Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

DATABASE DESIGN AND APPLICATIONS
ASSIGNMENT

PAYROLL MANAGEMENT SYSTEM

Submitted To
Prof Amit Dua
Birla Institute of Technology and Science

Submitted By
K Ravi Kumar Reddy
2020MT13010

Contents

Figures

1.Payroll Management System

Payroll Management System is a web-based application that enables a business to handle all their employee's financial records in a hassle-free automated fashion. A business's success heavily depends on two variables – Hiring the right team for the work and having a good system in place that helps managing the team efficiently. By having payroll management system in place, a business is essentially automating the micro administrative tasks that take the mental bandwidth to focus on the macro purposes.

1.1 Users of the System:

Different users that use this system are:

- **Admin:** The Admin user is the user that has access to all the data stores in the system, can create and manage all the financial data.

- **Employee:** The Employee user can access the payroll management system for his/her own financial information, tax computation and so on.

1.2 Benefits:

- **Ease of Operation:** Payroll function can be very cumbersome. If the software is intuitive, it reduces the need for software training and guidance. Payroll management system is comprehensive but would have straightforward workflows making it easy to compute financial data.

- **Scalability:** As organization size increases your software also need to serve you appropriately. The limitation can be in terms of employees' data it processes or in terms of the availability of features like leave and attendance management, reimbursement model, etc. Payroll Management system opts for advanced features at a reasonable price without much difficulty.

- **Employee Self-Service:** One of the primary payroll input providers is the employee. He provides information such as income tax saving investment declaration, type of flexible benefit opted, etc. The interaction between the payroll officer and employee is usually very event based. Hence Payroll Management System can reduce the manual intervention and automate payroll data collection for accurate tax computation.

- **Data Safety:** The on-premise systems have their limitations like threat of data loss by fire, flood, etc. Also, the setup cost of these systems is huge and comes with an initial operation limit. Often the annual maintenance cost is quite high. Whereas, the cloud solutions are built on such technology, so you can always opt for an upgraded plan that supports higher level of operations at any time. Since cloud solutions have data center at multiple locations, even during incidences of fire, flood, etc. your data remains safe.

- **Data Access:** If you use a cloud software, you do not need to be present in your office to be able to access your payroll data and employee data. You can login from anywhere at any time.

- **Improve the morale of employees**: If your employees feel like their job is not stable and they are not being paid on time, they might start to look elsewhere. A Payroll Management system will ensure that all last-minute glitches and manual errors are taken care of so that your employees' morale is always high.

- **Eliminates Payroll Errors:** No one notices payroll unless something is wrong, and then it's stressful and time-consuming to address any errors that arise. Even without errors, in-house payroll can take up a lot of HR's time and efforts. Using a Payroll Management System can free up time and ease stress by automating the process.

1.3Benefits:

- **Employee:**
  - Add and Update the personal information.
  - Declare the investments.
  - Obtain Pay slips on time.
  - Apply for reimbursements.
  - See the status of the reimbursement
- **Administrator:**
  - Pay slips' creation.
  - Process the reimbursements.
  - Verify the investments.

2. ER Diagram

In the Task Management System, we have following entities:

1. payslip
2. department
3. organization
4. reimbursement
5. investment
6. employee
7. administrator

The attributes corresponding each entity mentioned above are enlisted below:

1. payslip:
   a. id – Primary Key
   b. base_pay
   c. hra
   d. incentive

2. department:
   a. dept_id – Primary Key
   b. dept_name
   c. location

3. organization
   a. org_id – Primary Key
   b. org_name

4. reimbursement
   a. reimb_id – Primary Key
   b. reimb_type
   c. reimb_amount
   d. reimb_state

5. investment
   a. inv_id – Primary Key
   b. inv_amount

   c. inv_type
   d. inv_section

6. employee
   a. emp_id – Primary Key
   b. emp_name
   c. email_id
   d. phone_no
   e. emp_pan
   f. emp_dob
7. administrator:
   a. admin_id – Primary Key
   b. admin_name

The Enhanced Entity Relationship Diagram is a visual representation of data that help in designing databased with high-level models. The EER diagram for Payroll Management System is as shown below:

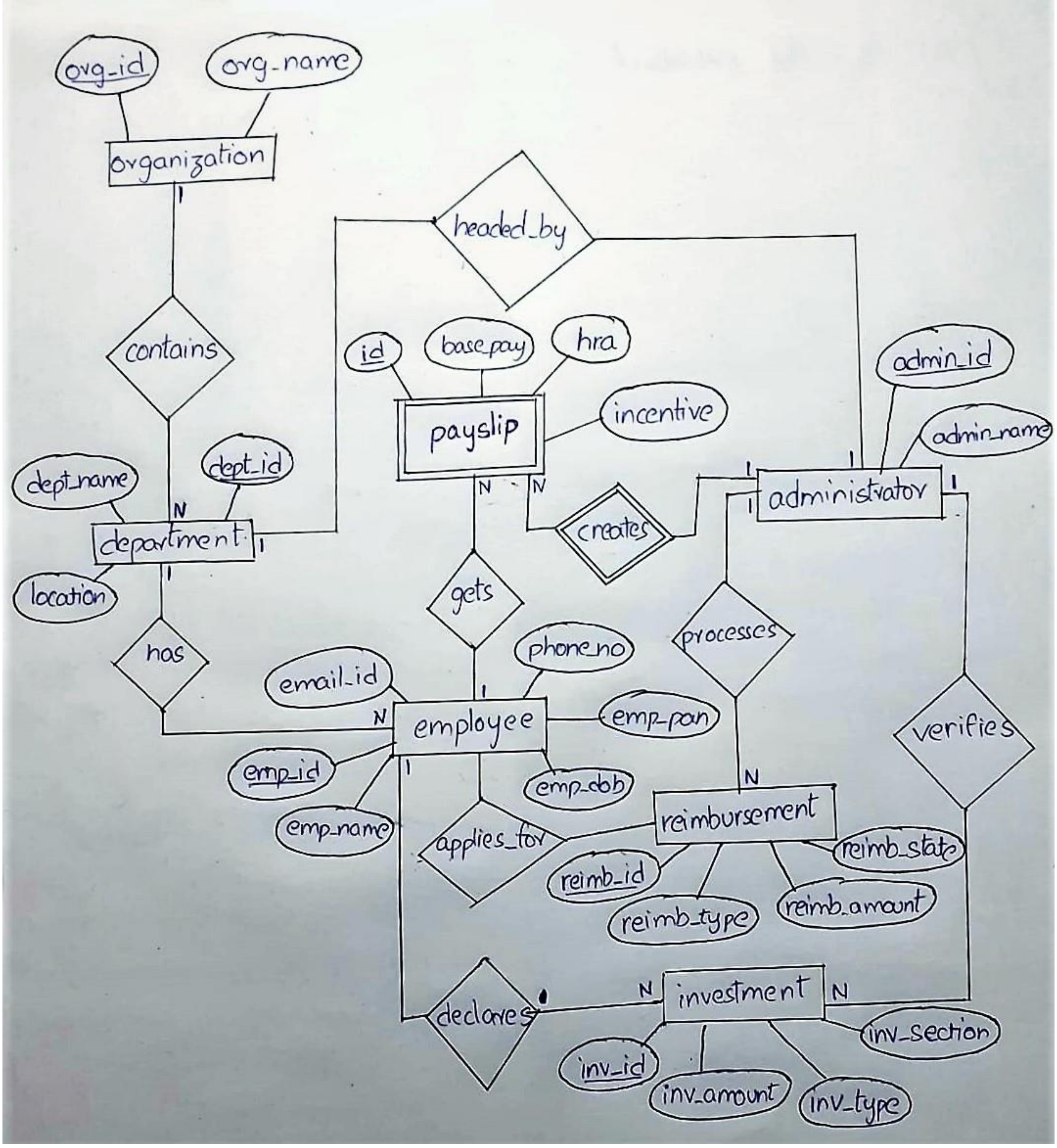


Figure-1: ER Diagram

3. ER to Relational Mapping

The relational model represents the database as a collection of relations. Various Relations between different entities
of Payroll Management system are illustrated in the relational model below:

1. An administrator creates multiple Pay slips for several employees. (1 to many relationship)
2. An employee gets multiple Pay slips over a time period (1 to many relationship)
3. An employee applies for several reimbursement claims (1 to many relationship)
4. An administrator processes several reimbursements of various employees (1 to many relationship)
5. An employee declares different investment claims over a time period (1 to many relationship).
6. An administrator verifies different investment claims of various employees (1 to many relationship)
7. Every department has several employees (1 to many relationship)
8. Every department is headed by one administrator (1 to 1 relationship)
9. Every organization contains several departments (1 to many relationship)

All Primary Keys have been underline in the model.

These relations are illustrated in the Relation model shown below:

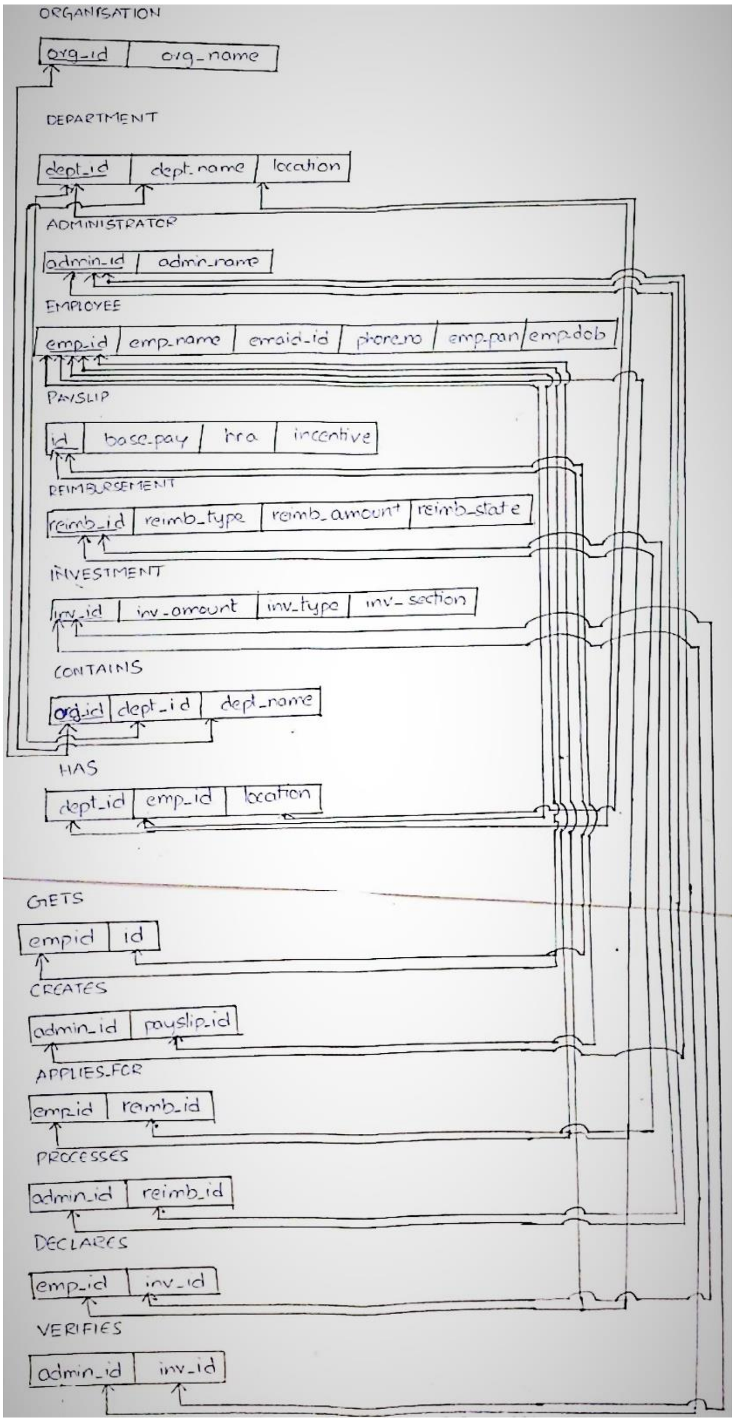

ORGANISATION
org_id | org_name
DEPARTMENT
dept_id | dept.name | location
ADMINISTRATOR
admin_id | admin_name
EMPLOYEE
emp_id | emp_name | email_id | phone_no | emp_pan | emp_dob
PAYSLIP
id | base_pay | hra | incentive
REIMBURSEMENT
reimb_id | reimb_type | reimb_amount | reimb_state
INVESTMENT
inv_id | inv_amount | inv_type | inv_section
CONTAINS
org_id | dept_id | dept_name
HAS
dept_id | emp_id | location
GETS
empid | id
CREATES
admin_id | payslip_id
APPLIES_FOR
emp_id | reimb_id
PROCESSES
admin_id | reimb_id
DECLARES
emp_id | inv_id
VERIFIES
admin_id | inv_id

Figure-2: ER to Relational Mapping

4. Normalization

1NF:

Relations which are already in 1NF:

- organization (id, org_name)
- department (dept id, dept_name, location)
- administrator (admin id, admin_name)
- payslip (base pay, hra, incentive)
- reimbursement (reimb id, reimb_type, reimb_amount, reimb_state)
- investment (inv id, inv_amount, inv_type, inv_section)
- contains (id, dept id, dept_name)
- has (dept id, emp id, location)
- gets (emp id)
- creates (admin id)

Relations that can be converted to 1NF:

- employee (emp id, emp_name, email_id, DoB, PAN_no, phone_no)
- applies_for (emp id, reimb id)
- processes (admin id, reimb_id)
- declares (emp id, inv id)
- verifies (admin id, inv id)

Example: employee (emp id, emp_name, email_id, DoB, PAN_no, phone_no):
(Without 1NF)

emp id	emp_name	email	DoB	PAN_no	phone_no
Emp-1	Rama	rama@gmail.com professor1@gmail.com	17/01/1994	ABCD1234	123456789
Emp-2	Krishna	krishna@gmail.com	17/02/1994	EFGH65678	987654321 456754379

As per 1NF rule, there should not be any multi-valued attribute in the relation. employee (emp id, emp_name, email_id, DoB, PAN_no, phone_no): **(With 1NF)**

emp id	emp_name	email	DoB	PAN_no	phone_no
Emp-1	Rama	rama@gmail.com	17/01/1994	ABCD1234	123456789
Emp-1	Rama	professor1@gmail.com	17/01/1994	ABCD1234	123456789
Emp-2	Krishna	krishna@gmail.com	17/02/1994	EFGH65678	987654321
Emp-2	Krishna	krishna@gmail.com	17/02/1994	EFGH65678	456754379

2NF:

Relations which are already in 2NF:

- organization (id, org_name)
- department (dept id, dept_name, location)
- administrator (admin id, admin_name)
- employee (emp id, emp_name, email_id, DoB, PAN_no, phone_no)
- payslip (base pay, hra, incentive)
- reimbursement (reimb id, reimb_type, reimb_amount, reimb_state)
- investment (inv id, inv_amount, inv_type, inv_section)
- gets (emp id)
- creates (admin id)
- applies_for (emp id, reimb id)
- processes (admin id, reimb_id)
- declares (emp id, inv id)
- verifies (admin id, inv id)

Relations that can be converted to 1NF:

- contains (id, dept id, dept_name)
- has (dept id, emp id, location)

Example:

has (dept id, emp id, location): **(Without 2NF)**

dept id	emp id	location

Dept-1	Emp-1	Bangalore
Dept-1	Emp-2	Bangalore
Dept-2	Emp-3	Hyderabad

has (dept id, emp id, location): **(With 2NF)**

As per 2NF rule, the above relation will be divided in two relations like below.

Table: has (dept id, emp id)

dept id	emp id
Dept-1	Emp-1
Dept-1	Emp-2
Dept-2	Emp-3

Table: dept_location (dept id, location)

dept id	location
Dept-1	Bangalore
Dept-1	Hyderabad

3NF:

None of the relations above have the transitive dependency, so they are already in 3NF form.

5. SQL Queries

1. **Create an account for new joinee.**
   - INSERT INTO employee (emp_id, emp_name, email, DoB, PAN_no, phone_no) VALUES ('Emp-3', 'Akash', 'akash@gmail.com', '17/03/1994', 'IJKL9012', 2345678912);

2. **Display all reimbursements applications and respective state by employee whose name is 'Rama'.**
   - SELECT reimb_type,reimb_state FROM reimbursement WHERE reimb_id=(SELECT reimb_id FROM applies_for WHERE emp_id=(SELECT emp_id FROM employee WHERE emp_name='Rama'));

3. **Display the work location of an employee whose name is "Krishna"**
   - SELECT location FROM department WHERE dept_id=(SELECT dept_id FROM has WHERE emp_id=(SELECT emp_id FROM employee WHERE emp_name=' Raquel Murillo'));

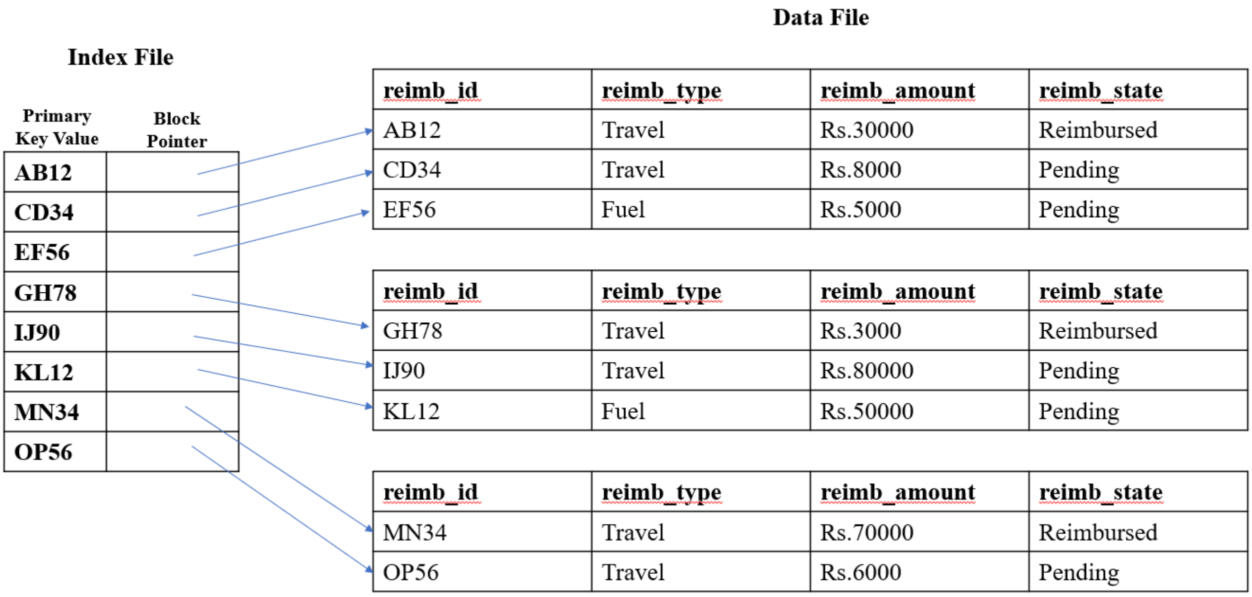
4. **Update the state of all the reimbursement with type 'travel' to 'Completed'**
   - UPDATE reimbursement SET reimb_type='travel' WHERE reimb_state='Completed';

5. **Delete the entry for the employee 'Krishna'**
   - DELETE FROM employee WHERE emp_name= 'Krishna'

6. Design Indices

Indexes are used to speed up the retrieval of records in response to search conditions. A primary index is an ordered file whose records are of fixed length and it acts like an access structure to efficiently search for and access the data records in a data file.

Primary Indexing for Payroll Management system's Reimbursement data:



Index File

Primary Key Value	Block Pointer
AB12	
CD34	
EF56	
GH78	
IJ90	
KL12	
MN34	
OP56	

Data File

reimb_id	reimb_type	reimb_amount	reimb_state
AB12	Travel	Rs.30000	Reimbursed
CD34	Travel	Rs.8000	Pending
EF56	Fuel	Rs.5000	Pending

reimb_id	reimb_type	reimb_amount	reimb_state
GH78	Travel	Rs.3000	Reimbursed
IJ90	Travel	Rs.80000	Pending
KL12	Fuel	Rs.50000	Pending

reimb_id	reimb_type	reimb_amount	reimb_state
MN34	Travel	Rs.70000	Reimbursed
OP56	Travel	Rs.6000	Pending

B+ Trees: A **B+ Tree** is primarily utilized for implementing dynamic indexing on multiple levels. Compared to B- Tree, the B+ Tree stores the data pointers only at the leaf nodes of the Tree, which makes search more process more accurate and faster.

Rules for B+ Tree

- Leaves are used to store data records.
- It stored in the internal nodes of the Tree.
- If a target key value is less than the internal node, then the point just to its left side is followed.

- If a target key value is greater than or equal to the internal node, then the point just to its right side is followed.
- The root has a minimum of two children

Suppose we insert 8 values into **reimbursement** table based on the values as shown in the data file above, the B+Tree in increasing order of **reimb_amount** would look as follows:

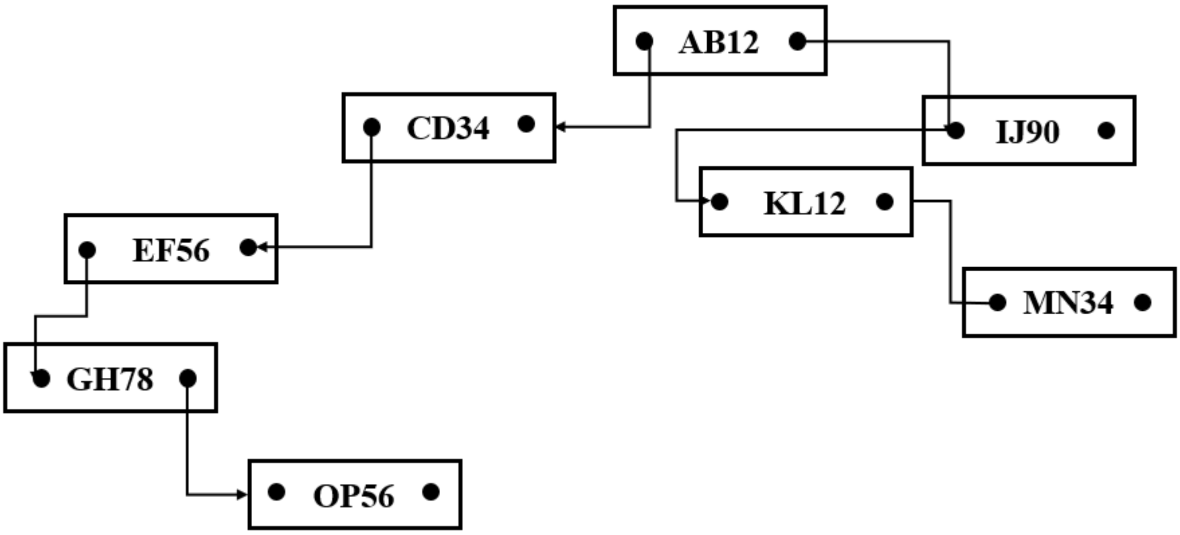


Figure-3: B+ Tree

7. References

1. Ramez Elmasri & Shamkant B. Navathe, Database Systems; Models, Languages, Design and Application Programming, Pearson Education, 7th Edition, 2017.
2. Abraham Silberschatz, Henry F Korth and S Sudarshan, Database System Concepts, McGraw Hill, 6th Ed., 2013.