

## HW 4 Nawras Rawas Qalaji

- 1.)
  - (1.) Index cannot be constructed because alternative (1) implies a clustered index
  - (2.) Index contains  $\langle k, rid \rangle$  pair with  $k$  being the age entry and  $rid$  being the record id associated with that age. The order would be random
  - (3.) Index contains  $\langle k, rid\text{-list} \rangle$  pair with  $k$  being the age and  $rid\text{-list}$  being the whole list of  $rid$ s stored in the index. The order would be random
  - (5.) Index contains  $\langle k, rid \rangle$  pair with  $k$  being the age entry and  $rid$  being the record id associated with that age. The order would be sorted based on the search key field
  - (8.) Index contains  $\langle k, rid \rangle$  pair with  $k$  being the gpa entry and  $rid$  being the record id associated with that gpa. The order would be random
  - (11.) Index contains  $\langle k, rid \rangle$  pair with  $k$  being the gpa entry and  $rid$  being the record id associated with that gpa. The order would be sorted based on the search key field
- 2.)
  - (1.) If system includes index only plans then Choice b, since the query gets the info on all employees' we just need an index only scan. If system doesn't include index only plans then Choice e, since the query gets info on all employees in the database there's no need to use an index to sort the data and a heap might be better
  - (2.) Choice c for both systems since c is not an index only plan because it is finding the  $dids$ . Since the query is looking for  $dids$  of departments based on the fields floor and budget index on the floor and budge fields. Clustered is better since you know one of the indexes will all be clustered together, for example all entries on floor 10, and you can just read in all those entries in the order of their budget. B+ tree is best since its structure allows quickly searching relevant entries