**Use Cases for Oracle**

1. For every schema, list Schema, # tables, # records

Create or replace procedure SchemaInformation is

begin

for w in (select s.SCHEMA\_NAME, count(distinct t.table\_id) as No\_Table, count(a.Attributes\_ID) as No\_Records

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

group by s.schema\_name)

loop

dbms\_output.put\_line(w.Schema\_name);

dbms\_output.put\_line(w.No\_Table);

dbms\_output.put\_line(w.No\_Records);

end loop;

1. For every schema, list table name, # attributes,# PK\_ attr, # FK\_ attr, # NN\_attr, # IDX\_attr, # CC\_attr, # UQ\_attrs, # PGM\_attr

create or replace procedure list\_table\_info is

begin

for l in (select schema\_name, t.table\_name, count(a.attributes\_Id) as NumberOfAttributes,

count(p.pk\_id) as NoOfPKs, count(f.fk\_id) NoOfFKs, count(c.cc\_id) NoOfCCs, count(i.idx\_id) NoOfIDXs,

count(n.nn\_id) NoOfNNs, count(pg.pgm\_id) NoOfPGMs, count(u.uqi\_id) NoOfUQIs

from DG\_Attributes a

left outer join DG\_Tables t on a.Table\_id = t.Table\_ID

left outer join DG\_Attributes\_ConstCheck c on c.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_FK f on f.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_Index i on i.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_NotNull n on n.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_Program pg on pg.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_Unique u on u.Attributes\_ID = a.Attributes\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

group by schema\_name, t.Table\_Name)

loop

dbms\_output.put\_line(l.schema\_name);

dbms\_output.put\_line(l.table\_name);

dbms\_output.put\_line(l.NumberOfAttributes);

dbms\_output.put\_line(l.NoOfPKs);

dbms\_output.put\_line(l.NoOfFKs);

dbms\_output.put\_line(l.NoOfCCs);

dbms\_output.put\_line(l.NoOfIDXs);

dbms\_output.put\_line(l.NoOfNNs);

dbms\_output.put\_line(l.NoOfPGMs);

dbms\_output.put\_line(l.NoOfUQIs);

end loop;

end;

exec list\_table\_info;

drop procedure list\_table\_info;

1. For every schema, list Table\_ID, Table\_Name, attribute\_id, attribute\_Name, PK\_ID, FK\_ID, NN\_ID, constraint\_type, and constraint\_Name

create or replace procedure Schema\_Table\_Information is

begin

for x in (select s.schema\_name, t.table\_id, t.table\_name, a.attributes\_id, a.attributes\_name,

p.pk\_id, p.constraint\_name as PK\_ConstName, f.fk\_id, f.constraint\_name as FK\_ConstName,

n.nn\_id, n.constraint\_name as NN\_ConstName,

c.cc\_id, c.constraint\_name as CC\_ConstName

from DG\_attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_FK f on f.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_ConstCheck c on c.Attributes\_ID = a.Attributes\_ID

left outer join DG\_Attributes\_NotNull n on n.Attributes\_ID = a.Attributes\_ID

group by s.schema\_name, t.Table\_ID, t.Table\_Name, a.attributes\_id, a.attributes\_name,

p.pk\_id, p.constraint\_name, f.fk\_id, f.constraint\_name, n.nn\_id, n.constraint\_name,

c.cc\_id, c.constraint\_name)

loop

dbms\_output.put\_line(x.schema\_name);

dbms\_output.put\_line(x.table\_id);

dbms\_output.put\_line(x.attributes\_id);

dbms\_output.put\_line(x.attributes\_name);

dbms\_output.put\_line(x.pk\_id);

dbms\_output.put\_line(x.PK\_ConstName);

dbms\_output.put\_line(x.fk\_id);

dbms\_output.put\_line(x.FK\_ConstName);

dbms\_output.put\_line(x.nn\_id);

dbms\_output.put\_line(x.NN\_ConstName);

dbms\_output.put\_line(x.cc\_id);

dbms\_output.put\_line(x.CC\_ConstName);

end loop;

end;

exec Schema\_Table\_Information;

drop procedure Schema\_Table\_Information;

1. For every schema, list tables with PK but without FK and IDX

create or replace procedure all\_pk\_without\_fk\_IDX is

begin

for n in (select distinct schema\_name, t.table\_name, Attributes\_Name, p.PK\_ID

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_FK) f on f.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_Index) i on i.Attributes\_ID = a.Attributes\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

where f.Attributes\_ID is null

and i.Attributes\_ID is null and p.Attributes\_ID is not null)

loop

dbms\_output.put\_line(n.schema\_name);

dbms\_output.put\_line(n.table\_name);

dbms\_output.put\_line(n.Attributes\_Name);

dbms\_output.put\_line(n.PK\_ID);

end loop;

end;

exec all\_pk\_without\_fk\_IDX;

drop procedure all\_pk\_without\_fk\_IDX;

1. For every database/schema, rank them according to number of NN (Not Null) have created

create or replace procedure RankOfSchemaOnNN is

begin

for z in (select s.schema\_name, rank () over (order by count (n.nn\_id)) NN\_Rank, count(n.nn\_id) CountNN

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

left outer join DG\_Attributes\_NotNull n on n.Attributes\_ID = a.Attributes\_ID

group by s.schema\_name)

loop

dbms\_output.put\_line(z.schema\_name);

dbms\_output.put\_line(z.NN\_Rank);

dbms\_output.put\_line(z.CountNN);

end loop;

end;

exec RankOfSchemaOnNN;

drop procedure RankOfSchemaOnNN;

1. For every database, list tables without PK

create or replace procedure all\_table\_no\_pk is

begin

for m in (select schema\_name, Table\_Name

from DG\_Tables t join DG\_Schema s on t.Database\_ID = s.schema\_ID

where t.Table\_Name not in (

select distinct t.table\_name

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_FK) f on f.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_Index) i on i.Attributes\_ID = a.Attributes\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

where p.Attributes\_ID is null

))

loop

dbms\_output.put\_line(m.schema\_name);

dbms\_output.put\_line(m.Table\_Name);

end loop;

end;

exec all\_table\_no\_pk;

drop procedure all\_table\_no\_pk;

1. For every database, list tables without IDX

create or replace procedure all\_table\_no\_idx is

begin

for o in (select schema\_name, Table\_Name

from DG\_Tables t join DG\_Schema s on t.Database\_ID = s.schema\_ID

where t.Table\_Name not in (

select distinct t.table\_name

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_FK) f on f.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_Index) i on i.Attributes\_ID = a.Attributes\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

where i.Attributes\_ID is null

))

loop

dbms\_output.put\_line(o.schema\_name);

dbms\_output.put\_line(o.Table\_Name);

end loop;

end;

exec all\_table\_no\_idx;

drop procedure all\_table\_no\_idx;

1. For every database, list tables with PK but without FK

create or replace procedure all\_pk\_without\_fk is

begin

for p in (select distinct schema\_name, t.table\_name, Attributes\_Name, p.PK\_ID, f.Attributes\_ID as FK\_ID, i.Attributes\_ID as IDX\_ID

from DG\_Attributes a

inner join DG\_Tables t on t.Table\_ID = a.Table\_id

left outer join DG\_Attributes\_PK p on p.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_FK) f on f.Attributes\_ID = a.Attributes\_ID

left outer join (select Attributes\_ID from DG\_Attributes\_Index) i on i.Attributes\_ID = a.Attributes\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

where f.Attributes\_ID is null and p.Attributes\_ID is not null)

loop

dbms\_output.put\_line(p.schema\_name);

dbms\_output.put\_line(p.table\_name);

dbms\_output.put\_line(p.Attributes\_Name);

dbms\_output.put\_line(p.PK\_ID);

dbms\_output.put\_line(p.FK\_ID);

dbms\_output.put\_line(p.IDX\_ID);

end loop;

end;

exec all\_pk\_without\_fk;

drop procedure all\_pk\_without\_fk;

1. For every database/schema, list tables with PK but without IDX

create or replace procedure all\_pk\_no\_idx is

begin

for q in (Select schema\_name, t.Table\_Name, a.Attributes\_Name, p.PK\_ID,u.IDX\_ID

From DG\_Attributes a

inner join DG\_Attributes\_PK p on a.Attributes\_ID=p.Attributes\_ID

left outer join DG\_Attributes\_Index u on a.Attributes\_ID=u.Attributes\_ID

join DG\_Tables t on a.Table\_id = t.Table\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

where u.Attributes\_ID is null

order by a.Table\_id)

loop

dbms\_output.put\_line(q.schema\_name);

dbms\_output.put\_line(q.Table\_Name);

dbms\_output.put\_line(q.Attributes\_Name);

dbms\_output.put\_line(q.PK\_ID);

dbms\_output.put\_line(q.IDX\_ID);

end loop;

end;

exec all\_pk\_no\_idx;

drop procedure all\_pk\_no\_idx;

1. For every database/schema, list tables with CC

create or replace procedure CC\_tables is

begin

for w in (SELECT s.schema\_name, t.Table\_Name, a.Attributes\_Name, c.CC\_ID

FROM DG\_Attributes a

inner join DG\_Attributes\_ConstCheck c on a.Attributes\_ID = c.Attributes\_ID

join DG\_Tables t on a.Table\_id = t.Table\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID)

loop

dbms\_output.put\_line(w.schema\_name);

dbms\_output.put\_line(w.Table\_Name);

dbms\_output.put\_line(w.Attributes\_Name);

dbms\_output.put\_line(w.CC\_ID);

end loop;

end;

exec CC\_tables;

drop procedure CC\_tables;

1. For every database/schema, list tables with PGM

create or replace procedure PGM\_tables is

begin

for e in (SELECT s.schema\_name, t.Table\_Name, a.Attributes\_Name, p.PGM\_ID

FROM DG\_Attributes a

inner join DG\_Attributes\_Program p on a.Attributes\_ID = p.Attributes\_ID

join DG\_Tables t on a.Table\_id = t.Table\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID)

loop

dbms\_output.put\_line(e.schema\_name);

dbms\_output.put\_line(e.Table\_Name);

dbms\_output.put\_line(e.Attributes\_Name);

dbms\_output.put\_line(e.PGM\_ID);

end loop;

end;

exec PGM\_tables;

drop procedure PGM\_tables;

1. For every database/schema, list all views (View name, View description)

create or replace procedure View\_tables is

begin

for r in (SELECT s.schema\_name, t.Table\_Name, v.View\_ID, i.V\_name, i.description

FROM DG\_Tables t

inner join DG\_View\_Table v on t.Table\_ID = v.Table\_ID

join DG\_Schema s on t.Database\_ID = s.schema\_ID

inner join DG\_Views i on v.View\_ID=i.View\_ID)

loop

dbms\_output.put\_line(r.schema\_name);

dbms\_output.put\_line(r.Table\_Name);

dbms\_output.put\_line(r.View\_ID);

dbms\_output.put\_line(r.V\_name);

dbms\_output.put\_line(r.description);

end loop;

end;

exec View\_tables;

drop procedure View\_tables;

1. For every database/schema, list all table with audit table

create or replace procedure TablesWithAuditTable is

begin

for t in (select s.schema\_name, t.table\_name

from DG\_Tables t

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

inner join DG\_Audit\_Table a on a.Table\_ID = t.Table\_ID

group by s.schema\_name, t.Table\_Name)

loop

dbms\_output.put\_line(t.schema\_name);

dbms\_output.put\_line(t.table\_name);

end loop;

end;

exec TablesWithAuditTable;

drop procedure TablesWithAuditTable;

1. For every database/schema, list schema, users, role, Dept

create or replace procedure UserInformation is

begin

for e in (select s.schema\_name, u.user\_id as users, r.role\_name as roles, un.unit\_name as Dept

from DG\_User\_Database d

inner join DG\_Schema s on s.schema\_ID = d.Database\_ID

inner join DG\_Users u on u.User\_ID = d.User\_ID

inner join DG\_User\_Unit\_Role ur on ur.User\_ID = u.User\_ID

inner join DG\_Role r on r.Role\_ID = ur.Role\_ID

inner join DG\_Unit un on un.Unit\_ID = ur.Unit\_ID

group by s.schema\_name, u.User\_ID, r.Role\_Name, un.Unit\_Name)

loop

dbms\_output.put\_line(e.schema\_name);

dbms\_output.put\_line(e.users);

dbms\_output.put\_line(e.roles);

dbms\_output.put\_line(e.Dept);

end loop;

end;

exec UserInformation;

drop procedure UserInformation;

1. For every database/schema, list schema equipped with triggers to manage change in users

create or replace procedure SchemaTableTriggers as

begin

for x in (select s.schema\_name, t.table\_name, tr.trigger\_id, tr.tname

from DG\_Base\_Trigger\_Table tr

inner join DG\_Tables t on t.Table\_ID = tr.Table\_ID

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

group by s.schema\_name, t.table\_name, tr.trigger\_id, tr.tname)

loop

dbms\_output.put\_line(x.schema\_name);

dbms\_output.put\_line(x.table\_name);

dbms\_output.put\_line(x.trigger\_id);

dbms\_output.put\_line(x.tname);

end loop;

end;

exec SchemaTableTriggers;

drop procedure SchemaTableTriggers;

1. For every database/schema, list schema, dept, role, tables

create or replace procedure DeptTableInformation is

begin

for t in (select t.table\_name as TableName, s.schema\_name, r.role\_name as Roles, un.unit\_name as Dept

from DG\_tables t

inner join DG\_Unit\_Role\_Table ur on ur.Table\_id = t.Table\_ID

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

inner join DG\_Role r on r.Role\_ID = ur.Role\_ID

inner join DG\_Unit un on un.Unit\_ID = ur.Unit\_ID

group by t.Table\_Name, s.schema\_name, r.Role\_Name, un.Unit\_Name)

loop

dbms\_output.put\_line(t.TableName);

dbms\_output.put\_line(t.schema\_name);

dbms\_output.put\_line(t.Roles);

dbms\_output.put\_line(t.Dept);

end loop;

end;

exec DeptTableInformation;

drop procedure DeptTableInformation;

1. For every database/schema, list schemas equipped with triggers to manage change in dept/tables

create or replace procedure TriggersInDept as

begin

for q in (select s.schema\_name, t.table\_name, tr.tname, u.unit\_name as Dept

from DG\_Unit u

inner join DG\_Unit\_Role\_Table un on un.Unit\_ID = u.Unit\_ID

inner join DG\_Base\_Trigger\_Table tr on tr.Table\_ID = un.Table\_id

inner join DG\_Tables t on t.Table\_ID = tr.Table\_ID

inner join DG\_Schema s on s.schema\_ID = t.Database\_ID

group by s.schema\_name, t.table\_name, tr.tname, u.unit\_name)

loop

dbms\_output.put\_line(q.schema\_name);

dbms\_output.put\_line(q.table\_name);

dbms\_output.put\_line(q.tname);

dbms\_output.put\_line(q.Dept);

end loop;

end;

exec TriggersInDept;

drop procedure TriggersInDept;