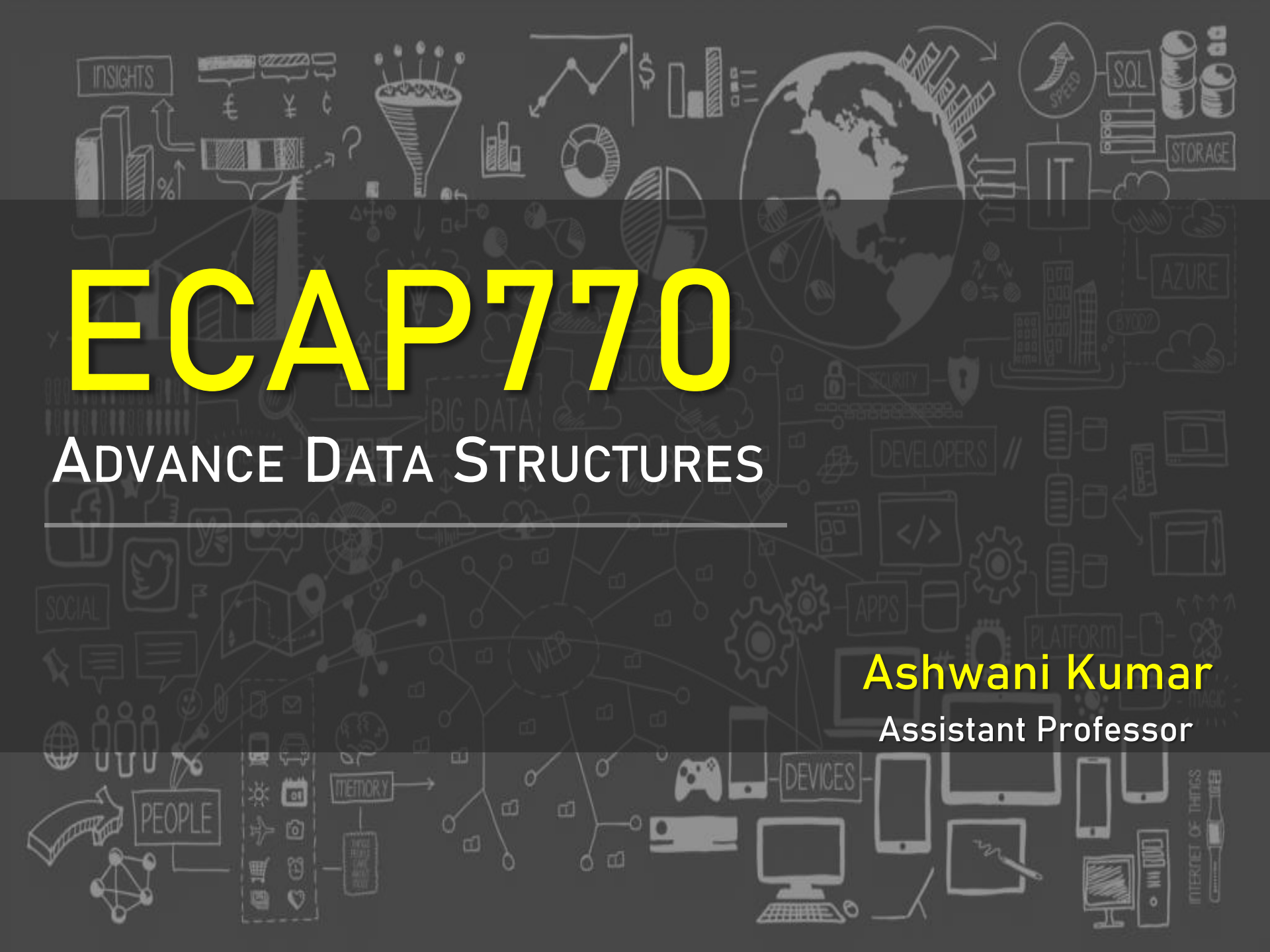


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Operations on linked List
- Insertion linked list
- Deletion in linked list
- Searching in linked list

Operations on linked List

List of operations are:

- Traversal - access each element of the linked list
- Insertion - adds a new element to the linked list
- Deletion - removes the existing elements
- Search - find a node in the linked list

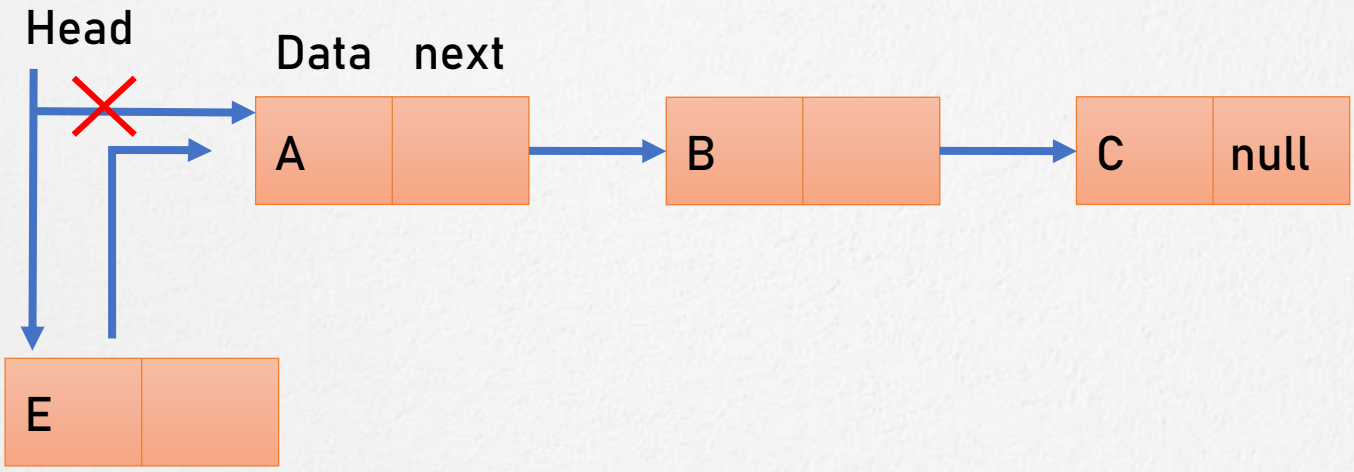
Insertion in linked list

- Insertion in Linked List can happen at following places:
- At the beginning of the linked list.
- At the end of the linked list.
- At a given position in the linked list.

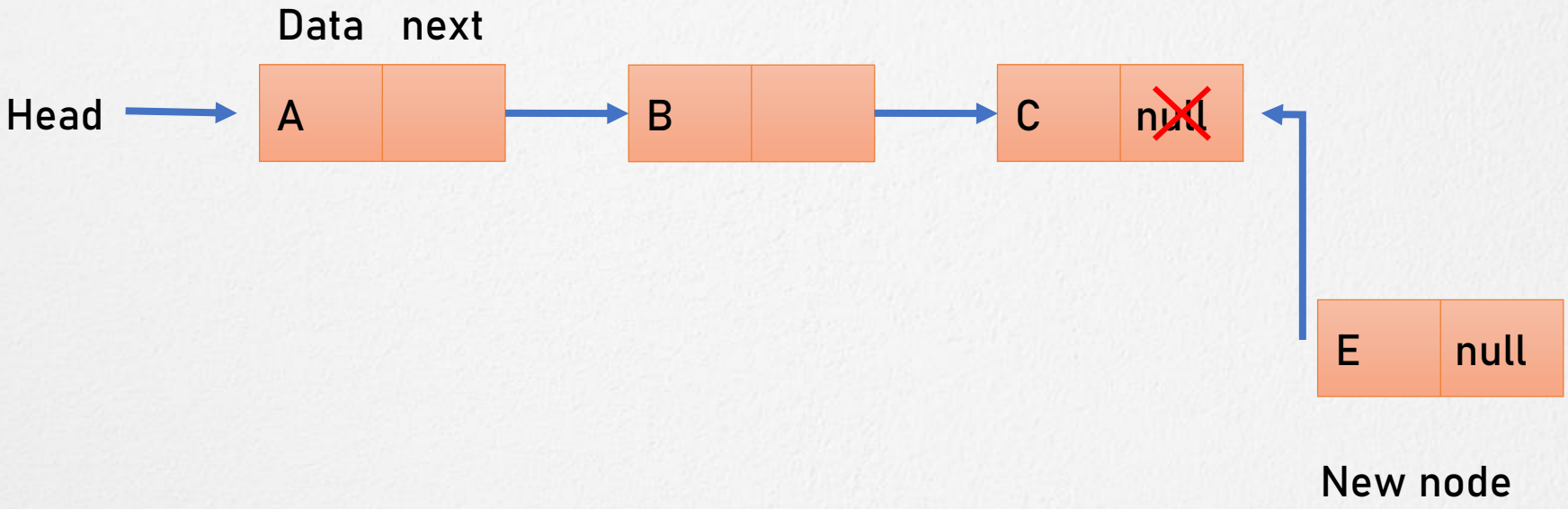
Insertion in linked list

Sr. No	Operation	Description
1	Insertion at beginning	It involves inserting any element at the beginning of the list. We just need to a few link adjustments to make the new node as the head of the list.
2	Insertion at end of the list	It involves insertion at the last of the linked list. The new node can be inserted as the only node in the list or it can be inserted as the last one.
3	Insertion after specified node	It involves insertion after the specified node of the linked list. We need to skip the desired number of nodes in order to reach the node after which the new node will be inserted. .

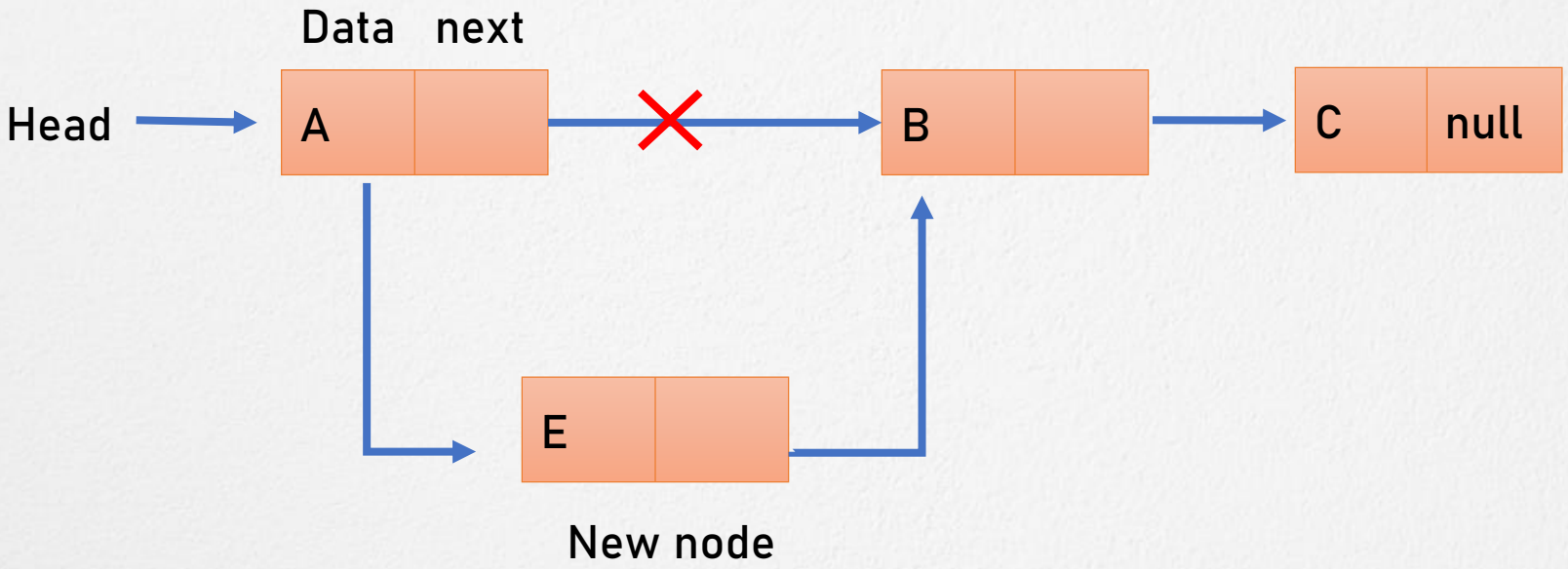
Insertion at beginning in linked list



Insertion at end in linked list



Insertion at given position in linked list



Algorithm : Insertion at beginning

- Step 1: IF PTR = NULL
- Write OVERFLOW
- Go to Step 7
- [END OF IF]
- Step 2: SET NEW_NODE = PTR
- Step 3: SET PTR = PTR → NEXT

Algorithm : Insertion at beginning

- Step 4: SET NEW_NODE → DATA = VAL
- Step 5: SET NEW_NODE → NEXT = HEAD
- Step 6: SET HEAD = NEW_NODE
- Step 7: EXIT

Deletion from a Linked List

Delete from beginning

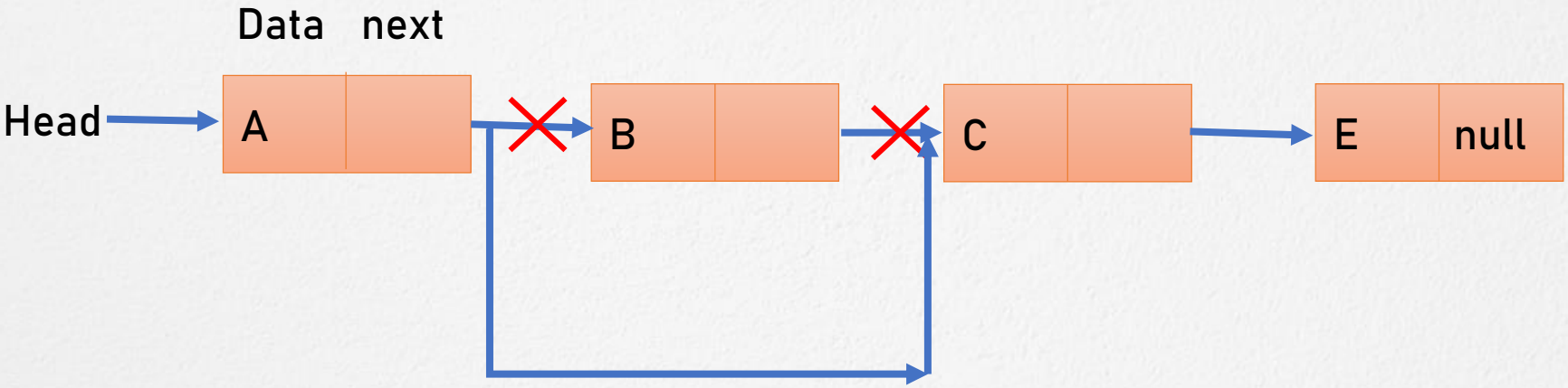
Delete from end

Delete from middle/
given position

Deletion from a Linked List

- Find the previous node of the node to be deleted.
- Change the next pointer of the previous node
- Free the memory of the deleted node.
- In case of first node deletion, we need to update the head of the linked list.

Deletion in linked list



Algorithm: Deletion at beginning

- Step 1: IF HEAD = NULL
- Write UNDERFLOW
- Go to Step 5
- [END OF IF]
- Step 2: SET PTR = HEAD
- Step 3: SET HEAD = HEAD -> NEXT
- Step 4: FREE PTR
- Step 5: EXIT

Searching in linked list

- Searching is performed to find the location of a particular element in the list.
- Traversing is performed in the list and make the comparison of every element of the list with the specified element.
- If the element is matched with any of the list element then the location of the element is returned from the function.

Algorithm: Searching in linked list

Step 1: SET PTR = HEAD

Step 2: Set I = 0

STEP 3: IF PTR = NULL

WRITE "EMPTY LIST"

GOTO STEP 8

END OF IF

STEP 4: REPEAT STEP 5 TO 7

UNTIL PTR != NULL

STEP 5: if ptr → data = item

write i+1

End of IF

STEP 6: I = I + 1

STEP 7: PTR = PTR → NEXT

[END OF LOOP]

STEP 8: EXIT

That's all for now...