Indexing

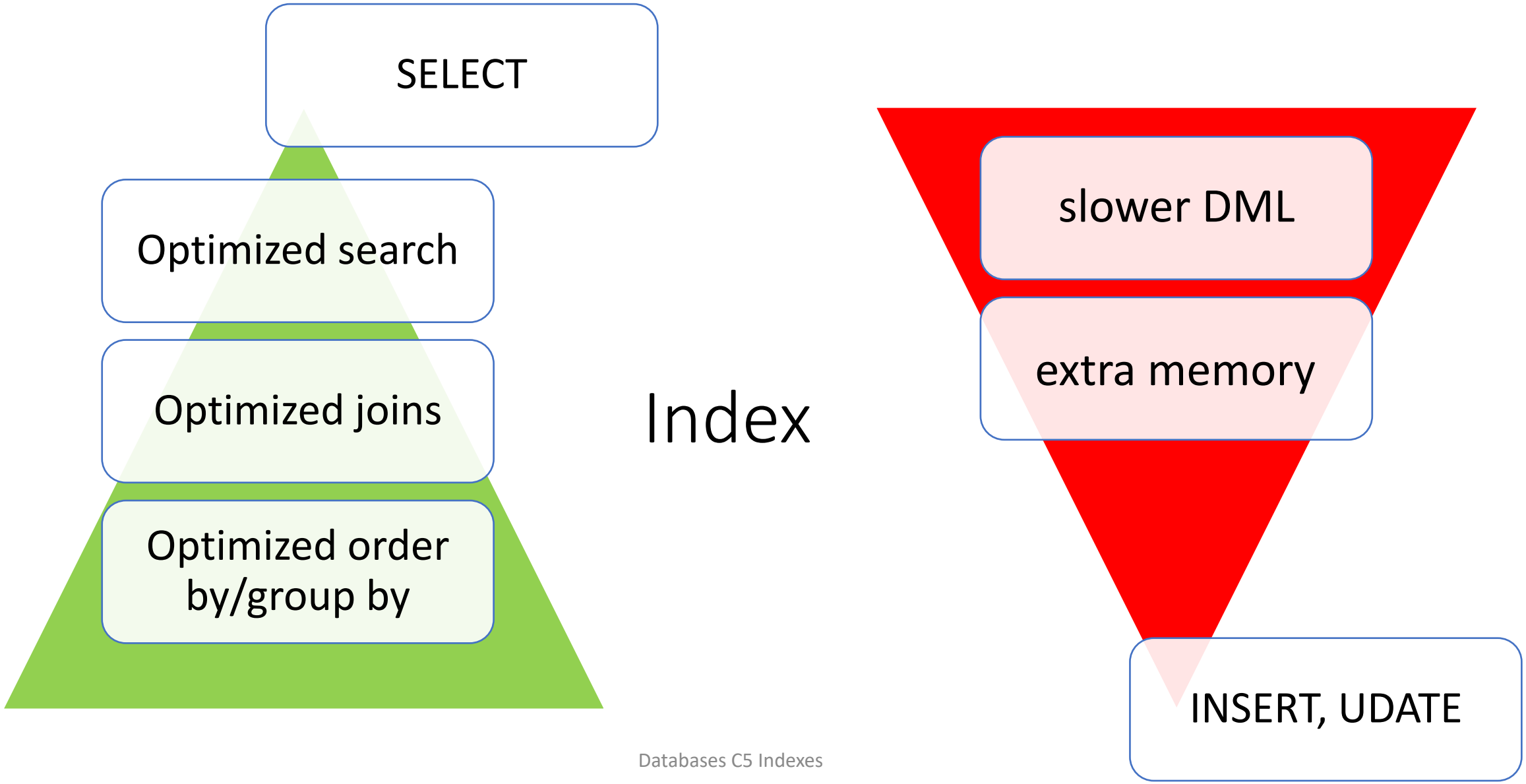
COURSE 5: Databases

Indexes

- Work like textbook indexes.

- Optimized search.
- Optimized joins (lookup in more than one table)
- Optimized order by/group by

- slower DML (insert and update operations).
- extra memory


SELECT
Optimized search
Optimized joins
Optimized order by/group by
Index
slower DML
extra memory
INSERT, UDATE

Ordered Indexes

Ordered Indexes

- Associated with a search key.
- Stores the values of the search key in order.
- Maps search key to data records using specific data structures.

- Index entry:

  search key value + pointers to records containing search key value.

- A table may have several ordered indexes.

Clustered index

- Defines the order in which data is physically stored in a table.
- Only one clustered index on a table (data can be stored in only one order)
- A cluster index is created automatically when a primary key is defined.
- *No additional data structure is needed to store the index. The index is the table itself.*

Clustered index - Oracle IoT (index organized tabled)

- Oracle: IOT index organized tables. Table is stored in a B-tree structure. (key and non-keys column are stored in leafs)

- In PotgreSQL

Non-clustered index

- Indexes that specify a different order from the sequential order of the file are non-clustering indexes or secondary indexes.
- Several secondary indexes may be created for a table.
- Slower than a clustered index.
- Requires additional disk space and additional data structures.

Non-clustered index

- A non-cluster index contains index key values.

- Each key value entry has a **row-locator**: pointer to the data-row that contains the key-value. The row-locator may indicate:

  - the key in the clustered index.
  - the pointer to the row.

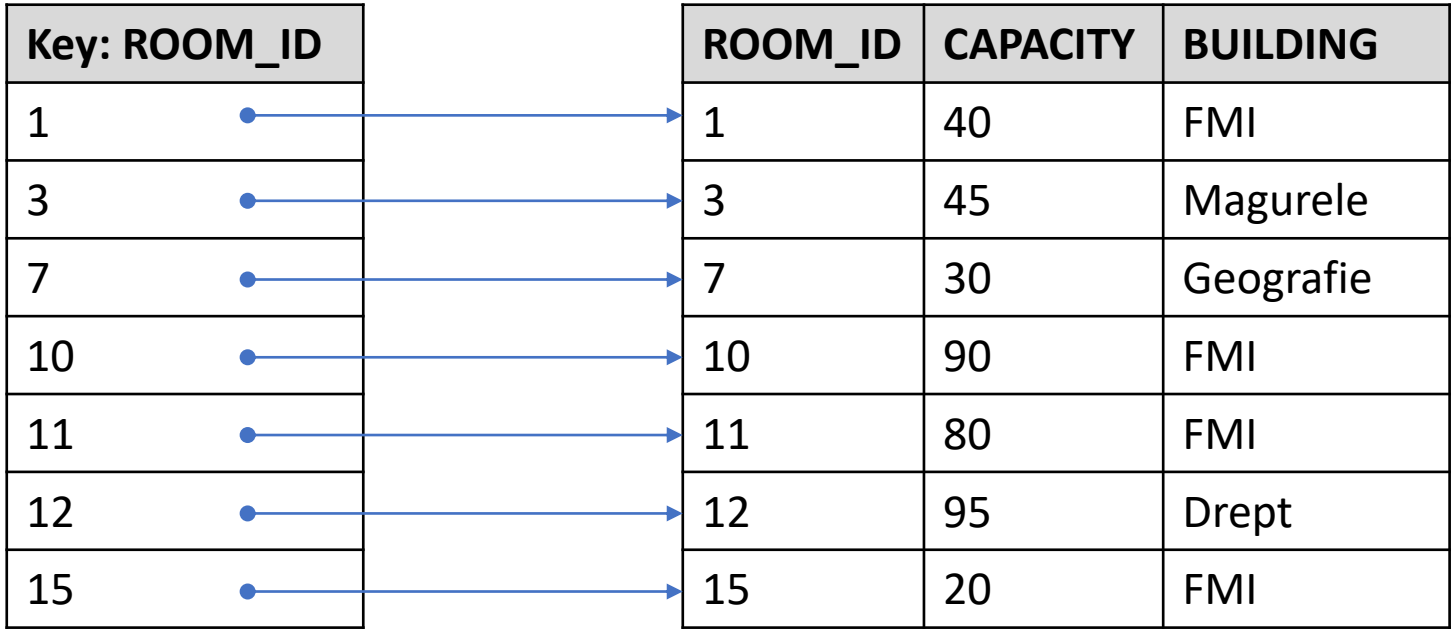
Sparse or dense index

Sparse indices: contains only some key values. Works only for clustering indices. If we want to find all records with search key value *v,* we located the first row with v, then we check the records in order, until key search is different than *v.*

Dense indices: contains a pair (*key_value*, *adr_first_record_with_value*) for each possible search key value.

Clustered index (SqlServer, MySql)

Dense indices

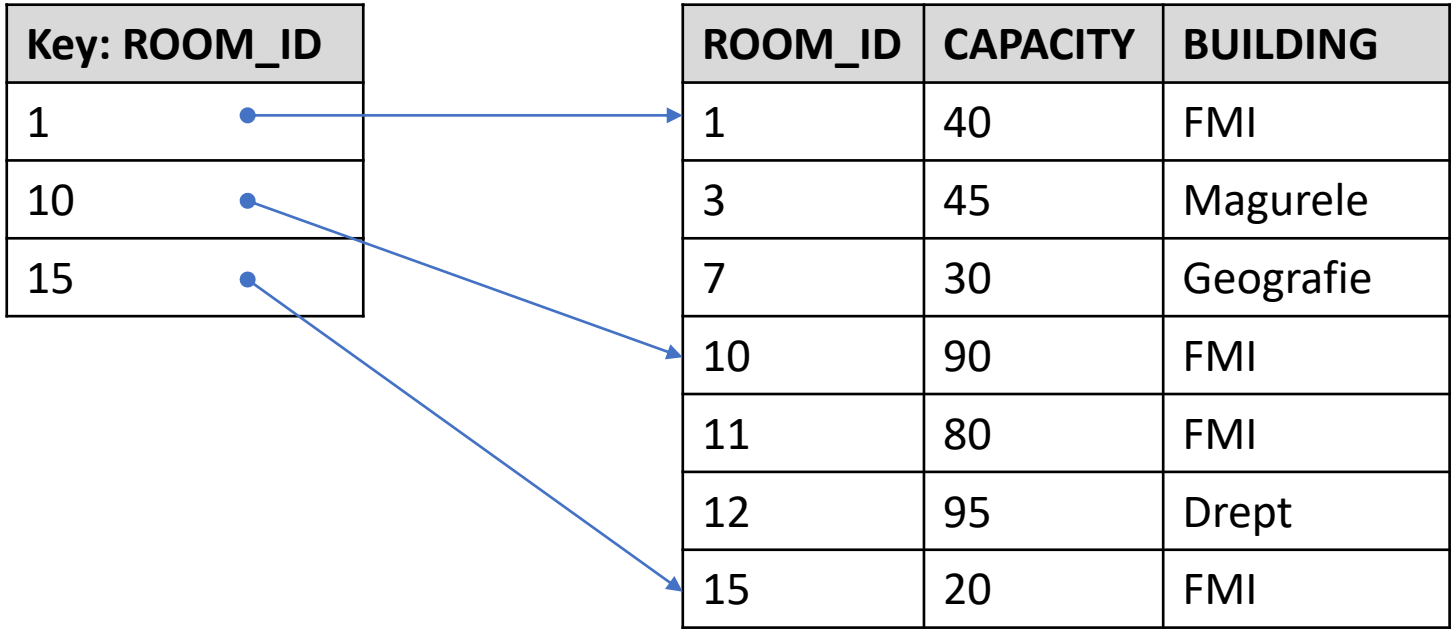


Key: ROOM_ID
1
3
7
10
11
12
15

ROOM_ID	CAPACITY	BUILDING
1	40	FMI
3	45	Magurele
7	30	Geografie
10	90	FMI
11	80	FMI
12	95	Drept
15	20	FMI

Clustered index (SqlServer, MySql)

sparse indices – non primary How does find(12) works?



Key: ROOM_ID
1
10
15

ROOM_ID	CAPACITY	BUILDING
1	40	FMI
3	45	Magurele
7	30	Geografie
10	90	FMI
11	80	FMI
12	95	Drept
15	20	FMI

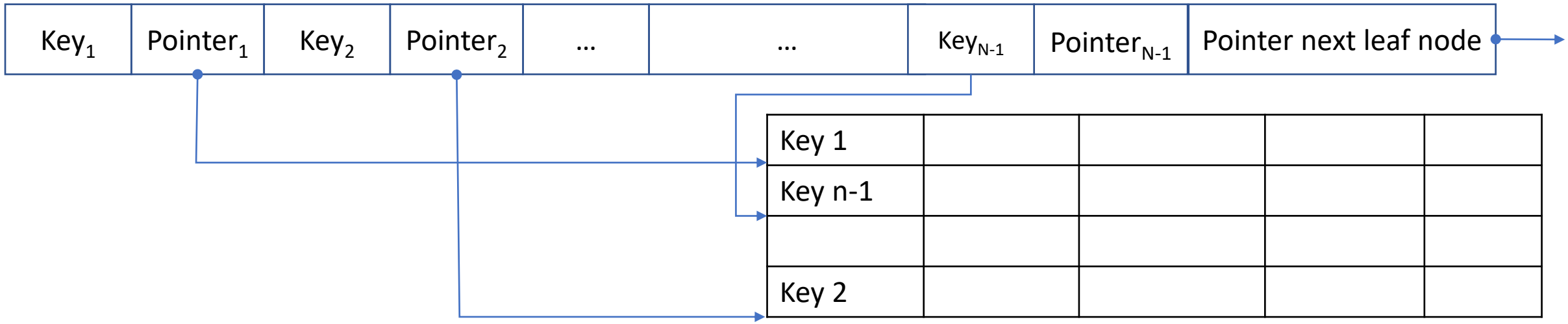
B^+ – Tree

- Balanced tree.
- The number of nodes traversed in order to reach a leaf block is the same for each leaf block.

- Default index type in Oracle.
- Two types of nodes: branch blocks and leaf blocks.
- Branch blocks -- pointers to lower levels.
- Leaf blocks contain rowids/physical address.

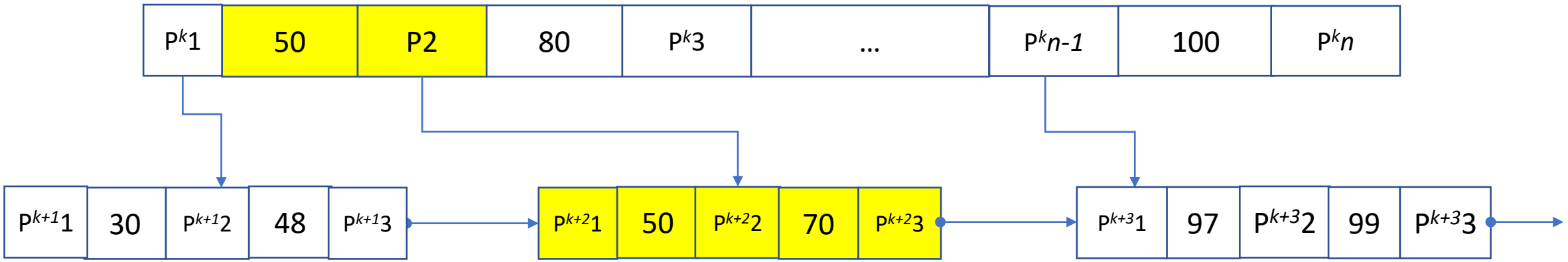
B^+ – Tree

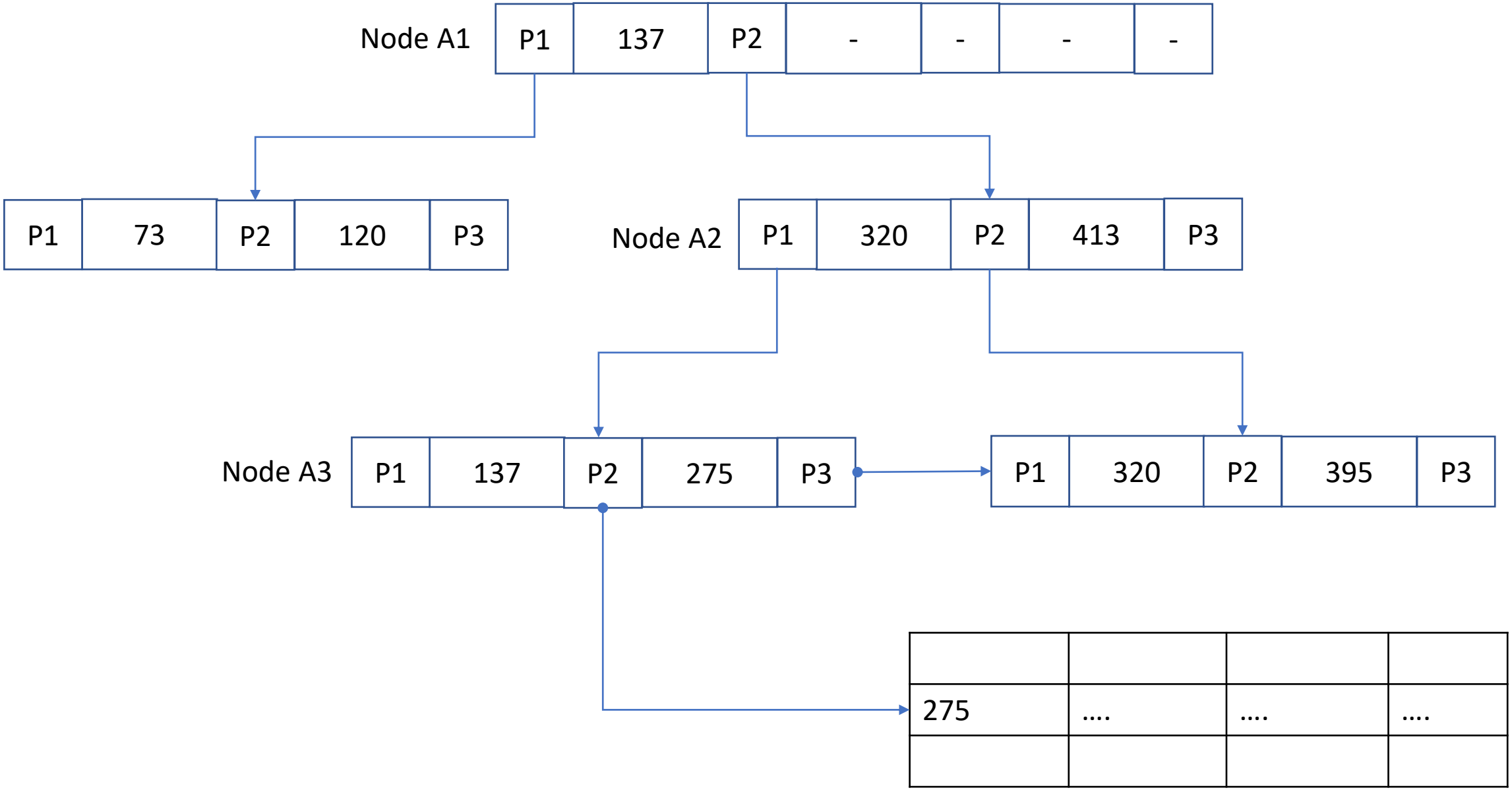
- Leaf node:



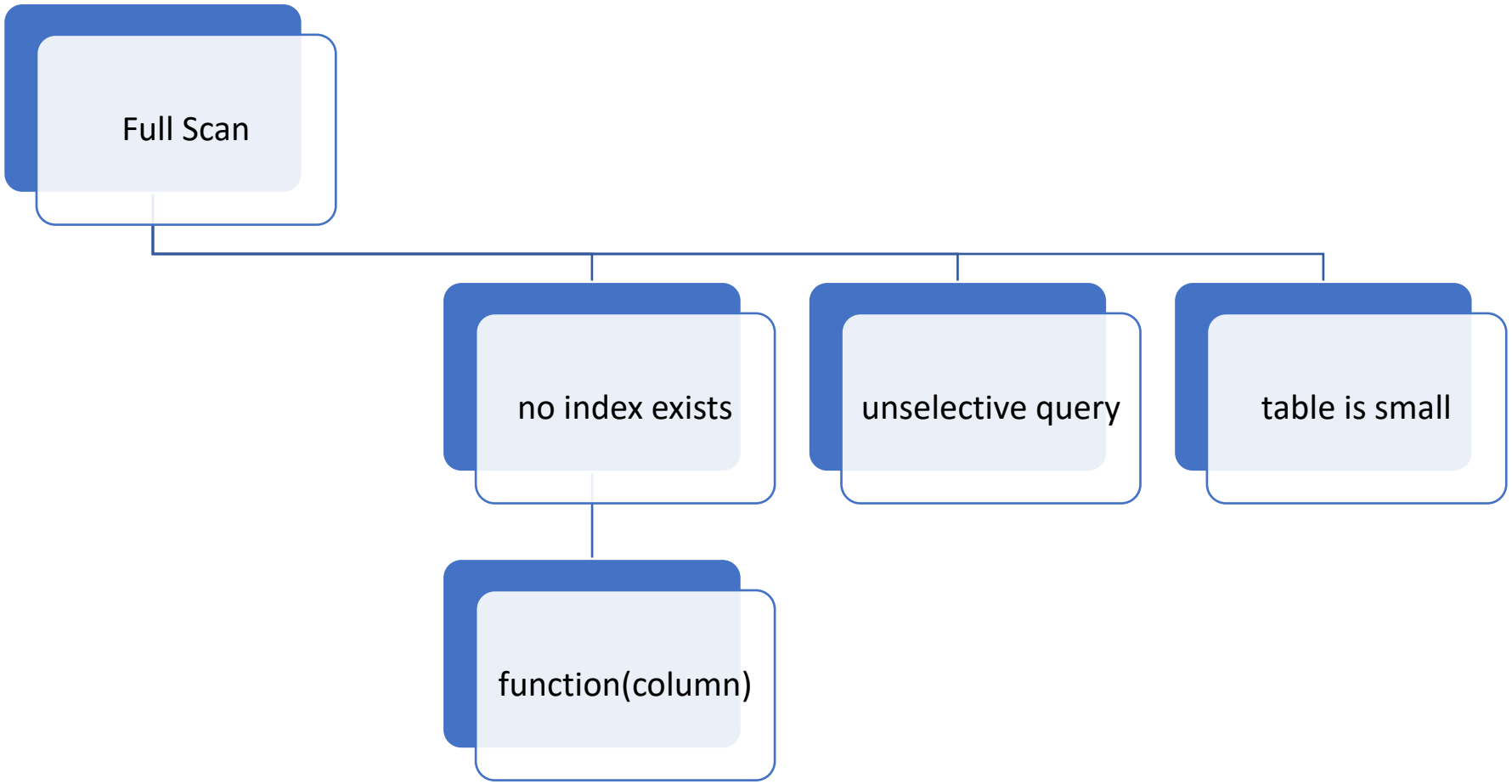
B^+ – Tree

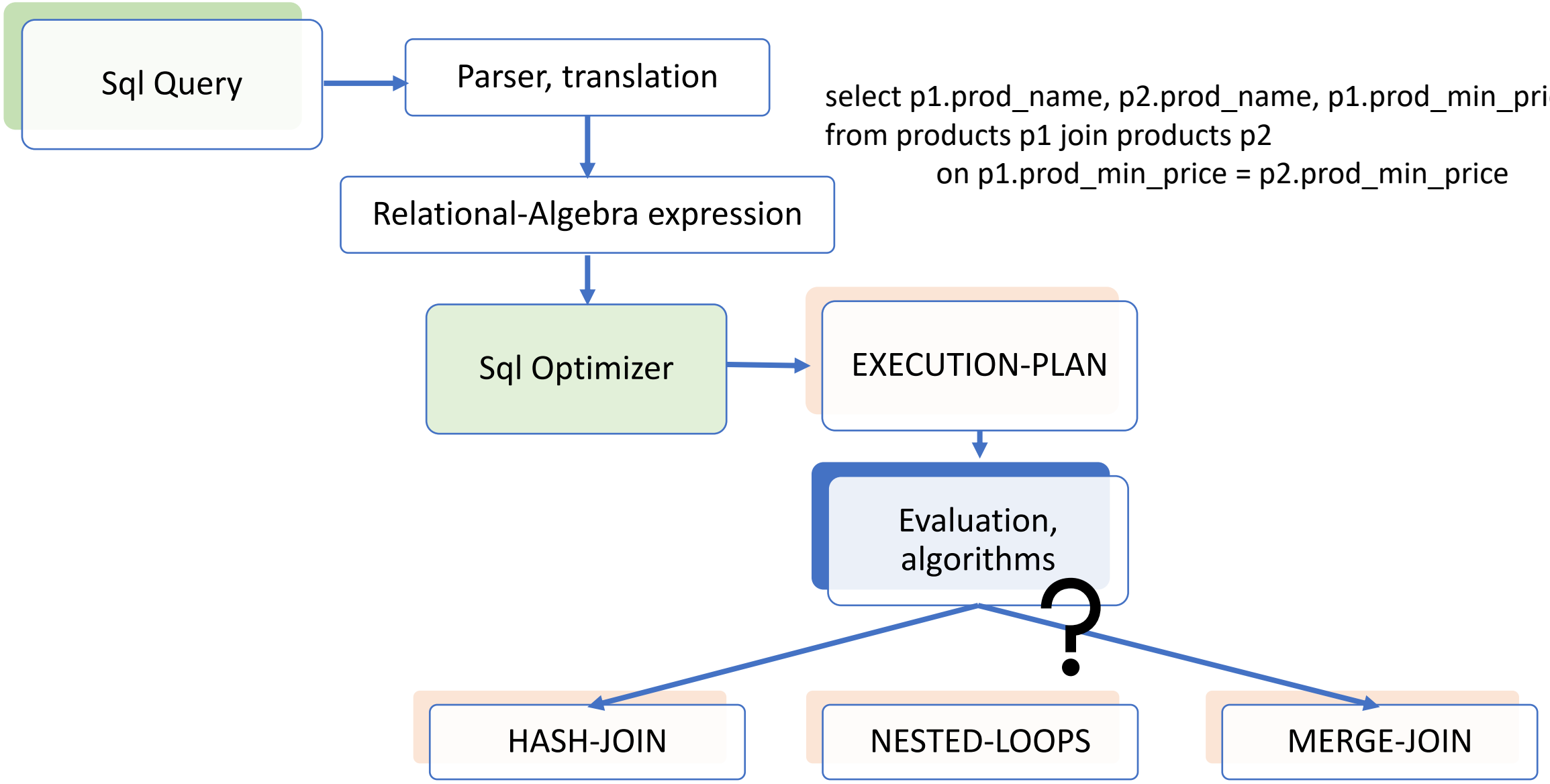
- Non-leaf node: keys --> intervals; pointers --> pointers to tree nodes.
- A non-leaf node may hold between $\lceil n/2 \rceil$ and n pointers.



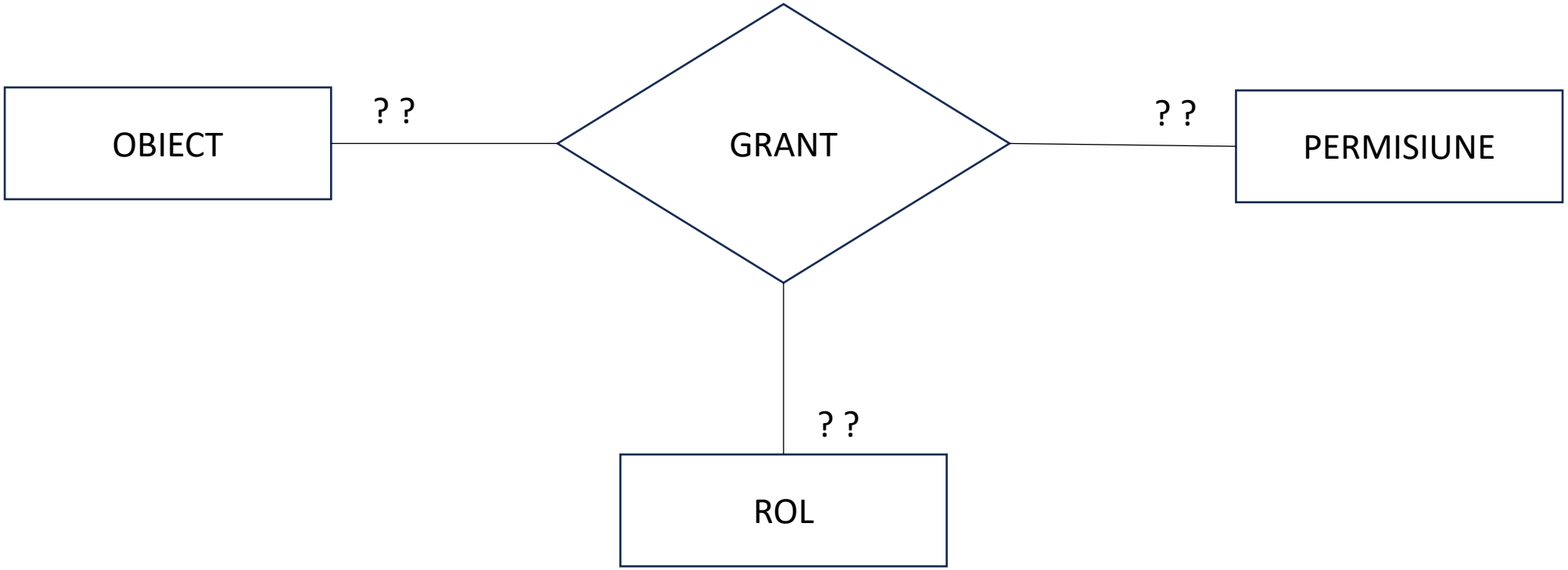

Node A1
P1
137
P2
-
-
-
-
P1
73
P2
120
P3
Node A2
P1
320
P2
413
P3
Node A3
P1
137
P2
275
P3
P1
320
P2
395
P3
275
....
....
....

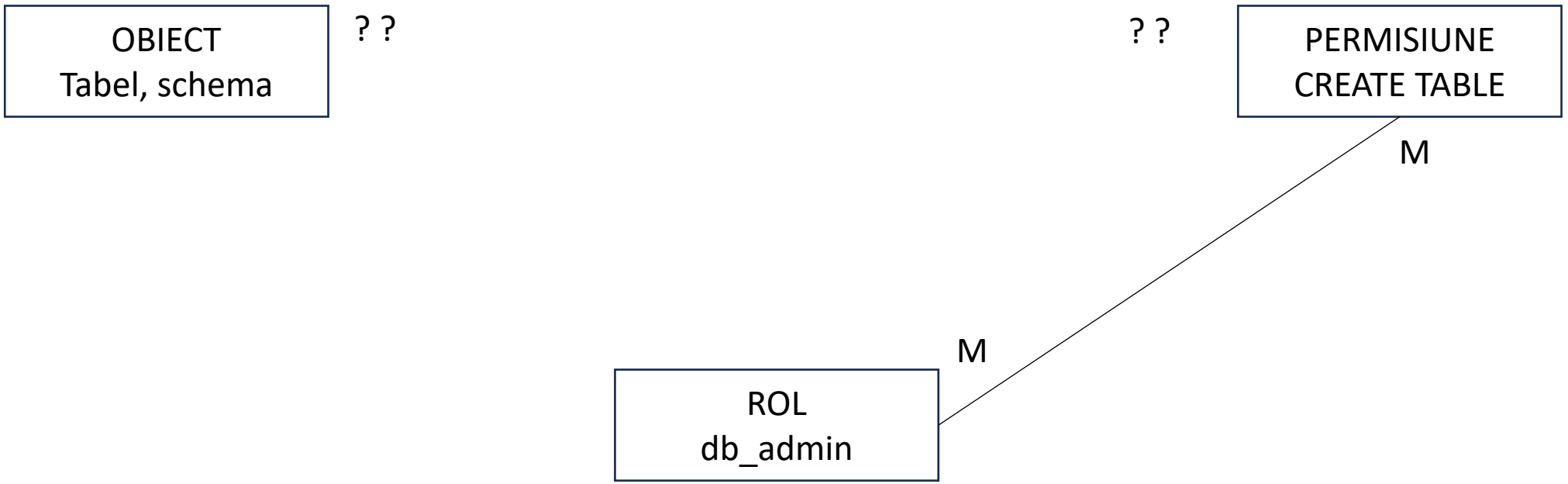
Sql Optimizer

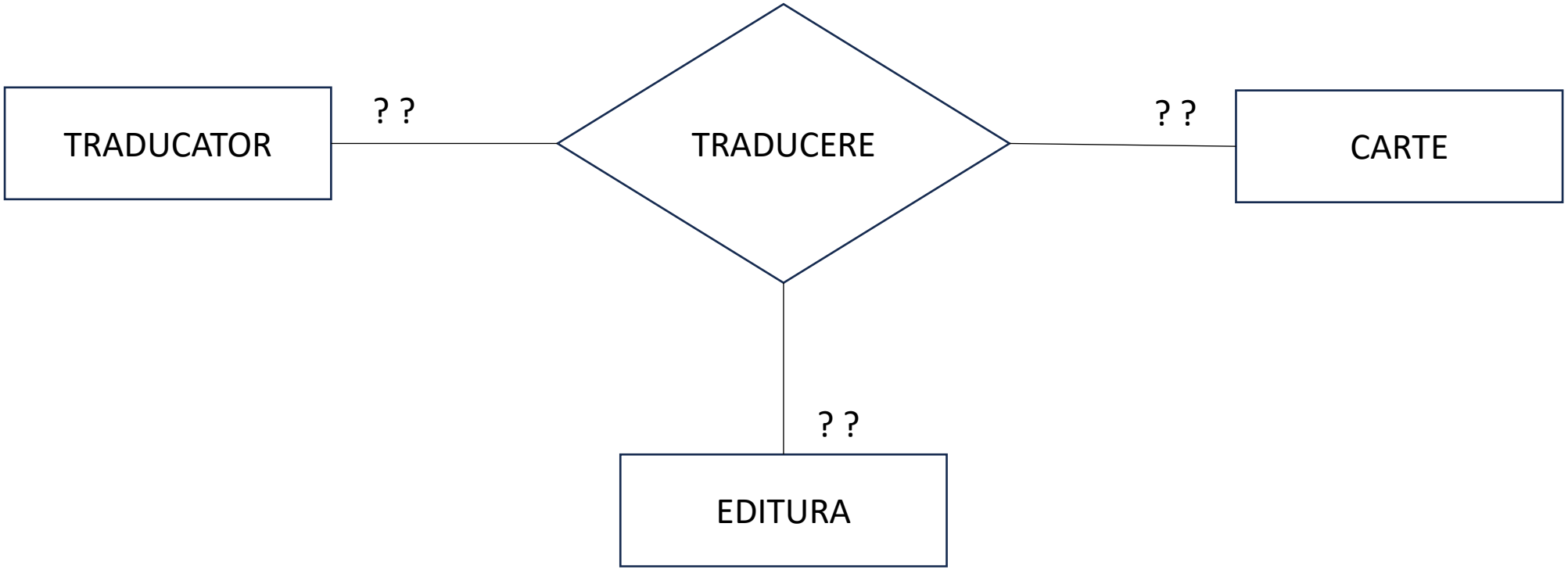


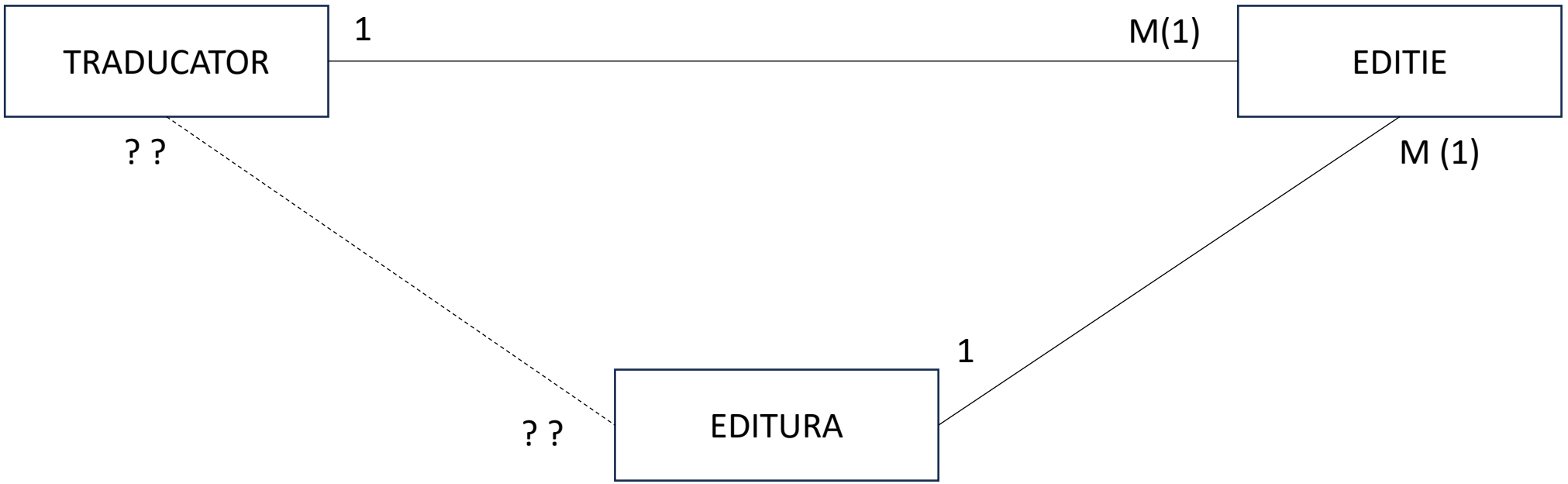


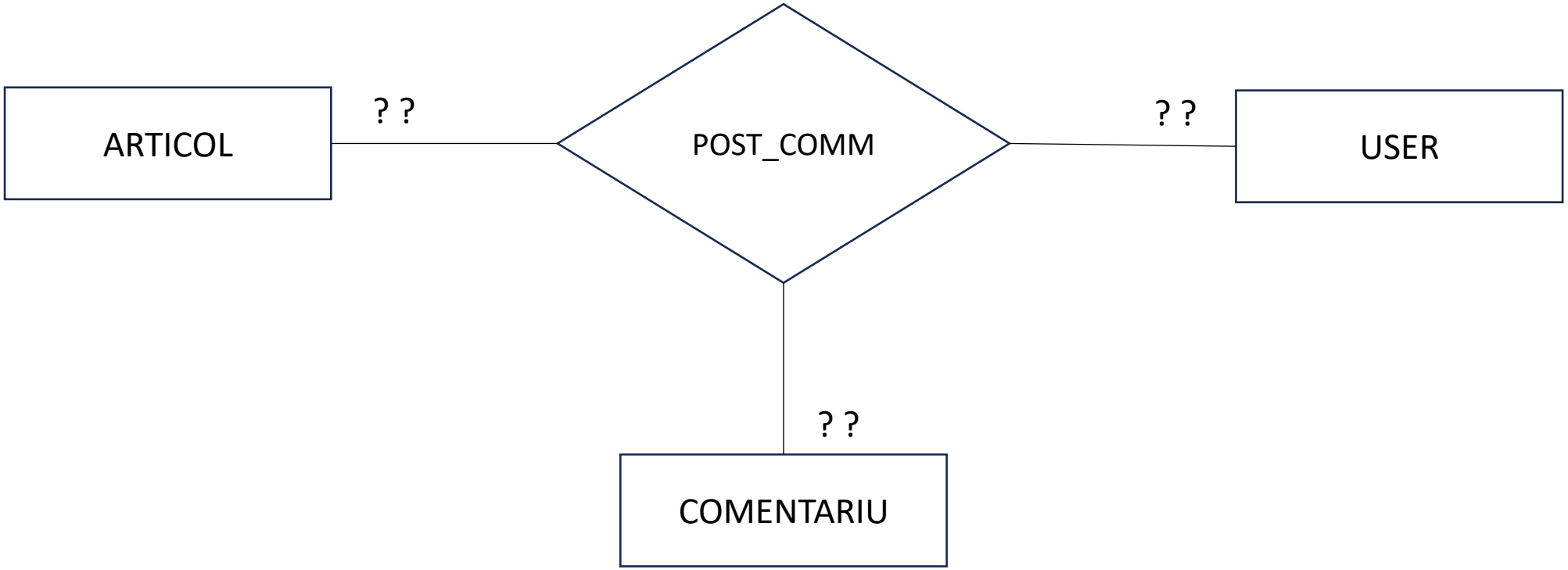

```
select p1.prod_name, p2.prod_name, p1.prod_min_price
from products p1 join products p2
        on p1.prod_min_price = p2.prod_min_price
```

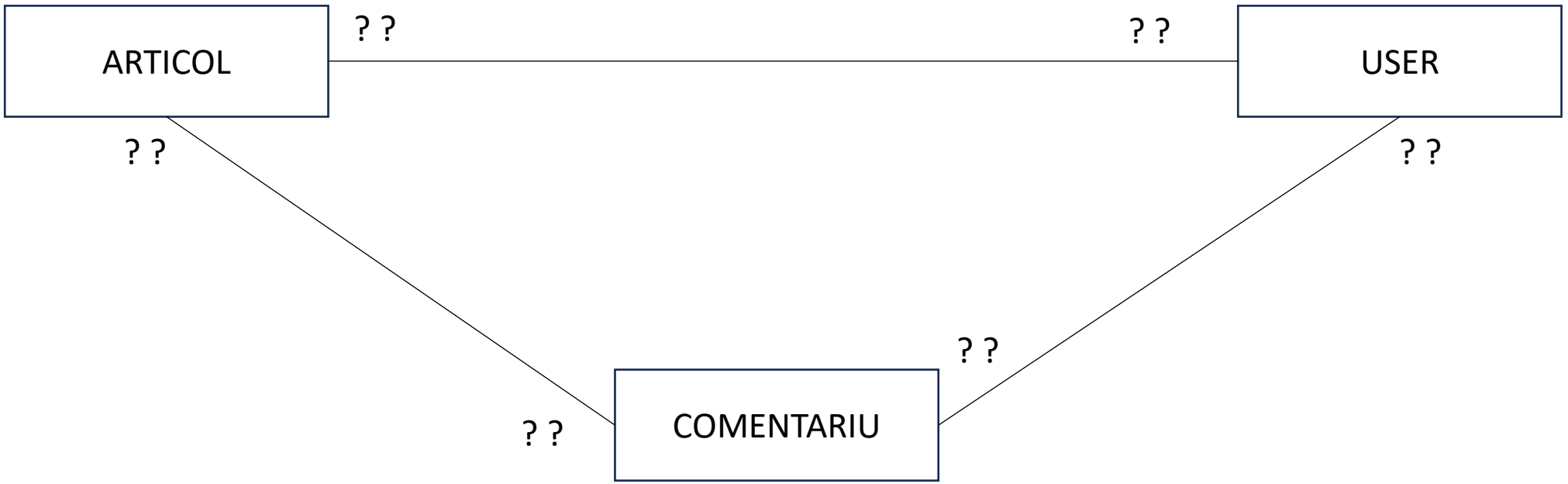

OBIECT
? ?
GRANT
? ?
PERMISIUNE
? ?
ROL

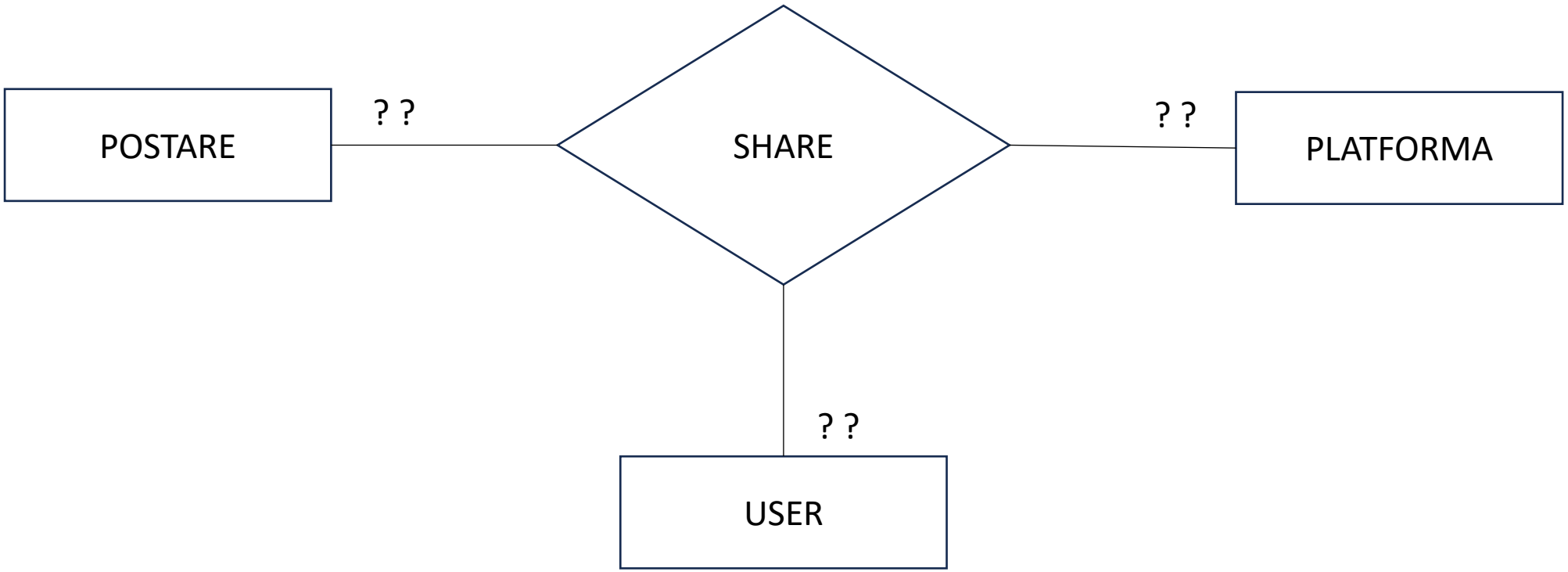

OBIECT
Tabel, schema
? ?
? ?
PERMISIUNE
CREATE TABLE
M
M
ROL
db_admin

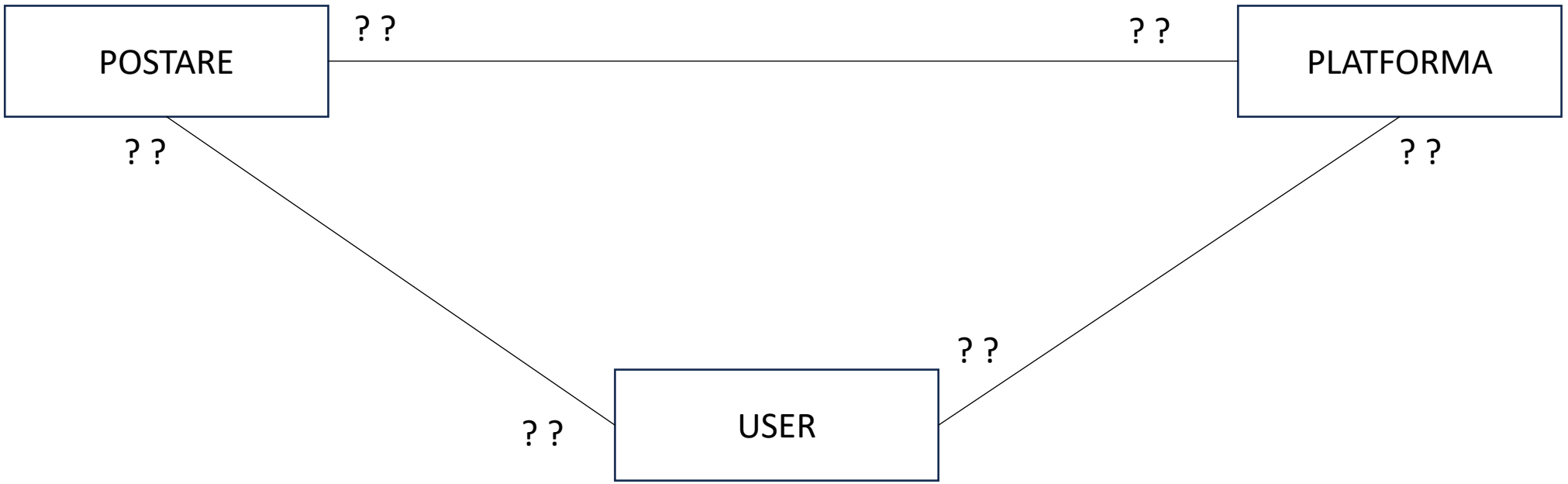

TRADUCATOR
? ?
TRADUCERE
? ?
CARTE
? ?
EDITURA


TRADUCATOR
1
M(1)
EDITIE
? ?
M (1)
1
EDITURA
? ?


ARTICOL
? ?
POST_COMM
? ?
USER
? ?
COMENTARIU


ARTICOL
? ?
? ?
USER
? ?
? ?
? ?
? ?
COMENTARIU


POSTARE
? ?
SHARE
? ?
PLATFORMA
? ?
USER


POSTARE
? ?
? ?
PLATFORMA
? ?
? ?
? ?
? ?
USER