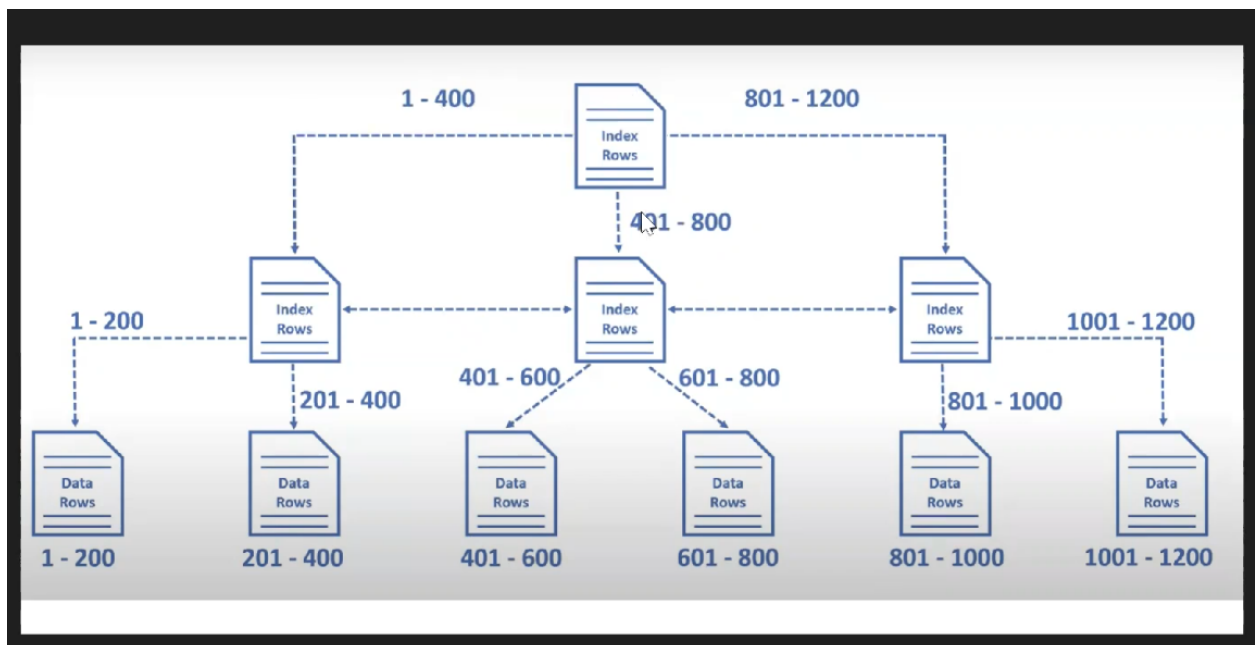


## Index

1. How data is stored in sql?



- 2.
3. The data in SQL is stored in the format of Btree in the ram...see the above picture
4. We have 1200 rows in our table...and we have index rows...which basically store the indexes of our data
5. Now if we want to query any id from the table.. that has an index to it...then first it searches index in index\_rows...then goes into table to retrieve that particular id
6. In order to indexes work...the data must be sorted
7. Let's do some practicals
8. Here we are creating an emp\_index table with some values into it

```
create table emp_index
(
  emp_id int ,
  emp_name varchar(20),
  salary int
);

insert into emp_index values(1,'Ankit',10000)
,(3,'rahul',10000),(2,'manish',10000),(4,'pushkar',10000)

select * from emp_index
```

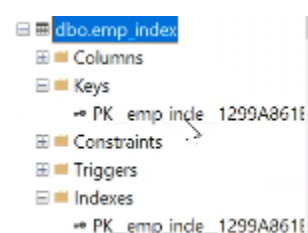
Results			
Messages			
	emp_id	emp_name	salary
1	1	Ankit	10000
2	3	rahul	10000

- 9.

10. If we create the same table with emp\_id as primary key..then ..clustered index will be automatically formed on primary key..see pic

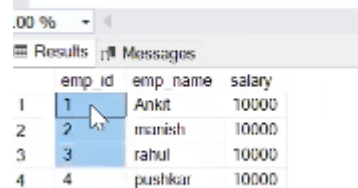
```
create table emp_index
(
  emp_id int primary key ,
  emp_name varchar(20),
  salary int
);

insert into emp_index values(1, 'Ankit', 10000)
, (3, 'rahul', 10000), (2, 'manish', 10000), (4, 'pushkar', 10000);
```



11. As we have primary key in our table and it has clustered index on it...the data will be sorted based on the primary key..it does not matter in which order we insert data...see

```
select * from emp_index
```



emp_id	emp_name	salary
1	Ankit	10000
2	manish	10000
3	rahul	10000
4	pushkar	10000

output of emp\_index with pk

12. In the emp\_index without primary key...the data will be stored as per the insertions and not be sorted...its the main diff between having a primary key and having not a pk
13. If we have a composite key...then the data will be sorted based on composite key's columns

A composite key in SQL can be defined as a combination of multiple columns, and these columns are used to identify all the rows that are involved uniquely. Even though a single column can't identify any row uniquely, a combination of over one column can uniquely identify any record. In

14. Difference bw cluster and noncluster index

### Clustered Index Example:

- If we create a clustered index on the "EmployeeID" column, the rows in the table will be physically sorted based on EmployeeID.
- This means that the data in the table itself is organized in the order of EmployeeID.
- Queries that involve range scans or sorting by EmployeeID will be very efficient. For instance, finding employees with EmployeeID between 100 and 200 would be fast.

### Non-Clustered Index Example:

- Now, let's create a non-clustered index on the "Department" column.
- This index does not change the physical order of rows in the table; it's like a separate list of pointers to the actual data rows.
- If you want to find all employees in the "HR" department, the non-clustered index helps by providing a quick lookup, but it won't change the physical order of rows in the table.

15.

16. Should learn more about cluster and non cluster index(for in interview)

17. How to create an index?

18. Here we have create non cluster index on emp\_index...see pic and learn syntax

```
create nonclustered index idx_name on emp_index(emp_name)
```

19. We have a inbuilt system procedure...which provides info on table

```
execute sp_helpindex emp_index
```

	index_name	index_description	index_keys
1	idx_name	nonclustered located on PRIMARY	emp_name
2	PK__emp_inde__1299A861E292A224	clustered, unique, primary key located on PRIMARY	emp_id

20. We have an orders\_index table which have 994000 rows...we will be using this ..for index practice

21. Interview ques @ 1:04:00

22. Refer online for INDEXES(too confusing)

## Delete Duplicates

emp	
emp_id	emp_name
1	ankit
2	rahul
3	vivek
1	rajnish

1. Lets take sample example of emp table
2. Basically emp\_id must be unique for every row...but somehow if we entered it by mistake we have to delete that duplicate
3. One way of deleting duplicates is to have a created\_time col...which contains created

emp_id	emp_name	create_time
1	ankit	12/22/2022 10:40
2	rahul	12/22/2022 10:40
3	vivek	12/22/2022 10:40
1	ankit	12/22/2022 10:41

time stamp of emp\_id ..now we can delete that duplicate emp\_id..with the help of create\_time col...basically we need a column to differentiate between duplicates entries...here we have create\_time col

4. Lets do it practically now...
5. Here we have created a emp\_dup ..which has one duplicate emp\_id ...see pic

```
select * from emp_dup
```

Results			
emp_id	emp_name	create_time	
1	Ankit	2022-12-22 10:40:01.000	
2	Vivek	2022-12-22 10:40:01.000	
3	Ankit	2022-12-22 10:42:01.000	

6. And if we observe the create time is different for these two records(emp\_id's = 1)
7. How to identify if there are duplicate values?

```
select emp_id from emp_dup group by emp_id having count(emp_id)>1
```

8. Now if we perform deletion without help of create\_time col..then we will be deleting all occurrences of that emp\_id including the duplicate

```
delete from emp_dup where emp_id in  
(select emp_id from emp_dup group by emp_id having count(emp_id)>1)
```

9. If we perform the above query ...then all occurrence of emp\_id = 1 will be deleted...but we don't want this...we only want to delete the duplicates

10. We will use getdate() function to get the create\_time of each emp\_id

```
insert into emp_dup  
values(1,'Ankit',getdate());
```

 we have ran this thrice...we got

	emp_id	emp_name	create_time
1	1	Ankit	2022-12-22 08:28:59.720
2	2	Vivek	2022-12-22 10:40:11.000
3	1	Ankit	2022-12-22 08:29:10.320

11. Now its our wish..for which id we want to delete..based on business requirements...like if we want to delete the emp\_id which was created first ...we can do that...or if we want to delete the emp\_id which was created at last..we can do that as well

12. Here we are deleting the id that we created first...we retrieved that using min(create\_time)...see pic

```
delete from emp_dup where emp_id in  
(select emp_id,min(create_time) as create_time from emp_dup group by emp_id having count(emp_id)>1)  
  
select * from emp_dup
```

Results Messages

emp_id	create_time
1	2022-12-22 08:28:59.720

13. Same thing using joins

```
delete emp_dup from emp_dup e  
inner join (select emp_id,min(create_time) as create_time from emp_dup group by emp_id having count(emp_id)>1)  
on e.emp_id=d.emp_id and e.create_time=d.create_time
```

14. Now let's say we don't have create\_time col..then we cannot delete the duplicates... instead, we can delete all entries of that emp\_id

15. But in ORACLE there is row\_id in the backend ...with the help of it we can delete the cols...it works as a helper col for deleting duplicates

16. The other workaround for us to delete duplicates without helper col is...to take a backup of distinct values from emp\_dup1 to emp\_dup1\_back

```
select distinct * into emp_dup1_back from emp_dup1
```

17. Next, we run the emp\_dup1\_back | select \* from emp\_dup1\_back we get  
now we don't have any duplicates in our table

	emp_id	emp_name
1	1	Ankit
2	2	Vivek

18. Then we delete all the data from emp\_dup1 using truncate `truncate table emp_dup1`

19. Now we will insert the data of emp\_dup1\_back into emp\_dup1...Now this is one of the ways to delete duplicates..without a helper col

20. Now what if have 3 duplicates..then in this case..we will store the id..which we want to keep in our table

21. Here in the below query ..we have kept the latest id..and deleted the duplicates which was created before it

```
delete emp_dup from emp_dup e
inner join (select emp_id,max(create_time) as create_time from emp_dup group by emp_id ) d
on e.emp_id=d.emp_id and e.create_time=d.create_time;
```

22. If we perform left join instead of inner join..we get

```
select *
from emp_dup e
LEFT join (select emp_id,max(create_time) as create_time from emp_dup group by emp_id ) d
on e.emp_id=d.emp_id and e.create_time=d.create_time
```

emp_id	emp_name	create_time	emp_id	create_time
1	Ankit	2022-12-22 08:42:06.107	NULL	NULL
2	Vivek	2022-12-22 10:40:01.000	2	2022-12-22 10:40:01.000
1	Ankit	2022-12-22 08:42:14.550	1	2022-12-22 08:42:14.550
1	Ankit	2022-12-22 08:41:06.380	NULL	NULL

23. Now if want to delete the duplicates....we need to retrieve rows that are null

```
select *
from emp_dup e
LEFT join (select emp_id,max(create_time) as create_time from emp_dup group by emp_id ) d
on e.emp_id=d.emp_id and e.create_time=d.create_time
where d.emp_id is null
```

emp_id	emp_name	create_time	emp_id	create_time
1	Ankit	2022-12-22 08:42:06.107	NULL	NULL
1	Ankit	2022-12-22 08:41:06.380	NULL	NULL

24. Now with the help of below query we deleted duplicates ..using the above idea

```
delete emp_dup
from emp_dup e
LEFT join (select emp_id,max(create_time) as create_time from emp_dup group by emp_id ) d
on e.emp_id=d.emp_id and e.create_time=d.create_time
where d.emp_id is null
```

25.