CBCS SCHEME

USN 15CS651

Sixth Semester B.E. Degree Examination, Dec.2019/Jan.2020
Data Mining and Data Warehousing

Time: 3 hrs. Max. Marks: 80

Note: *Answer any FIVE full questions, choosing ONE full question from each module.*

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. 2. Any revealing of identification, appeal to evaluator and /or equations written eg. 42+8 = 50, will be treated as malpractice.

Module-1

1 a. What is Data warehouse? Explain three tier architecture of data warehouse. (08 Marks)
b. Explain the schemas of multidimensional data models. (08 Marks)

OR

2 a. What is Data cube measure? Explain the categorization of measures. (08 Marks)
b. Explain data cube operations with examples. (08 Marks)

Module-2

3 a. Explain data cube computation and curse of dimensionality. (08 Marks)
b. Explain different methods of indexing OLAP data. (08 Marks)

OR

4 a. State and explain various data mining tasks. (08 Marks)
b. Define Similarity and dissimilarity between the objects. Find SMC and Jaccord's coefficient of two binary vectors.
X = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0) Y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1). (08 Marks)

Module-3

5 a. What is Association Analysis? Explain Association rule , Support and Confidence. (08 Marks)
b. State Apriori principle. Write apriori algorithm for frequent itemset. (08 Marks)

OR

6 a. Construct an FP tree for the following dataset.

TID	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

(08 Marks)

b. Explain the strategies used in frequent itemset generation. (08 Marks)

Module-4

7 a. Explain the general approach for solving classification problem. (08 Marks)
b. Write the algorithm for decision tree induction. (08 Marks)

OR

8 a. Explain the methods of comparing classifiers. (08 Marks)
b. Write the characteristics of nearest neighbor classifier. (08 Marks)

Module-5

9 a. Explain the requirements of cluster analysis. (08 Marks)
b. State and explain K – means algorithm. (08 Marks)

OR

10 a. Write DBSCAN clustering algorithm and estimate time and space complexity. (08 Marks)
b. State and explain the issues in cluster evaluation. (08 Marks)

* * * * *

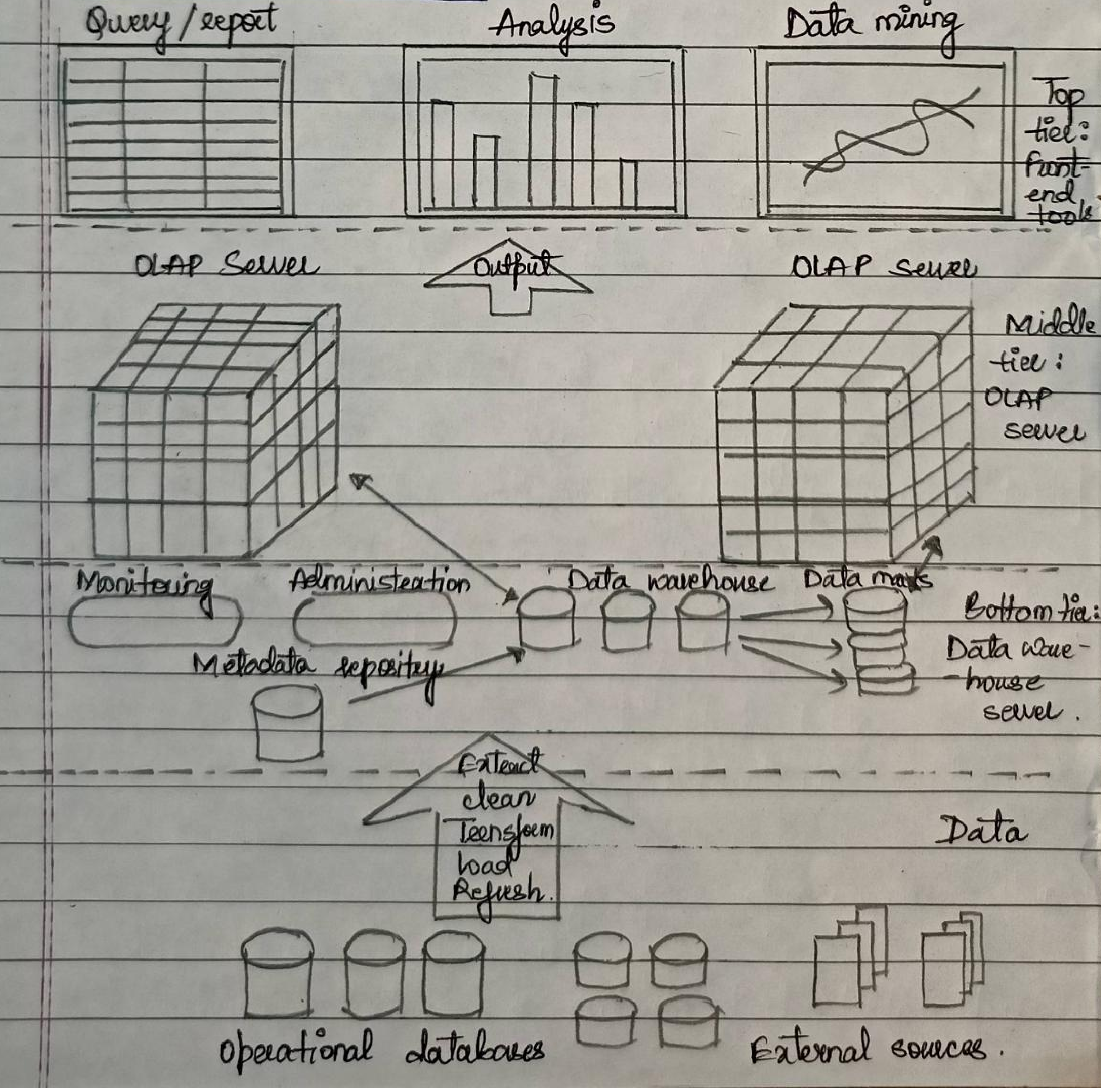
Data Mining and Data Warehousing.

MODULE - 1

1) (a) What is Data Warehouse ? Explain three tier architecture of data warehouse.

Ans:- A Data Warehouse is a subject - oriented, integrated, time - varient, and non - volatile collection of data in support of managment's decision making process.

→ Three tier architecture



Data Warehouse often adopt a three-tier architecture.

1). <u>The Bottom tier</u> :- is a warehouse database server, that is always a relational database system.

- Back-end tools and utilities are used to feed data into the bottom-tier from operational databases or other external sources. [Eg, Customer profile information provided by external consultants].
- These tools & utilities perform data extraction, cleaning and transformation as well as load & refresh functions to update data warehouse.
- The data are extracted using application program interfaces known as gateways.
- A gateways is supported by the underlying DBMS & allows client programs to generate SQL code to be executed to a server.

Eg : gateways include ODBC [Open Database Connection] and OLEDB [Object Linking and Embedding Database] by Microsoft and JDBC [Java Database Connection]

2). <u>The middle tier</u> :- is an OLAP server, that is typically implemented using either,

a) relational OLAP [ROLAP] model (i.e, an extended relational DBMS that maps operations on multidimensional data to standard relational operations).

Scanned by TapScanner

b) Multidimensional OLAP (MOLAP) model i.e, a special purpose server that directly implements multidimensional data and operations.

3). The top tier :- is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools. eg:- trend analysis, prediction and so on.

1) (b) Explain the schemas of multidimensional data models.

Ans:- Schemas for Multidimesional data models.

i) STAR SCHEMA

- The most commonly modelling paradigm is the star schema, in which the data warehouse contains,

a) a large central table [fact table] containing the bulk of data, with no redundancy.

b). a set of smaller attendant tables [dimension table], one for each dimension.

- The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Eg :- A star schema for AllElectronics sales is shown in fig.

- Sales are considered along four dimensions : item, time, branch & location.
- The schema contains a central fact table for

sales that contains keys to each of the four dimensions with two measures : dollars_sold & units_sold.

- To minimize the size of the fact table dimension identifiers [eg. time_key and item_key] are system generated identifiers.

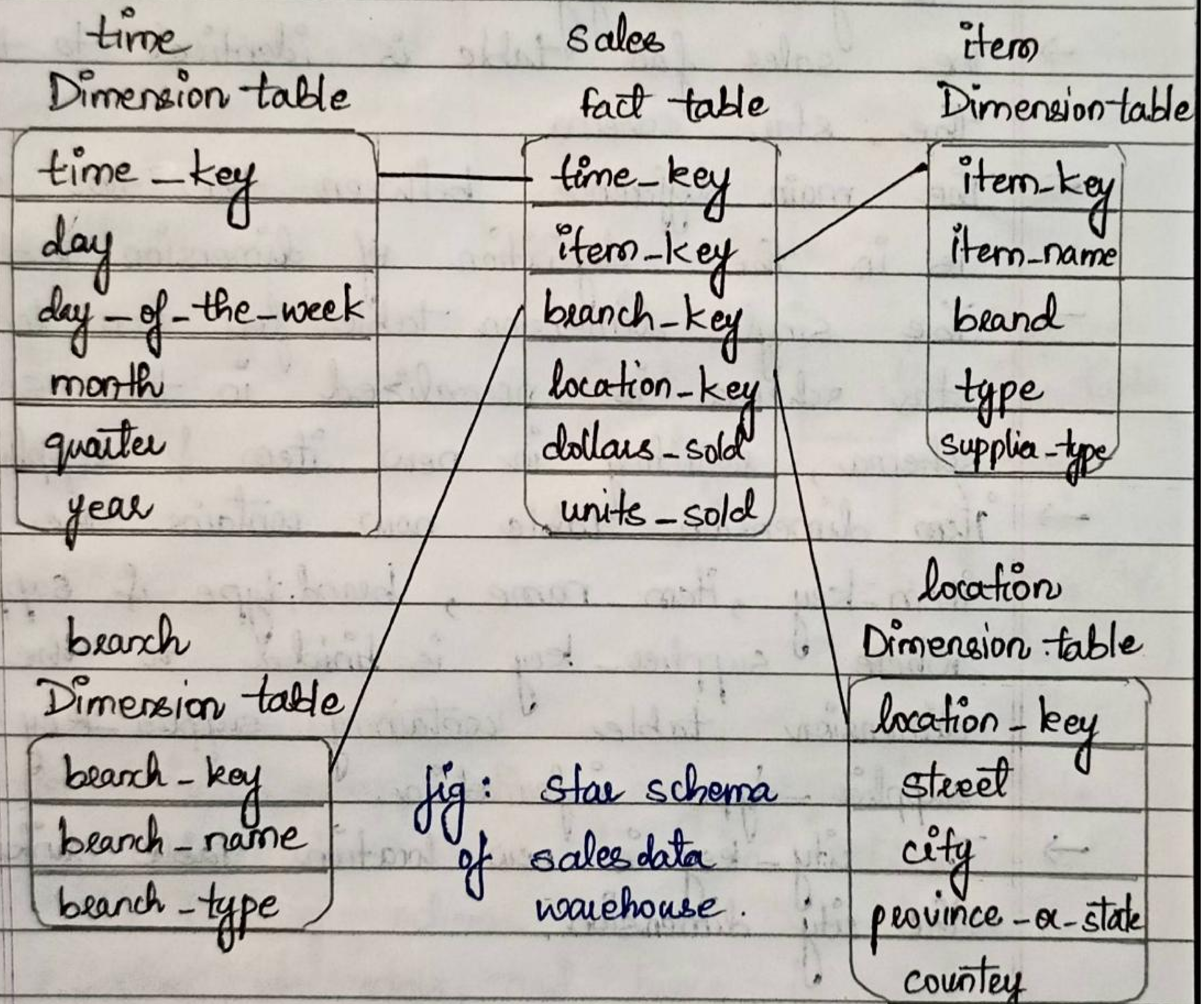


fig: star schema of sales data warehouse.

ii) <u>SNOWFLAKE SCHEMA</u>.

- Snowflake schema is a variant of the star schema model, where some dimension tables are normalized, there by further splitting the data into additional tables.
- The resulting schema graph forms a slope similar to snowflake.

- The snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query.
- The system performance may be adversely impacted.

Eg :→ A snowflake schema for AllElectronics sales is given in fig.

→ The sales fact table is identical to that of the star schema.

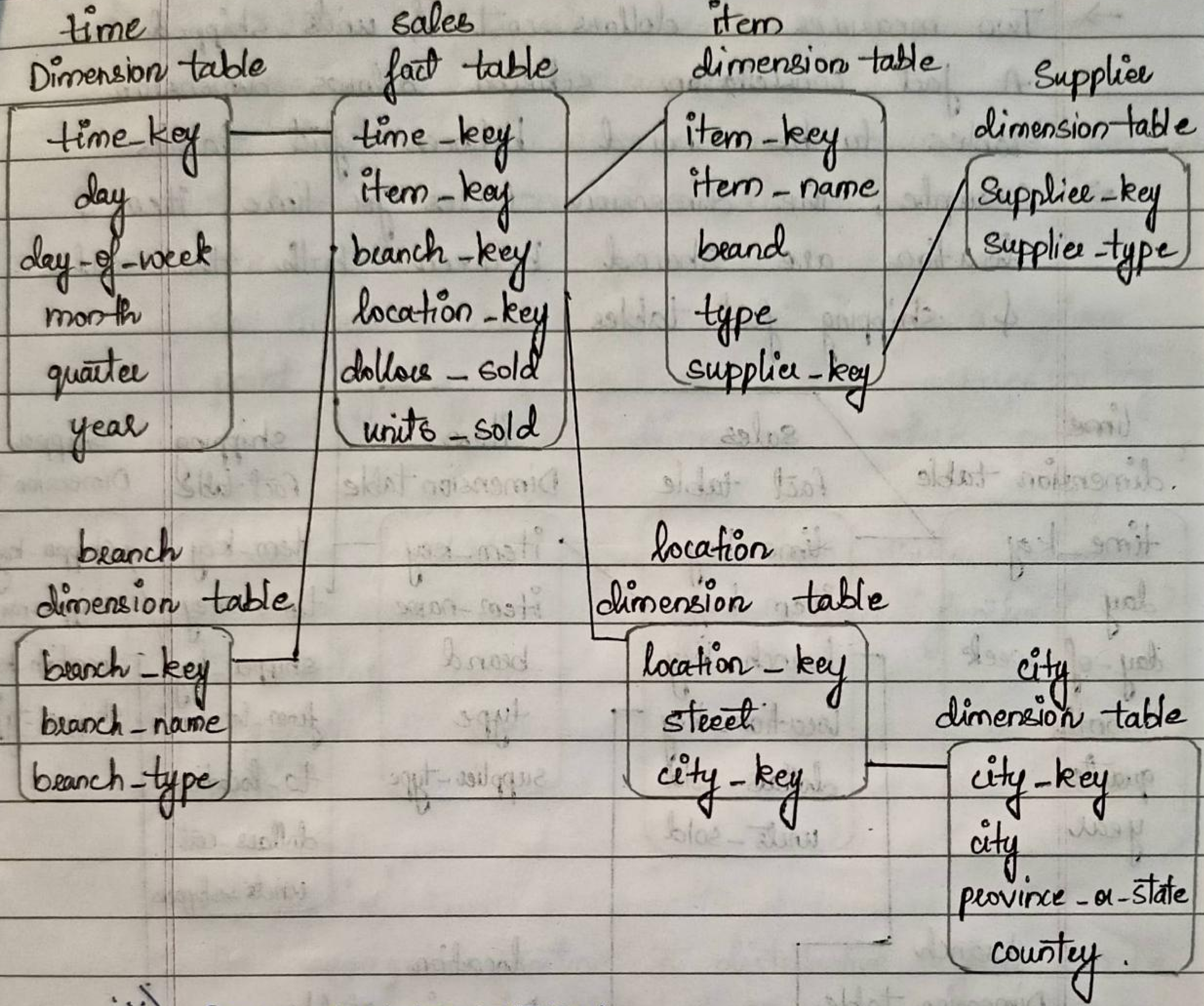
→ The main difference between the two schemas is in the definition of dimension tables.

→ The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item & supplier tables.

→ item dimension table now contains the attributes item_key, item_name, brand, type & supplier_key where supplier_key is linked to the supplier dimension table containing supplier_key & supplier_type information.

→ The city_key in new location table links to the city dimension.

fig: Snowflake Schema of a Sales data warehouse.



iii) <u>FACT CONSTELLATION</u>

- Sophisticated applications may require multiple fact table to share dimension tables.
- This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or fact constellation.

Eg : → A fact constellation is shown in fig.

→ This schema specifies two fact tables, sales & shipping.

→ The shipping table has five dimensions or keys : item_key, time_key, shipper_key, from_location & to_location.

→ Two measures : dollars_cost & units_shipped.
→ A fact constellation schema allows dimension tables to be shared between fact tables.
→ Example, the dimension tables for time, item & location are shared between both the sales & shipping fact tables.

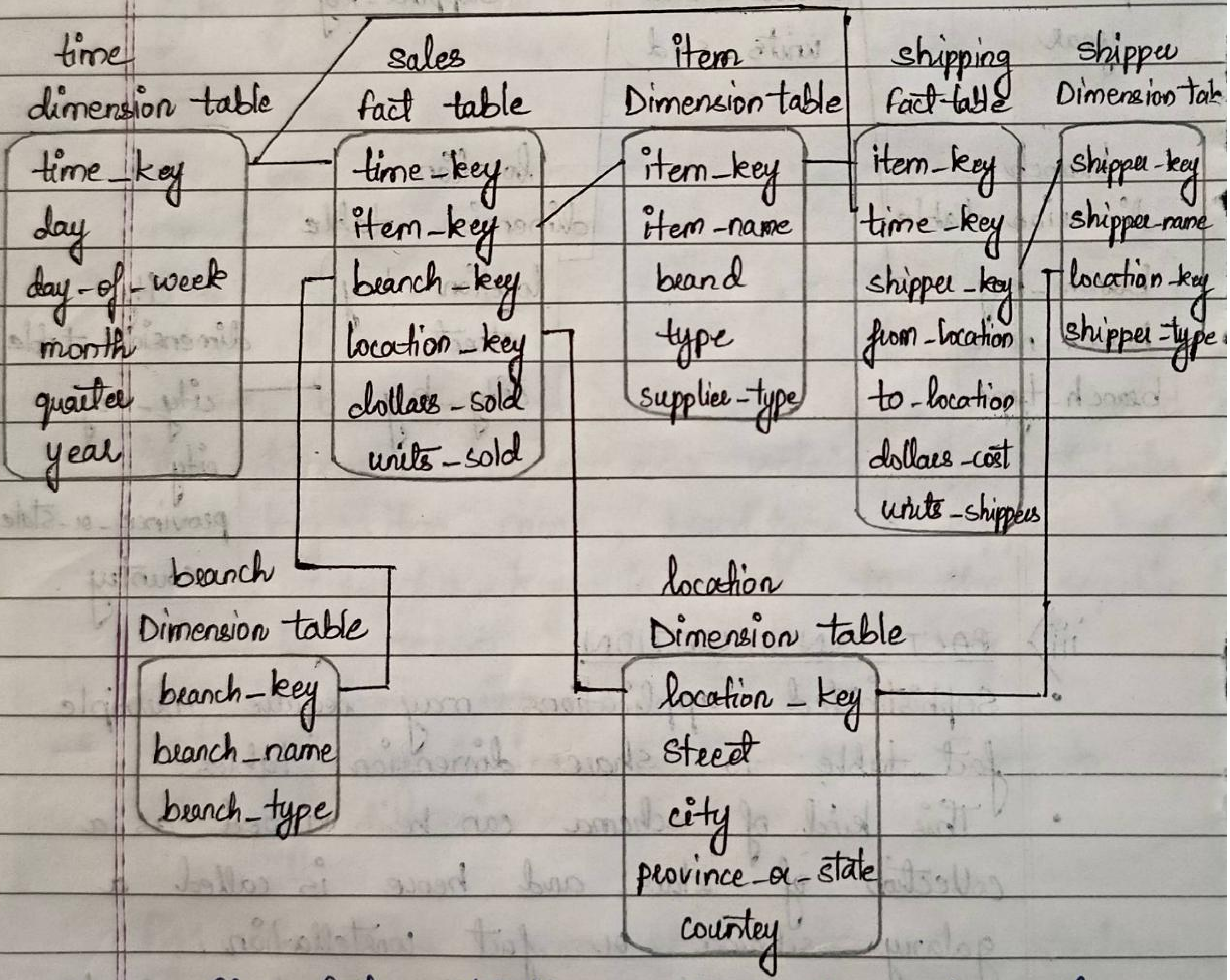


fig : Fact constellation schema of a sales and shipping data warehouse.

2) (a) What is Data cube measure? Explain the categorization of measures.

Ans:- → A Data cube measure is a numerical functions that can be evaluated at each point in the data cube space.

→ A measure value is computed for a given point by aggregating the data corresponding to the respective dimension value pairs defining the given point.

Measures can be organized into three categories based on the kind of aggregate functions used;

i) Distributive
ii) Algebraic
iii) Holistic.

i) Distributive

→ An aggregate function is distributive if it can be computed in a distributed manner as follows. Suppose the data are partitioned into n sets, we apply function to each partition, resulting in n aggregate values.

→ If the result derived by applying the function to the n aggregate values in the same as the derived by applying the function to the entire data set, the function can be computed in a distributed manner.

Eg: count() can be computed for a data cube by first partitioning the cube into a set of sub cubes, computing count() for each subcube & then summing up the counts obtained for each subcube.

→ Hence, count() is distributive aggregate function.

→ sum(), min() & max() are distributive aggregative functions.

→ A measure is distributive if it is obtained by applying a distributive aggregate function.

ii) Algebraic

→ An aggregate function is algebric, if it can be computed by an algebraic function with M arguments [where M is bounded positive integer], each of which is obtained by applying a distributive aggregate function.

Eg: avg() [average] can be computed by $\frac{sum()}{count()}$, where both sum() & count() are distributive aggregate functions.

→ min_N() and max_N() and standard-deviation() are algebraic aggregate functions.

→ A measure is algebric if it is obtained by applying an algebric aggregate functions.

iii) Holistic

→ An aggregate function is holistic if there is no constant bound on the storage size needed to describe a subaggregate i.e, there doesnot exist an algebric function with M arguments [where M is constant] that characterizes the computation.

Eg : median ( ), mode ( ) and rank ( ).

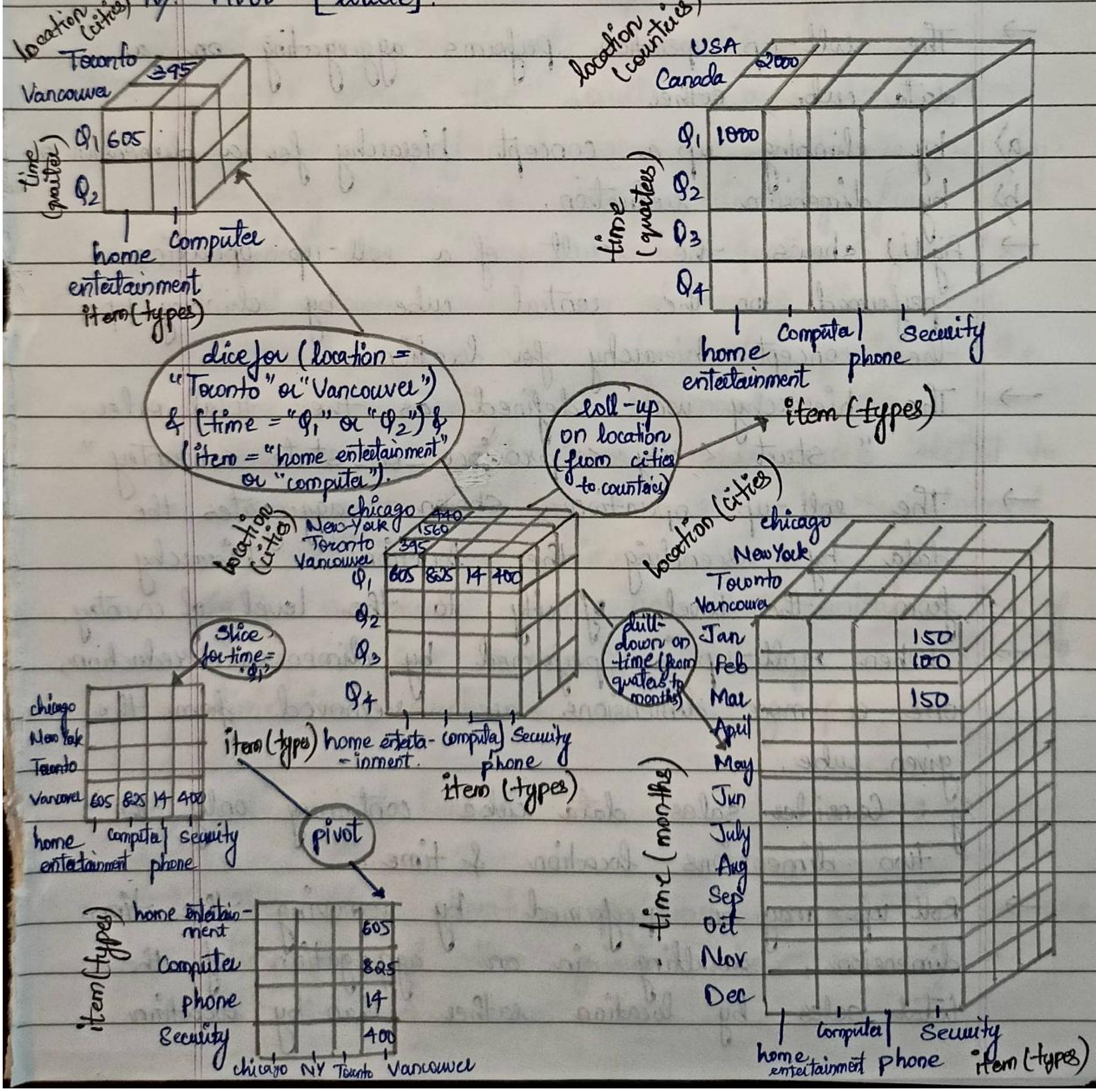
→ A measure is holistic if it is obtained by applying a holistic aggregate fun.

Q) (b) Explain data cube operations with examples.

Ans:- The typical OLAP operation in multidimensional data are as follows :-

i) Roll-up ii) Drill-down iii) Slice and dice

iv) Pivot [rotate].



→ At the center of the figure is a data cube for all electronics sales.

→ The cube contains the dimensions : location, time, item ; where location is aggregated w.r.t city values , time is aggregated w.r.t quarters & item is aggregated w.r.t item types.

i) <u>Roll-up</u> [also called drill-up operation]

→ The roll-up operation performs aggregating on a data cube, either

a) by climbing up a concept hierarchy for a dimension or

b) by dimension reduction.

→ Fig(i) shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for location.

→ The hierarchy was defined as the total order,
" street < city < province_or_state < country "

→ The roll-up operation shown aggregates the data by ascending the location hierarchy from the level of city to the level of country.

→ When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube.

Eg : Consider sales data cube containing only the two dimensions location & time..

→ Roll-up may be performed by removing the time dimension, resulting in an aggregation of the total sales by location rather than by location and by time.

ii) Drill-down

→ Drill-down is the reverse of roll-up.
→ It navigates from less detailed data to more detailed data.
→ Drill-down can be realized by either,
a). Stepping down a concept hierarchy for a dimension
b). introducing additional dimensions.
→ fig(i) shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy from time defined as "day < month < quarter < year".
→ Drill-down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month.
→ Because a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube.
Eg: a drill-down on the central cube of fig(i) can occur by introducing an additional dimension, such as customer group.

iii). Slice and Dice

→ The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube.

→ Fig(i) shows a slice operation where the sales are selected from the central cube for the dimension time using the criterion time = "Q_1".

→ The dice operation defines a subcube by performing a selection on two or more dimensions.

→ Fig(i) shows a dice operation on central cube based on the following selection criteria that involve the dimensions:
(location = "Toronto" or "Vancouver") & (time = "Q_1" or "Q_2") & (item = "home entertainment" or "computer")

iv) Pivot (rotate)

→ Pivot is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data.

→ Fig(i) shows a pivot operation where the item & location axes in a 2-D slice are rotated.

MODULE - 2

3) ⓐ Explain data cube computation and curse of dimensionality.

Ans:- • Multidimensional data Analysis is the efficient Computation of Aggregation across many sets of dimensions.

• In SQL terms, these aggregation are referred as "group-by". where each group-by can be represented by a "Cuboid". Set of group by's forms a lattice of Cuboids defining a data cube.

→ Curse of Dimensionality.

- A major challenge related to precomputation is that the required storage space may explode if all cuboids in data cube are precomputed. Especially when the cube has many dimensions.

The Storage requirements are even more excessive when many of dimensions have associated concept hiarchies. This problem is refered as "Curse of Dimensionality".

Eg :- Time is usually explored not at only one conceptual level (eg : year).
But rather at multiple conceptual levels such as in the hiearchy.
"day < month < quarter < Year".

- For an n-dimensional Data cube, the total no. of Cuboids that can be generated is,

$$\boxed{\text{Total no. of Cuboids} = \prod_{i=1}^{n} (L_i + 1).}$$

where L_i is the number of levels associated with the dimension i. One is added to L_i in the equation above to include the virtual top level.

3) (b). Explain different methods of indexing OLAP data.

Ans:- To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views. There are two methods of indexing OLAP, they are;

i) Bitmap Indexing ii) Join indexing.

i) Bitmap Indexing :- is a popular indexing method, because it allows quick searching in data cubes.

→ In the bitmap index for a given attribute, there is a distinct bit vector (BV), for each value 'v' in the attribute domain.

→ If a given attribute's domain consists of n values, then 'n' bits are needed for each entry in the bitmap index.

→ If the attribute has the value v for a given row in the data table, then the bit representing that value is set to '1' in corresponding row of bitmap index.
All other bits for that row are set to '0'.
Eg:- "AllElectronics" data warehouse, Suppose the dimension 'item' at the top levels has four values such as "Home Entertainment", "Computer", "phone" & "Security".

→ Bitmap indexing is advantageous compared to 'hash' or 'Tree indices'. Because comparison, Join & aggregation operations are reduced to bit Arithmatic which substantially reduce the processing time.

→ Bitmap Indexing provide a significant reduction in space & input/output (I/O) since a string of character can be represented by a single bit.

Base table

RID	item	city
R_1	H	V
R_2	C	V
R_3	P	V
R_4	S	V
R_5	H	T
R_6	C	T
R_7	P	T
R_8	S	T

item bitmap table

RID	H	C	P	S
R_1	1	0	0	0
R_2	0	1	0	0
R_3	0	0	1	0
R_4	0	0	0	1
R_5	1	0	0	0
R_6	0	1	0	0
R_7	0	0	1	0
R_8	0	0	0	1

city bitmap table

RID	V	T
R_1	1	0
R_2	1	0
R_3	1	0
R_4	1	0
R_5	0	1
R_6	0	1
R_7	0	1
R_8	0	1

ii) Join Indexing :- Method gained popularity from its use in relational database query processing.

→ Traditional Indexing Maps the values in a given column to a list of rows having the value.

→ Join indexing register the joinable rows of two relations from a relational database.

Eg: if two relation R(RID, A) and S(B, SID) Join on the attribute 'A' & 'B', then the Join index record contains the pair (RID, SID), where RID & SID are record identifier from the R and S relations.

→ Join index records can identify joinable tuple without perfoming costly join operations.
→ Join indexing maintain relationship between attribute values of a dimension (within dimension table) & the corresponding row in the fact table.
→ Consider Eg: "All Electronics" Star Schema of a Join index relationship between the "Sales" fact table & the 'location' & 'item' Dimension table.

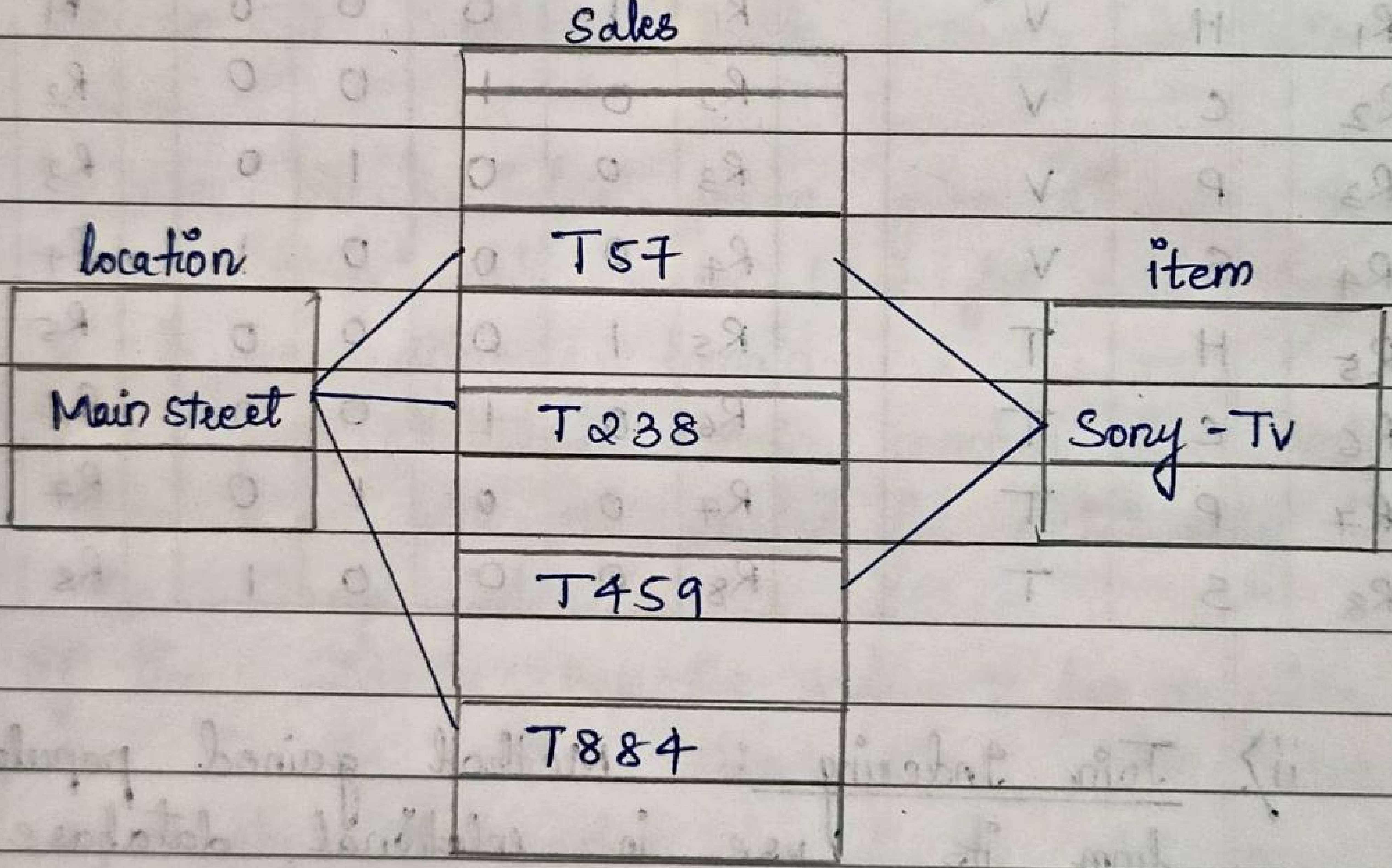


fig: Linkage between sales fact table & location & item dimension table.

location	Sales - key
.....	......
Main Street	T57
Main Street	T238
Main Street	T884
.....	......

item	Sales - key
.....	.....
Sony-Tv	T57
Sony-Tv	T459
......	.....

location	item	Sales-key
.....	.....	....
Main Street	Sony-Tv	T57
.....	...	.....

→ To speed-up query processing, the Join indexing & bitmap indexing methods can be integrated to form 'bitmapped Join indices'.

4) (a) State and explain various data mining tasks.

Ans :- Data mining tasks are generally divided into two major categories :-

* Predictive tasks
* Descriptive tasks.

There are number of Data mining tasks such as classification prediction, Associative analysis, Cluster analysis, Anamoly detection. etc.

All these tasks are either predictive tasks or Descriptive task.

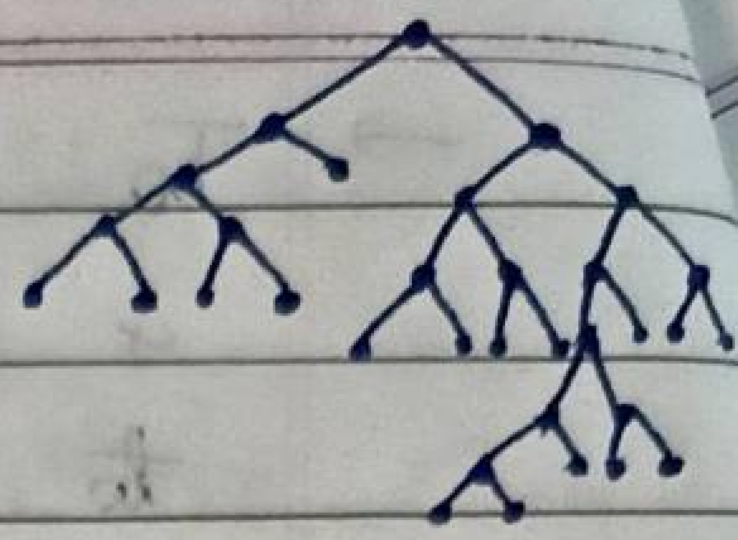
Fig. below illustrates four of the core datamining tasks:

i) Predictive Modelling.
ii) Associative analysis.
iii) Cluster analysis.
iv) Anamoly detection.

i) Predictive Modelling :- It refers to the task of building a model for the target variables as a function of the explanatory variables.

There are two types of predictive modeling tasks.

- Classification :- used for discrete target variables.
- Regression :- used for continuous target variables.



Data

Cluster Analysis

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	80K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Predictive Modelling

Association Analysis.

Anamoly Detection

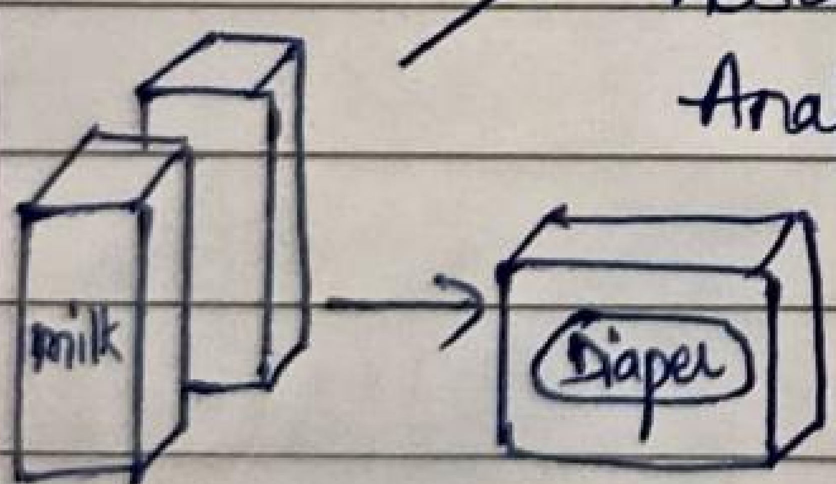


fig: four of the core data mining tasks.

ii) <u>Association Analysis</u> :- It is used to discover patterns that describe strongly associated features in the data.

→ The discovered patterns are typically represented in the form of implication rules or feature subsets.

→ Because of the exponential size of its search space, the goal of association analysis is to extend the most interesting patterns in an efficient manner.

iii). Cluster Analysis :- It seeks to find groups of closely related observations, so that belongs to the same cluster are more similar to eachother than observations that belong to other clusters.

→ Clustering has been used to group sets of related customers.

iv). Anamoly detection :- It is the task of identifying observations whose characteristics are significantly different from the rest of the data. Such observations are known as anamolis or outlines.

→ The goal of an anamoly detective algorithm is to discover the real anamolies & avoid falsely labeling normal objects as anomolies.

→ A good anamoly detector must have high detection rate & low false rate.

4) (b) Define Similarity and dissimilarity between the objects. Find SMC and Jaccard's coefficient of two binary vectors.

$X = (1,0,0,0,0,0,0,0,0,0)$

$Y = (0,0,0,0,0,0,1,0,0,1)$.

Ans:- Similarity

- Similarity between two objects is a numerical measure of the degree to which the two objects are alike.
- Similarity is higher for pair of objects that are more alike.
- Similarities are usually non-negative & are often between 0 and 1.

Dissimilarity

- Dissimilarity between two objects is a numerical measure of the degree to which the two objects are alike.
- Dissimilarity are lower for more similar pair of objects.
- Dissimilarity fall in the interval [0, 1] or in the range [0, ∞].

→ Simple Matching Coefficient [SMC].
It counts both presence and absence equally.

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}}$$

$$SMC = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

→ Jaccard Coefficient

Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes.

$$J = \frac{\text{number of matching presences}}{\text{no. of attributes not involved in 00 matches}}$$

$$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}}$$

Given : $x = (1,0,0,0,0,0,0,0,0,0)$
$y = (0,0,0,0,0,0,1,0,0,1)$

$f_{00} = 7$
$f_{01} = 2$
$f_{10} = 1$
$f_{11} = 0$

$$SMC = \frac{f_{11}+f_{00}}{f_{11}+f_{00}+f_{01}+f_{10}}$$

$$SMC = \frac{0+7}{0+7+2+1} = \frac{7}{10}$$

$$\boxed{SMC = 0.7}$$

$$J = \frac{f_{11}}{f_{11}+f_{01}+f_{10}} = \frac{0}{0+2+1} = 0$$

MODULE - 3

5) (a) What is Association Analysis ? Explain Association rule, Support and Confidence.

Ans :- Association Analysis

- Association discovers the connection among a set of items in large data sets. Association identifies the relationship among objects. Association is used for commodity Managment, Advertising, Direct Marketing etc.
  Eg :- Market basket Transactions.

TID	Item.
1	{Bread, Milk}
2	{Bread, Diaper, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

- Each row in this table Corresponds to a transaction, which contains a unique identifier labeled "TID" & a set of items bought by a given Customer.
- The Association Analysis is useful for discovering interesting relationships hidden in large data sets.

⇒ Association rule :- An Association rule is an implication expression form $X \rightarrow Y$, where x & y are disjoint itemsets.

- The strength of an Association rule can be

measured in terms of its
- Support
- Confidence.

Eg :- {Milk, Diapers} → {Beer}.

⇒ Support :- fraction of transaction that contains both X & Y.

$$S = \frac{\sigma\{Milk, Diaper, Beer\}}{|T|}$$

$$= \frac{2}{5} = 0.4$$

⇒ Confidence :- Measure how often items in 'y' appear in transaction that contains X.

$$C = \frac{\sigma\{Milk, Diaper, Beer\}}{\sigma\{Milk, Diaper\}}$$

$$C = \frac{2}{3} = 0.67$$

5) (b) State Apriori principle. Write apriori algorithm for frequent itemset.

Ans:-
- Apriori Algorithm is the first Association rule Mining Algorithm that pioneered the use of Support-Based pruning to systematically control the exponential growth of candidate itemsets.
- Apriori uses a "Bottom-up" Approach, where frequent subsets are extended one item at a time.

Apriori Algorithm

1). $k = 1$
2). Find all frequent 1-itemsets $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq MinSup \}$
3). repeat.
4). $k = k+1$
5). F_k = Apriori-gen (F_{k-1}) //Generate Candidate itemset.
6). for each transaction $t \in T$ do
7). Generate Subset (C_k, t)
8). End for
9). $F_k = \{ c \mid c \in C_k \geq MinSup \}$ //Extract k-frequent itemset.
10). until $F_k = 0$.
11). return F_k.

Eg :-

TID	Items
1	{Bread, Milk}
2	{Bread, Diaper, Beer, Eggs}
3	{Milk, Diaper, Beer, Cola}
4	{Bread, Milk, Diaper, Beer}
5	{Bread, Milk, Diaper, Cola}

MinSup Count = 3.

→ Firstly, Generate 1-itemset by counting the components.

Bread	4
Diaper	4
Milk	4
Beer	3
Egg	1
Cola	2

(1-itemset)

Eleminate the candidate that are 'infrequent by comparing with 'MinSup Count = 3' such as 'Egg & Cola'

→ Next, Generate 2-itemset by forming subset of two components from 1-itemset.

{Bread, Milk}	3
{Bread, Diaper}	3
{Bread, Beer}	2
{Milk, Diaper}	3
{Milk, Beer}	2
{Diaper, Beer}	3

(2-itemset).

Get the count of Subset's by scanning Base-DB, Eliminate the Candidates that have less count compare to "MinSup Count".

→ Generate 3-itemset by following subset of three components from 2-itemset.

{Bread, Milk, Diaper}	2
{Bread, Diaper, Beer}	2
{Milk, Diaper, Beer}	2
{Bread, Milk, Beer}	1

(3-itemset).

All 3-itemset are infrequent, Because count is less than Min-Sup Count = 3.

6) (a) Construct an FP tree for the following dataset.

TID	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

Ans:- Step 1 :- Count all item in whole Transactions to get support count.

items	Support
a	8
b	7
c	6
d	5
e	3

Step 2 :- Apply the Threshold (MinSup) & remove item which posses Support count less than MinSup.

In the above table, all items Support count is greater than "MinSup = 2", So all items are considered in FP-tree.

Step 3 :- Arrange items in Descending order of there Support Count.

items	Support
a	8
b	7
c	6
d	5
e	3

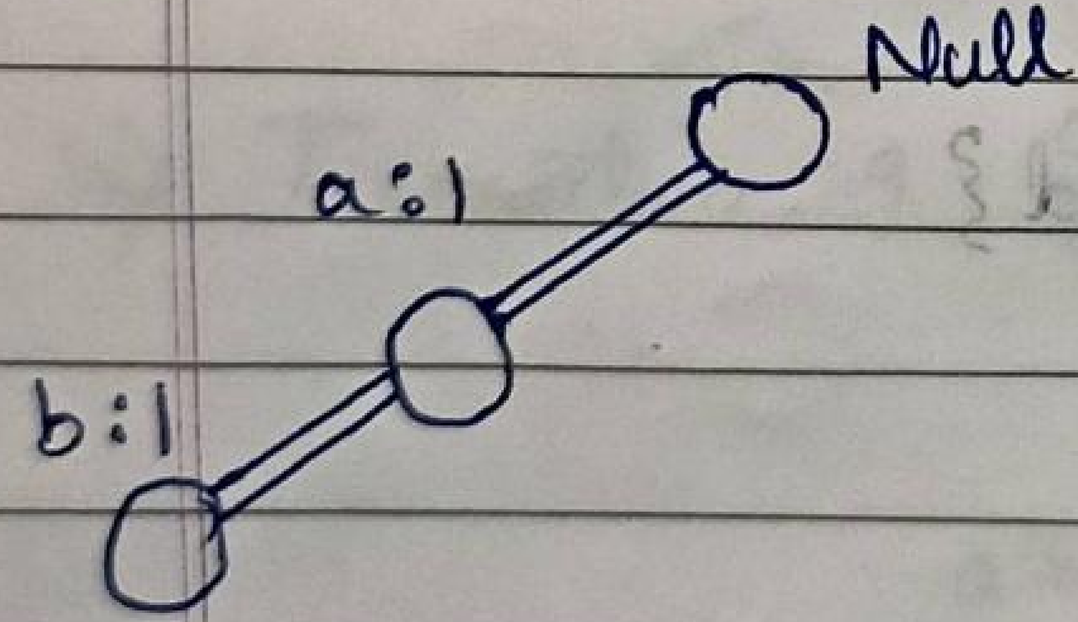
∴ "a, b, c, d, e" will be the FP-tree construction pattern.

Step 4 :- FP-Tree Construction, we go through each of transaction with following two conditions.

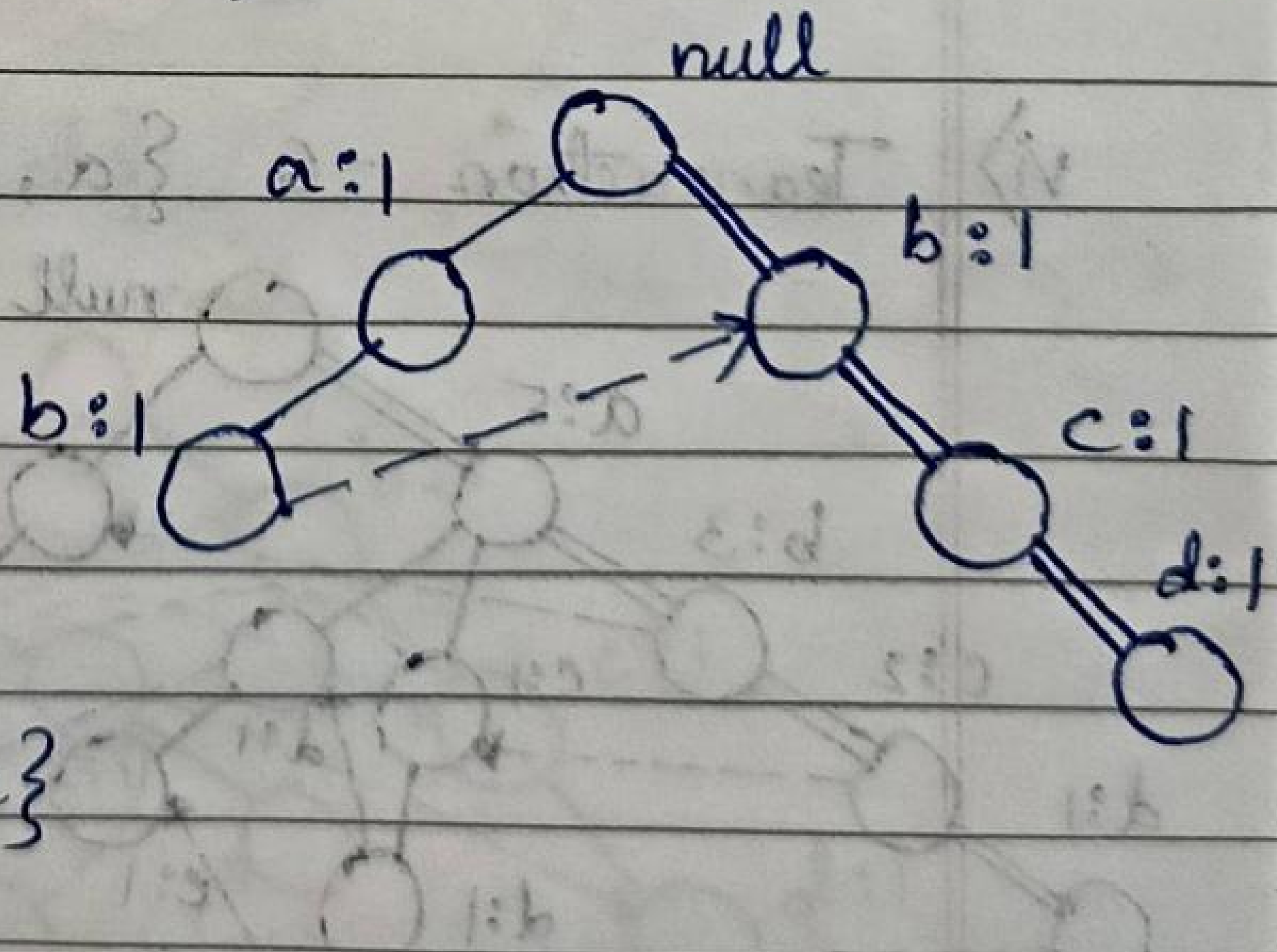
→ items whose support count is less than Min-Sup should be removed from Transaction.

→ Descending-order item list for FP-Tree construction should be maintained.

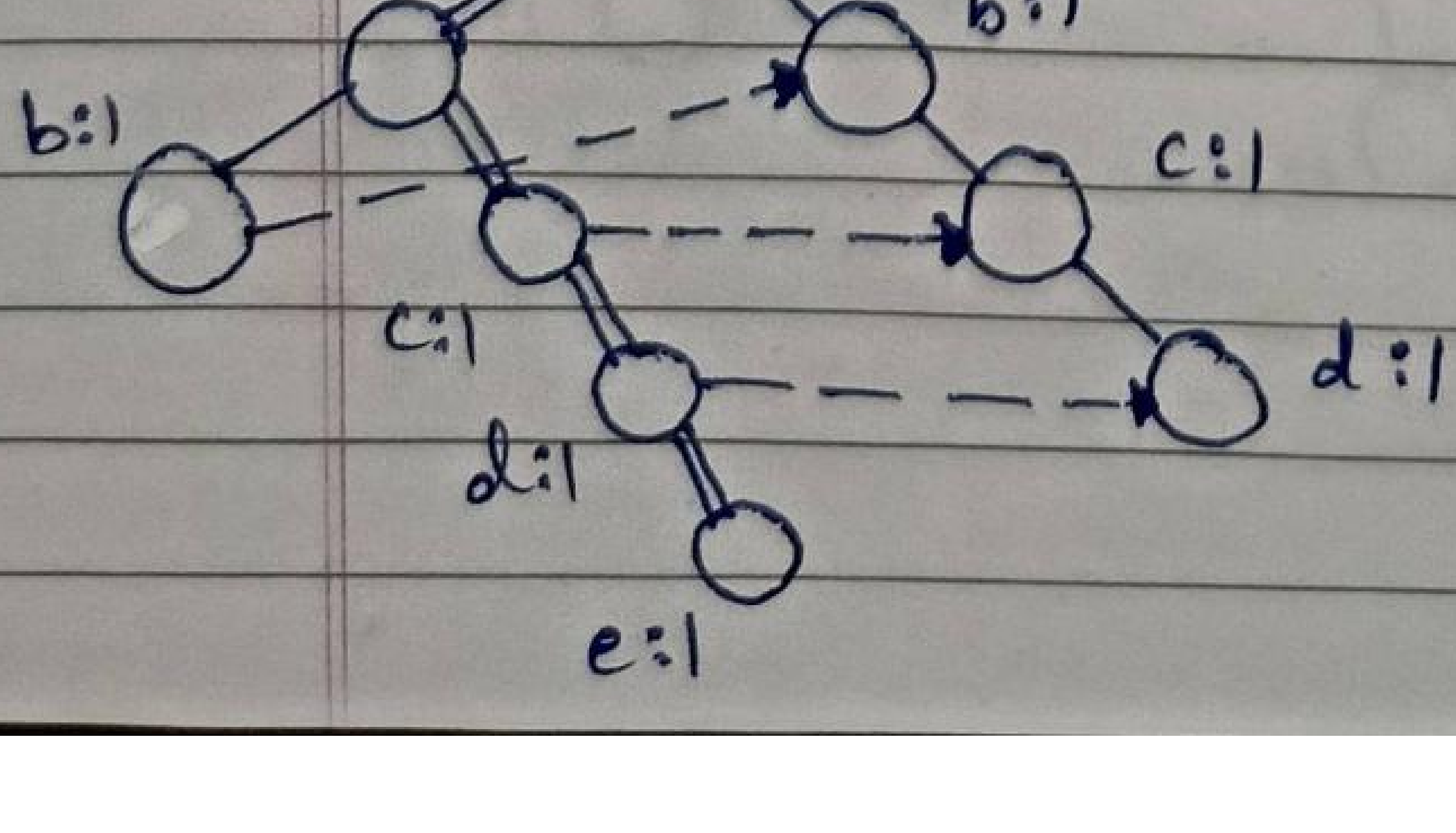
i) Transaction -1 {a, b}



ii) Transaction -2 {b, c, d}

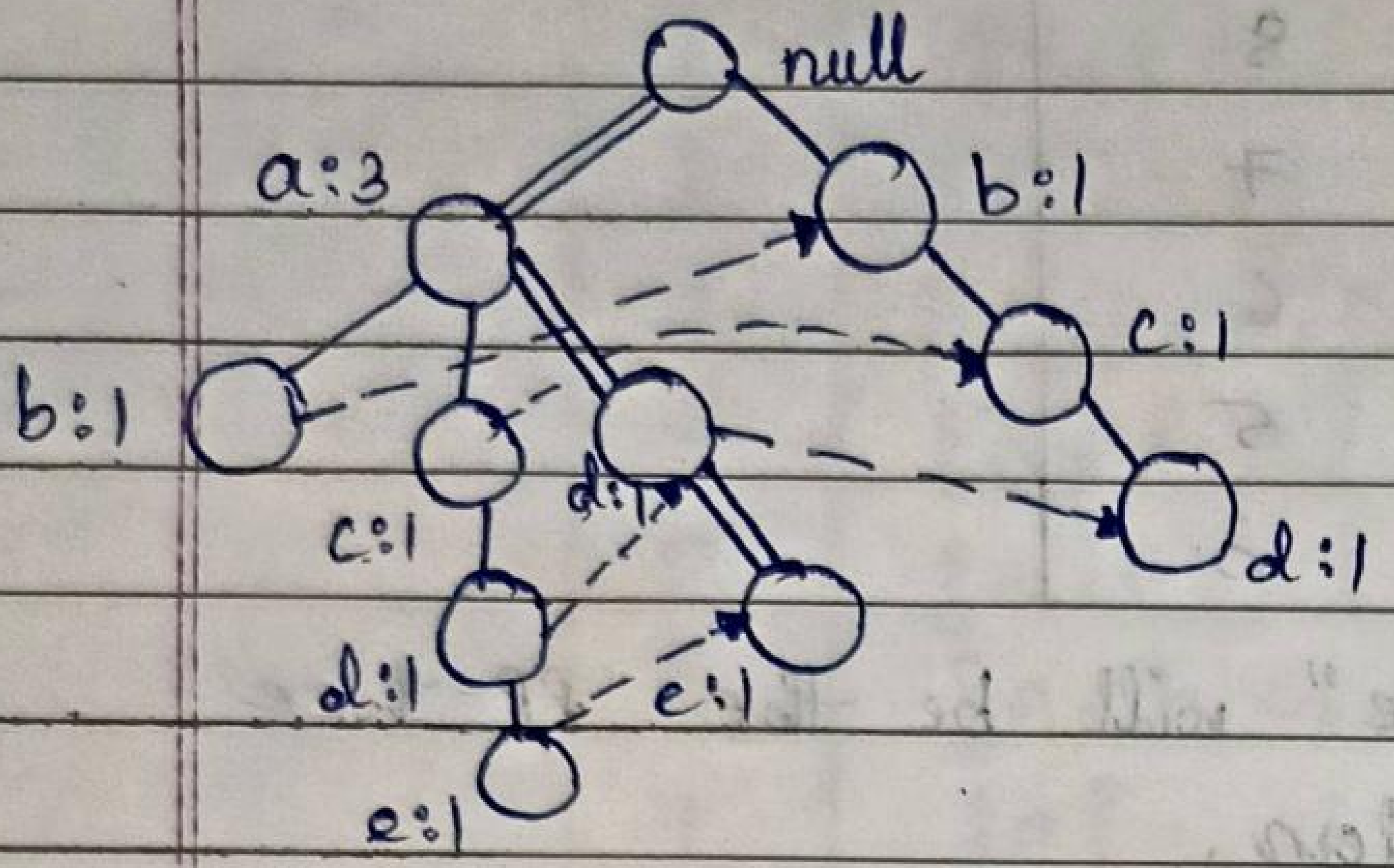


iii) Transaction -3 {a, c, d, e}

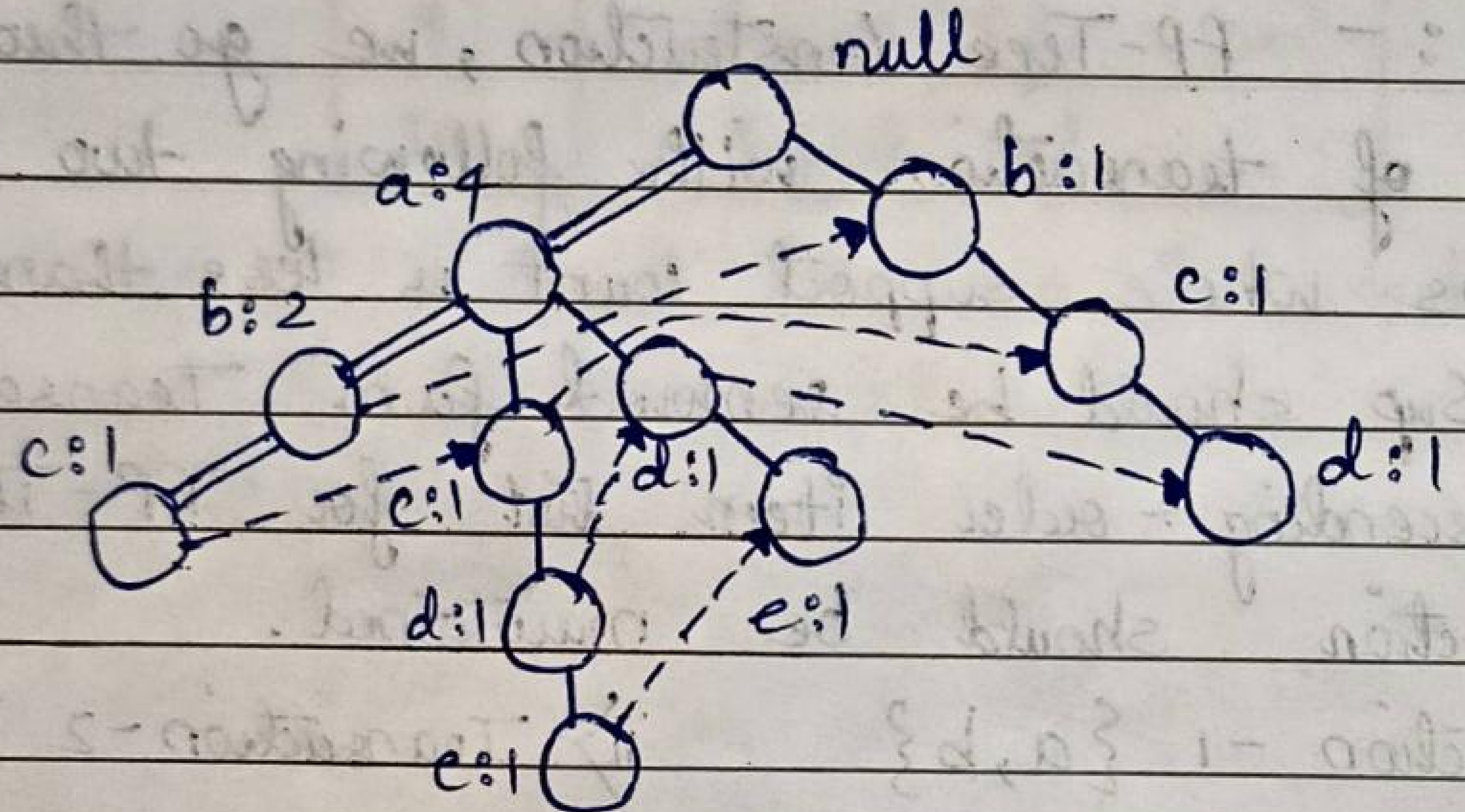


Scanned by TapScanner

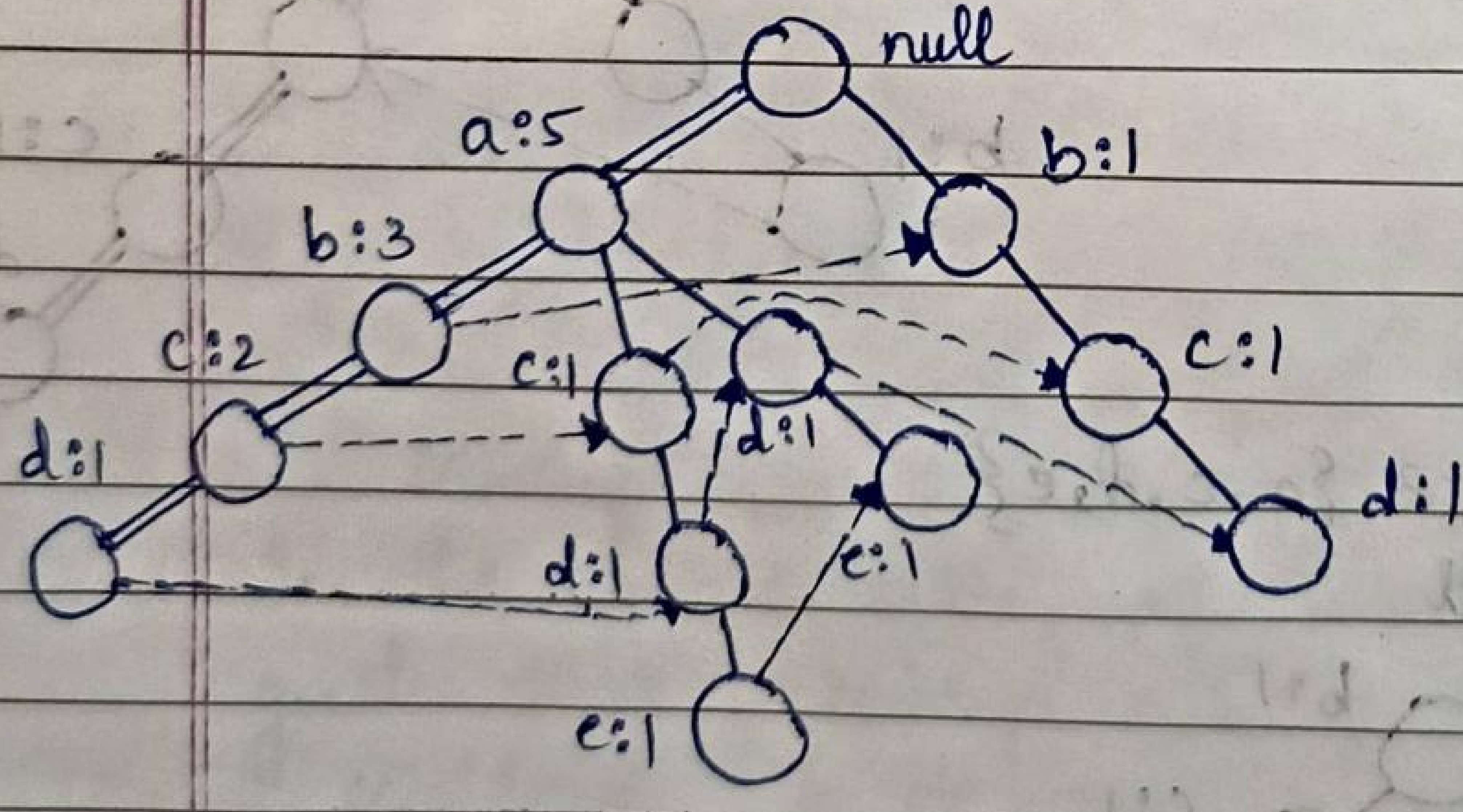
iv). Transaction -4 {a, d, e}



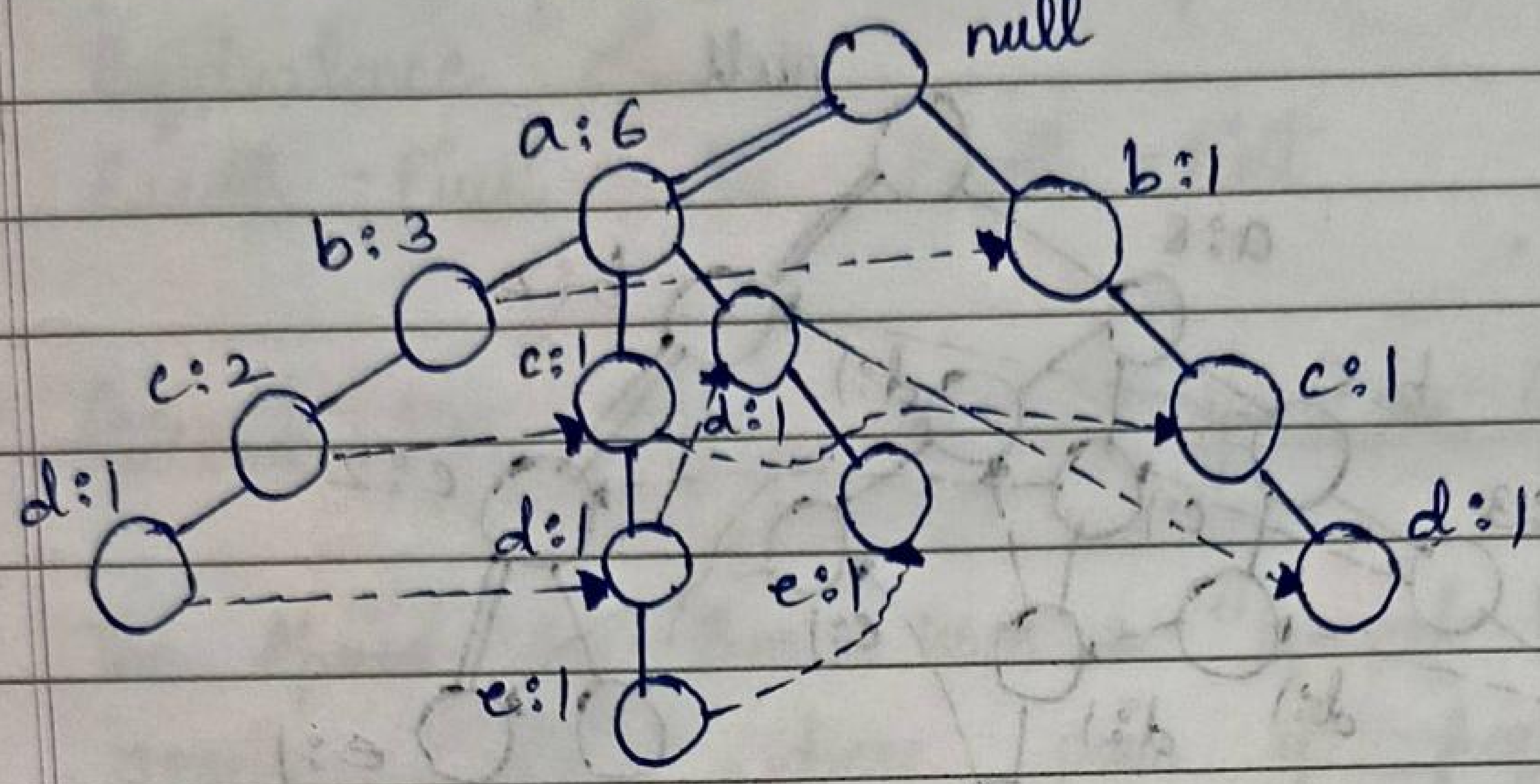
v). Transaction -5 {a, b, c}



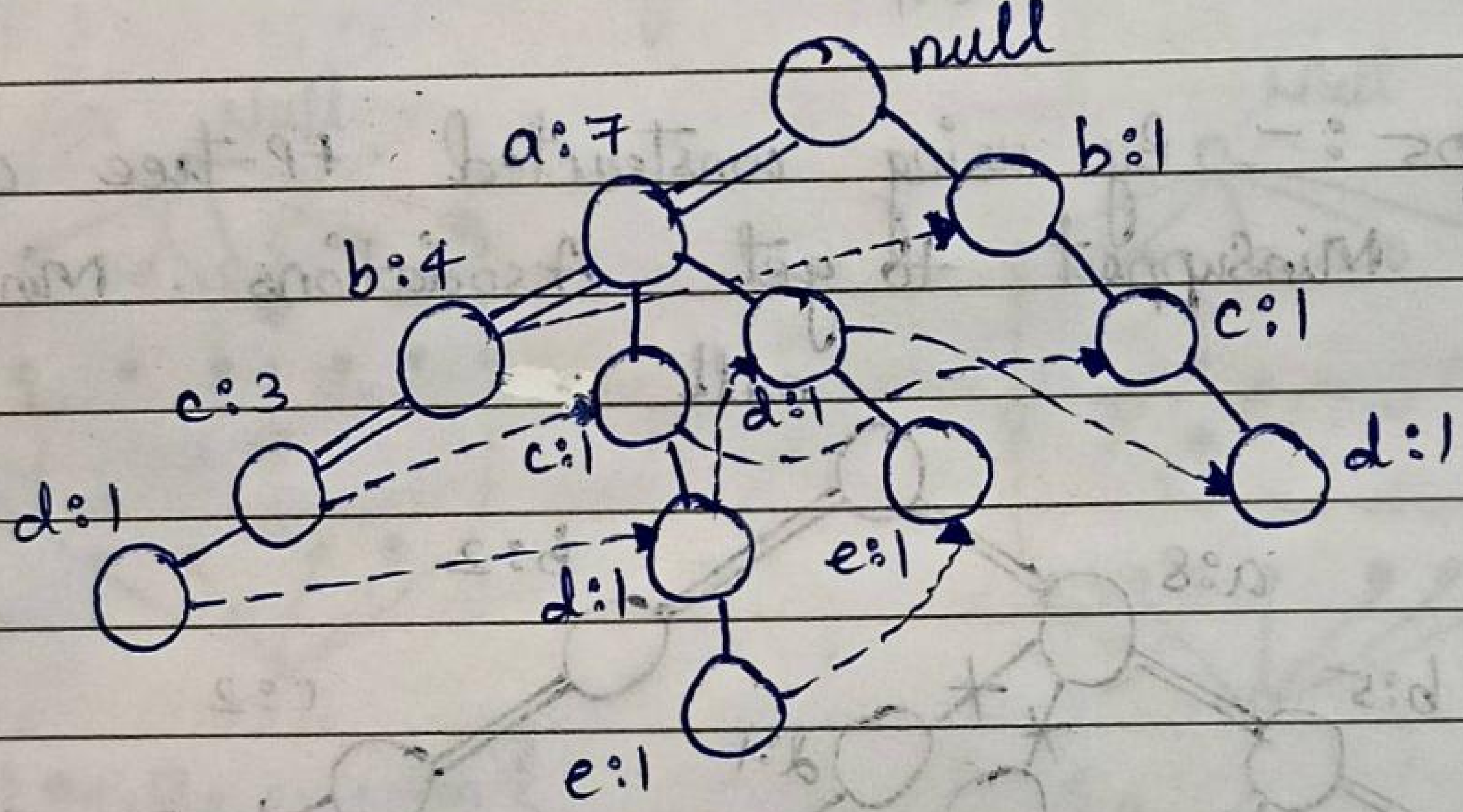
vi) Transaction -6 {a, b, c, d}



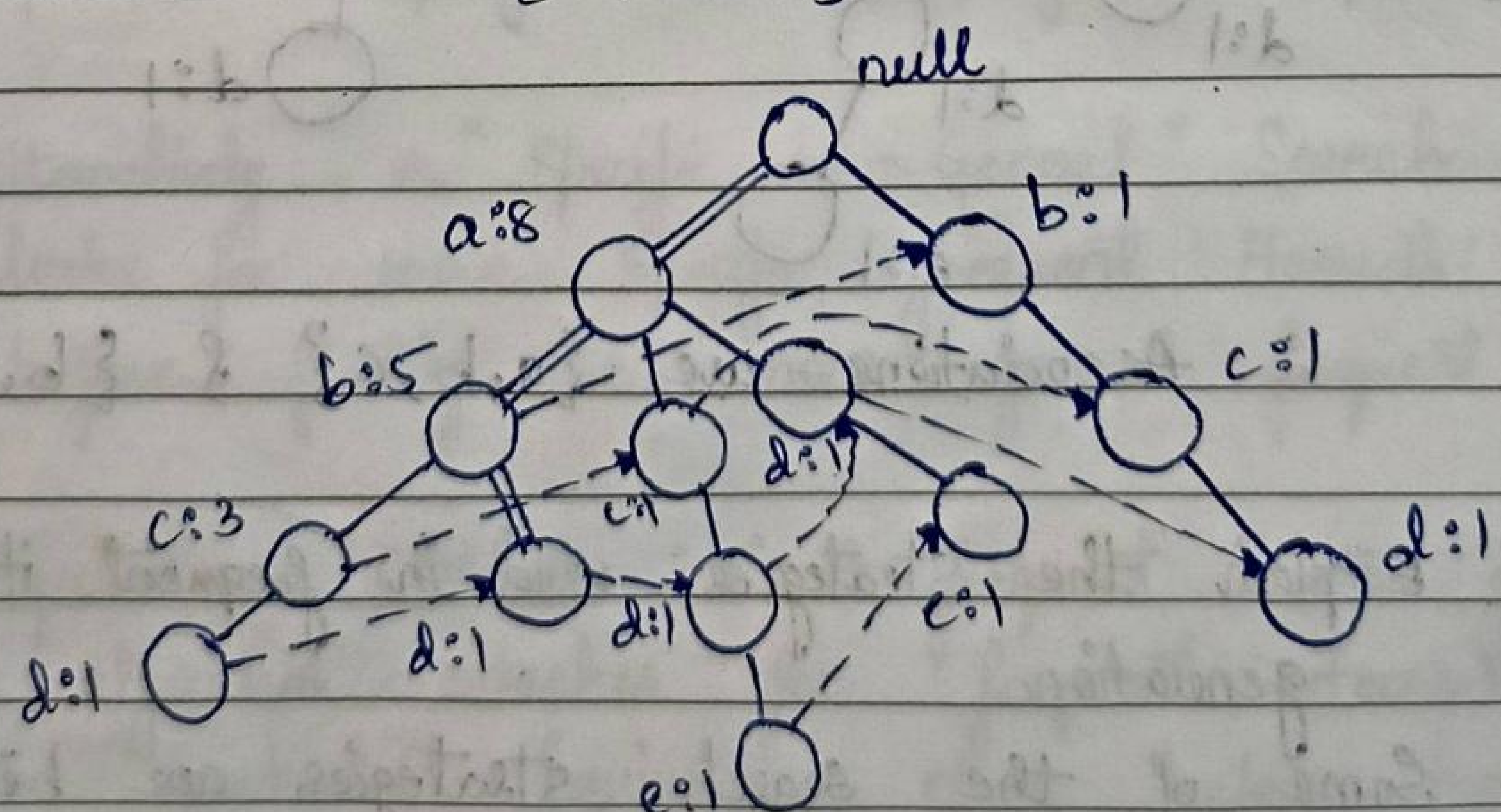
vii) Transaction - 7 {a}



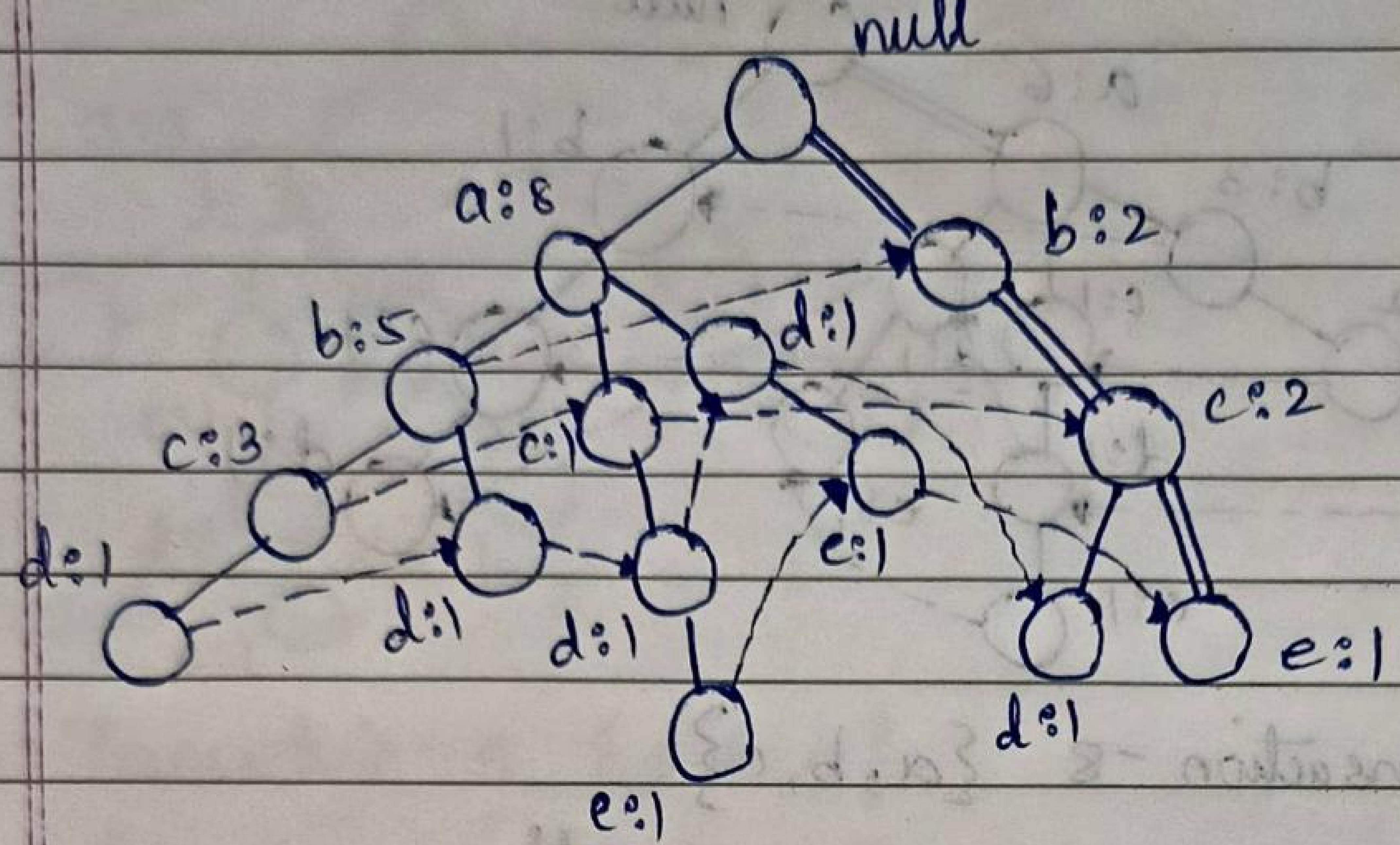
viii) Transaction - 8 {a, b, c}



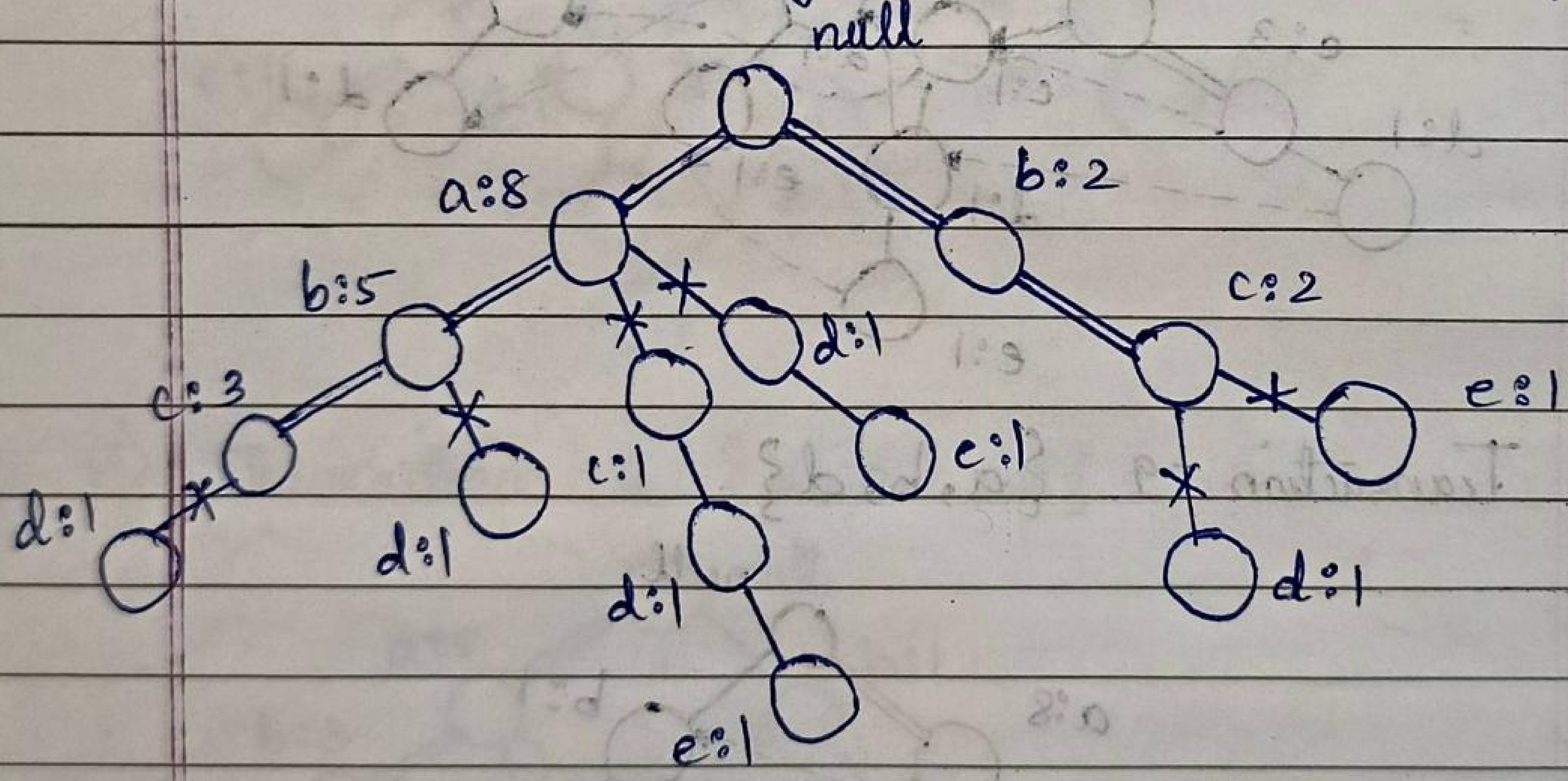
ix) Transaction - 9 {a, b, d}.



x) Transaction - 10 {b, c, e}.



Steps :- By using constructed FP-tree compare MinSupport to get Associations. MinSup = 2.



∴ Associations are {a, b, c} & {b, c}.

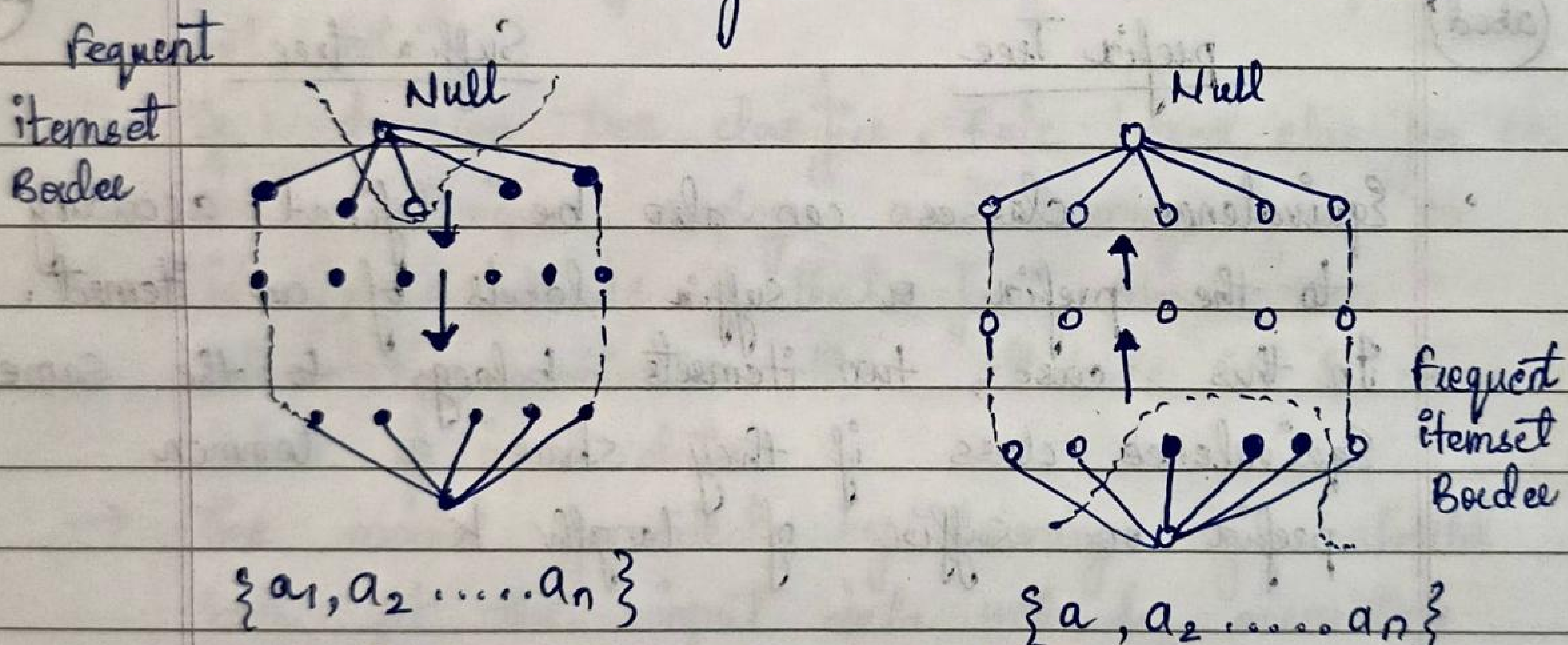
6) (b). Explain the strategies used in frequent itemset generation.

Ans :- Some of the search strategies are better than Apriori depending on the configuration of frequent itemsets.

- General-to-specific Vs Specific-to-General
- Equivalence classes
- Breadth-First Vs Depth-first.

→ <u>General-to-specific Vs Specific-to-General.</u>

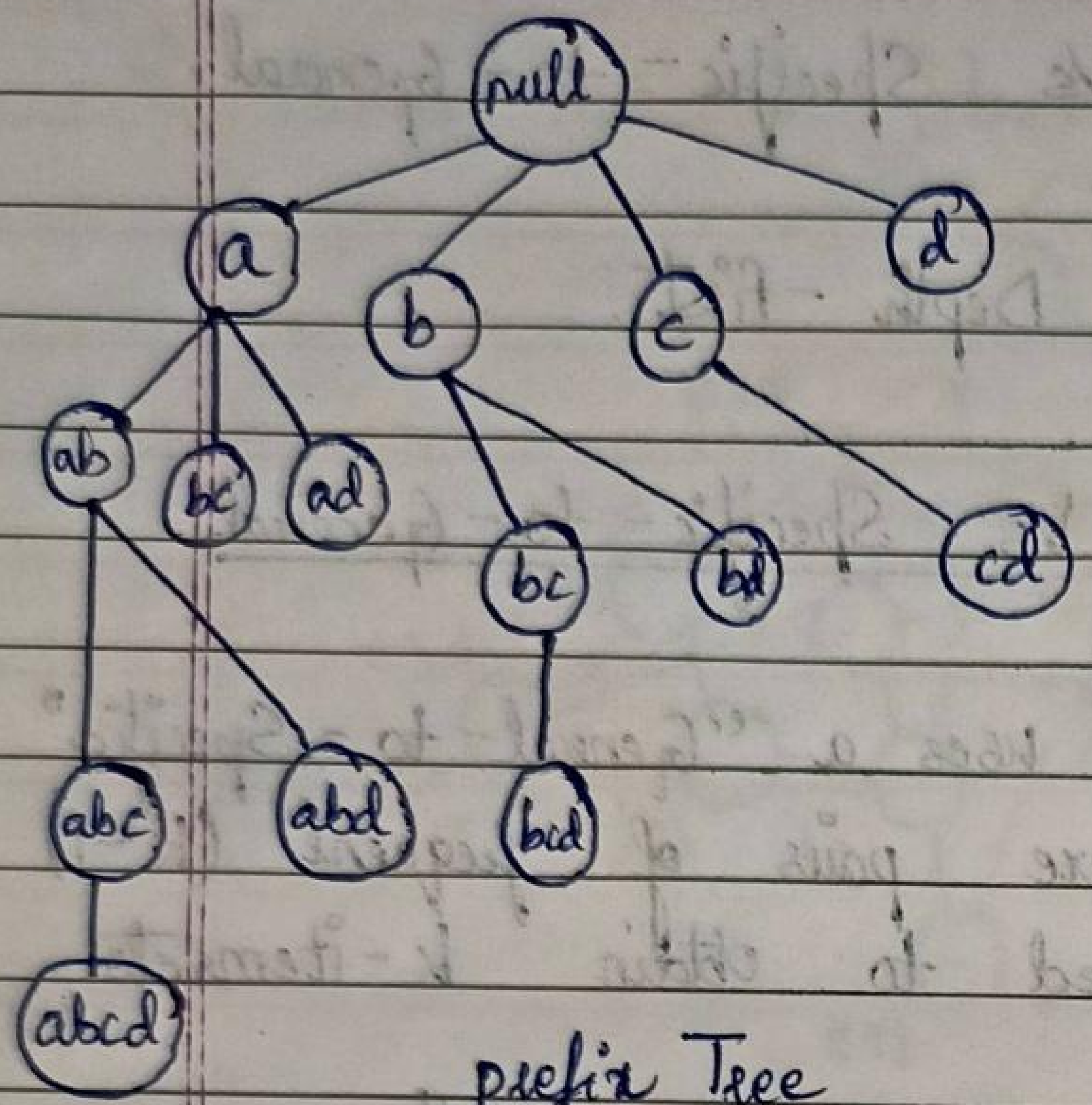
- The Apriori Algorithm uses a "General-to-Specific" search strategy, where pairs of frequent (k-1) itemsets are merged to obtain k-itemsets.



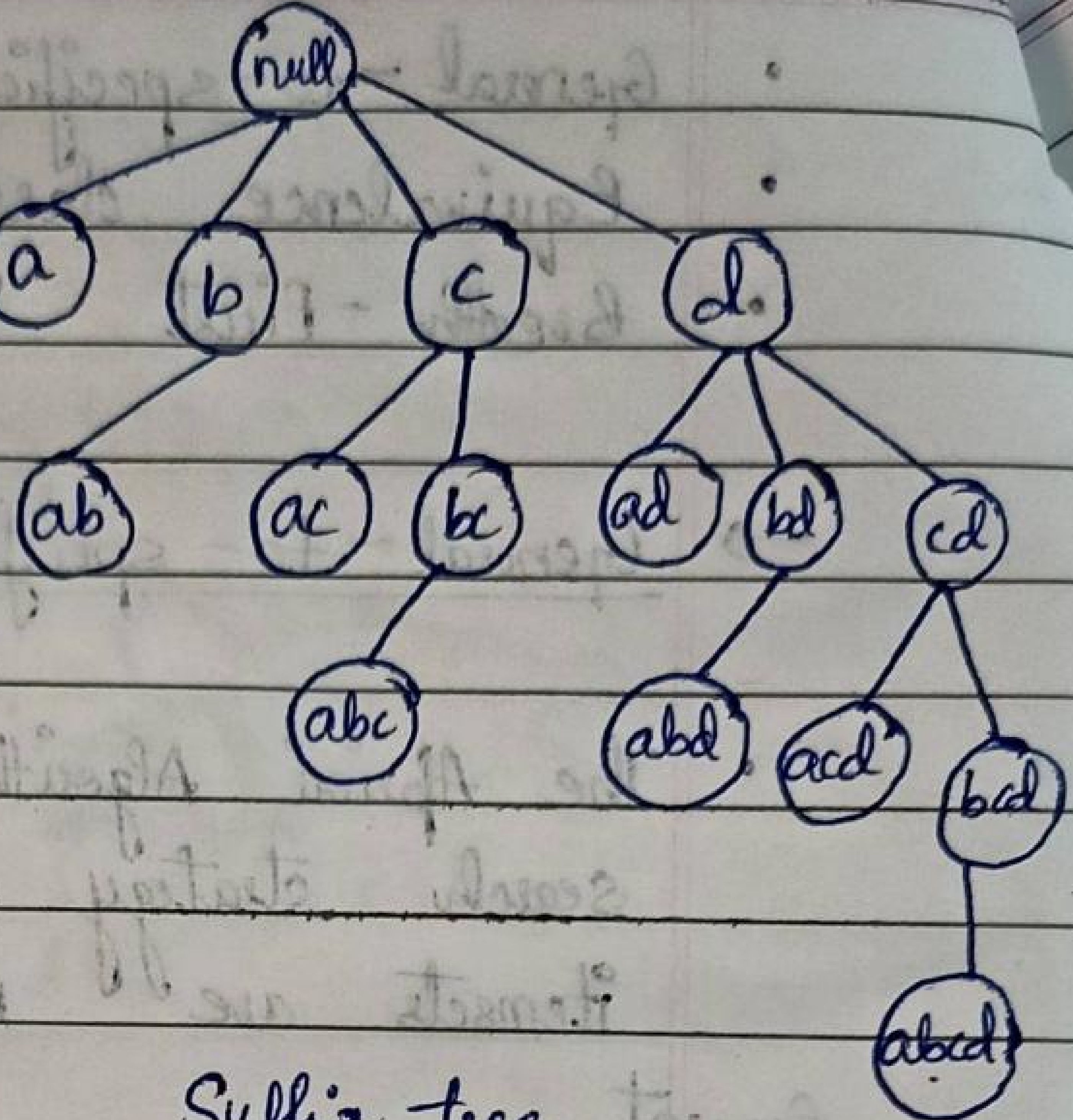
General-to-Specific. Specific-to-General.

- Alternatively, a "Specific-to-General" Search strategy looks for more Specific frequent itemsets first, before finding the more general frequent itemsets.

→ <u>Equivalence Classes</u> :- A frequent itemset generation algorithm searches for frequent itemsets within a particular Equivalence class before moving to another Equivalence class.



prefix Tree

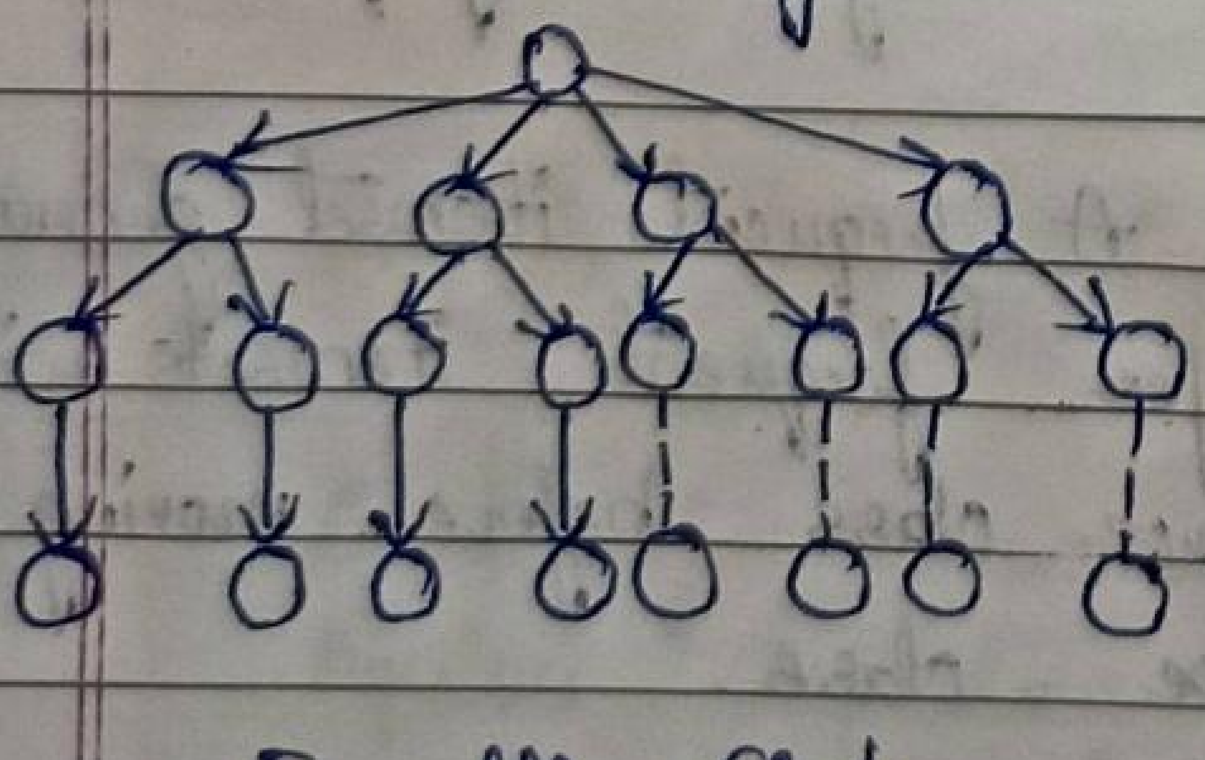


Suffix tree

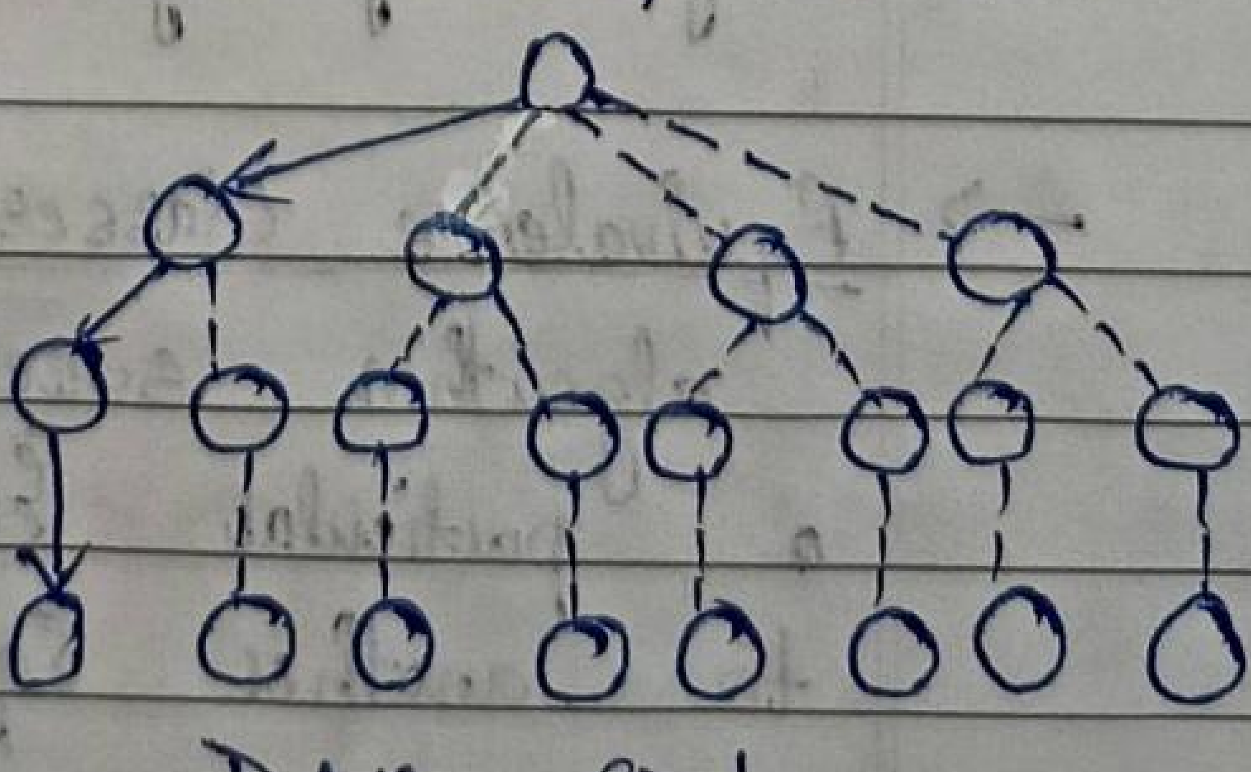
- Equivalence classes can also be defined according to the prefix or suffix labels of an itemset. In this case, two itemsets belongs to the same Equivalence class if they share a common prefix or suffix of length k.

→ Breadth - First V/s Depth - First

The Apriori Algorithm traverses in a "Breadth-First" Manner, it first Discovers all the frequent 1-itemsets followed by 2-itemsets & so on.



Breadth - First



Depth - first

- The "Depth-First" Approach is often used by algorithm designed to find Maximal Frequent itemsets.

MODULE-4

7) (a). Explain the general approach for solving classification problem.

Ans:- A classification technique is a systematic approach to building clarification model from an input set data.

Eg: decision tree classifier, Rule-based classifier etc.

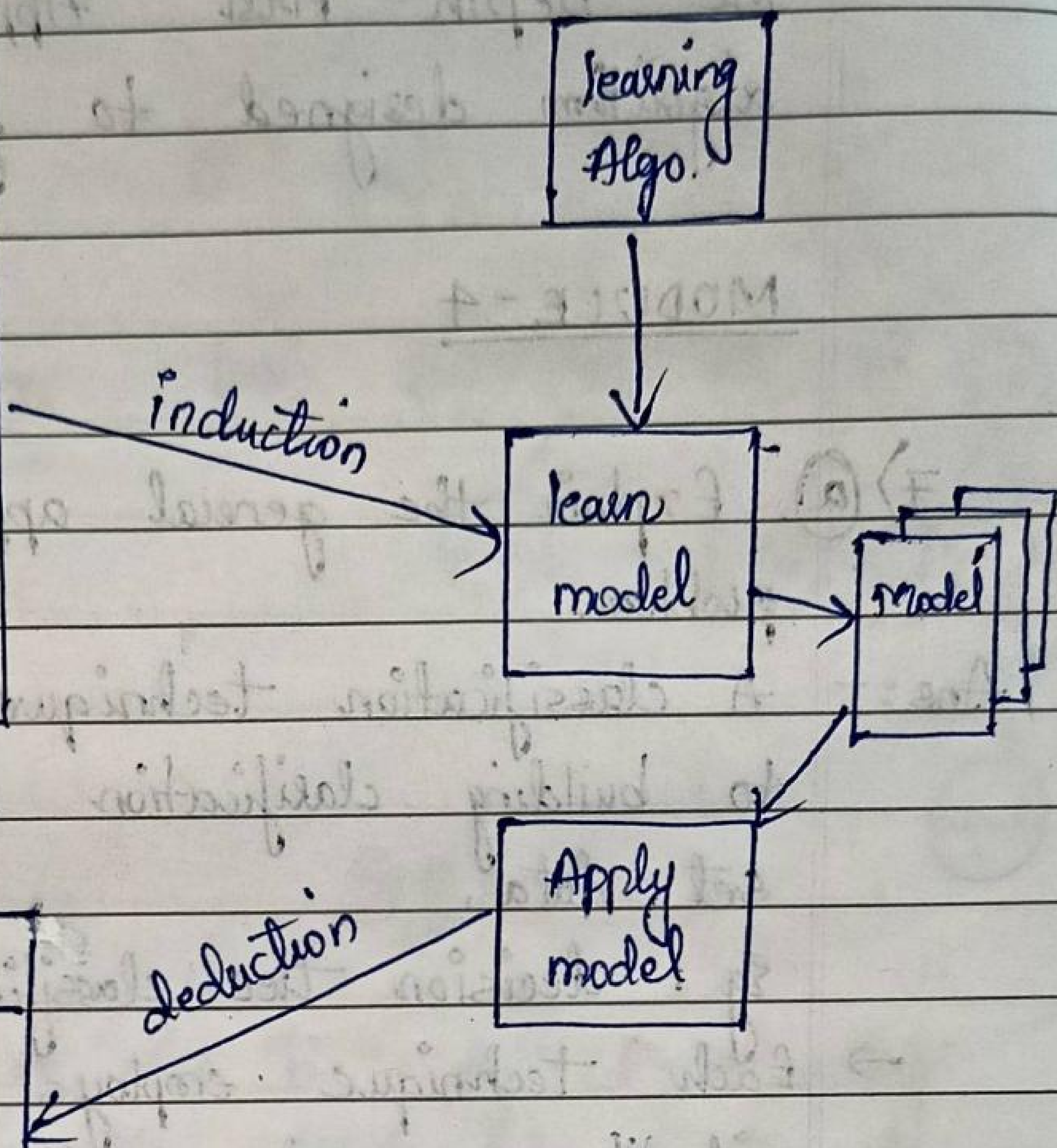
→ Each technique employs a "learning Algo" to identify a mode that best file the relationships between the attribute set & class label of input data.

→ The model generated by learning algo. should both fit the input data well & accurately predict the class label of records it has never before.

→ The Evaluation of performance of a classification model is based on the counts of test records correctly & incorrectly predicted by the model.

Tid	Attr 1	Attr 2	class
1	Yes	large	No
2	No	Small	No
3	Yes	Medium	No
4	No	large	Yes
5	Yes	large	No
6	Yes	Small	No

Training Set



Tid	Attr 1	Attr 2	class
7	Yes	large	?
8	No	Small	?
9	Yes	Small	?

Test Set

→ These counts are fabulated in a table known as "Confusion Matrix"

		Predict class	
		class = 1	class = 0
Actual class	class = 1	f_{11}	f_{10}
	class = 0	f_{01}	f_{00}

Table depicts the confusion matrix for a binary classification problem, Entry f_{ij} in the table denotes the no. of records from class, predicted to be of class.

→ The performance matrix such as accuracy which is defined as follows.

* Accuracy $= \dfrac{\text{No. of correct predictions}}{\text{total no. of predictions}}$

$$= \frac{f_{11} + f_{00}}{f_{10} + f_{11} + f_{00} + f_{01}}$$

The performance of a model can be expressed in terms of its error-rate which is given by,

* Error-rate $= \dfrac{\text{No. of wrong predictions}}{\text{total no. of predictions.}}$

$$= \frac{f_{10} + f_{01}}{f_{01} + f_{10} + f_{11} + f_{00}}$$

Most classification Algo. seeks models that attain the "highest accuracy or Equivalently lowest error rate" when applied to test set.

7) (b). Write the algorithm for decision tree induction.

Ans:- Algorithm for decision tree induction.

The input to the Algorithm consists of training records "E" & attribute set "F".

Tree Growth (E, F).

1). if stopping-cond (E, F) = true then
2). leaf = Create Node ()
3). leaf. label = classify (E
4). return leaf
5). Else

6). root = CreateNode( )
7). root.test_cond = find_test_split (E, F)
8). let V = { v/v is a possible outcome of root.test_cond }
9). for each v ∈ V do
10). E_v = { e / root.test_cond (e) = v ∈ e ∈ E }
11). child = Tree Growth (E_v, F)
12). add child as descendent of root & label the edge as v.
13). End for
14). End IF.
15). return Root.

After building the decision tree "Tree-pruning" step can be performed to reduce the size of decision tree.

→ decision tree that are too large are susceptible to phenomenon known as "Overfitting"

8) (a) Explain the methods of comparing classifiers.

Ans:- i) Estimating a confidence interval for accuracy.

To determine the confidence interval, we need to establish the probability distribution that governs the accuracy measure.

→ following is a list of characteristics of binomial Exp.

- Exp. consists of N independent trails, where trail has two possible outcomes → success
 → failure.
- The probability of success p, in each trail outcome.

→ Eg : Binomial Exp in counting no. of heads that turn up when coin flipped N times.

$$p(X = v) = \binom{N}{p} p^{v} (1-p)^{N-v}$$

where X is no. of success observed in N trails.

→ Based on Normal distribution, the following confidence interval for acc can be derived.

$$P\left(-Z_{\alpha/2} \leq \frac{acc - P}{\sqrt{P(1-P)/N}} \leq Z_{1-\alpha/2}\right) = 1-\alpha$$

ii) <u>Comparing the performance of two models.</u>

Consider a pair of models M_1 & M_2 that are evaluated on two independent test sets D_1 & D_2.

- Let n_1 denote no. of records in D_1 & n_2 denote no. of records in D_2

Suppose the error rate for M_1 on D_1 is ε_1 the Error rate for M_2 on D_2 is ε_2.

∴ the variance of d can be computed as follows,

$$\sigma_d^2 \simeq \hat{\sigma}_d^2 = \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}$$

where $e_1(1-e_1)/n_1$ & $e_2(1-e_2)/n_2$ are the variance of error rate finally at $(1-\alpha)\%$ confidence level.

iii). <u>Comparing performance of two classifier.</u>

Suppose we want to compute the performance of two classifier using k-fold cross-validation approach.

- Initially data set D is divided into k-equal sized partions, we then apply each classifier to construct a model from k-1 of partition & test it on remaining partition. This step repeated k times.

$$\hat{\sigma}^2_{d^{cv}} = \frac{\sum_{j=1}^{k} (d_j - \bar{d})^2}{k(k-1)}$$ is the obsered difference.

Where $\bar{d}$ is average difference, for this approach we need to use a t-distribution to compute the confidence interval for d_t^{cv}

$$d_t^{cv} = \bar{d} \pm t_{(1-\alpha), k-1} \hat{\sigma}_{d^{cv}}$$

8) (b) Write the characteristics of nearest neighbor classifier.

Ans: * Nearest-Neighbor classification is part of a more general technique known as "instance-based learning" which uses specific training instances to make prediction without having to maintain an abstraction derived from data.

* lazy learners such as nearest-neighbor classifier donot require model building.

* Nearest-Neighbour classifier make their prediction lead on local information.

* Nearest-Neighbour classifer can produce arbitarily shaped decision boundries.

The decision boundries of nearest-neighbor classifier have "high variability" because they depend on the composition of training examples.

* Nearest - Neighbor classifier can produce wrong prediction unless the appropriate proximity measure & data preprocessing steps are taken.

MODULE - 5

9) ⓐ. Explain the requirements of cluster analysis.

Ans:- Scalability - We need highly scalable clustering algorithms to deal with large databases.

Ability to deal with different kinds of attributes - Algorithms should be capable to be applied on any kind of data to be applied on any kind of data such as interval - based data, categorical, & binary data.

Discovery of clusters with attribute shape - The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small size.

High dimensionality - The clustering algorithm should not only be able to handle low - dimensional data but also the high dimensional space.

Ability to deal with noisy data - Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.

Interpretability – The clustering results should be interpretable, comprehensible, and usable.

9) (b). State and explain K-means algorithm.

Ans :- K-means clustering intends to partition 'n' objects into k-cluster in which each object belongs to the cluster with the nearest mean. This Method produces exactly "K" different cluster of greatest possible distinction.

- It is a prototype-based clustering technique creates a one-level partitioning of data objects. There are no. of such techniques but two of most prominent are "K-Means" & "K-Mediod".
- The objective of K-mean clustering is to minimize total "intra-cluster variance" or "squared error function".
- In K-mean, we first choose "K" initial centroids, where "K" is a user-specified parameter, namely the no. of clusters desired. Each point is then assigned to closest centroid & each collection of points assigned to a centroid is a cluster.
- The Centroid of each cluster is then updated based on the points assigned to the cluster. we repeat the assignment & update steps until no point changes clusters or Equivalently

until the centroide remain the same.

K-Means Algorithm

1. Select K points as initial centroids.
2. repeat
3. form K cluster by assigning each point to its closest centroid.
4. recompute the centroid of each cluster.
5. Until centroide do not change.

K-means algorithm was stated generally as "recompute the centroid of each cluster".

10) (a). Write DBSCAN clustering algorithm and estimate time and space complexity.

Ans:- DBSCAN algorithm is based on this intuitive notion of "clusters" & "noise". The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

DBSCAN Algorithm

1. label all points as Core, Border and Noise
2. Eliminate Noise points.
3. put an Edge between all core points that are within Eps.
4. Make each group of connected core points to seperate cluster.
5. Assign each Border point to one of cluster of its associated core points.

Time Complexity : $O(n^2)$

- For each point it has to be determined if it is a core point.
- can be reduced to $O(n * \log(n))$ in lower dimensional spaces by using efficient data structures.

Space Complexity : $O(n)$.

- It is only necessary to keep a small amount of data for each point i.e, the cluster label & the identification of each point as a core, border or noise point.

10) (b). Explain different types of clusters ?

Ans :- Different types of Clusters :-

Clustering Aims to find useful groups of objects (clusters), where usefulness is defined by the goals of Data Analysis.

- Well-Separated :- The data objects with a cluster must have small distance & distance between two cluster must be high.

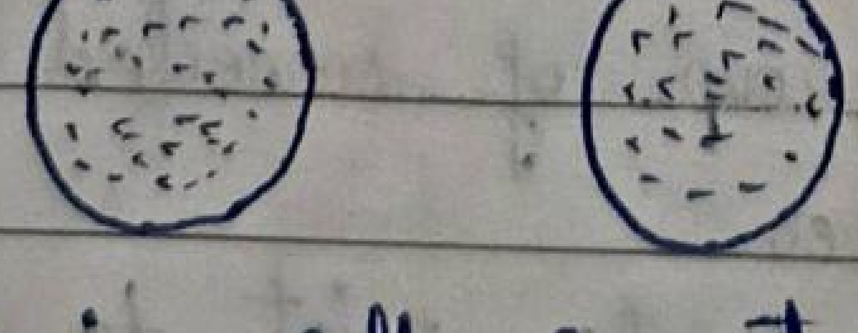


fig : Well-Separated cluster.

Sometimes a threshold is used to specify that all the objects in a cluster must be sufficiently close to one another.

- **Prototype-Based clusters (Center-Based)** :- A cluster is a set of objects in which each object is closer to the prototype that defines the cluster than to the prototype of any other clusters.

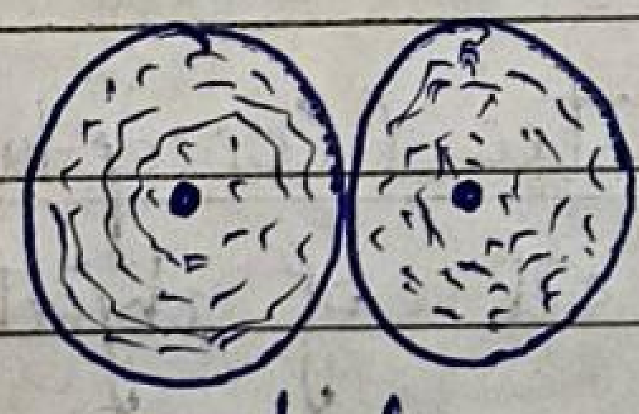


fig: proto type - Based cluster.

- **Graph-Based clusters (Contiguity-Based)** :- If the data is represented as graph, where the nodes are objects & the links represent connection among objects then a cluster can be defined as connected component.

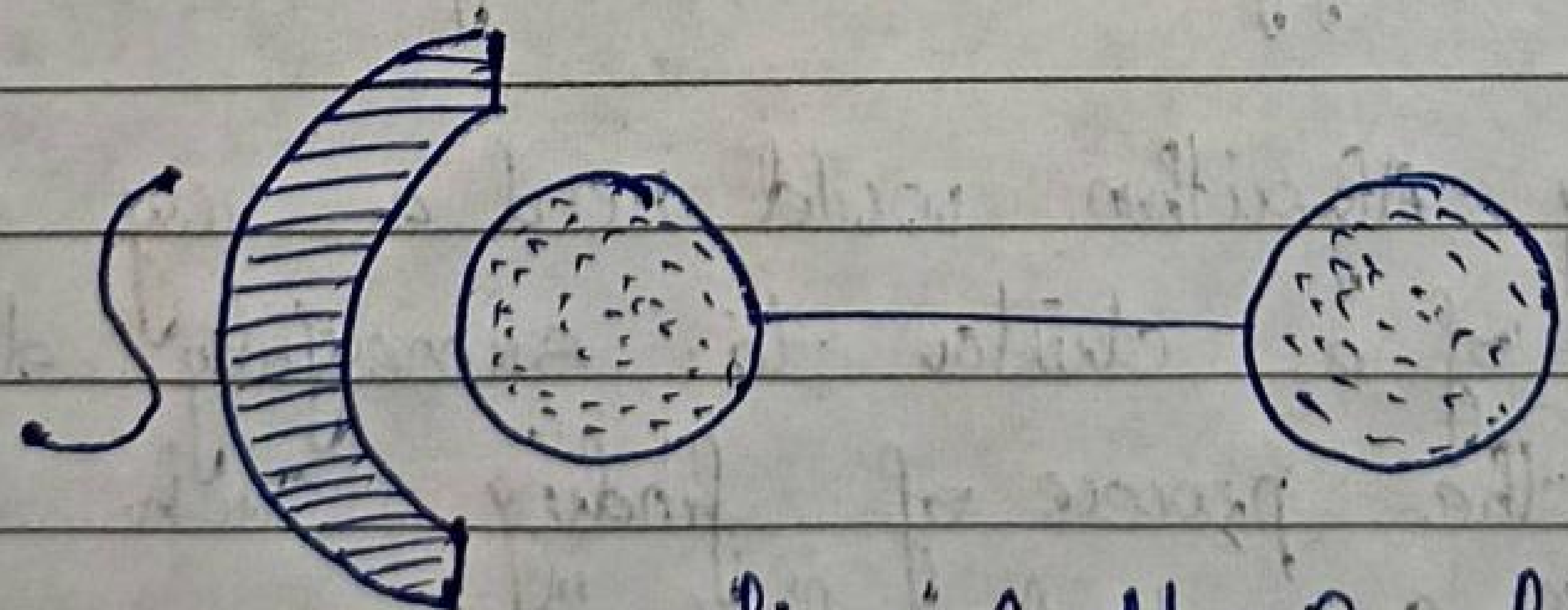


fig: Graph-Based cluster.

- **Density-Based cluster** :- A cluster is a dense region of objects that is surrounded by a region of low density. A density Based defination of cluster is often employed when the cluster are irregular or intertwined & when noise & outliers are present.

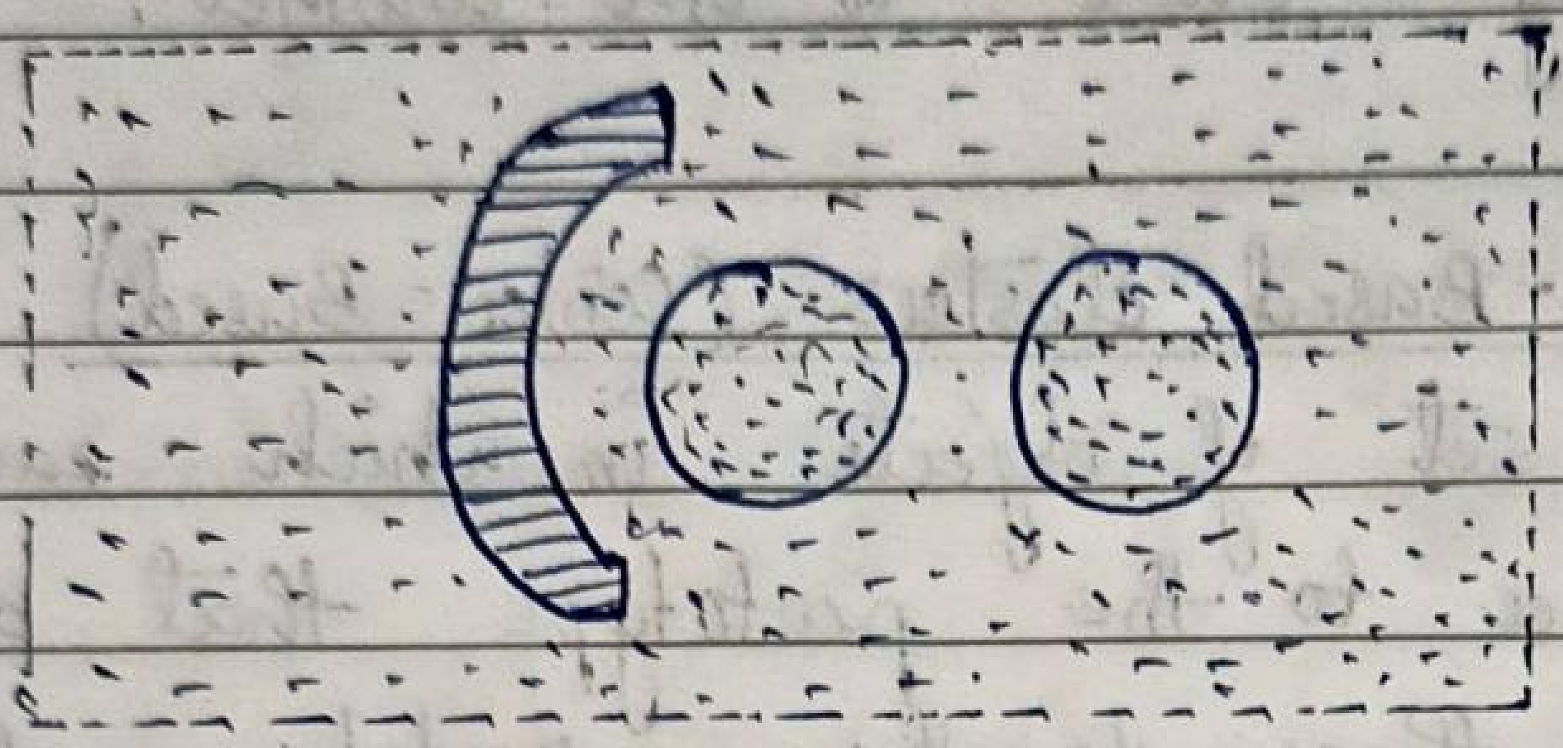


fig : Density - Based cluster.

- Shared - property cluster (Conceptual) :- More Generally, A cluster as a set of objects that share same property.

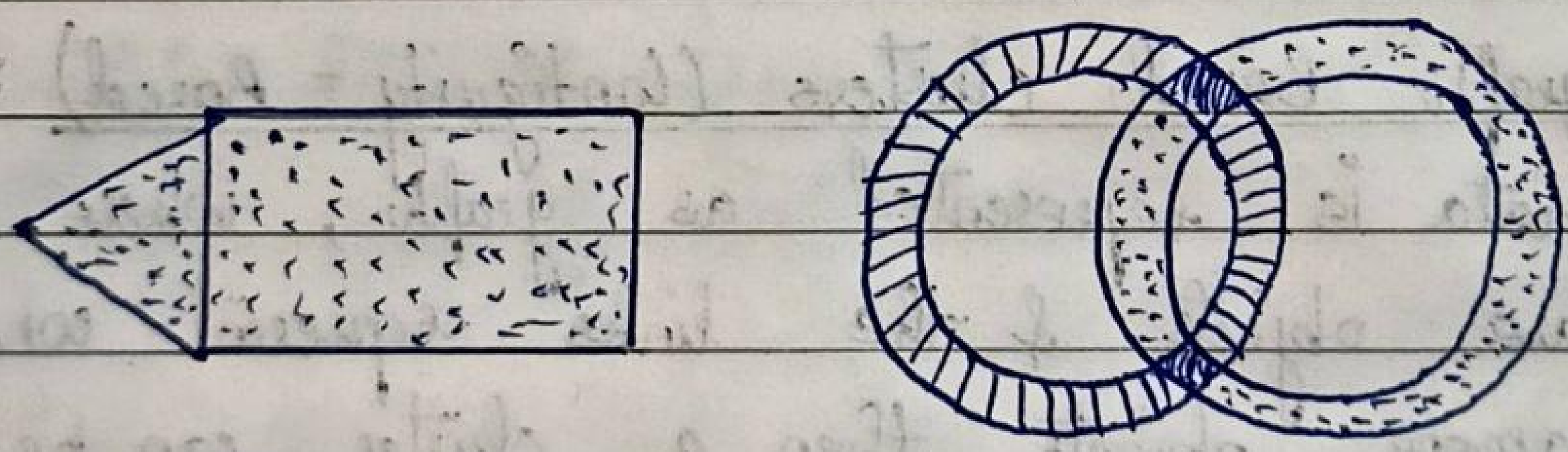


fig : shared - property clusters.

A clustering Algorithm would need a very specific concept of a cluster to successfully detect these clusters, the process of finding such clusters is called " Conceptual clusters".