Database joins

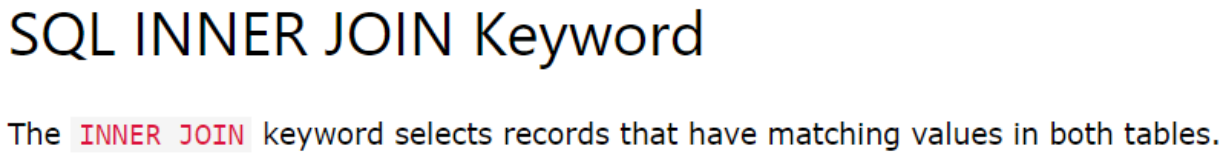
1.
```
select o.order_id,o.order_date,r.return_reason
from orders o
inner join returns r on o.order_id=r.order_id
```

100 %

Results | Messages

	order_id	order_date	return_reason
1	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
2	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
3	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
4	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
5	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
6	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
7	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
8	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
9	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
10	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
11	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
12	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
13	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
14	CA-2020-145583	2020-10-13 00:00:00.000	wrong item

2. Here we are trying to join the two tables on order id



3. From orders o….here we have given an alias name **'o'** to our orders table and we have given alias **r** to our returns table
4. We are using alias because `o.order_id=r.order_id` here order_id is present in both the tables…so to differentiate which order_id belongs to **orders** table and which order_id belong to **returns** table we use aliases for them..
5. Instead of giving aliases we can give table names..but sometimes table names will be very long ..and it doesn't look good
6. Here we have joined orders table with return tables..on order_id
7. In our returns table ..there 296rows…which says that there are 296 returns from our orders
8. And after we join orders table with returns table..we get 800 rows because a single order_id can have multiple products

9.

```
select o.order_id,o.product_id,r.return_reason
from orders o
inner join returns r on o.order_id=r.order_id

select * from returns
```

	order_id	product_id	return_reason
1	CA-2018-143336	OFF-AR-10003058	wrong item
2	CA-2018-143336	TEC-PH-10001949	wrong item
3	CA-2018-143336	OFF-BI-10002215	wrong item
4	CA-2020-111682	OFF-ST-10000604	wrong item
5	CA-2020-111682	OFF-PA-10001569	wrong item
6	CA-2020-111682	FUR-CH-10003968	wrong item
7	CA-2020-111682	OFF-PA-10000587	wrong item
8	CA-2020-111682	TEC-AC-10002167	wrong item
9	CA-2020-111682	OFF-BI-10001460	wrong item
10	CA-2020-111682	OFF-AR-10001868	wrong item
11	CA-2020-109806	OFF-AR-10004830	wrong item

```
select * from returns
```

	order_id	return_reason
1	CA-2018-100762	bad quality
2	CA-2018-100867	bad quality
3	CA-2018-102652	bad quality
4	CA-2018-103373	bad quality
5	CA-2018-103744	bad quality
6	CA-2018-103940	bad quality
7	CA-2018-104829	bad quality
8	CA-2018-105270	bad quality
9	CA-2018-108609	bad quality
10	CA-2018-108861	bad quality
11	CA-2018-109918	bad quality
12	CA-2018-110786	bad quality
13	CA-2018-111871	bad quality
14	CA-2018-116785	bad quality

Query executed successfully. | 8LR135CG0276ZST (15.0 RTM) | ANT\ankiban (52) | namastesql | 00:00:00 | 296 rows

10.

```
select o.order_id,o.order_date,r.return_reason
from orders o
inner join returns r on o.order_id=r.order_id
```

	order_id	order_date	return_reason
1	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
2	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
3	CA-2018-143336	2018-08-27 00:00:00.000	wrong item
4	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
5	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
6	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
7	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
8	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
9	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
10	CA-2020-111682	2020-06-17 00:00:00.000	wrong item
11	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
12	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
13	CA-2020-109806	2020-09-17 00:00:00.000	wrong item
14	CA-2020-145583	2020-10-13 00:00:00.000	wrong item

Query executed successfully. | 8LR135CG0276ZST (15.0 RTM) | ANT\ankiban (52) | namastesql | 00:00:00 | 800 rows

11.

```
select *
from orders o
inner join returns r on o.order_id=r.order_id
```

12. This will return all the columns from two tables…we can see return's table columns at last
13.
```
select o.*
from orders o
inner join returns r on o.order_id=r.order_id
```
this gives all the columns from orders table
14.
```
select o.*,r.return_reason
from orders o
inner join returns r on o.order_id=r.order_id
```
this will give all the columns from orders table and return_reason column from returns table
15. Whenever we are using joins..and we have multiple tables..we have to use aliases for the table
16.
```
select o.order_id,o.product_id,r.return_reason, r.order_id as return_order_id
from orders o
left join returns r on o.order_id=r.order_id;
```

100 %

Results | Messages

	order_id	product_id	return_reason	return_order_id
7	CA-2018-115812	OFF-AR-10002833	NULL	NULL
8	CA-2018-115812	TEC-PH-10002275	NULL	NULL
9	CA-2018-115812	OFF-BI-10003910	NULL	NULL
10	CA-2018-115812	OFF-AP-10002892	NULL	NULL
11	CA-2018-115812	FUR-TA-10001539	NULL	NULL
12	CA-2018-115812	TEC-PH-10002033	NULL	NULL
13	CA-2021-114412	OFF-PA-10002365	NULL	NULL
14	CA-2020-161389	OFF-BI-10003656	NULL	NULL
15	US-2019-118983	OFF-AP-10002311	NULL	NULL
16	US-2019-118983	OFF-BI-10000756	NULL	NULL
17	CA-2018-105893	OFF-ST-10004186	NULL	NULL
18	CA-2018-167164	OFF-ST-10000107	NULL	NULL
19	CA-2018-143336	OFF-AR-10003056	wrong item	CA-2018-143...
20	CA-2018-143336	TEC-PH-10001949	wrong item	CA-2018-143
21	CA-2018-143336	OFF-BI-10002215	wrong item	CA-2018-143...

SQL LEFT JOIN Keyword

The `LEFT JOIN` keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

17.
18. Here if we do left join return table with order table..then…wherever there is a match…we will get the return reason and return_order_id…if there is no match then in SQL it assigns NULL to them..which also signifies that there are no returns on that order id

19.
```
select o.order_id,o.product_id,r.return_reason, r.order_id as return_order_id
from orders o
left join returns r on o.order_id=r.order_id;
```

100 %

Results | Messages

	order_id	product_id	return_reason	return_order_id
1	CA-2020-152156	FUR-BO-10001798	NULL	NULL
2	CA-2020-152156	FUR-CH-10000454	NULL	NULL
3	CA-2020-138688	OFF-LA-10000240	NULL	NULL
4	US-2019-108966	FUR-TA-10000577	NULL	NULL
5	US-2019-108966	OFF-ST-10000760	NULL	NULL
6	CA-2018-115812	FUR-FU-10001487	NULL	NULL
7	CA-2018-115812	OFF-AR-10002833	NULL	NULL
8	CA-2018-115812	TEC-PH-10002275	NULL	NULL
9	CA-2018-115812	OFF-BI-10003910	NULL	NULL
10	CA-2018-115812	OFF-AP-10002892	NULL	NULL
11	CA-2018-115812	FUR-TA-10001539	NULL	NULL
12	CA-2018-115812	TEC-PH-10002033	NULL	NULL
13	CA-2021-114412	OFF-PA-10002365	NULL	NULL
14	CA-2020-161389	OFF-BI-10003656	NULL	NULL
15	US-2019-118983	OFF-AP-10002311	NULL	NULL

20. So here if we want everything from the left table …we use left join…so here whatever table is there before left join(we have orders table) is our left table..and whatever matching data is present..we get in the output..if its not there..we get NULL
21. Most of the interview questions will rely on inner join and left join
22.
```
select o.order_id,o.product_id,r.return_reason, r.order_id as return_order_id
from orders o
left join returns r on o.order_id=r.order_id
where r.return_reason is null
```

23. Here we have retrieved the order_ids..which have not been returned…first we have applied left join on Orders and Return table…and by seeing the output…we have applied filter using where
24. The order of execution ..in the above statement ..if first orders ..then join…where and select
25.
```
select r.return_reason,sum(sales) as total_sales
from orders o
left join returns r on o.order_id=r.order_id
group by r.return_reason
```

100 %

Results | Messages

	return_reason	total_sales
1	others	22587.426
2	NULL	2116696.5823999
3	wrong item	143280.3699
4	bad quality	14636.482

we can perform group by on joins…here we have retrieved the total sales of returned items(so we have to refund this sales)

SQL INNER JOIN Keyword

The `INNER JOIN` keyword selects records that have matching values in both tables.

26.

```
select r.return_reason,sum(sales) as total_sales
from orders o
inner join returns r on o.order_id=r.order_id
group by r.return_reason
```

100 %

Results | Messages

	return_reason	total_sales
1	bad quality	14636.402
2	others	2258.7426
3	wrong item	143280.3699

27. And if we do inner join then we get matching values from both the table…so in here we have retrieved the total sales for the returned items
28. If we do not specify any columns for join…then it performs cross join..where each row in first table..matches with every row in the second table

order_id	sales
1	100
2	200
6	

order_id	return reaso
1	bad item
3	bad item
4	bad item

29. So here we will get 6 rows in total
30. Example is this

```
select * from
employee,dept
;
```

100 %

Results | Messages

	emp_id	emp_name	dept_id	salary	manager_id	emp_age	dep_id	dep_name
1	1	Ankit	100	10000	4	39	100	Analytics
2	2	Mohit	100	15000	5	48	100	Analytics
3	3	Vikas	100	10000	4	37	100	Analytics
4	4	Rohit	100	5000	2	16	100	Analytics
5	5	Mudit	200	12000	6	55	100	Analytics
6	6	Agam	200	12000	2	14	100	Analytics
7	7	Sanjay	200	9000	2	13	100	Analytics
8	8	Ashish	200	5000	2	12	100	Analytics
9	9	Mukesh	300	6000	6	51	100	Analytics
10	10	Rakesh	500	7000	6	50	100	Analytics
11	1	Ankit	100	10000	4	39	200	IT
12	2	Mohit	100	15000	5	48	200	IT
13	3	Vikas	100	10000	4	37	200	IT
14	4	Rohit	100	5000	2	16	200	IT
15	5	Mudit	200	12000	6	55	200	IT
16	6	Agam	200	12000	2	14	200	IT
17	7	Sanjay	200	9000	2	13	200	IT
18	8	Ashish	200	5000	2	12	200	IT

Query executed successfully. | BLR135CG0276ZST (15.0 RTM) | ANT\ankiban (52) | namastesql | 00:00:00 | 40 rows

31. Here initially in employee table we have 10 rows and in dept table we have 4 rows…so now total we got 40 rows

32. The other way of doing cross joins is

```
select * from
employee
inner join dept on 100=100
order by employee.emp_id
;
```

here 100=100 will always return true..it signifies that there

```
select * from employee;
select * from dept;
```

100 %

Results | Messages

	emp_id	emp_name	dept_id	salary	manager_id	emp_age
2	2	Mohit	100	15000	5	48
3	3	Vikas	100	10000	4	37
4	4	Rohit	100	5000	2	16
5	5	Mudit	200	12000	6	55
6	6	Agam	200	12000	2	14
7	7	Sanjay	200	9000	2	13
8	8	Ashish	200	5000	2	12
9	9	Mukesh	300	6000	6	51
10	10	Rakesh	500	7000	6	50

	dep_id	dep_name
1	100	Analytics
2	200	IT
3	300	HR
4	400	Text Analytics

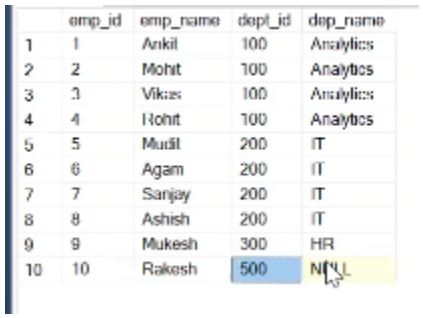
33. here in our tables if we can see…in first dept_id = 500 is not present in the second table…and dept_id = 400 ..is not present in the first table..
34. So if we apply inner join on first table and second table..we get 9 records

```
select * from
employee e
inner join dept d on e.dept_id=d.dep_id
```

	emp_id	emp_name	dept_id	salary	manager_id	emp_age	dep_id	dep_name
1	1	Ankit	100	10000	4	39	100	Analytics
2	2	Mohit	100	15000	5	48	100	Analytics
3	3	Vikas	100	10000	4	37	100	Analytics
4	4	Rohit	100	5000	2	16	100	Analytics
5	5	Mudit	200	12000	6	55	200	IT
6	6	Agam	200	12000	2	14	200	IT
7	7	Sanjay	200	9000	2	13	200	IT
8	8	Ashish	200	5000	2	12	200	IT
9	9	Mukesh	300	6000	6	51	300	HR

35. And also if we see the column name on which we are performing join..need not to be same

36.
```
select e.emp_id,e.emp_name,e.dept_id,d.dep_name from
employee e
left join dept d on e.dept_id=d.dep_id
```



	emp_id	emp_name	dept_id	dep_name
1	1	Ankit	100	Analytics
2	2	Mohit	100	Analytics
3	3	Vikas	100	Analytics
4	4	Rohit	100	Analytics
5	5	Mudit	200	IT
6	6	Agam	200	IT
7	7	Sanjay	200	IT
8	8	Ashish	200	IT
9	9	Mukesh	300	HR
10	10	Rakesh	500	NULL

and if we perform left join on employee and dept tables..we get 10 rows

SQL RIGHT JOIN Keyword

The `RIGHT JOIN` keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

37.
38. Right join is the opposite of left join..if there is a matching record in the 2nd table..then it joins ..or else it just keep NULL and returns the first table
39. So in the real world..we don't use right join..because while using left join ..if we just swap the tables..then it becomes the right join

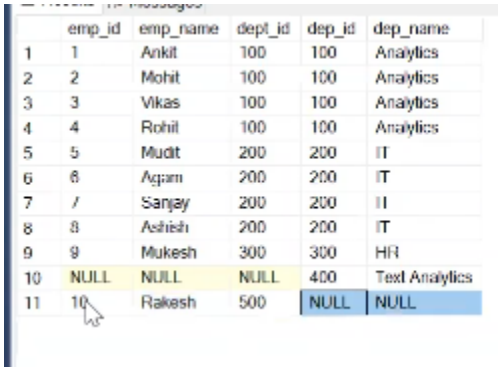
40.
```
select e.emp_id,e.emp_name,e.dept_id,d.dep_id,d.dep_name from
employee e
right join dept d on e.dept_id=d.dep_id;

select e.emp_id,e.emp_name,e.dept_id,d.dep_name from
dept d
left join employee e on e.dept_id=d.dep_id
```

41. Here in the left join…dept d is considered as first table..and it returns everything from dept d..which matches from the employee e
42. And in the right join…the table which is in right is considered as first table..and it returns everything from dept d..which matches from the employee e
43. Once practice right and left join
44.

SQL Full Join creates a new table by joining two tables as a whole. The joined table contains all records from both the tables and fill in NULLs for missing matches on either side. In short, full join is a type of outer join that combines the results of both left and right joins.

45.
```
select e.emp_id,e.emp_name,e.dept_id,d.dep_id ,d.dep_name from
dept d
full outer join employee e on e.dept_id=d.dep_id;
```



	emp_id	emp_name	dept_id	dep_id	dep_name
1	1	Ankit	100	100	Analytics
2	2	Mohit	100	100	Analytics
3	3	Vikas	100	100	Analytics
4	4	Rohit	100	100	Analytics
5	5	Mudit	200	200	IT
6	6	Agam	200	200	IT
7	7	Sanjay	200	200	IT
8	8	Ashish	200	200	IT
9	9	Mukesh	300	300	HR
10	NULL	NULL	NULL	400	Text Analytics
11	10	Rakesh	500	NULL	NULL

46. So here …till row 9 we get inner join…if we perform left join..we get all the records except the 10th row…if we perform right join in dept table..we get first 10 rows as output..and if we perform full join we get 11 rows
47. Next here we r creating a sample table see pic

```
create table people
(
manager varchar(20),
region varchar(10)
)

insert into people
values ('Ankit','West')
,('Deepak','East')
,('Vishal','Central')
,('Sanjay','South')
```

48.
```
select o.order_id,o.product_id,r.return_reason,p.manager
from orders o
inner join returns r on o.order_id=r.order_id
inner join people p on o.region=p.region
```

49. Here we have applied 2 inner joins..first it joins two tables(orders o,returns r) on order_id…and then this new table will join with people p(table) on region..see pic and understand
50.
```
select o.order_id,o.product_id,r.return_reason,p.manager
from orders o
left join returns r on o.order_id=r.order_id
inner join people p on o.region=p.region
```

try this

51. We can join any number of tables…using joins..theres no limitation