Joints

ARUN'S PAGE NO. DATE

Displaying rows & columns from multiple table

here one single table view is called as "Joints"

Joints retrieval

1) cross join
2) inner join / simple join
3) outer join
   - Left outer join
   - Right outer join
   - Full outer join
4) self join

Syntax: Cartesian, ANSI

Ex: cross joint
1) select * from emp, dept; 56 row
2) select * from emp cross join dept; 56 row

1) cross Joint

* cross join displays the record from the all the table in combination.

Cartesian syntax :- select * from table1, table2, ---

ANSI syntax :- select * from table1 cross join table3 ----

2) INNER join

cartesian
select * ---- Ex: 1) select * from emp, dept
from table1, table2 ... where emp.deptno = dept.deptno and emp.deptno = 20; 5 rows

ANSI -> select * ----
from table1
Inner join table2
on join condition (?) (INNER,) join table3
on join condition (),
where condition (?)

Keyword

Ex: 2) select * from emp inner join dept on emp.deptno = dept.deptno where emp.deptno = 20; 5 rows

SQL> select * from DEPT;

SQL> select * from emp, dept
where emp.DEPTNO = dept.deptno;

SQL> select emp.*, DEPT.DNAME, loc from EMP, DEPT
where EMP.DEPTNO = DEPT.DEPTNO;

=> Display all the employee details along with salary grade

PAGE NO.
DATE / /

SOL> Select EMP.*, SALGRADE.Grade from emp, salgrade
where EMP.SAL BETWEEN SALGRADE.LOSAL AND SALGRADE.
HISAL;

=> Left outer join

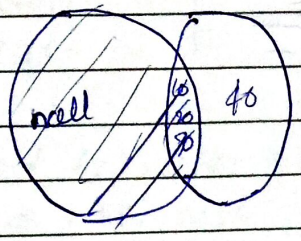


Table 1 Table 2
left Right
emp dept

Cartesian
Select * ---- 1 ...
from table1, table2
where table1.column(?) table2.column(+)
and filter(?)
(+) placed

ANSI
Select * ---- 1 ----
from table1
left outer join table2
on join condition
≡ where condition(?)

> Select * from emp, salgrade
where emp.deptno = salgrade.grade;
no row selected no relation

Sol> Select * from emp, dept
where emp.DEPTNO = DEPT.DEPTNO(+);

left outer join
Select all the records from the left table irrespective of the condition but only selects the records that satisfy condition from the right table

Right outer join

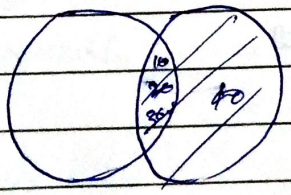


Table 1 Table 2
left Right
emp dept

Select * ---- 1 ----
from table1, table2
where table1.column(+) ? table2.column
≡ AND filter(?)

Select * ----
from table1
Right outer join table2
on join condition
≡ where condition(?)

Sol> Select * from emp, dept
where emp.DEPTNO(+) = DEPT.DEPTNO;

Full outer join

ARUN'S
PAGE NO.
DATE / /

select * ——
from table1
full outer join table2
on join condition

Full outer join

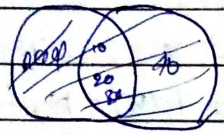


Table1 — left emp
Table2 — Right dept

SQL> select * from emp
Full outer join dept
on emp.deptno = dept.deptno; - 15 rows 40 deptno extra → row

=> Display location wise no of employees
select loc, count(*) from emp, dept
where emp.deptno = dept.deptno group by loc

=> Display all the employee details given in employee name, job, department name location for the employees working in location dallas & chicago

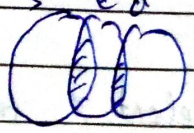
SQL> select ename, job, dname, loc 11 rows
from emp, DEPT
where emp.DEPTNO(+) = dept.DEPTNO AND loc in ('dallas', 'chicago')

=> Display all the employee details salary with their salary grades for the employee with salary grade 4 & 5.

SQL> select emp.*, grade → 6 rows
from emp, sal grade
where SAL between LOSAL and HISAL
AND grade IN(4,5);

=> Display employee details such as employee name, loc, job, dept name for the employees sal grade 3 & 4

SQL> select ename, job, loc, dname, grade → 7 rows
from emp, DEPT, SALGRADE
where emp.DEPTNO = dept.deptno
AND Sal between losal AND HISAL
AND grade IN(3,4);



SQL> select loc, count (*) Display loc wise no of employee
2 from emp, dept
where emp.deptno(+) = dept.deptno
group by loc;

newyork	3
chicago	6
boston	1
dallas	5

Here it's count records

SQL> select loc, count(empno) Here it's count employee
from emp, dept
=> where emp.deptno(+) = dept.deptno
group by loc;

Newyork	3
chicago	6
boston	0
DALLAS	5

SQL> select loc, count(empno)
from emp
Full outer JOIN DEPT
on emp.deptno = DEPT.DEPTNO
group by loc;

newyork	3
chicago	6
BOSTON	0
DALLAS	5
	2

- Self join :-
it a join operation is perform on same table itself is called self join

SQL> Select * from emp employee, emp manager
where employee.mgr = manager.empno;

=> display employee number and their respected to manager names after each employee?

Full outer join

SQL> select employee.ename employee, manager.ename manager
from emp employee, emp manager
where employee.mgr = manager.empno(+);

SQL> select employee.ename employee, manager.ename manager
from emp employee, emp manager
where employee.mgr = MANAGER.EMPNO;

⇒ Display the employee name along with their salary manager name along with their salary for the employee salary more than their manager?

SQL> Select Employee.ename emp-name, Employee.SAL ~~SAL~~, ~~MANAGER~~ emp-sal, manager.ename ~~MANAGER~~ MAN-name,

manager.SAL MAN-SAL
from emp employee, emp manager
where employee.mgr = manager.empno
and employee.SAL > manager.SAL;

=> Display employee names, the deptno, manager names their deptno for the employees working in dept same as their manager

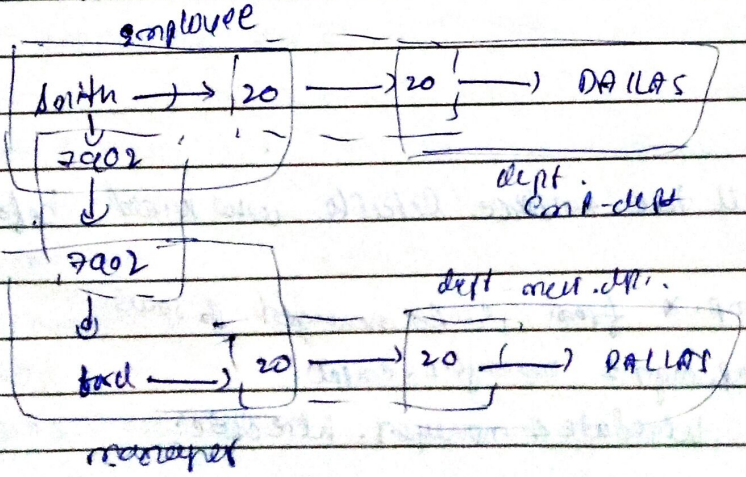
SQL> select employee.ename, employee.deptno, manager.ename manager.deptno
from emp employee, emp manager
where employee.mgr = manager.empno;

SQL> Select employee.ename, employee.deptno, manager.ename, manager.deptno
from emp employee, emp manager
where employee.mgr = manager.empno
and employee.deptno = manager.deptno;

=> Display manager wise employee wrt their manager name

SQL> Select manager.ename, count(manager.ename)
from emp employee, emp manager
where employee.mgr = manager.empno
group by manager.ename;

=> Display all the employee detail such as their names their department names and location along with their manager name, dept name & loc for the employee working in loc same as their manager.



SQL> select employee.ename, emp_dept.DNAME, DEPT.LOC,
3 manager.ENAME, MGR_DEPT.DNAME, MGR_DEPT.LOC

FROM EMP EMPLOYEE, DEPT EMP_DEPT,
EMP MANAGER, DEPT MGR_DEPT
WHERE EMPLOYEE.DEPTNO = EMP_DEPT.DEPTNO
AND MANAGER.DEPTNO = MGR_DEPT.DEPTNO
AND EMPLOYEE.MGR = MANAGER.EMPNO
AND EMP_DEPT.LOC = MGR_DEPT.LOC;

I do full one

Upto here 1st 11 rows

=> Display employee names their location, their salary, and their salary grade along with their manager name, location, salary & their salary grade

23/

corelated sub query

* performing joint operation in a sub query such that one of the table belongs to outer query & other belongs to inner query.
* In corelated sub query both inner query & outer query execute both w.r.t to each other but ~~result~~ displayed result w.r.t. outer query.

```
select * from table
where column (?) (select column
from tabled
where table1.col(?) tabled.column);
```

```
SQL> select *
from emp
where sal > (select sal
  from emp
  manager
where emp.mgr = manager.empno);
```

=> Display all the employee details who joined before their manager

```
SQL> select emp.* from emp, manager 6 rows
where emp.mgr = manager.empno
and emp.hiredate < manager.hiredate.
```

(or) select * from emp
where hiredate < (select hiredate from emp
where emp.mgr = manager.empno);

=> Display all the employee details working in the same as their manager

sol> select * from emp, dept // cross
where emp.deptno = dept.deptno
and loc = (select loc from emp manager, dept non_dept
where manager.deptno = non_dept.deptno
and emp.mgr = manager.empno);

sol> select * from emp x
where (select count(distinct y.sal) from emp y
where x.sal <= y.sal) = 4;

sol> select * from emp x
where (select count(distinct y.sal) from emp y
where x.sal <= y.sal) = # -> if is used to take dynamic i/p
(user i/p)

=) Display jobwise no of employees without using group by class

sol> select job, (select count(job) from emp x where x.job = y.job)
from emp y;

sol> select distinct job, (select count(job) from emp x
where x.job = y.job) from emp y;

→ out come of any DML statement is result set
or processed data from data base (record set)

Table name
countries
Departments
employees
Jobs
Jobs_history
locations
reg_emp

ARUN'S
PAGE NO.
DATE / /

Joints

1) Cross Joints

① select * from emp, dept, salgrade ; -> 280 rows
cartesion

② select * from emp cross joins dept cross join salgrade ; -> 280 rows

2) Inner Joints
cartsian

① select * from emp, dept, salgrade → for salgrade we don't have any relation
where emp.deptno = dept.deptno ;
70 rows

② select * from emp, dept, salgrade
where emp.deptno = dept.deptno and emp.deptno=20;
without salgrade Table 5 row will get select
But Here we using salgrade so to get proper result we have to create relation there salgrade doesn't have any relationship with other table so after filtering result of emp.deptno it take combination of with salgrade 35 row get selected.

3) Outer Joints :- all the content of both tables are integrated either they are matched or not

a) Left outer joints
cartesian

* select * from emp, dept
where emp.deptno = dept.deptno (+) AND
Job in ('ANALYST', 'Manager'); -5 row

> ANSI
> select
on e
where
> select
where
b) Right
> select
where
Job
> select
where
> select
on e
> select
where

ANSI

> select * from emp Left outer join dept
on emp.deptno = dept.deptno
where Job in ('Analyst', 'manager'); -5 row selected

> select * from emp, salgrade same as ANSI also
where emp.deptno = salgrade.grade(+);
↳ 14 records all null values bcz no relation

b) Right outer Joints
cartesian

> select * from emp, dept
where emp.deptno(+) = dept.deptno AND
Job in ('salesman', 'manager') -> 7 row

ANSI

> select * from emp Right outer join dept
~~where~~ on emp.deptno(+) = dept.deptno
where Job in ('salesman', 'manager'); -7 row

> select * from emp Right outer join salgrade
on emp.deptno(+) = salgrade.grade; -55 row
null values ↳ 5 rows
bcz no relation ship

> select * from emp, salgrade
where emp.deptno(+) = salgrade.grade; 5 row
null values 5 rows

ARJUN'S
PAGE NO.
DATE / /

c) Full outer Joins

> select * from emp Full outer Join dept
on emp.deptno = dept.deptno ; -> 15 rows
Here we have only one type of syntax ANSI
BCz cartesian syntax it's not there BCz
to works as inner joins cartesian syntax
so we don't have

> select * from emp Full outer Join dept
on emp.deptno = dept.deptno
where dept.deptno in (20, 40) ; 6 rows
BCz deptno column is present in both table
so column ambiguously will happen so we need
to specify the column belongs to which treble you
want

> select * from emp Full outer Join dept
on emp.deptno = dept.deptno
where emp.deptno in (20, 40) ; 5 rows
BCz in emp table deptno column does not contain
40 deptno.

problems
1) Display loc wise no of employees

> select loc, count(*) from emp, dept
where emp.deptno = dept.deptno -> Inner Join
group by loc ; -> 3 rows

loc	count(*)
new york	3
chicago	6
dallas	5

ARUN'S
PAGE NO.
DATE / /

> select loc, count(*) from emp, dept
where emp.deptno = dept.deptno (+) → left join
group by loc; → 3 rows

loc	count(*)
new york	3
chicago	6
Dallas	5

→ use empno bcz it's count only employees

> Select loc, count(*) from emp, dept
where emp.deptno (+) = dept.deptno Right join
group by loc; → 4 rows

loc	count(*)
new york	3
chicago	6
BOSTON	1 → we dont have any employee
Dallas	5

Here we joined using Right outer joint in that how ~~much~~ much loc have that table it's provide the result for each value for that column even if they don't have any value it's going add the emp table to that Right join table

> select * from dept;

> select * from dept emp;

Self JOINTS

> select employee.empno, employee.ename, employee.deptno, employee.deptnew, manager.empno, manager.ename, manager.deptno from emp employee, emp manager where employee.mgr = manager.empno and employee.deptno = manager.deptno; 11 row selected.

ARUN'S
PAGE NO.
DATE / /

> select Employee.empno, ename, deptno, mgr, manager.empno, dept no, ename from emp employee, emp manager where ~~manager~~. employee.mgr = manager.empno and employee.deptno = ~~employee~~ manager.deptno;

→ column ambiguous will happen because not specified from which column belongs to which table so it will be ambiguous

problems

Q) Display employee names, their location, salary and salary grade along with manager name, location, salary and salary grade based on ~~loc~~ manager

> select employee.empno emp-empno, employee.name emp-name employee.mgr emp-mgr, empdept.dname emp-dname employee.sal emp-sal, empsalgrade.grade emp-grade, empdept.deptno emp-deptno, empdept.loc emp-loc, manager.empno mgr-empno, manager.ename mgr-ename, ~~manager~~ mgrdept.~~dept~~name mgr-dname, manager.sal mgr-sal, mgrsalgrade.grade mgr-grade; mgrdept.loc mgr-loc.
from emp employee, emp manager
dept empdept, dept mgrdept
salgrade empsalgrade, salgrade mgrsalgrade
where employee.deptno = empdept.deptno and ~~mgrdept~~ manager.deptno = mgrdept.deptno and employee.sal between empsalgrade.losal and empsalgrade.hisal
employee.mgr = manager.empno and empdept.loc = mgrdept.loc and manager.sal between mgrsalgrade.losal and mgrsalgrade.hisal