

## Assignment IV

### Aim:

Design atleast 10 SQL queries for suitable databases application using SQL DML statement, all types of join, sub-query, perform.

- 1) cartesian product
- 2) natural join
- 3) inner join
- 4) left outer join
- 5) right outer join
- 6) Full outer join.

### Theory:

In cartesian product, each row of one table combines with each row of the other.

A cartesian product is formed when

- 1) a join condition is omitted
- 2) a join condition is invalid
- 3) all rows in the first table are joined to all rows in the second table.

ID	Name	dept-name	total_cred
00128	Zhang	comp. science	102
12345	Shamkar	comp. science	32
19991	Brandt	history	80
23121	Chavez	finance	110
44553	Peter	physics	56
45678	William	physics	46

The student relation.



ID	course.id	sec.id	semester	year	grade
00128	CS-101	1	fall	2009	A
00128	CS-347	1	fall	2009	A
12345	CS-101	1	fall	2009	C
12345	CS-190	2	spring	2009	A
12345	CS-313	1	spring	2010	A
19991	CS-347	1	fall	2009	A
23121	MIS-351	1	spring	2010	-
44553	EIN-201	1	spring	2010	C+

The -takes relation

Syntax

select  $A_1, A_2, A_3, \dots, A_n$  from  $r_1, r_2, \dots, r_n$  where  $P;$

$A_i \rightarrow$  attribute

$r_i \rightarrow$  relation

$p \rightarrow$  predicates

example

select ID, name, ID, sec.id from student,

takes where student.ID = takes.ID;



ID	name	ID	sec-id
10128	zhang	00128	1
10128	zhang	00128	1
10128	zhang	12345	1
10128	zhang	12345	2
.	.	.	.
.	.	.	.
.	.	.	.
54321	Williams	23121	1
54321	Williams	44553	1

## Join

Join is used to fetch data from two or more tables, which is joined to appear as single set of data.

## Types of join

I Natural join or inner join

II Outer join

a) left outer join

b) right outer join

c) Full outer join

## I) Natural join or inner join

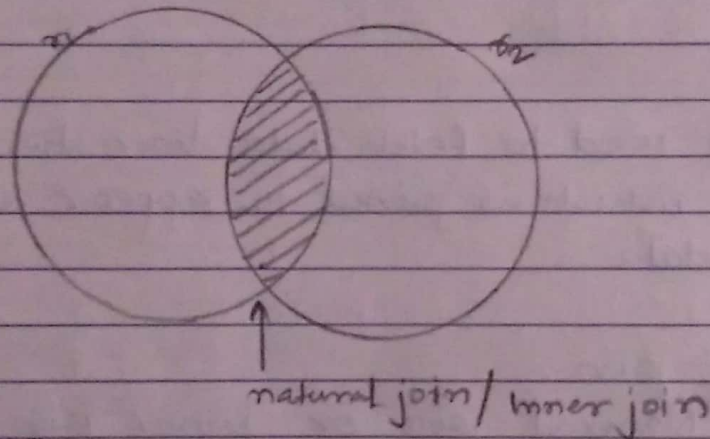
The natural join operation operates on two relations and produces a relation as the result. Unlike the cartesian product of two relations; which concatenates each tuple of the first relation with every tuple of the second, natural join considers



only those pairs of tuples with the same value on those attribute that appears in the schemas of both relations.

### Syntax

select  $A_1, A_2, A_3, \dots, A_n$  from  $r_1$   
natural join  $r_2$  natural join  $\dots$  natural join  $r_m$   
where  $p$ ;



### example

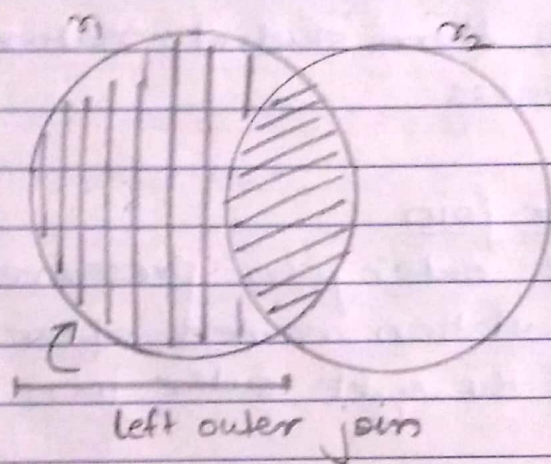
select student.ID, student.name, takes.semester,  
takes.grade from student inner join takes using(id);



## III) Outer join

### i) left outer join

The left outer join preserves tuple only in the relation named before. (to the left of) the left outer join equation.



This is how outer join operates:

- 1) First, compute the result of inner join.
- 2) then for every tuple  $t$  in the left hand side relation that does not match any tuple in the right hand side relation in the inner join, add a tuple  $r$  to the result of the join constructed as follows.

The attributes of tuple  $r$  that are derived from the left hand side relation are filled in with the values from tuple  $t$ .

The remaining attributes of  $r$  are filled with null values.



syntax

select  $A_1, A_2, \dots, A_n$  from  $r_1$  natural left  
outer join  $r_2$ , natural left outer join  $r_3 \dots$   
natural left outer join  $r_m$  where predicate;

example

select student.ID, student.name, takes.semester,  
takes.grade from student natural left  
outer join takes;

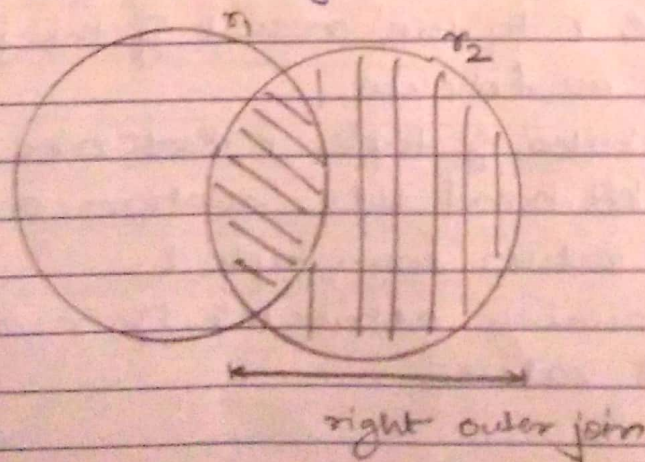
ii) right outer join

The right outer join preserves tuples only  
in the relation named after (to the  
right of) the right outer join operation.

It is symmetric to the left outer join

syntax

select  $A_1, A_2, \dots, A_n$   
from  $r_1$  natural right outer join  $r_2 \dots$   
natural outer join  $r_m$  where predicate;





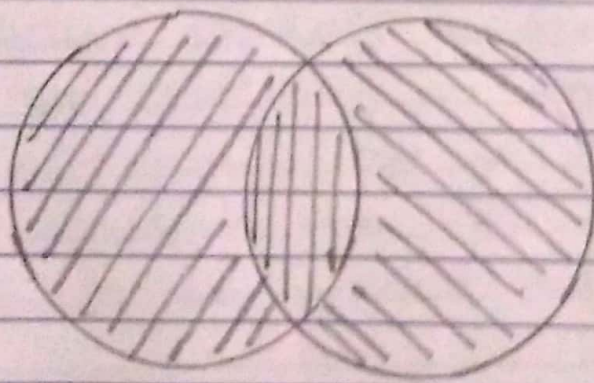
example

```
select student.ID, student.name, takes.semester,  
takes.grade from takes right outer join student  
on student.ID = takes.ID;
```

iii) full outer join

The full outer join preserves tuples in both relation.

It extends with null those from the left hand side relation that did not match which have those tuples from right hand side relation that did not match with any tuple, from left hand relation and add them to result.



full outer join

syntax/example

```
select student.ID, student.names, takes.semester,  
takes.grade from student full outer join on  
takes where student.ID = takes.ID;
```