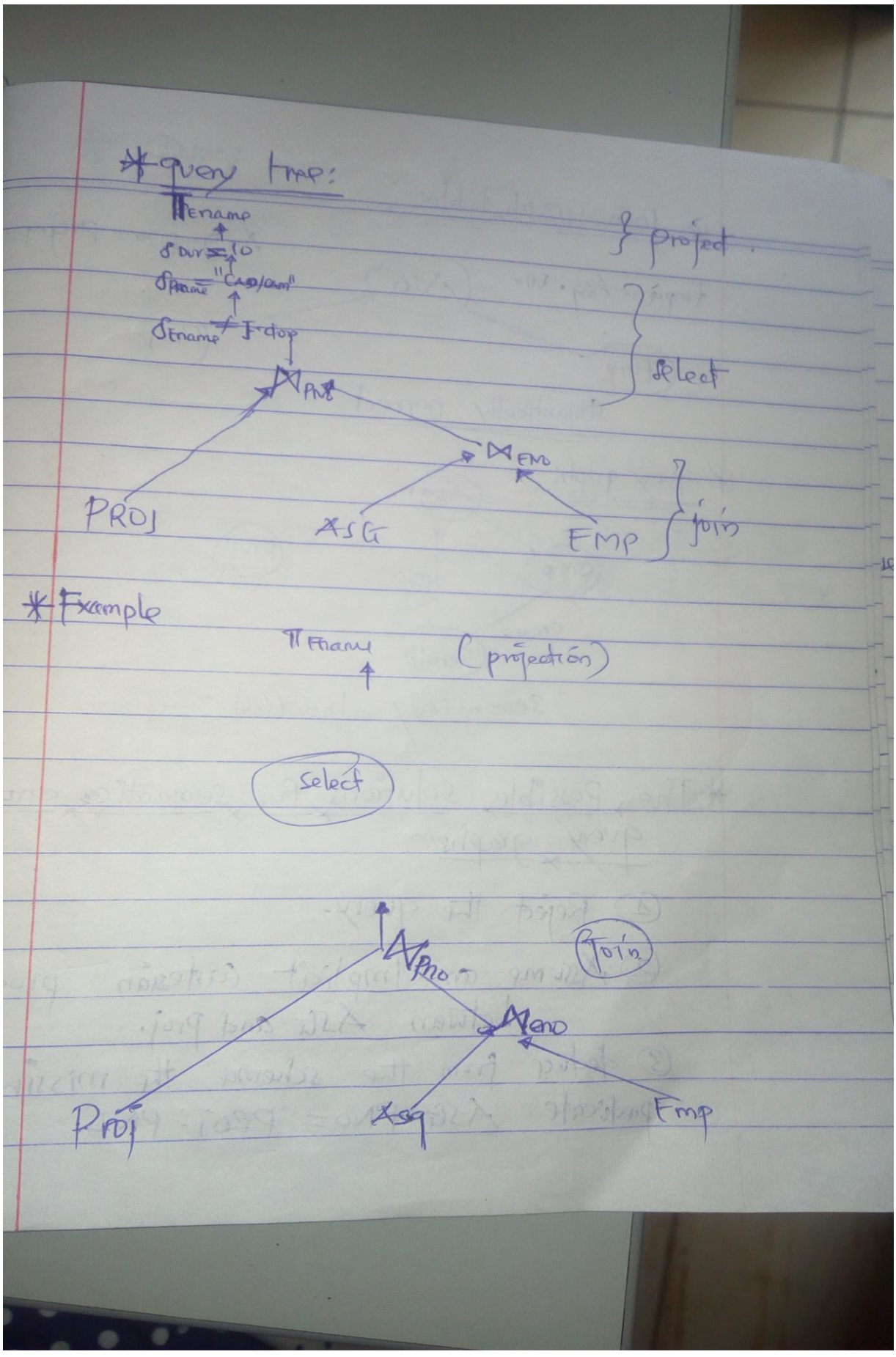
* Query graph:

Semantically Incorrect.

* The possible solutions for semantic error query graphs:

1. Reject the query.
2. Assume an Impliait cartesian produ between ASG and proj.
3. deduce from the schema the missing predicate ASG.PNO = PROJ.PNO.

Desa la kijasi



Desa la kijasi