



UNDERSTANDING KEYS, CONSTRAINTS, AND INDEXING IN DATABASE

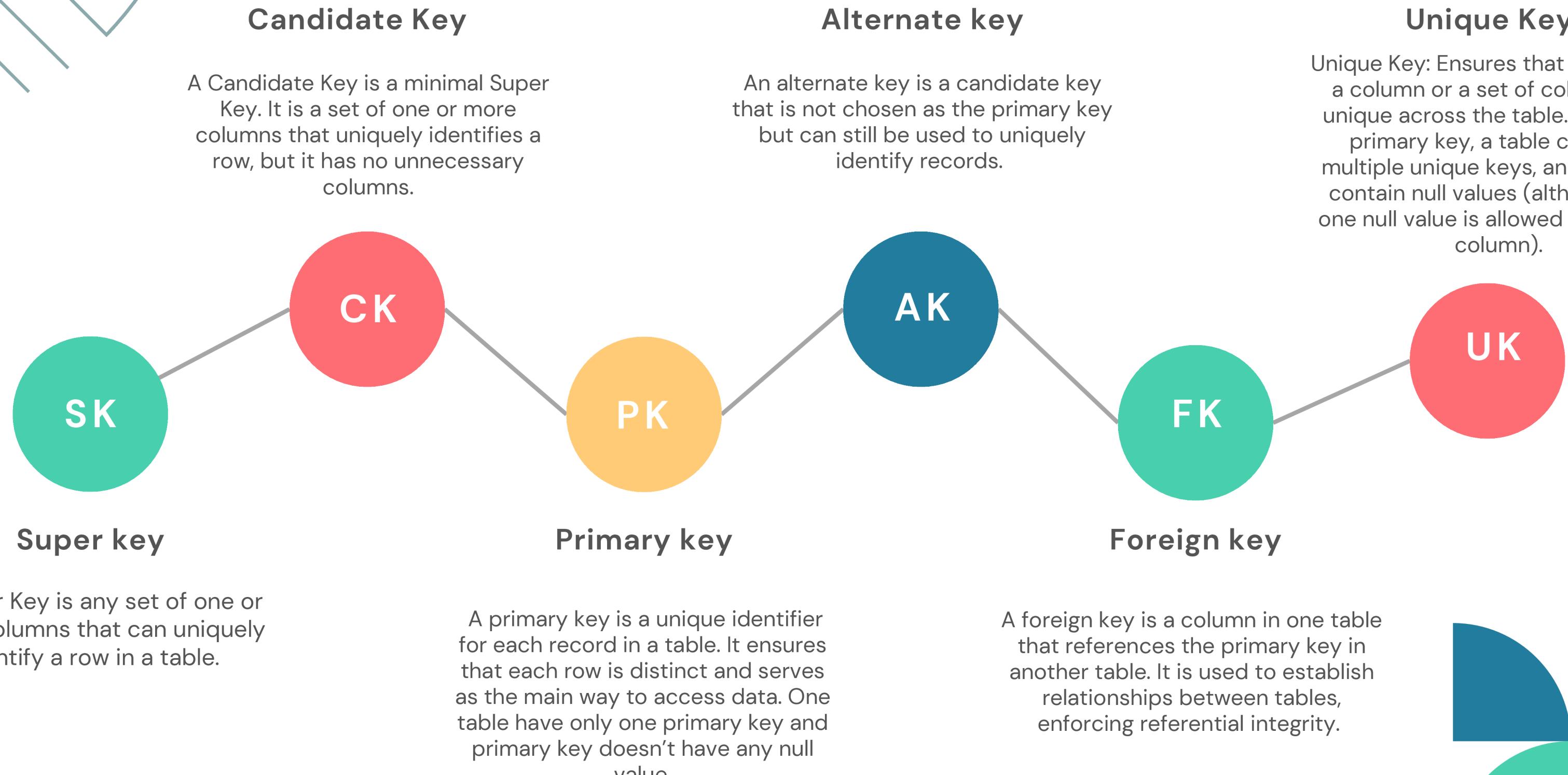
Majd Riyad

**"STOP DREAMING
AND START
DOING"**

KEYS

In a database, a **key** is an attribute or a group of attributes that **uniquely** identify each record in a table. **Keys** are needed for establishing and enforcing **relationships** between tables and ensuring **data integrity**.

TYPES OF KEYS



HOW TO CHOOSE PRIMARY KEY?

Employees

EmployeeID

Email

FirstName

LastName

SUPRE KEY

- {EmployeeID}
- {Email}
- {EmployeeID, Email}
- {EmployeeID, FirstName, LastName}...

CANDIDATE KEY

- {EmployeeID}
- {Email}

PRIMARY KEY

- {EmployeeID}

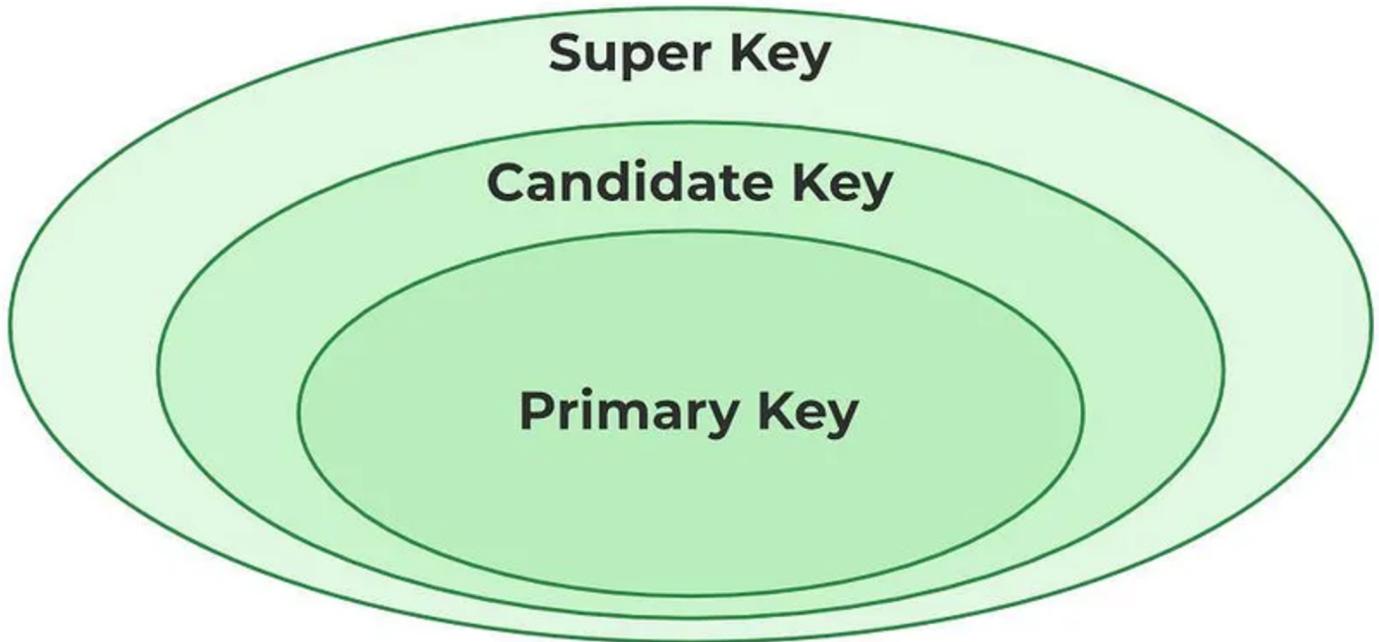


TABLE-1

Empld	EmpName	EmpLicence	EmpPassport	Dld
1001	Matt	LC1201	MA100LC	1
1002	Maxy	LC2078	XY100LC2	2
1003	Roy	LK00928	LK100RO	3

Primary Key

Candidate Key

Alternate Key

Super Key

Alternate Key

Unique Key

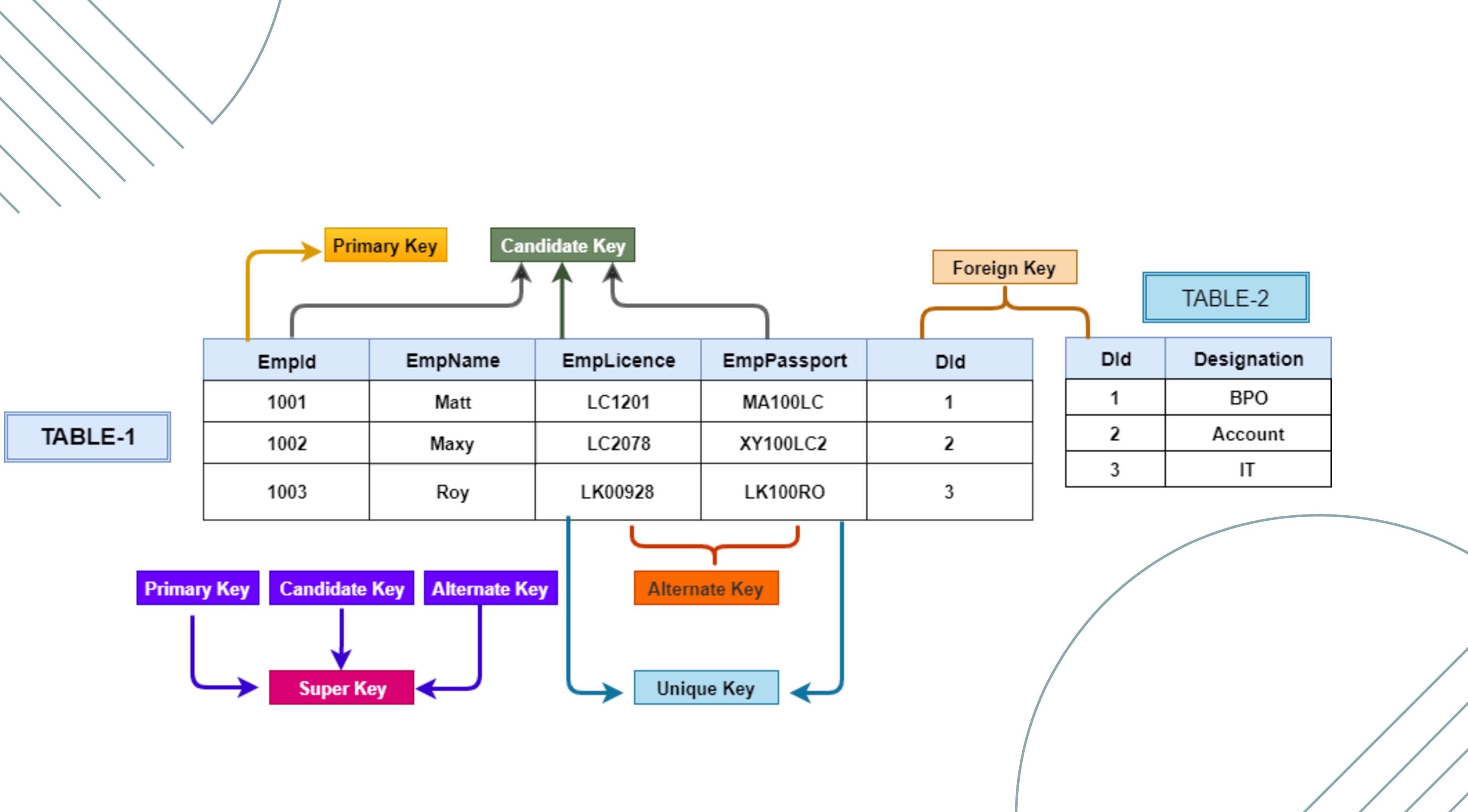


TABLE-2

Dld	Designation
1	BPO
2	Account
3	IT

**WHAT KEEPS EACH VALUE IN A COLUMN UNIQUE AND MAKES
SURE TABLES ARE RELATED TO EACH OTHER?**



INTEGRITY CONSTRAINTS

Integrity constraints are rules applied to database columns and tables to ensure data accuracy and consistency. These constraints help maintain the integrity of the database by preventing invalid data from being entered and by enforcing relationships between tables.

TYPES OF INTEGRITY CONSTRAINTS

01 - PRIMARY KEY

- Ensures each record in a table is uniquely identifiable.

02 - UNIQUE

- Ensures all values in a column or a set of columns are unique across the table

03 - CHECK

- Enforces specific conditions or rules on the values in a column.

04 - FOREIGN KEY

- Maintains referential integrity between two tables by ensuring that a value in one table matches a value in another table's primary key or unique key.

05 - NOT NULL

- Ensures all values in a column or a set of columns are unique across the table

06 - DEFAULT

- Provides a default value for a column if no value is specified during data insertion.

EXAMPLE OF CONSTRAINTS

```
4 CREATE TABLE Employees (
5     EmployeeID INT PRIMARY KEY, -- Entity Integrity
6     FirstName VARCHAR(50) NOT NULL, -- NOT NULL Constraint
7     LastName VARCHAR(50) NOT NULL,
8     Email VARCHAR(100) UNIQUE, -- Unique Constraint
9     Salary DECIMAL(10, 2) CHECK (Salary > 0), -- CHECK Constraint
10    DepartmentID INT,
11    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID) -- Referential Integrity
12 );
```

INDEXING



WAIT!!



Before seeing the indexing, we have to ask important question.

HOW TABLE DATA(ROWS) ARE ACTUALLY STORED?



id	name	address
12	Majd	44-55
44	Ieen	22-33-

THIS IS JUST A LOGIC REPRESENTATION

- Actual Data is not stored in this way

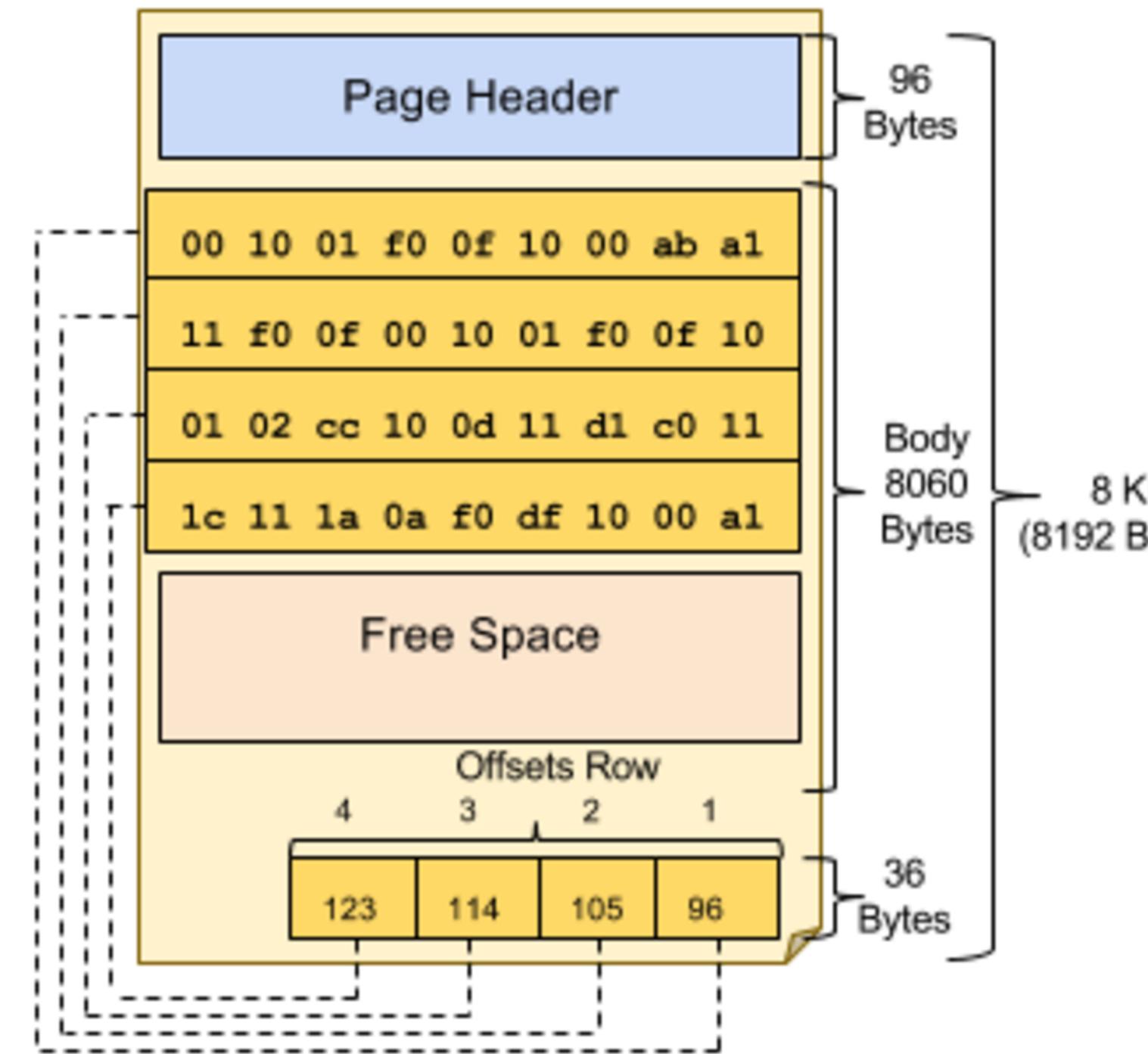
- DBMS creates Data Pages (generally its 8KB).
- Each Data Pages can store multiple table row in it.

DATA PAGES

- DBMS creates and manage these data pages. As for storing 1 table it can create many data pages.
- These data pages ultimately gets stored in **Data Block** physical memory like disk.

WHAT IS DATA BLOCK?

- Data Block is minimum amount of data which can be read/write by any I/O.
- DBMS is can not manage the Data Block ,it manage by storage system like Disk.
- Data Block size can range form 4kb to 32kbt (so based on the data block size, it can hold 1 or many Data page).



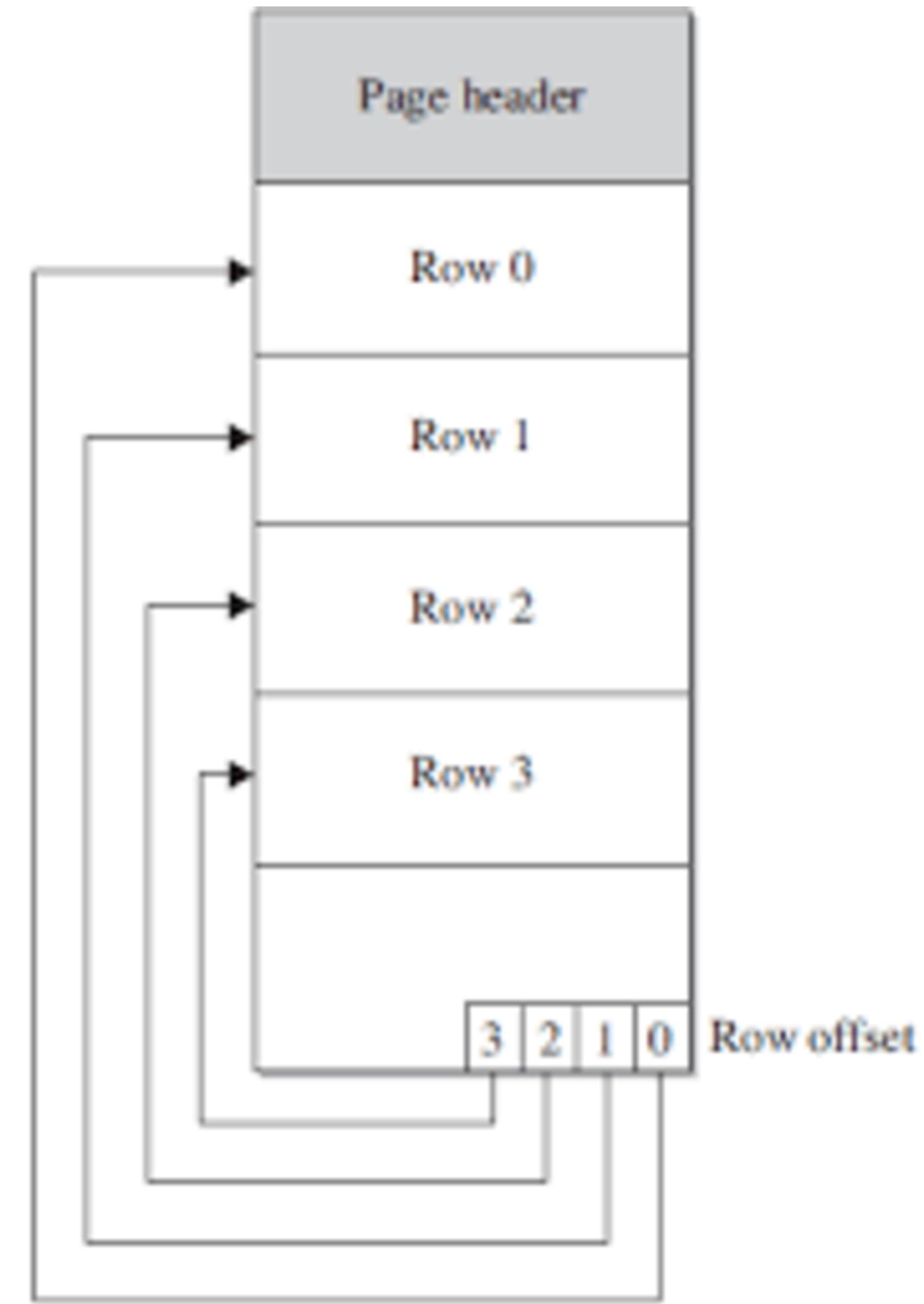
DBMS MAINTAINS THE MAPPING OF DATA PAGE AND DATA BLOCK

Data Page 1	Data Block 1
Data Page 2	Data Block 1
Data Page 3	Data Block 2
Data Page 4	Data Block 2

HOW DATA RETRIEVA WORK WITHOUT INDEXING?



- When a query is executed, the DBMS performs a full table scan to find the relevant rows.
- This involves reading every row in the table to check if it matches the query criteria.
- Full table scans can be very slow, especially for large tables.



INDEXING

Indexing in a database is a technique used to improve the speed of data retrieval operations. An index is a data structure that allows the database management system (DBMS) to quickly locate and access the rows in a table based on the values of one or more columns.

Search Key	Data Reference
------------	----------------

Structure of an index

TYPES OF INDEXING

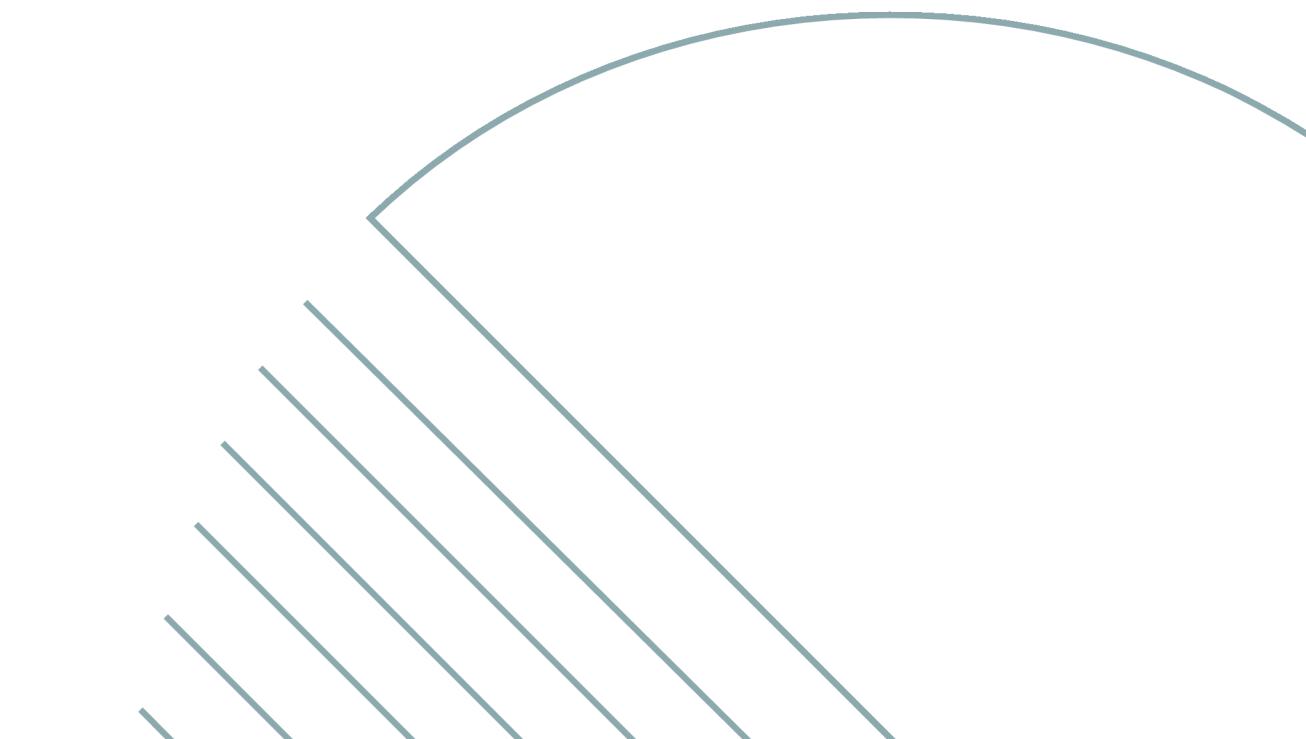
CLUSTERED

- A *clustered index determines the physical order of data in a table*
- *There can be only one clustered index per table.*

NON-CLUSTERED

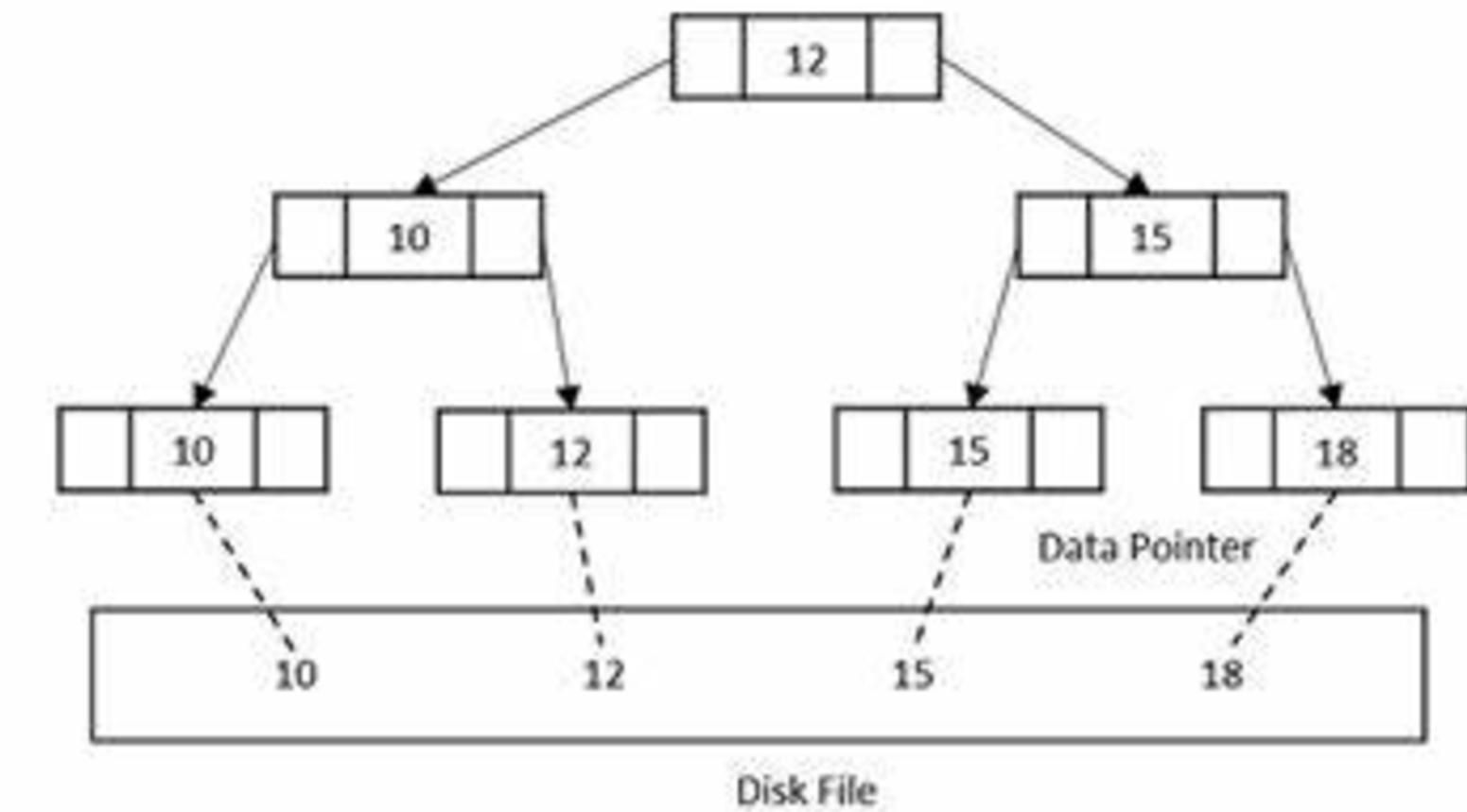
- *Non-clustered indexes do not determine the physical order of data in a table*
- *They store the indexed values and pointers to the actual data rows.*

DATA STRUCTUR USED FOR INDEXING



B+ TREE

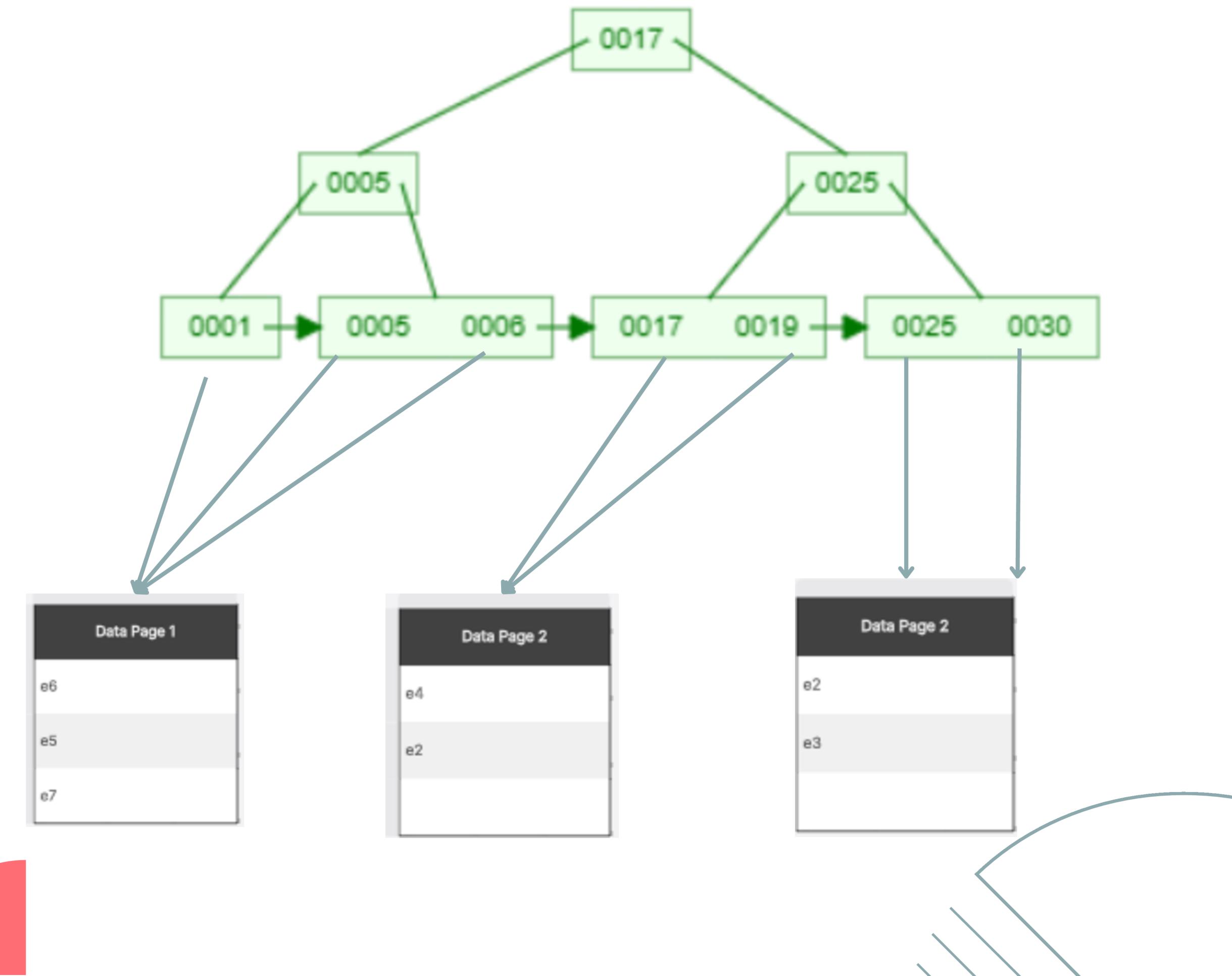
- DBMS uses B+ tree to manage its Data Page.
- Root node hold the value which used for faster searching the data.



- leaf node actually holds the indexed column value

EXAMPLE OF B+ TREE

Emp_id	Emp_name
19	e1
25	e2
30	e3
17	e4
6	e5
1	e6
5	e7



THANK YOU