

## BahwanCybertek Second Round Interview Q&A

**Q. Write a query to increase a column size of table**

```
create table course
(  
cname varchar2(10)  
);
```

```
alter table course modify cname varchar2(30);
```

**Q. Two tables tab1 and tab2.**

**Tab 1 contain 100 record , tab 2 contain 150 record common 30 record.**

**Tell me the output row count of the following:**

**Inner join**

**Left out join**

**Right outer join**

**Inner Join:**

It will provide only matched records of both the tables.

```
select *  
from tab1 join tab2  
on tab1.cid=tab2.cid;
```

30 records

**Left Outer Join:**

It will provide matched and unmatched records of first table. + sign is on the right.

```
select *  
from tab1 join tab2  
on tab1.cid=tab2.cid(+);
```

100 records

**Right Outer Join:**

It will provide matched and unmatched records of second table. + sign is on the left.

```
select *  
from tab1 join tab2  
on tab1.cid(+)=tab2.cid;
```

150 records.

**Q. What is cursor and refcursor and why we use both and give example.**

**Cursor**

Cursor is a sql private work area. It only process one row at a time.

**Types:**

Implicit Cursor:

DML statements written in plsqlblock

Explicit Cursor:

Cursor declared in the declaration part.

**Cursor Attributes:**

%found

%notfound

%rowcount

%isopen

```
declare
l varchar(30);
cursor c1 is
select first_name from employees;
begin
open c1;
loop
fetch c1 into l;
dbms_output.put_line(i);
exit when c1%rowcount=4;
end loop;
if c1%isopen=true then
close c1;
end if;
end;
/
```

**Refcursor:**

Refcursor is a datatype used for string the query output.

```
create or replace procedure sp_emp(a out sys_refcursor)
as
begin
open a for select * from employees;
end;
/
```

**Q.**  
**Begin**  
**Alter query**  
**End;**  
**Begin**  
**Update query**  
**End;**  
**Both block will execute or not ? And reason for that.**

DML statements can be runned in a plsql. So, the update statement will work.

```
Begin  
update employees set first_name='ELON' where employee_id=100;  
end;  
/
```

DDL (Data Definition Language) statements like CREATE, ALTER, RENAME,TRUCATE and DROPcannot be run in a plsql block. To use DDL in a plsql, we must use EXECUTE IMMEDIATE.

```
begin  
execute immediate 'alter table employees modify first_namevarchar(50)';  
end;  
/
```

**Q. Have you worked in performance tuning.**

Yes, I've worked n performance tuning. I used these to do performance tuning.

Bind variables to switch from hard parsing to soft parsing

DBMS\_PROFILER with START\_PROFILER AND STOP\_PROFILER functions. Then, PLSQL\_PROFILER\_RUNS, PLSQL\_PROFILER\_DATA and PLSQL\_PROFILER\_UNITS to get the problem in the exact line.

XML Flat file to load multiple records at once.

Check last Analyzed\_date and gather statistics using dbms\_gather.gather\_table\_stats.

### Q. Explain plan. What you will check in that?

Explain plan is used to check the problem in the select statement. By examining this plan, you can find out if Oracle is picking the right indexes and joining your tables in the most efficient manner.

```
Explain plan for
select e.first_name,e.department_name
from employees e, departments d
where e.department_id=d.department_id;
```

```
select * from table(dbms_xplan.display);
```

Joins:

Nested loop  
Column1 High volume  
Column2 Low volume  
where Unique key

Hash  
Column1 High volume  
Column2 High volume  
where no indexed column

Merge  
Column1 High volume  
Column2 High volume  
where non unique column

Index:

Unique scan  
where Unique index

Range scan  
where non unique index

Full scan  
where no index

### Q. What is correlated subquery and inline view ?

correlated sub query:

The inner query depends on the outer query

```
select * from employees where department_id in
(
select department_id from departments where
departments.department_id=employees.department_id
);
```

Inline view:

If the sub query is written in from clause.

```
select * from
select first_name,salary,dense_rank() over(order by salary desc)RK from employees)
where RK=1;
```

### Q. Utl file. How it works?

UTL\_FILE is a oracle supplied package. With it, plsql programs can read and write in operating system text files.

```
sqlplus sys/admin as sysdba
create directory NEW as 'D:\NEW';
grant read,write on NEW to HR;
grant execute on utl_file to HR;
```

```
declare
F utl_file.file_type;
l varchar2(200);
begin
F:=utl_file.fopen('NEW','EMPDETAILS.TXT','W',32767);
for x in (select first_name,salary from employees where rownum<=10) loop
i:=x.first_name||'|'||x.salary;
utl_file.put_line(F,i);
utl_file.new_line(F);
end loop;
utl_file.fclose(F);
end;
/
```

