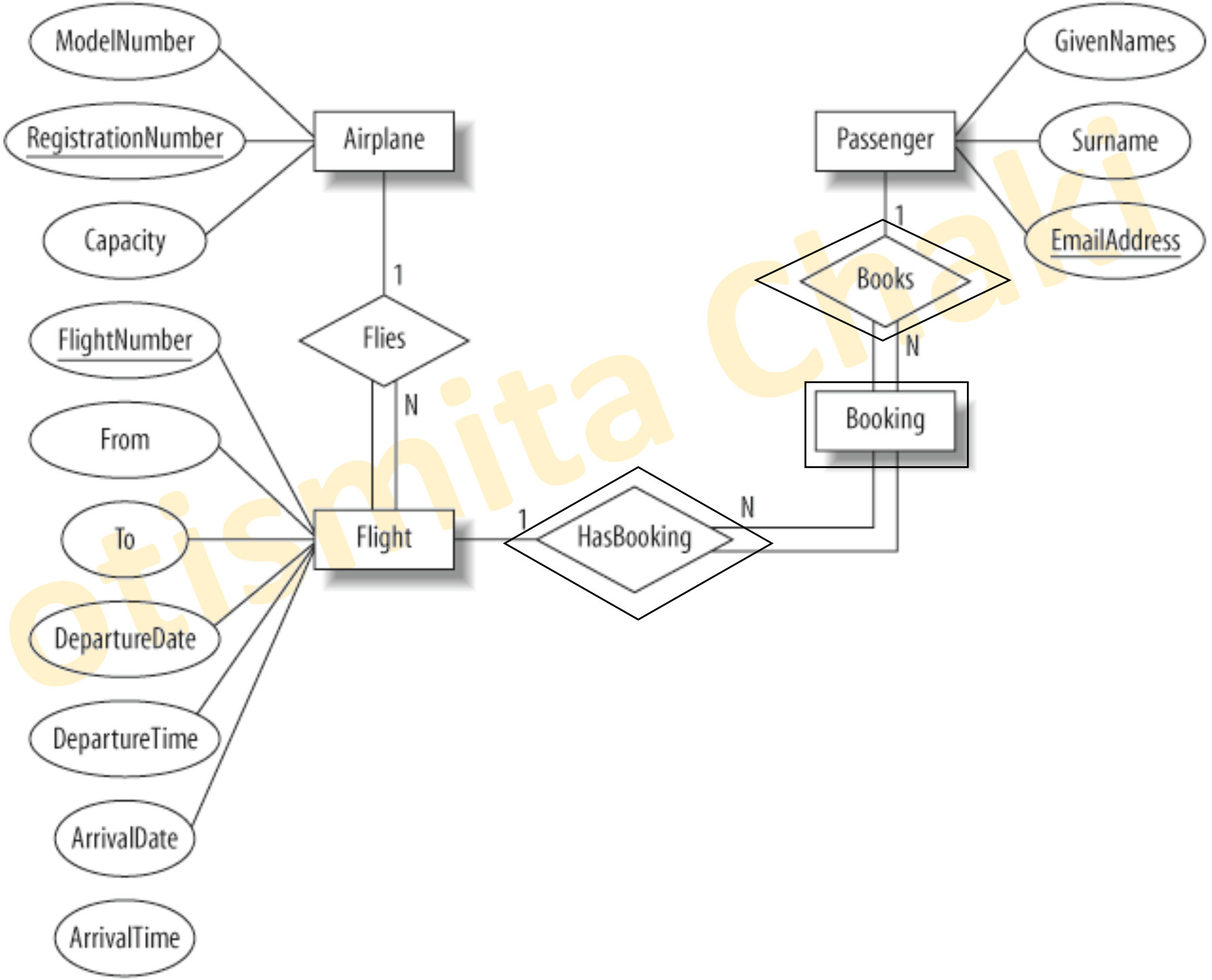
Examples

Dr. Jyotismita Chaki

ER Diagram: 1

- Consider the following requirements list:
  - The airline has one or more airplanes.
  - An airplane has a model number, a unique registration number, and the capacity to take one or more passengers.
  - An airplane flight has a unique flight number, a departure airport, a destination airport, a departure date and time, and an arrival date and time.
  - Each flight is carried out by a single airplane.
  - A passenger has given names, a surname, and a unique email address.
  - A passenger can book a seat on a flight.

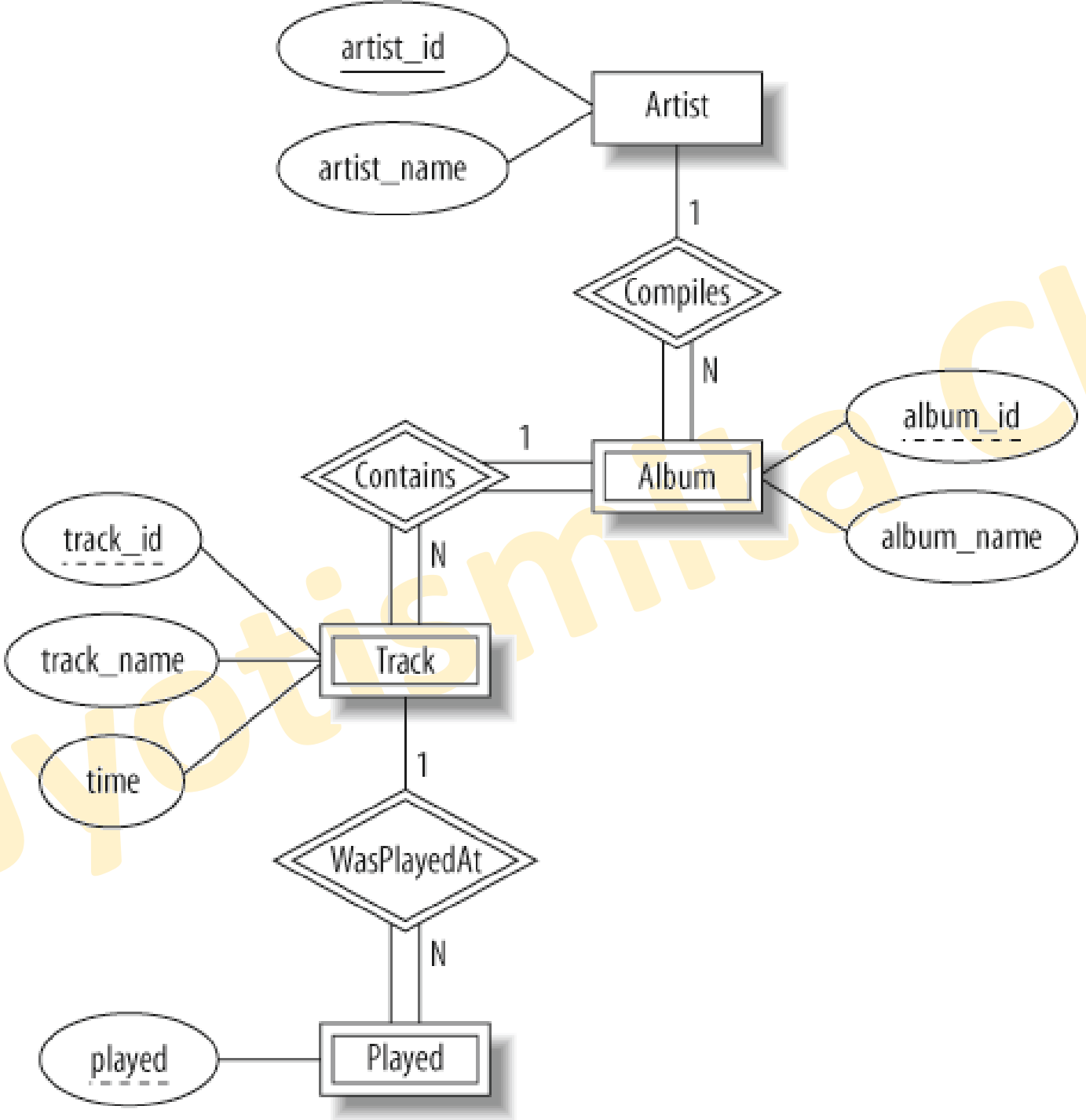
Solution



ER Diagram: 2

- Requirements for the music database:
  - The collection consists of albums.
  - An album is made by exactly one artist.
  - An artist makes one or more albums.
  - An album contains one or more tracks
  - Artists, albums, and tracks each have a name.
  - Each track is on exactly one album.
  - Each track has a time length, measured in seconds.
  - When a track is played, the date and time the playback began (to the nearest second) should be recorded; this is used for reporting when a track was last played, as well as the number of times music by an artist, from an album, or a track has been played.

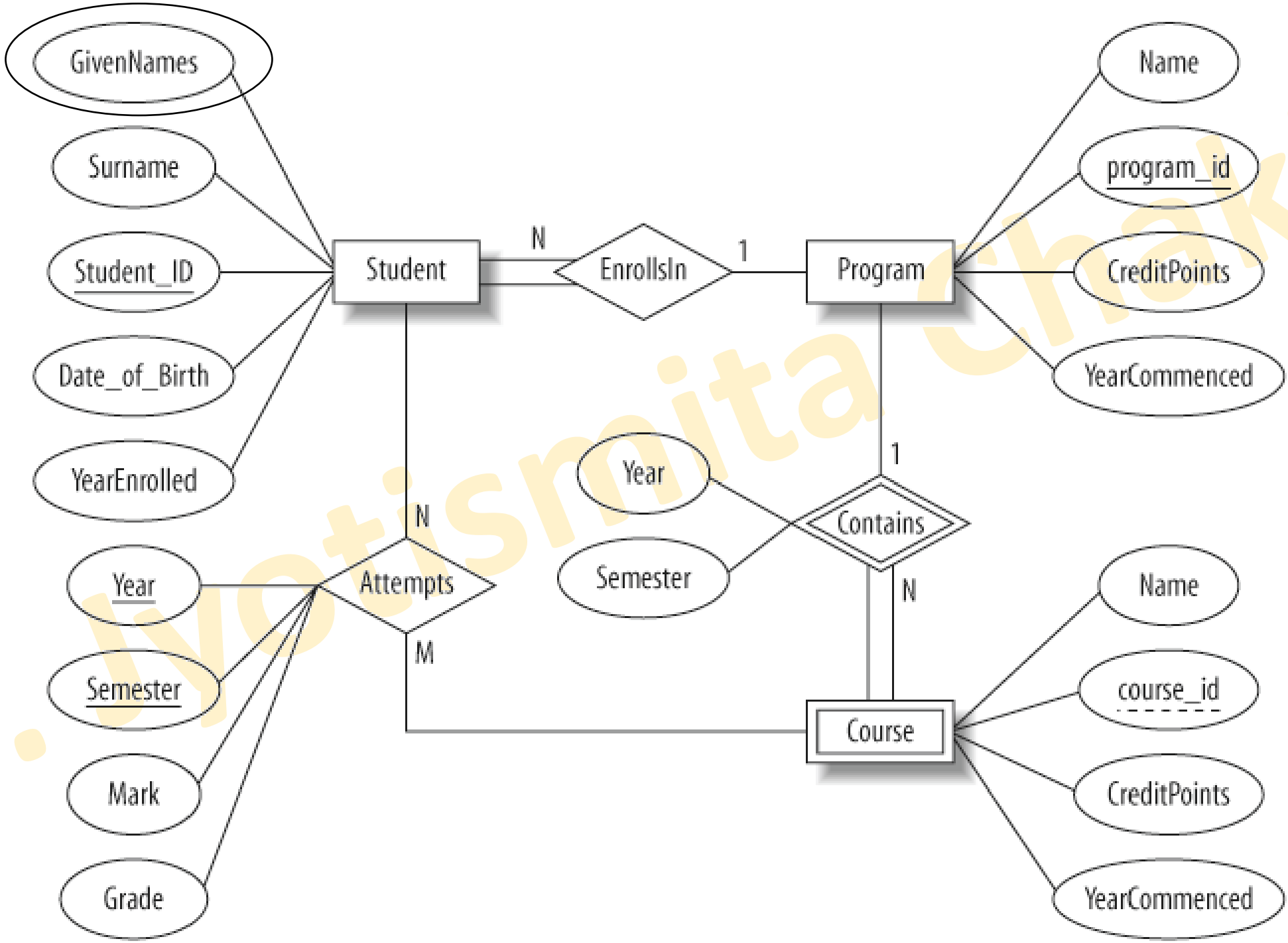
Solution



ER Diagram: 3

- Consider the following requirements list:
  - The university offers one or more programs.
  - A program is made up of one or more courses.
  - A student must enroll in a program.
  - A student takes the courses that are part of her program.
  - A program has a name, a program identifier, the total credit points required to graduate, and the year it commenced.
  - A course has a name, a course identifier, a credit point value, and the year it commenced.
  - Students have one or more given names, a surname, a student identifier, a date of birth, and the year they first enrolled. We can treat all given names as a single object—for example, “John Paul.”
  - When a student takes a course, the year and semester he attempted it are recorded. When he finishes the course, a grade (such as A or B) and a mark (such as 60 percent) are recorded.
  - Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).

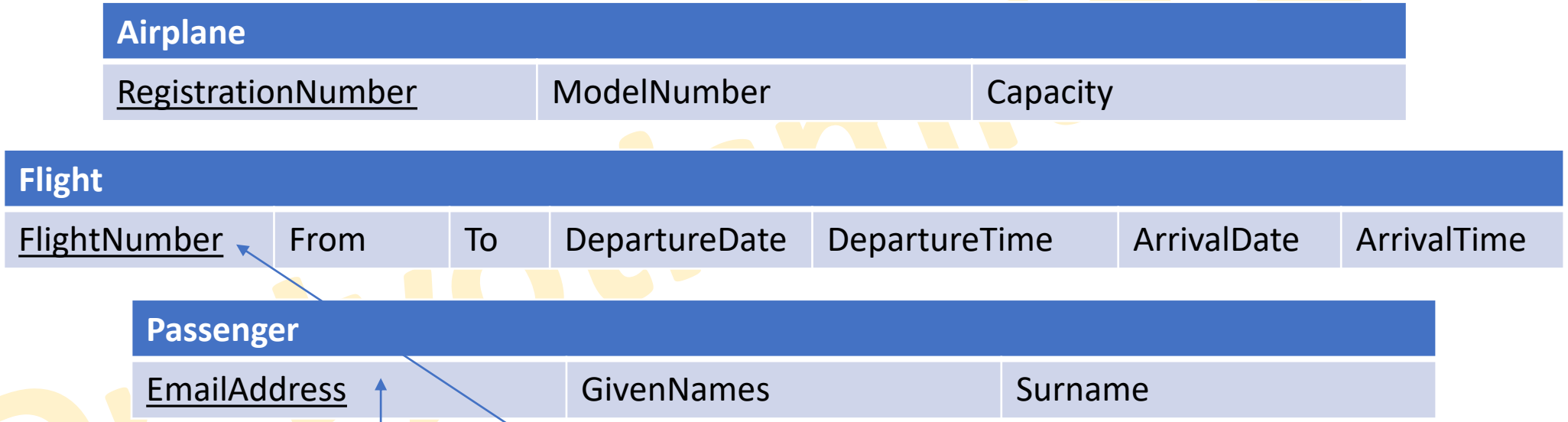
Solution



ER to Relational Model: 1

Solution

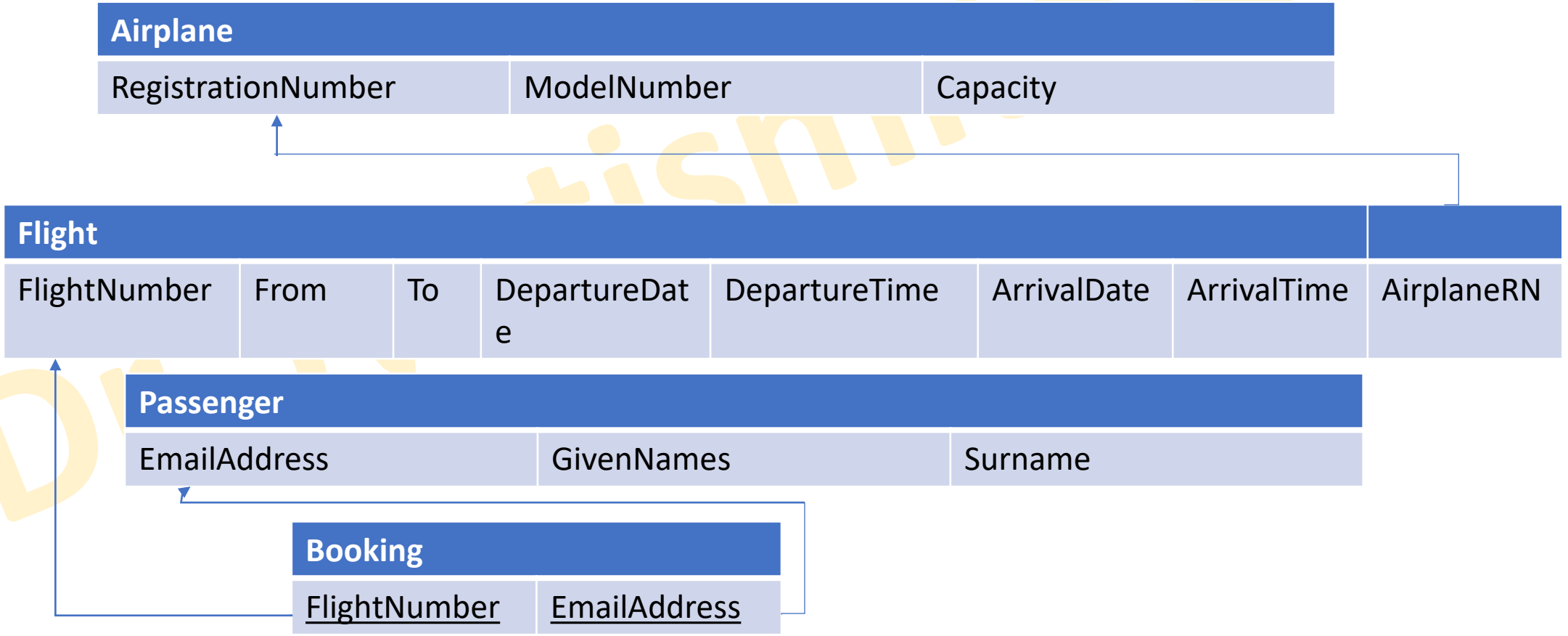
- Step 1: Mapping of Regular Entity Types.



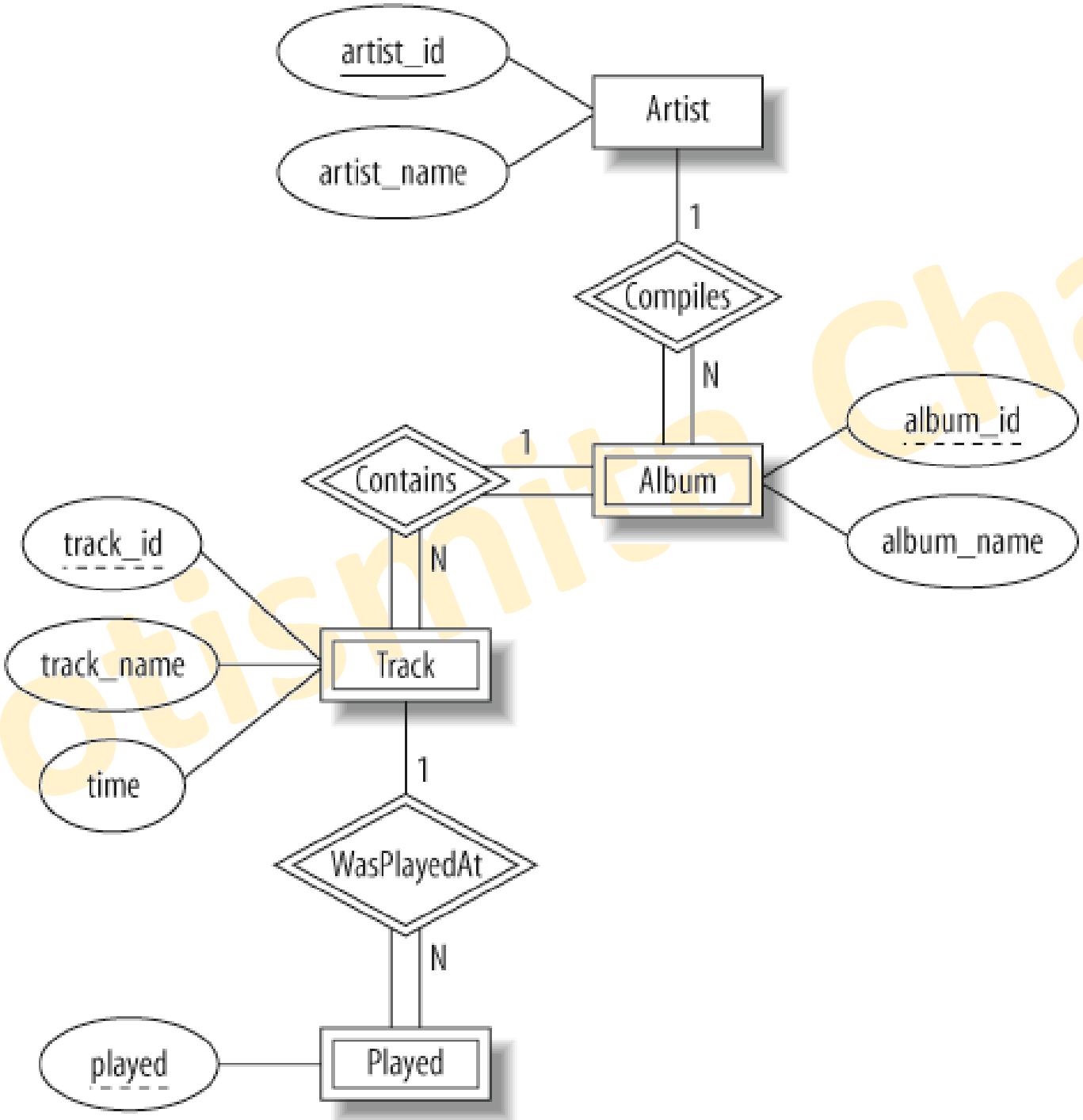
- Step 1: Mapping of Week Entity Types

Solution

- Step 2: Mapping of Binary 1:N Relationship Types



ER to Relational Model: 2



Solution

- Step 1: Mapping of Regular Entity Types

Artist	
Artist_ID	Artist_Name

- Step 2: Mapping of Weak Entity Types

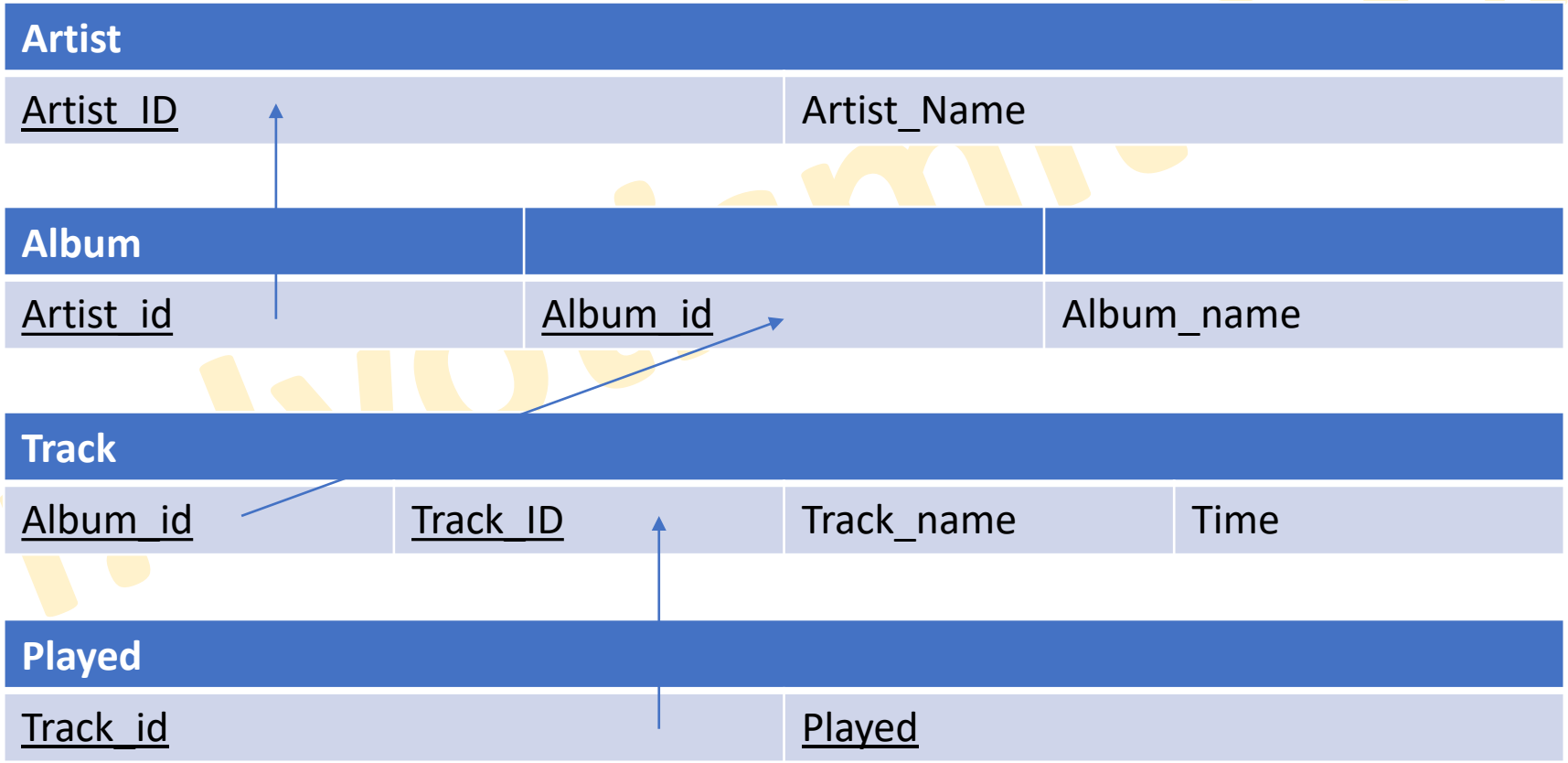
Album		
Artist_id	Album_id	Album_name

Track			
Album_id	Track_ID	Track_name	Time

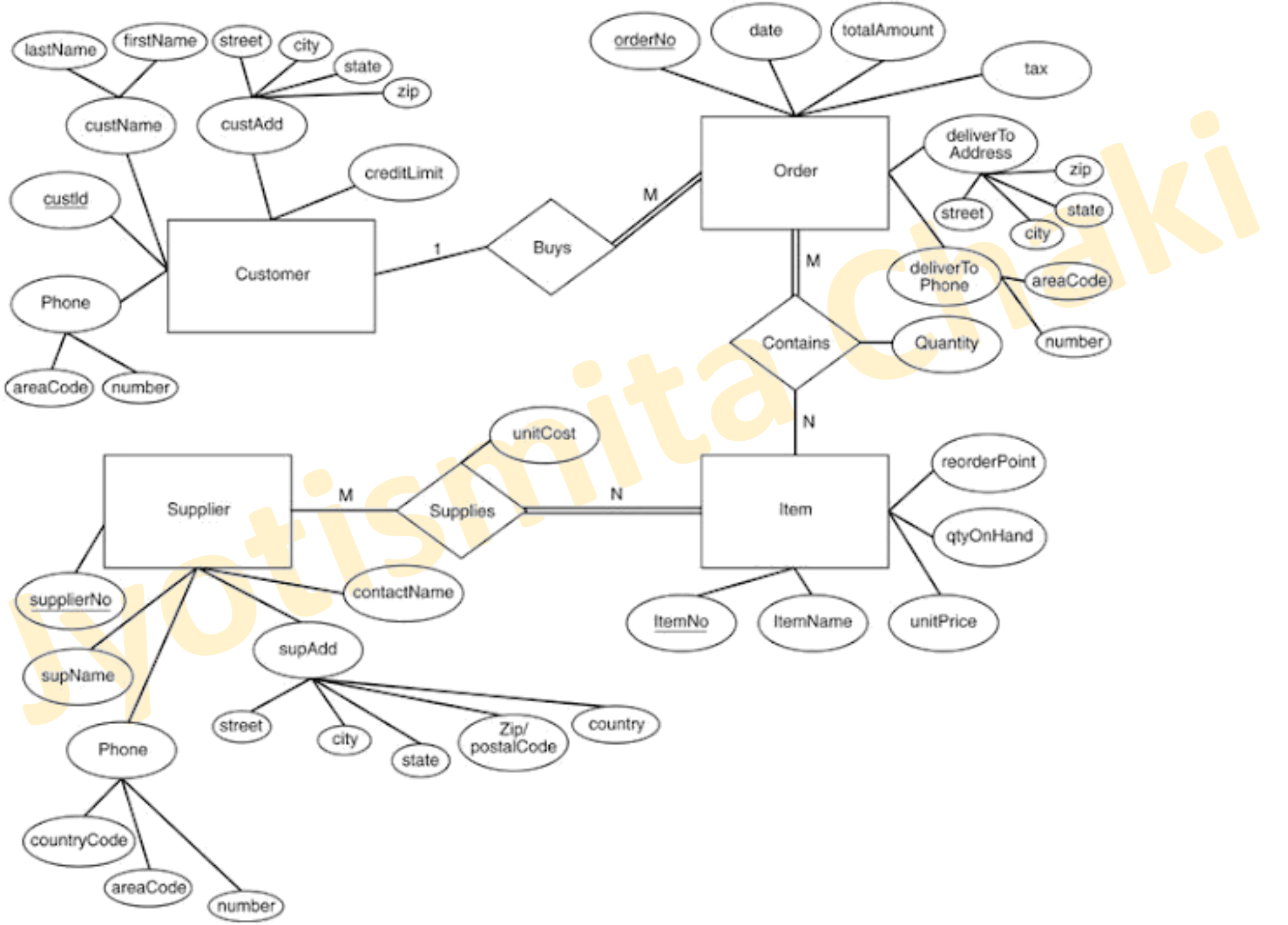
Played	
Track_id	Played

Solution

- Step 2: Mapping of Binary 1:N Relationship Types



ER to Relational model: 3



Solution

- Customer (custID, lastName, firstName, street, city, state, zip, creditLimit, areaCode, number)
- Order (orderNo, date, totalAmount, tax, street, city, state, zip, areaCode, number, custId)
- Supplier (supplierNo, supName, street, city, state, zip, country, contactName, countryCode, areaCode, number)
- Item (ItemNo, ItemName, unitPrice, qtyOnHand, reorderPoint)
- Contains (orderNo, ItemNo, Quantity)
- Supplies (supplierNo, ItemNo, unitCost)

Normalization: 1

Emp_Dept_Skill

EID	EName	DeptID	Phone	DeptName	SkillID	SkillName	SkillYears
73775	Alex	CU5	1111, 2222, 3333	Billing	782, 691, 783	Cleark, Planner, Annalyst	5, 1, 3
96347	Bill	41F	4444, 5555, 6666	Shipping	783, 316, 780	Analyst, Typist, Designer	2, 2, 10
87346	Cindy	QP2	7777, 8888, 9999	Marketing	005, 099, 316	Sales, Secretary, Typist	2, 15, 4

Solution: 1NF

EID	DeptID	EName	DeptName
73775	CU5	Alex	Billing
96347	41F	Bill	Shipping
87346	QP2	Cindy	Marketing

Emp_Dept

EID	DeptID	Phone	SkillID	SkillName	SkillYears
73775	CU5	1111	782	Cleark	5
73775	CU5	2222	691	Planner	1
73775	CU5	3333	783	Annalyst	3
96347	41F	4444	783	Analyst	2
96347	41F	5555	316	Typist	2
96347	41F	6666	780	Designer	10
87346	QP2	7777	005	Sales	2
87346	QP2	8888	099	Secretary	15
87346	QP2	9999	316	Typist	4

Emp_Dept_Skill

Solution: 2NF

Emp1

EID	EName
73775	Alex
96347	Bill
87346	Cindy

Department

DeptID	DeptName
CU5	Billing
41F	Shipping
QP2	Marketing

Emp_Dept

EID	DeptID
73775	CU5
96347	41F
87346	QP2

Emp2

EID	Phone	SkillYears
73775	1111	5
73775	2222	1
73775	3333	3
96347	4444	2
96347	5555	2
96347	6666	10
87346	7777	2
87346	8888	15
87346	9999	4

Skill

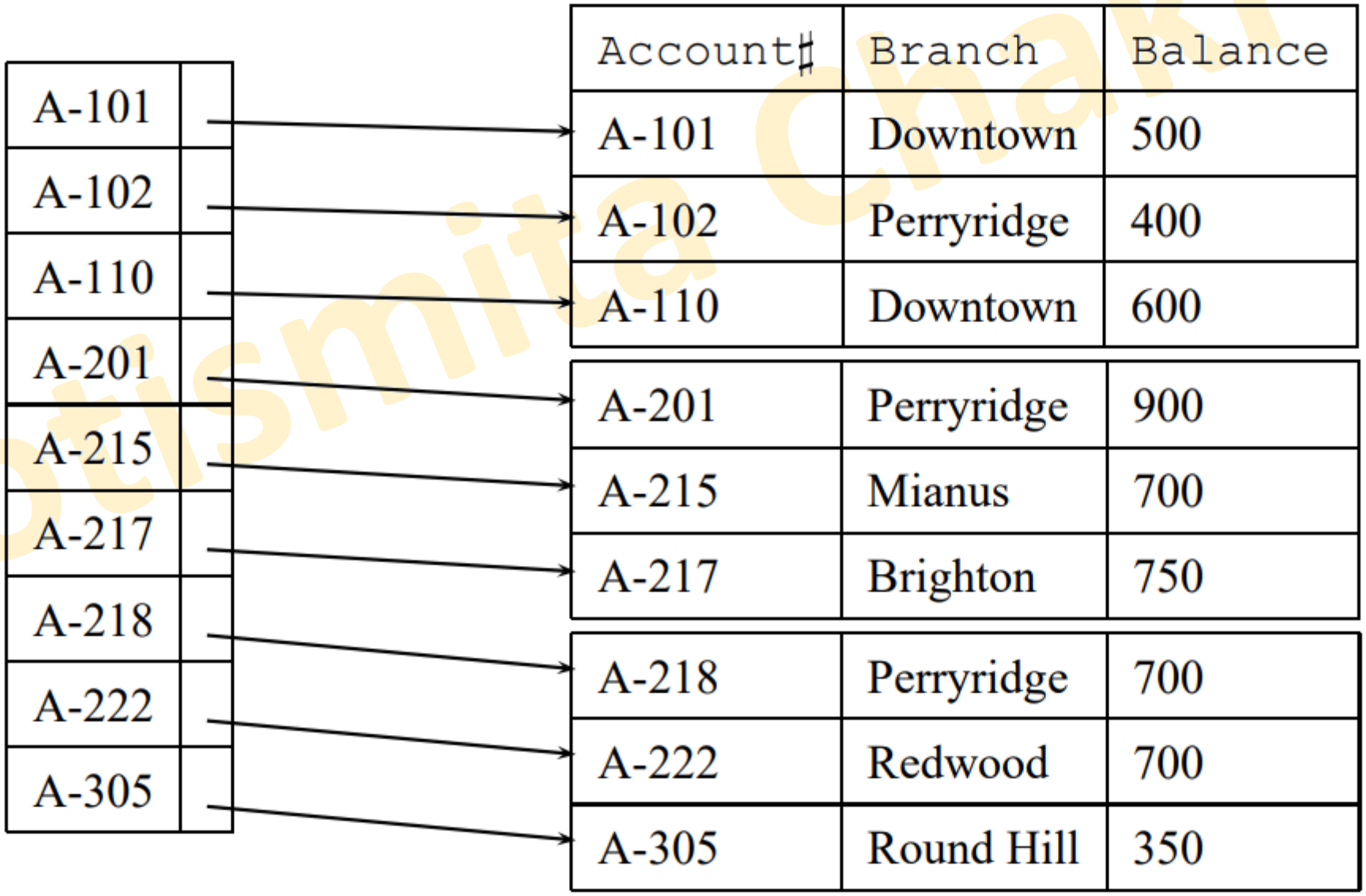
SkillID	SkillName
782	Cleark
691	Planner
783	Annalyst
783	Analyst
316	Typist
780	Designer
005	Sales
099	Secretary
316	Typist

Emp_Skill_Dept

EID	SkillID	DeptID
73775	782	CU5
73775	691	CU5
73775	783	CU5
96347	783	41F
96347	316	41F
96347	780	41F
87346	005	QP2
87346	099	QP2
87346	316	QP2

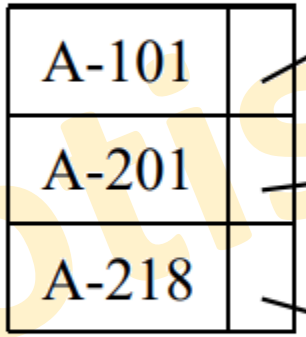
Primary Index: Dense

- Every value of the search key has a representative in a Dense Index.
- The index maintains the keys in the same order as in the data file.
- Primary Dense Index with Search Key=Account#.



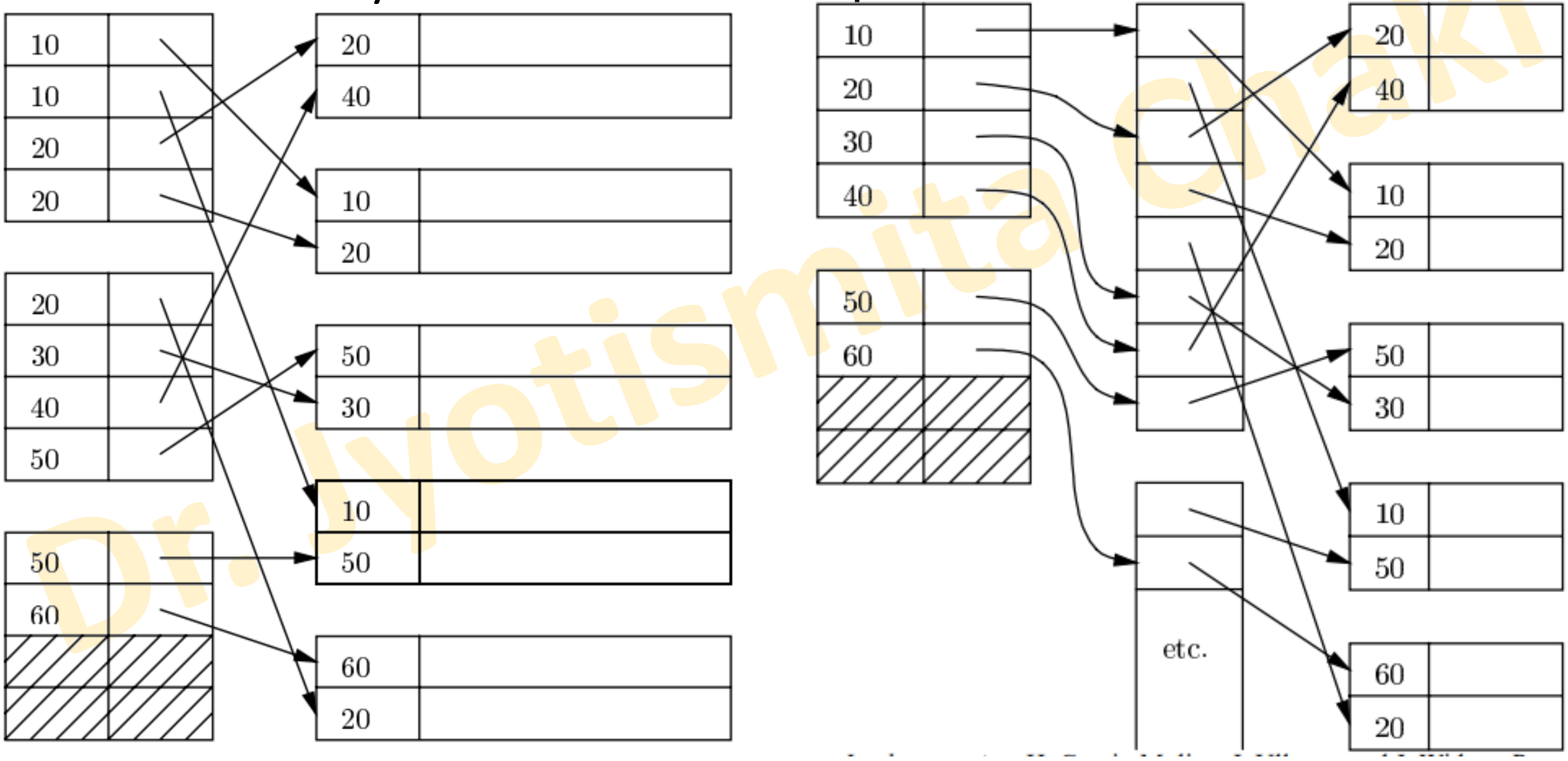
Primary Index: Sparse

- Used when dense indexes are too large: A Sparse Index uses less space at the expense of more time to find a record given a key.
- A sparse index holds one key-pointer pair per data block, usually the first record on the data block.
- Primary Sparse Index with Search Key=Account#.

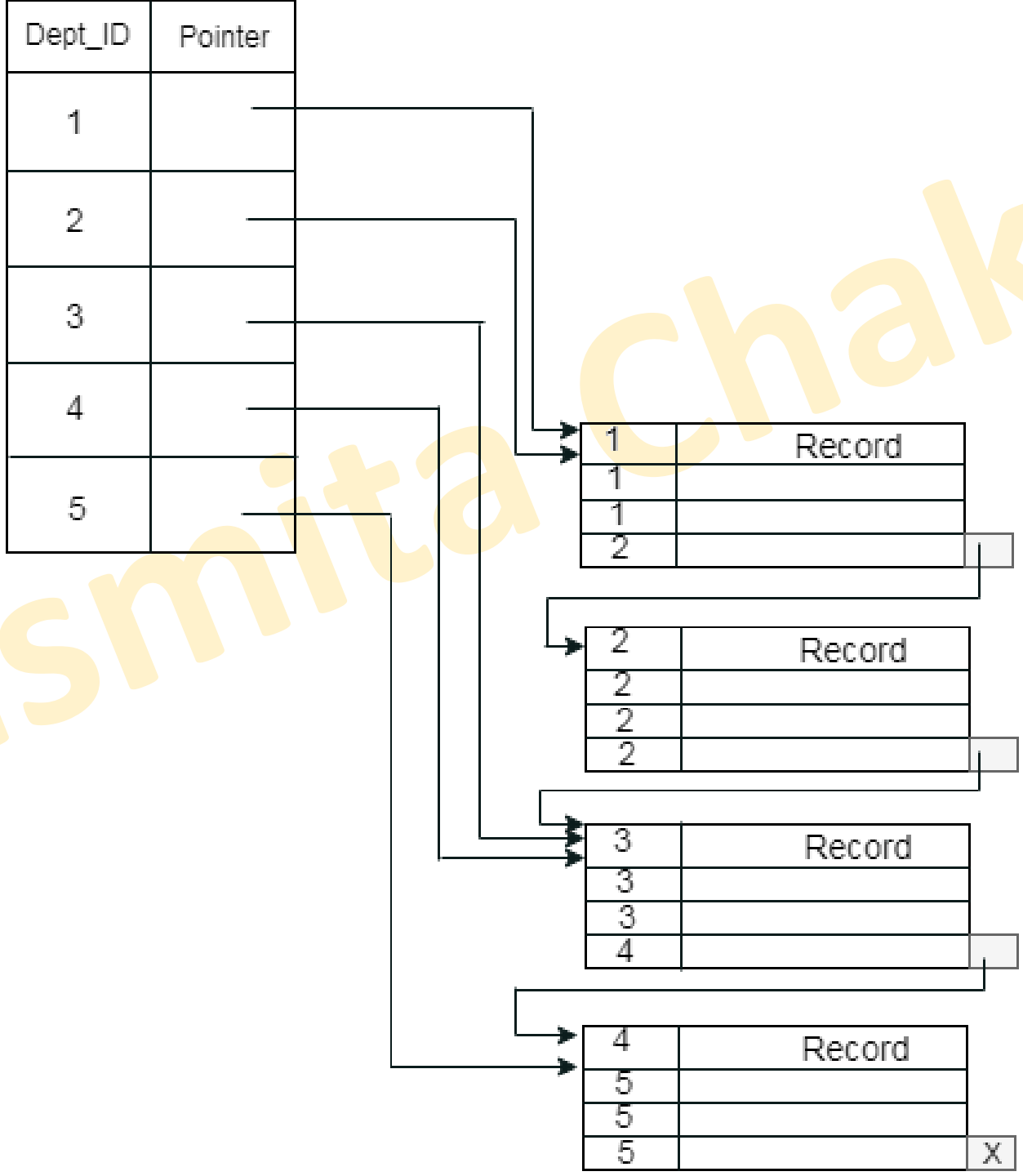


Account#	Branch	Balance
A-101	Downtown	500
A-102	Perryridge	400
A-110	Downtown	600
A-201	Perryridge	900
A-215	Mianus	700
A-217	Brighton	750
A-218	Perryridge	700
A-222	Redwood	700
A-305	Round Hill	350

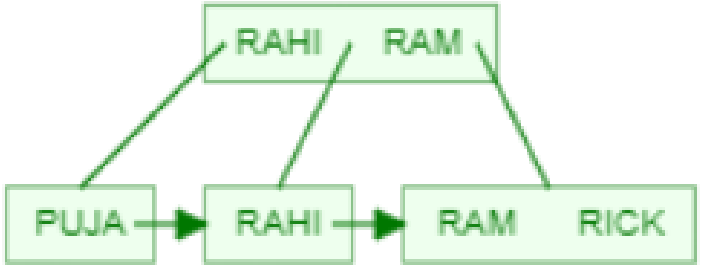
Secondary Index: Record pointer

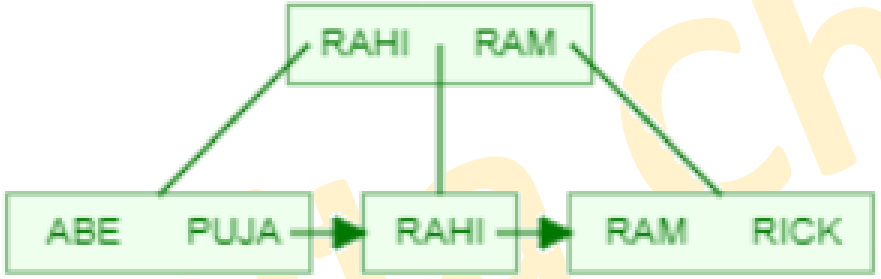


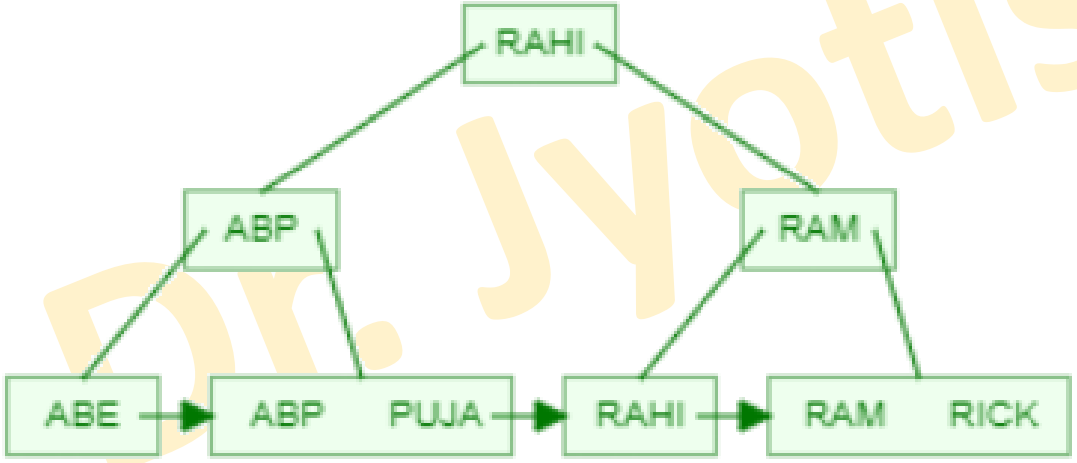
Clustering Index

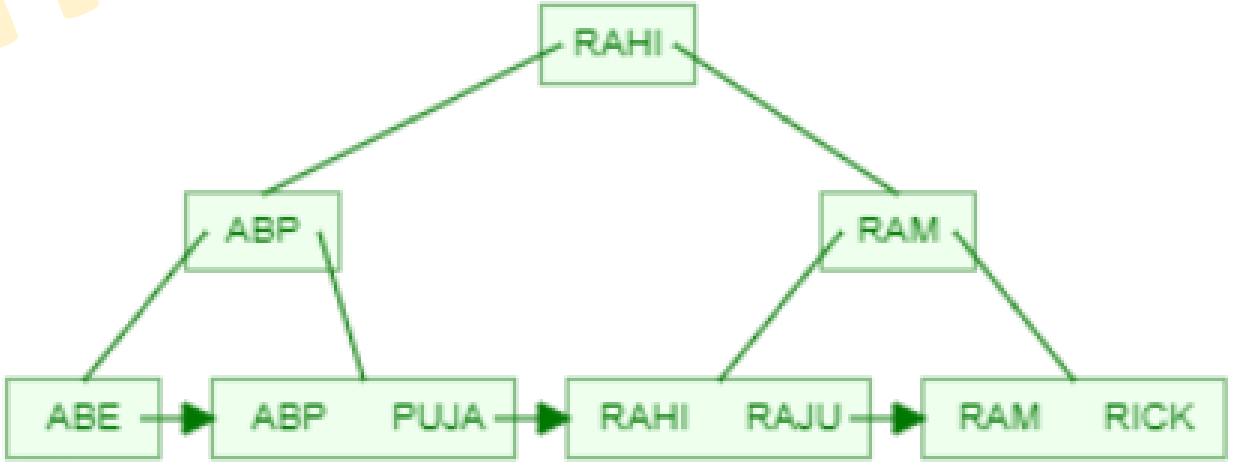


Insertion using B+ tree: Max degree = 3: Insert ABE, ABP, RAJU

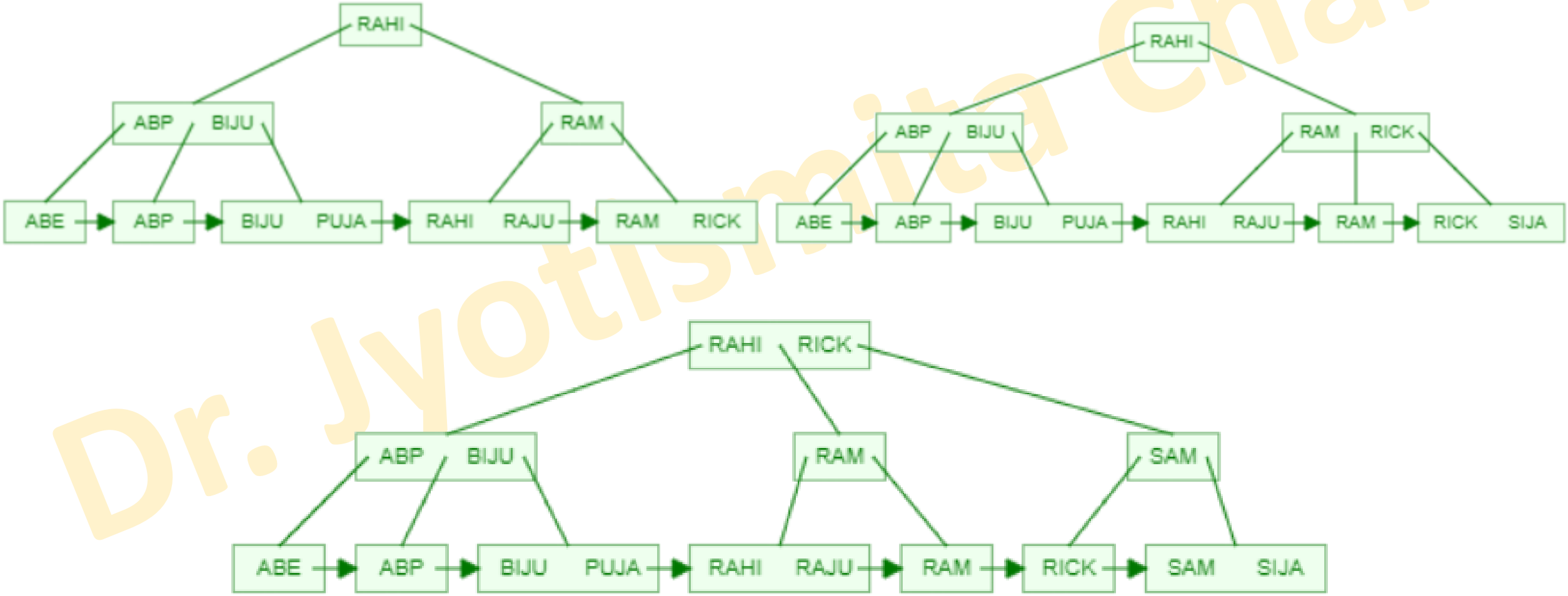




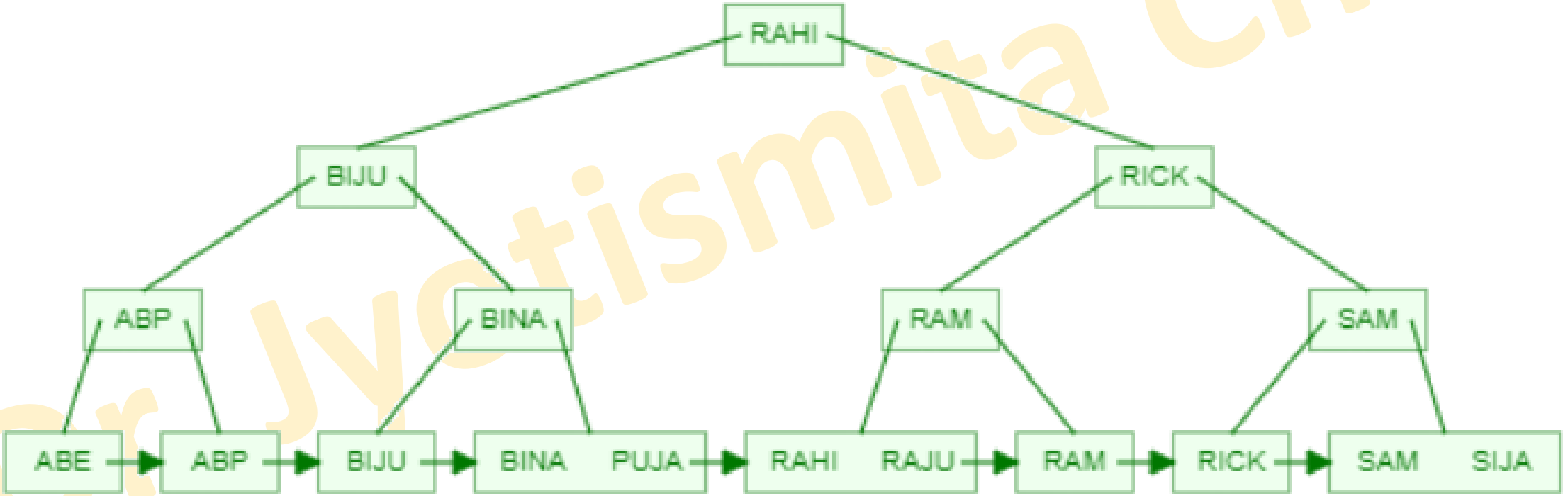




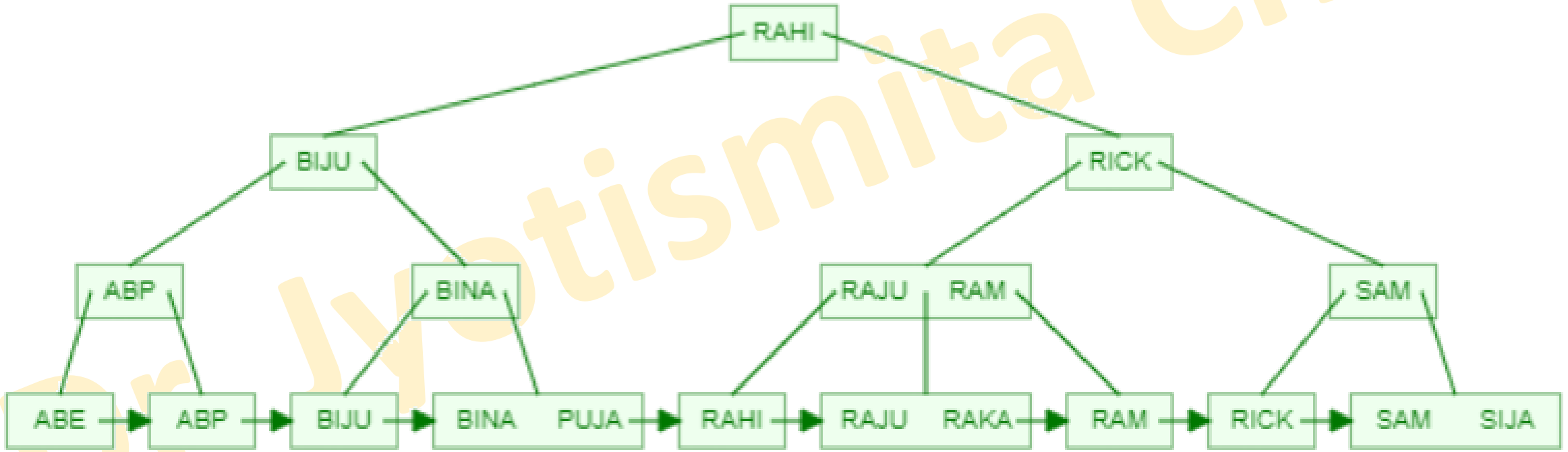
Insertion using B+ tree: Max degree = 3: Insert BUJU, SIJA, SAM



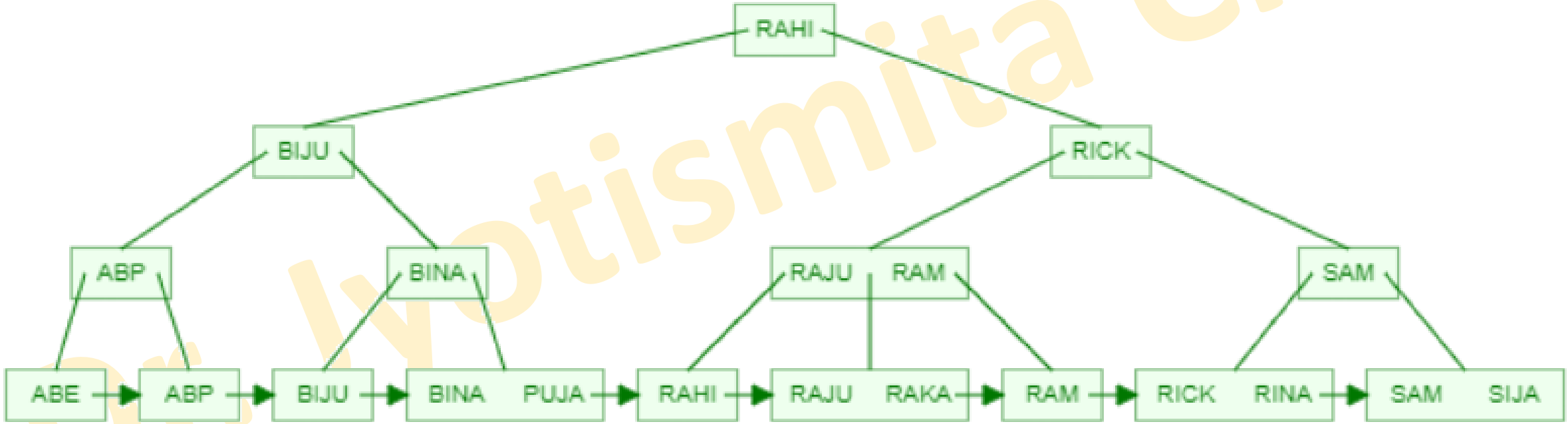
Insertion using B+ tree: Max degree = 3: Insert BINA



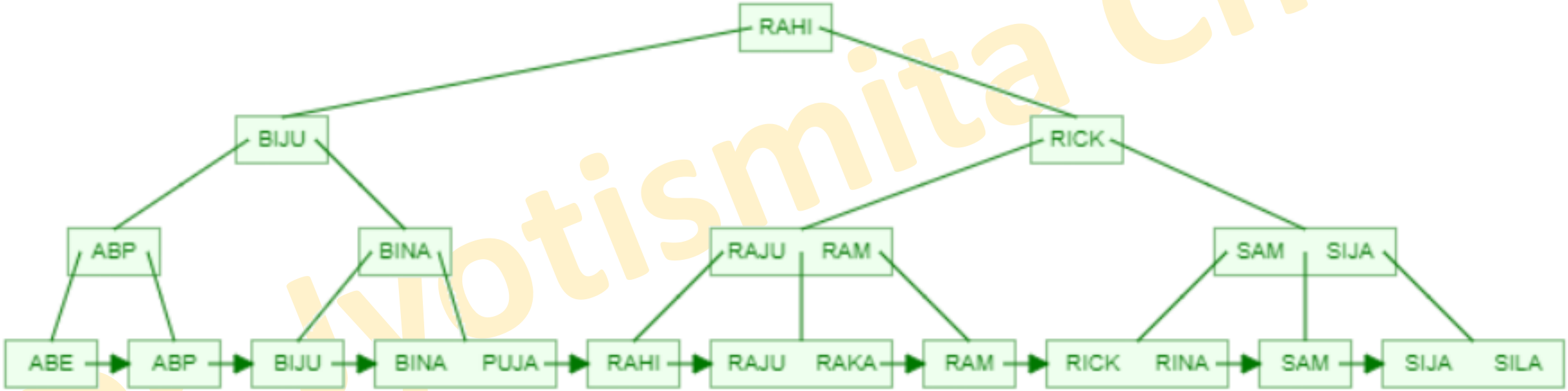
Insertion using B+ tree: Max degree = 3: Insert RAKA



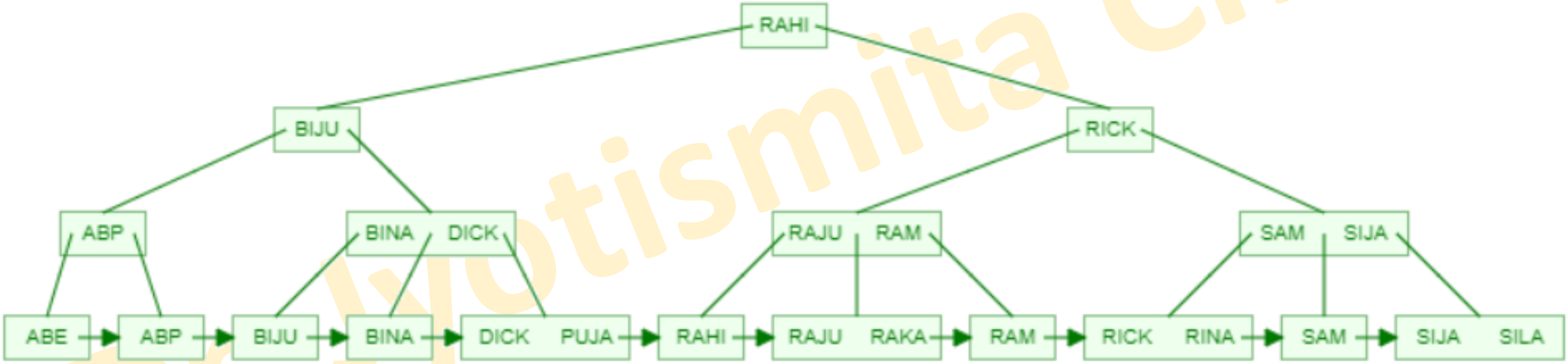
Insertion using B+ tree: Max degree = 3: Insert RINA



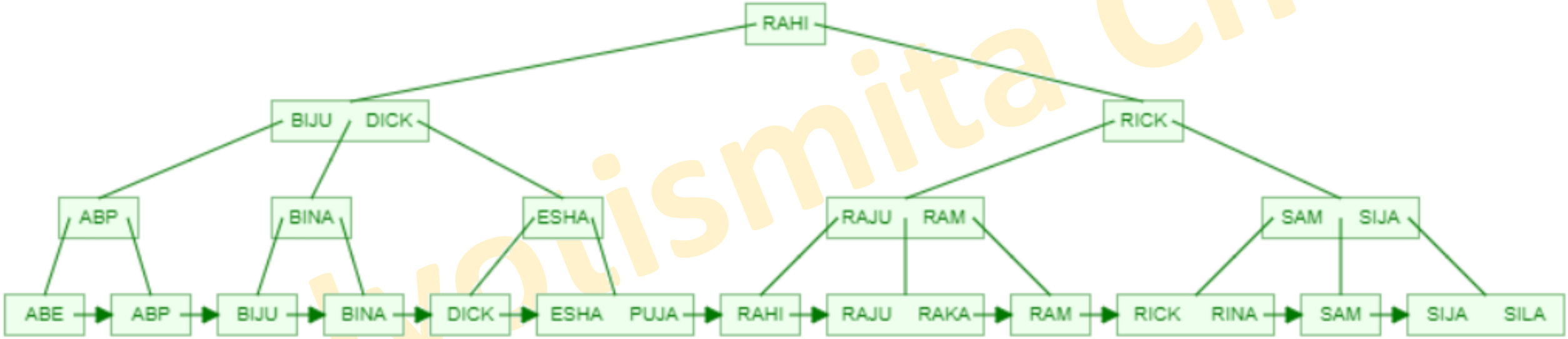
Insertion using B+ tree: Max degree = 3: Insert SILA



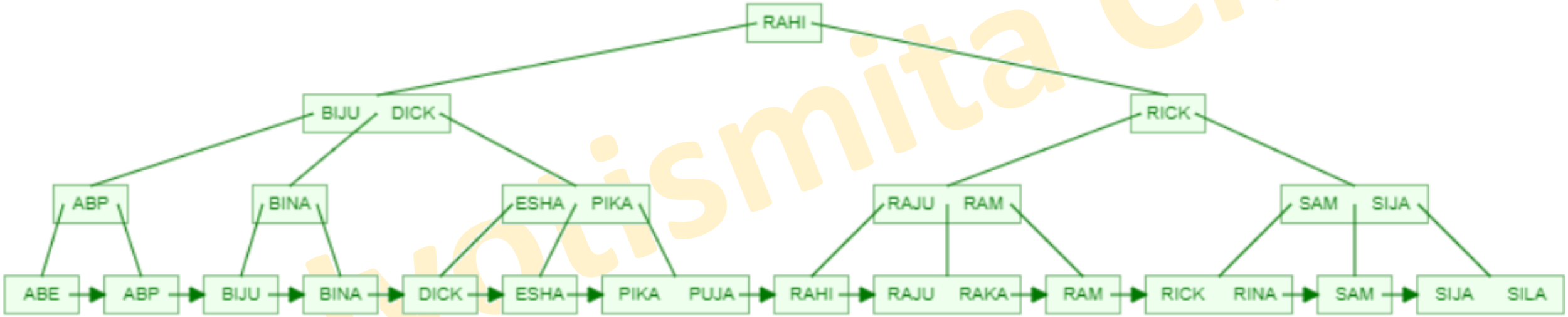
Insertion using B+ tree: Max degree = 3: Insert DICK



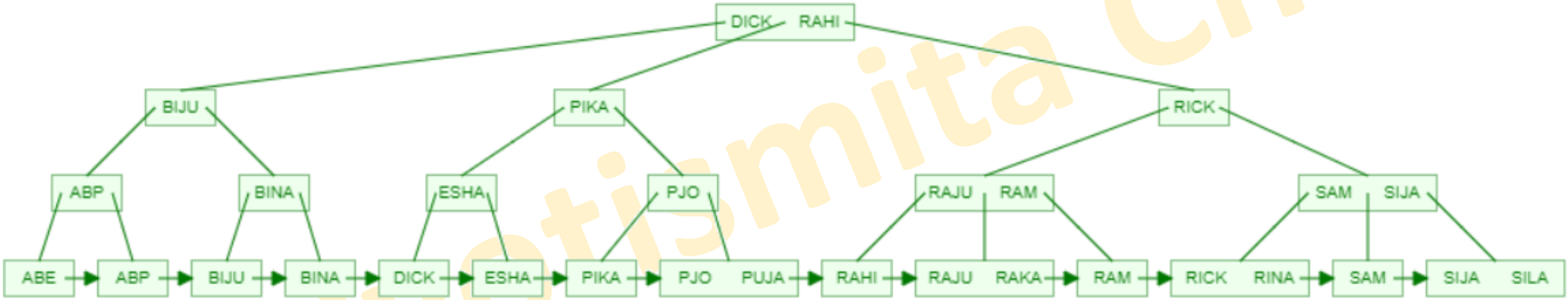
Insertion using B+ tree: Max degree = 3: Insert ESHA



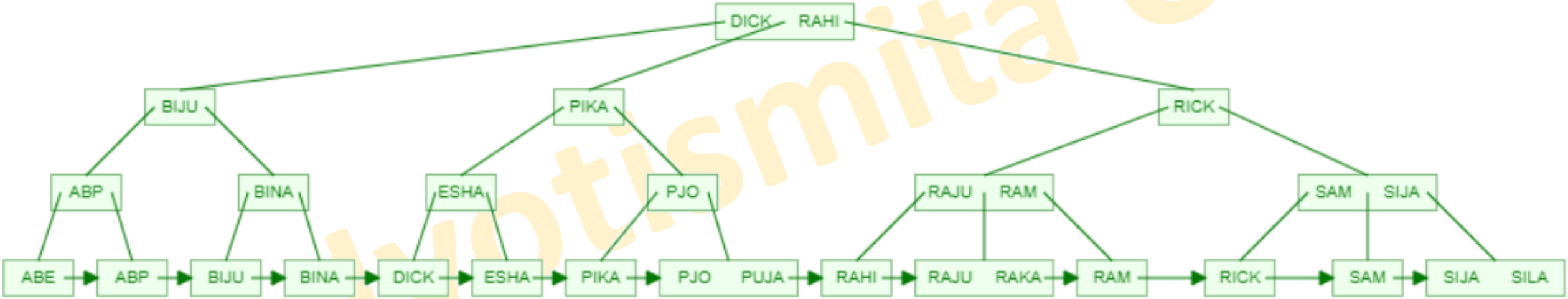
Insertion using B+ tree: Max degree = 3: Insert PIKA



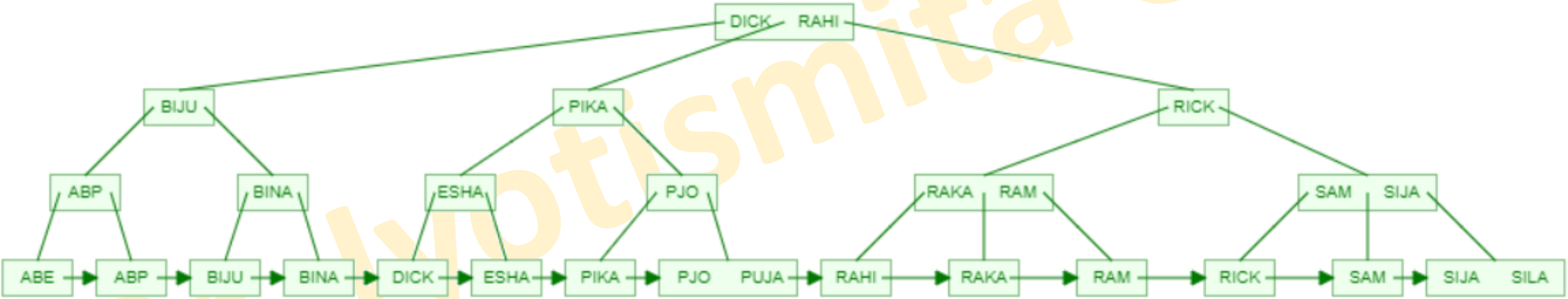
Insertion using B+ tree: Max degree = 3: Insert PJO



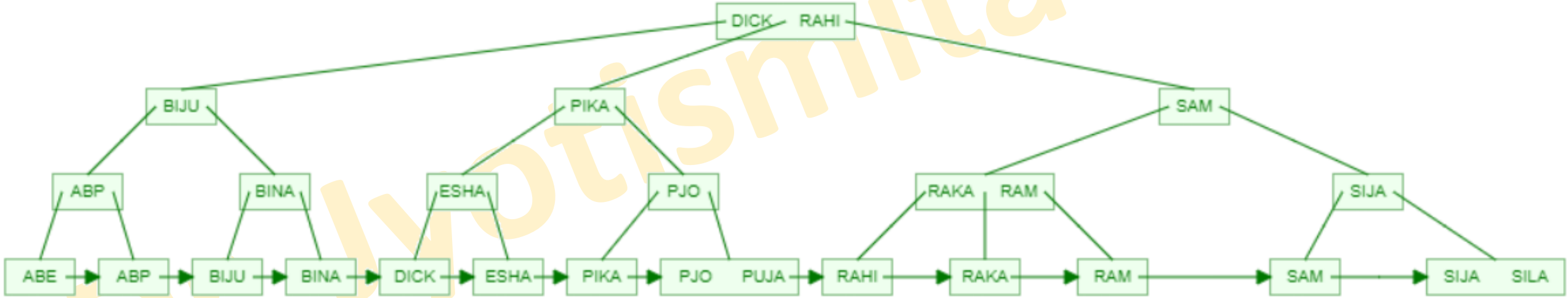
Deletion from B+ tree: Maximum degree = 3: Delete RINA



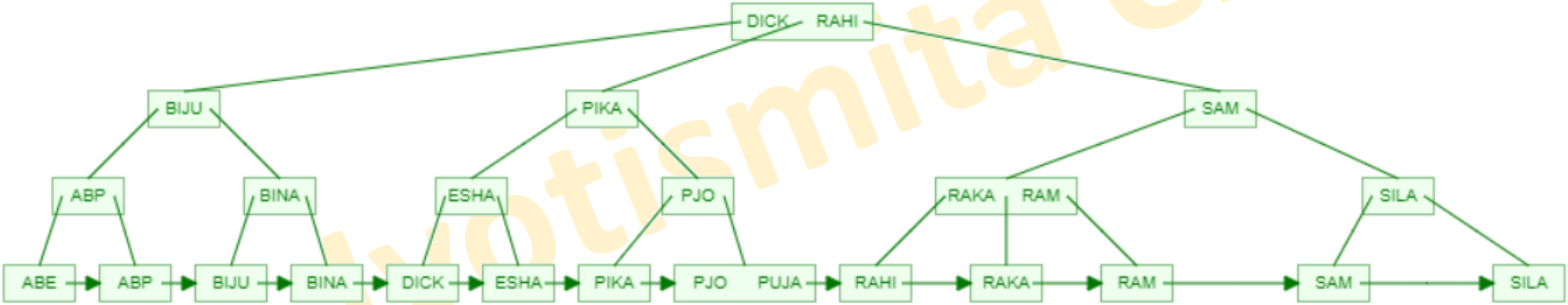
Deletion from B+ tree: Maximum degree = 3: Delete RAJU



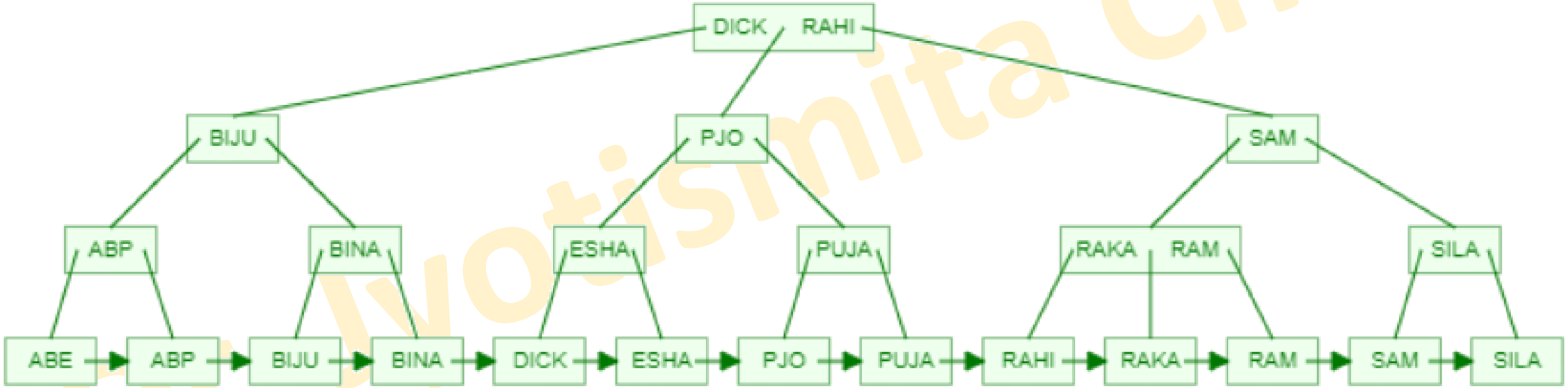
Deletion from B+ tree: Maximum degree = 3: Delete RICK



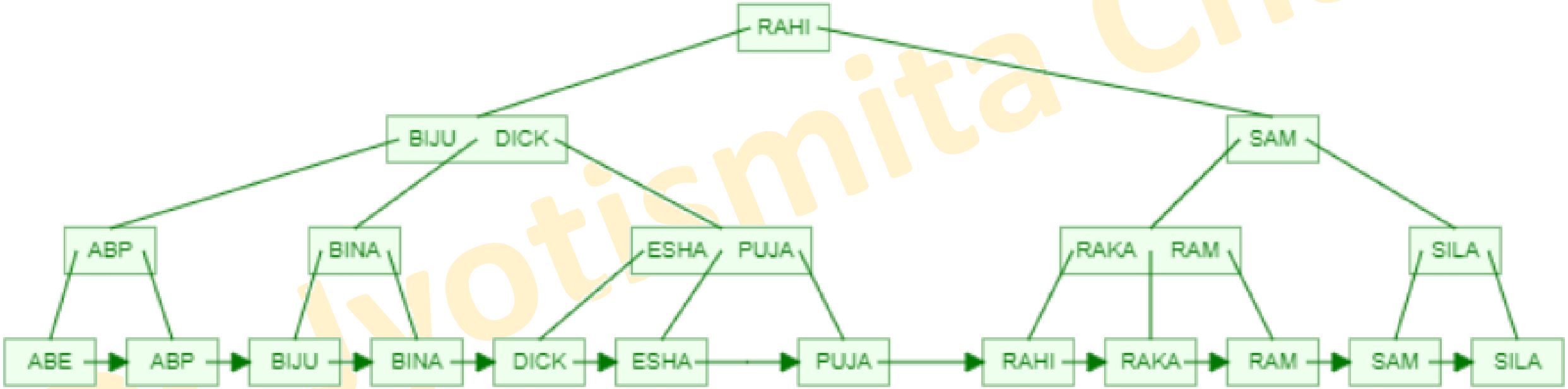
Deletion from B+ tree: Maximum degree = 3: Delete SIJA



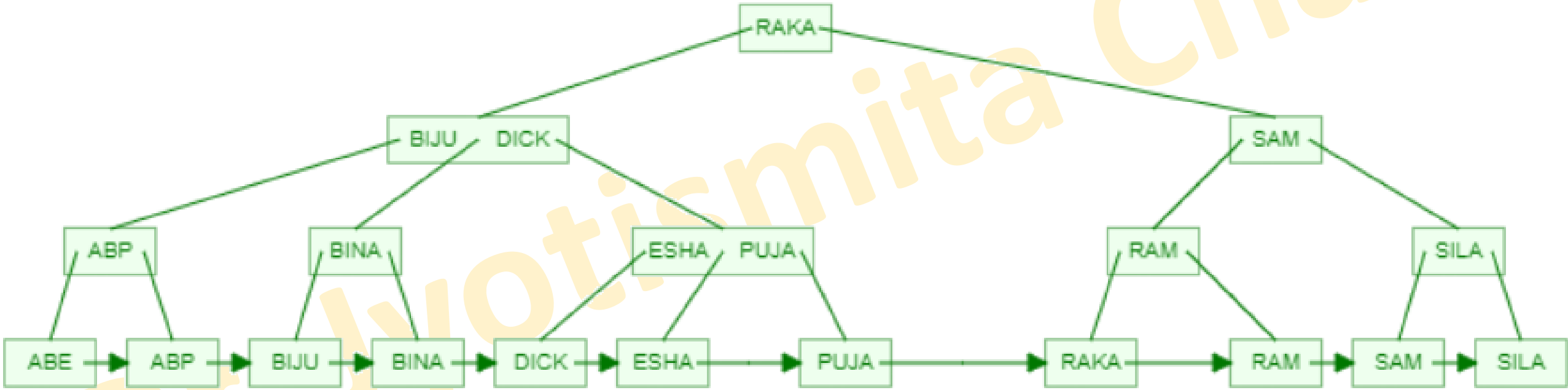
Deletion from B+ tree: Maximum degree = 3: Delete PIKA



Deletion from B+ tree: Maximum degree = 3: Delete PJO

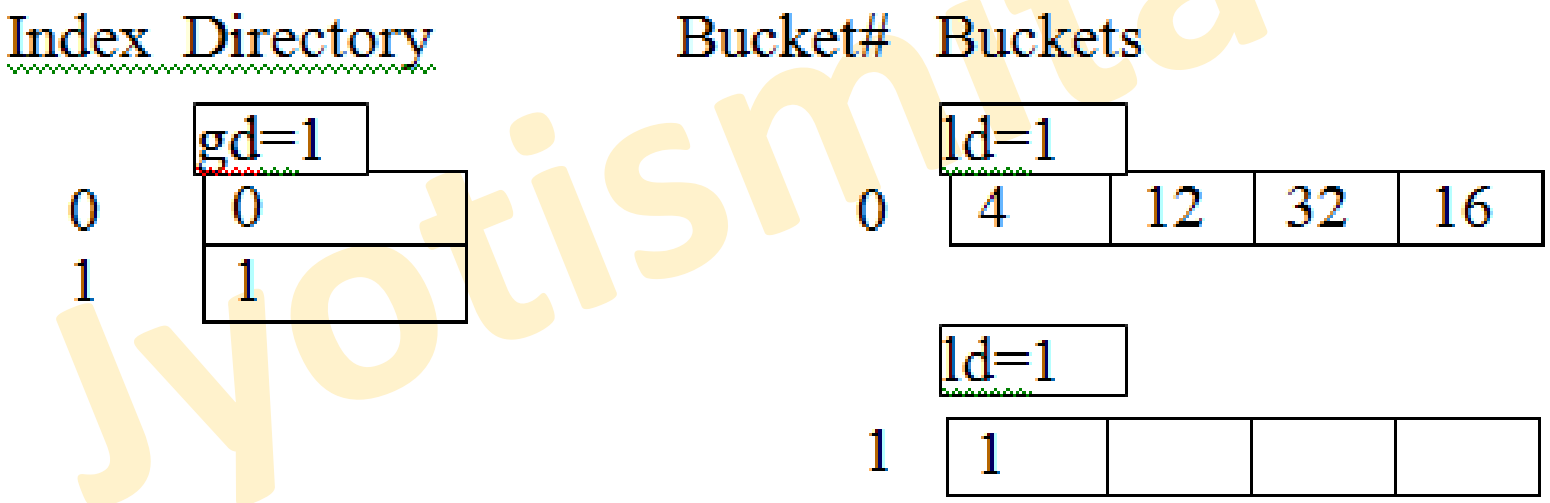


Deletion from B+ tree: Maximum degree = 3: Delete RAHI



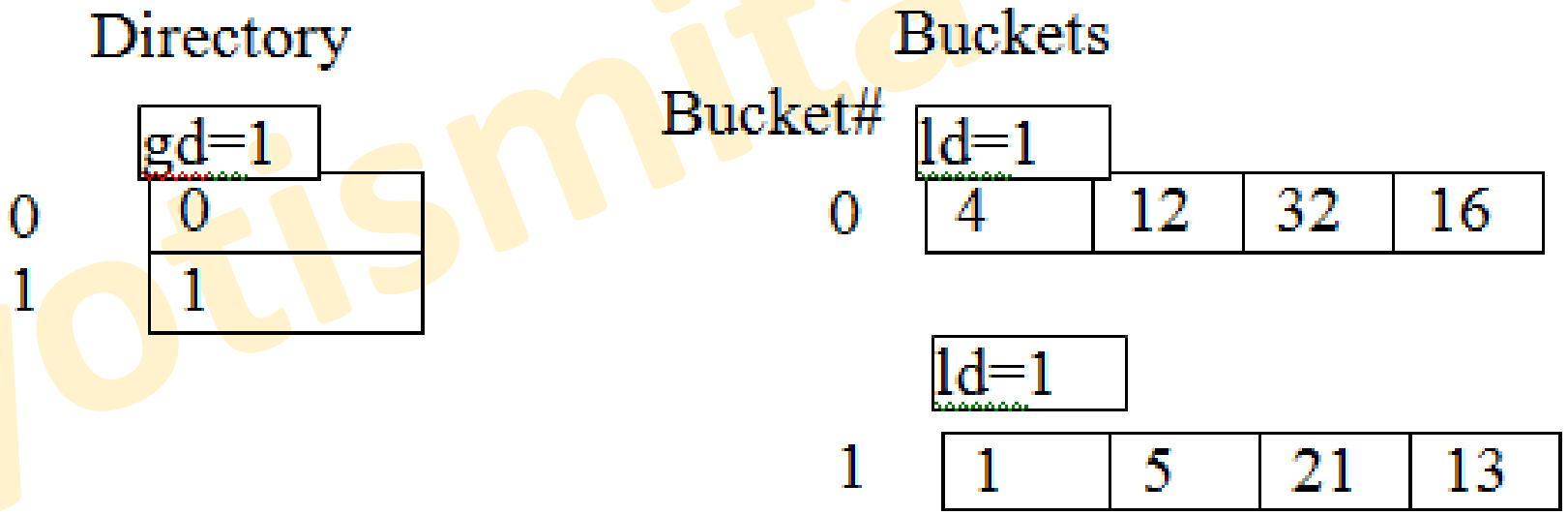
Insert using Extendible Dynamic Hashing

- Store five records with key values 1, 4, 12, 32 and 16.



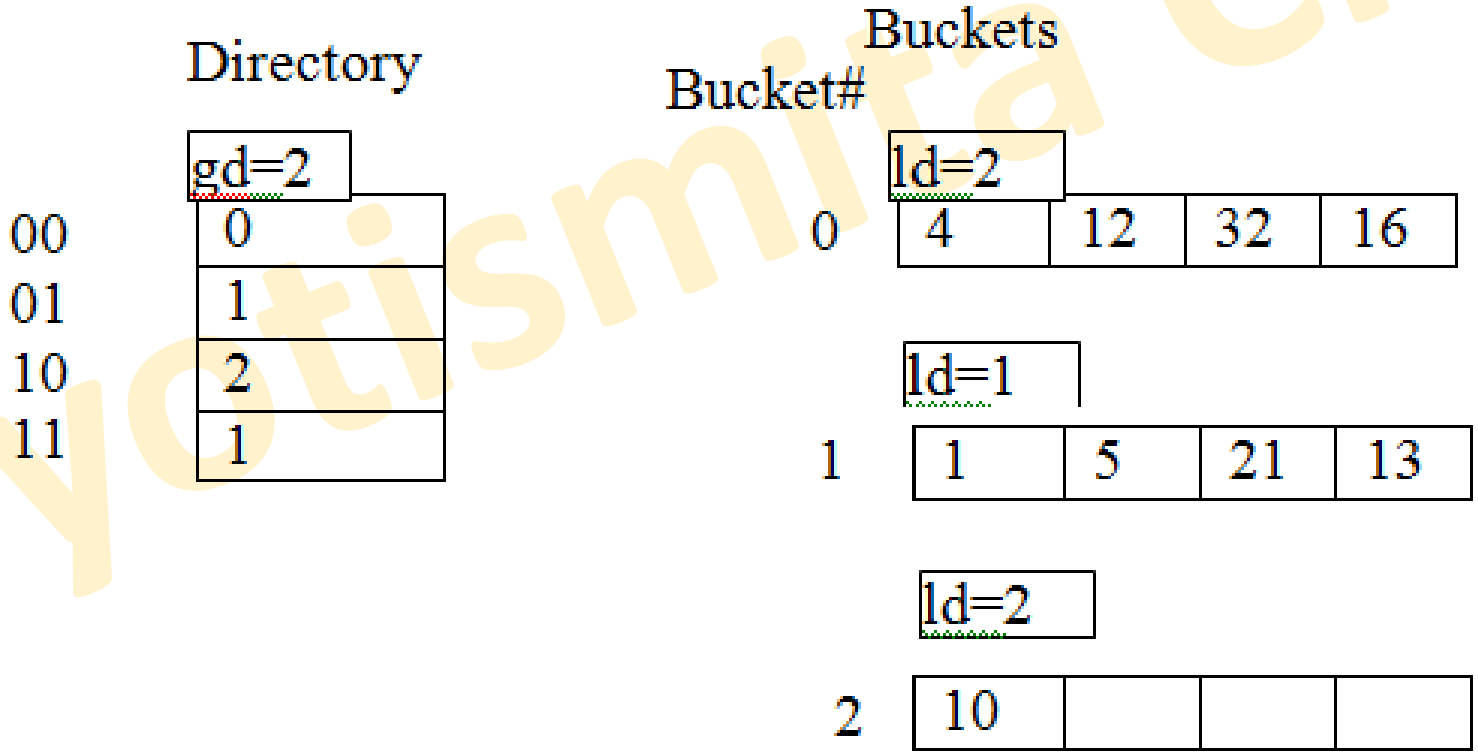
Insert using Extendible Dynamic Hashing

- Insert record with keys 5, 21 and 13



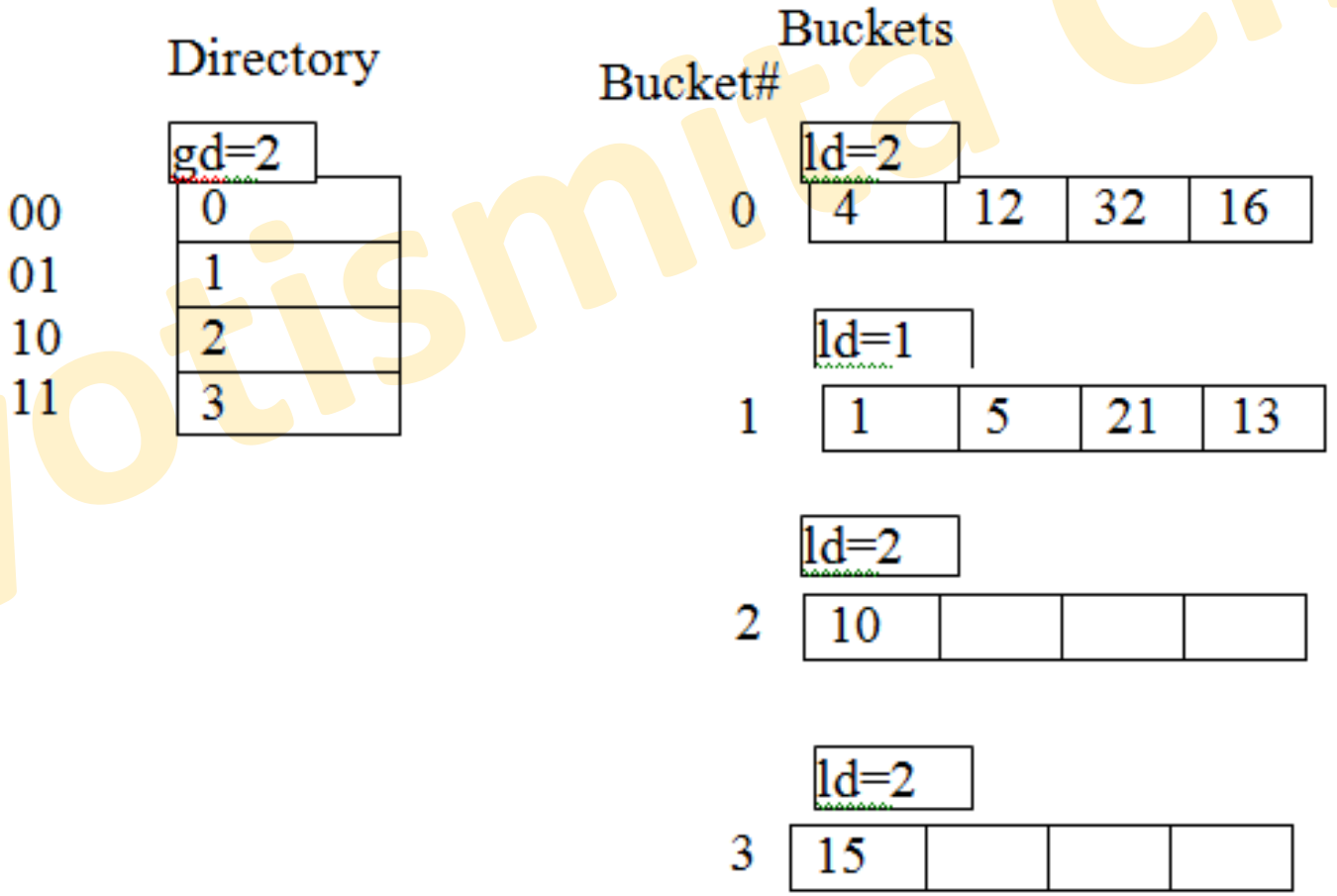
Insert using Extendible Dynamic Hashing

- Insert record with key 10



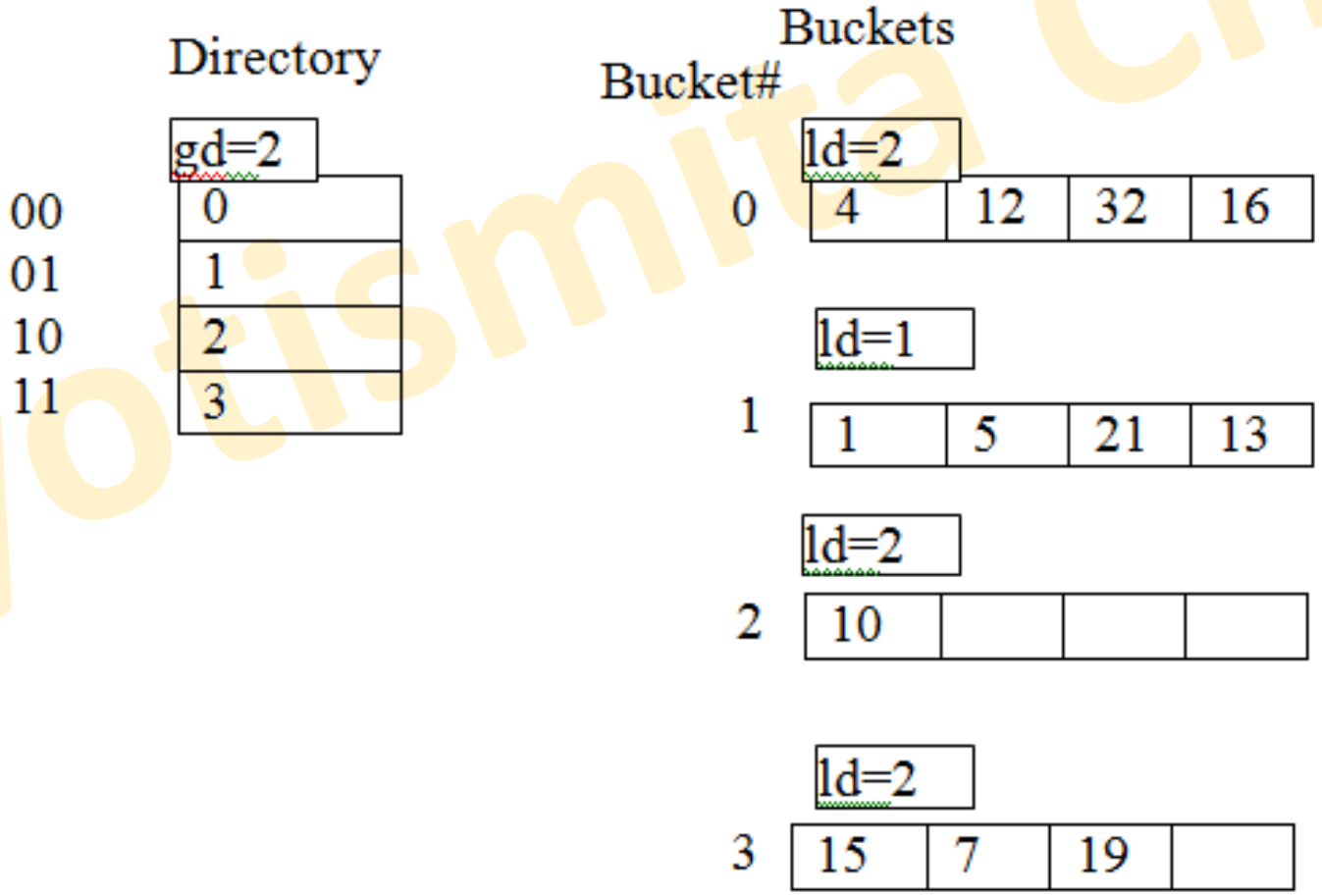
Insert using Extendible Dynamic Hashing

- Insert record with key 15



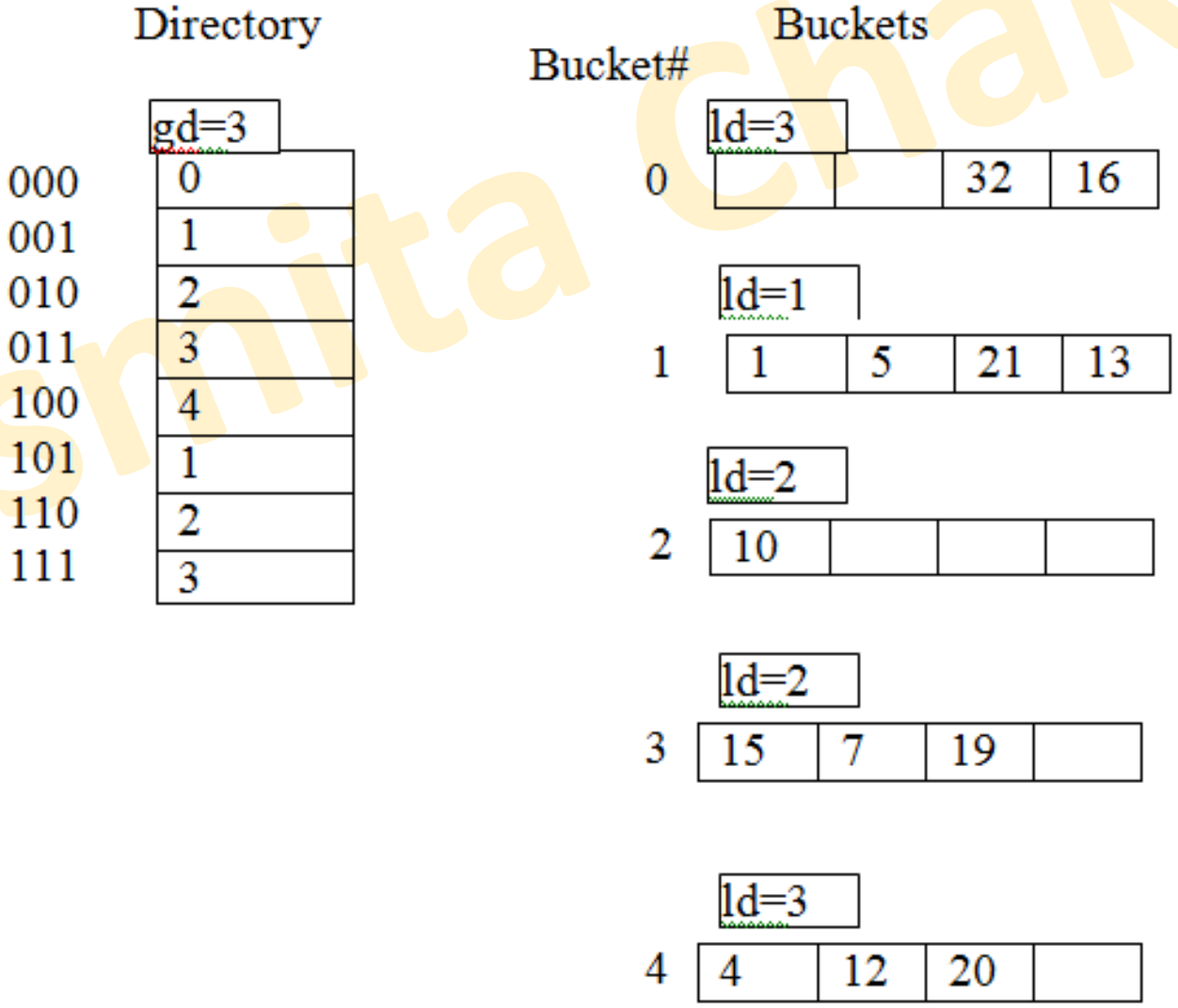
Insert using Extendible Dynamic Hashing

- Insert record with key 7 and 19



Insert using Extendible Dynamic Hashing

- Insert record with key 20



Insert using Extendible Dynamic Hashing

- Insert record with key 9