

```
create view EMP_WINTER as select empno, ename from emp where extract(month from hiredate) in ('01', '02', '12')
```

View created. 0.03 seconds

```
create view TOP_MANAGERS as select emp.ename from ( select mgr, count(mgr) as num from emp group by mgr having count(mgr) >= 3) mgr_ids join emp on emp.empno = mgr_ids.mgr
```

View created. 0.03 seconds

```
select * from emp_winter
```

EMPNO	ENAME
7788	SCOTT
7902	FORD
7369	SMITH
7499	ALLEN
7521	WARD
7876	ADAMS
7900	JAMES
7934	MILLER

8 rows selected. 0.01 seconds

```
select * from top_managers
```

ENAME
BLAKE
KING

2 rows selected. 0.00 seconds

```
drop view emp_winter
```

View dropped. 0.07 seconds

```
drop view top_managers
```

View dropped. 0.07 seconds

```
create sequence DEPT1_SEQ start with 10 increment by 10 maxvalue 9000 minvalue 1 cache 20 nocycle
```

Sequence created. 0.01 seconds

```
create table DEPT1 ( DEPTNO number(3,0) not null enable, DNAME varchar2(50), LOC varchar2(50), constraint DEPT1_PK primary key (DEPTNO) using index enable ) organization index
```

Table created. 0.05 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'ACCOUNTING', 'NEW YORK')
```

1 row(s) inserted. 0.04 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'RESEARCH', 'DALLAS')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'SALES',      'CHICAGO')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'OPERATIONS', 'BOSTON')
```

1 row(s) inserted. 0.01 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'FINANCE',    'LOS ANGELES')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'HR',          'HOUSTON')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'PURCHASE',    'PHOENIX')
```

1 row(s) inserted. 0.01 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'SUPPLY',      'PHILADELPHIA')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'IT',          'SAN ANTONIO')
```

1 row(s) inserted. 0.00 seconds

```
insert into dept1(deptno, dname, loc) values(dept1_seq.nextval, 'ANALYSIS',    'SAN DIEGO')
```

1 row(s) inserted. 0.01 seconds

```
select * from dept1
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	FINANCE	LOS ANGELES
60	HR	HOUSTON
70	PURCHASE	PHOENIX
80	SUPPLY	PHILADELPHIA
90	IT	SAN ANTONIO
100	ANALYSIS	SAN DIEGO

10 rows selected. 0.00 seconds

```
select * from user_sequences
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER	SCALE_FLAG	EXTEND
DEPT1_SEQ	1	9000	10	N	N	20	210	N	N
DEPT_SEQ	1	9000	10	N	N	0	50	N	N
EMP_SEQ	1	90000	10	N	N	0	8000	N	N

3 rows selected. 0.00 seconds

```
drop table dept1
```

Table dropped. 0.53 seconds

Sequence dropped. 0.02 seconds

drop sequence dept1_seq

Sequence dropped. 0.02 seconds

```
create or replace function fac(a in number) return number is
result := 1; else result := a * fac(a - 1); end if; result number; begin
if a <= 1 then end fac; return result; end if;
```

Function created. 0.01 seconds

select fac(5) from dual

FAC(5)
120

Statement processed. 0.01 seconds

```
create or replace function get_work_days(id number) return number is
round((select sysdate from dual) - hiredate), 0) as WORK_DAYS into result from emp where empno = id;
return result; end get_work_days;
```

Function created. 0.02 seconds

select get_work_days(7839) from dual

GET_WORK_DAYS(7839)
14206

Statement processed. 0.01 seconds

```
create or replace procedure get_statistics (num_emps out number, num_depts out number, num_jobs out number,
sum_sal out number) is begin select count(*) into num_emps from emp; select count(*) into num_depts
from dept; select count(distinct job) into num_jobs from emp; select sum(sal) into sum_sal from emp; end
get_statistics;
```

Procedure created. 0.01 seconds

```
declare num_emps number; num_depts number; num_jobs number; sum_sal number; begin get_statistics(num_emps,
num_depts, num_jobs, sum_sal); dbms_output.put_line('Total number of employees: ' || num_emps);
dbms_output.put_line('Total number of departments: ' || num_depts); dbms_output.put_line('Number of various jobs: ' ||
num_jobs); dbms_output.put_line('Summary salary: ' || sum_sal); end;
```

Total number of employees: 14
Total number of departments: 4
Number of various jobs: 5
Summary salary: 29025

Statement processed. 0.00 seconds

```
create table DEBUG_LOG ( ID number(4,0) not null enable, LOG_TIME date, MESSAGE varchar2(50),
IN_SOURCE varchar2(50), constraint DEBUG_LOG_PK primary key (ID) using index enable ) organization index
```

Table created. 0.05 seconds

```
create sequence DEBUG_LOG_SEQ start with 1 increment by 1 maxvalue 9999 minvalue 1 cache 20 nocycle
```

Sequence created. 0.03 seconds

```
create or replace procedure get_date_bounds (oldest_date out date, newest_date out date) is begin select
min(hiredate) into oldest_date from emp; select max(hiredate) into newest_date from emp; end
get_date_bounds;
```

Procedure created. 0.00 seconds

```
declare oldest_date date; newest_date date; begin get_date_bounds(oldest_date, newest_date); insert into
debug_log(id, log_time, message, in_source) values(debug_log_seq.nextval, sysdate, 'Oldest date: ' || oldest_date || ' ' || 'Newest
date: ' || newest_date, 'get_date_bounds'); --dbms_output.put_line('Hire date of employee that works most: ' || oldest_date);
--dbms_output.put_line('Hire date of employee that works least: ' || newest_date); end;
```

1 row(s) inserted. 0.06 seconds

```
declare oldest_date date; newest_date date; begin get_date_bounds(oldest_date, newest_date); insert into
debug_log(id, log_time, message, in_source) values(debug_log_seq.nextval, sysdate, 'Oldest date: ' || oldest_date || ' ' || 'Newest
date: ' || newest_date, 'get_date_bounds'); --dbms_output.put_line('Hire date of employee that works most: ' || oldest_date);
--dbms_output.put_line('Hire date of employee that works least: ' || newest_date); end;
```

1 row(s) inserted. 0.00 seconds

select * from debug_log

ID	LOG_TIME	MESSAGE	IN_SOURCE
1	10/09/2020	Oldest date: 12/17/1980 Newest date: 01/12/1983	get_date_bounds
2	10/09/2020	Oldest date: 12/17/1980 Newest date: 01/12/1983	get_date_bounds

2 rows selected. 0.00 seconds

```
create or replace procedure log_info      (in_info_msg in varchar2,      in_source in varchar2      )      is      PRAGMA      AUTONOMOUS_TRANSACTION;      begin      insert into debug_log(id, log_time, message, in_source)      values(debug_log_seq.nextval,      sysdate, in_info_msg, in_source);      commit;      exception      when others then      return;      end log_info;
```

Procedure created. 0.09 seconds

```
create or replace function division(a in number, b in number)      return number      is      result number;      begin      result      := a / b;      log_info(a || ' / ' || b || ' = ' || result, 'division');      return result;      exception      when others      then      log_info(substr(sqlerrm, 1, 100) || ' (' || a || ' / ' || b || ')', 'division');      end division;
```

Function created. 0.03 seconds

select division(10, 5) from dual

DIVISION(10,5)
2

Statement processed. 0.01 seconds

select division(25, 5) from dual

DIVISION(25,5)
5

Statement processed. 0.01 seconds

select division(25, 0) from dual

ORA-06503: PL/SQL: Function returned without value

select division(0, 0) from dual

ORA-06503: PL/SQL: Function returned without value

select division(-1, 0) from dual

ORA-06503: PL/SQL: Function returned without value

select * from debug_log

ID	LOG_TIME	MESSAGE	IN_SOURCE
1	10/09/2020	Oldest date: 12/17/1980 Newest date: 01/12/1983	get_date_bounds
2	10/09/2020	Oldest date: 12/17/1980 Newest date: 01/12/1983	get_date_bounds
3	10/09/2020	10 / 5 = 2	division
4	10/09/2020	25 / 5 = 5	division
5	10/09/2020	ORA-01476: divisor is equal to zero (25 / 0)	division
6	10/09/2020	ORA-01476: divisor is equal to zero (0 / 0)	division
7	10/09/2020	ORA-01476: divisor is equal to zero (-1 / 0)	division

7 rows selected. 0.00 seconds

drop table debug_log

Table dropped. 0.51 seconds

drop sequence debug_log_seq

Sequence dropped. 0.03 seconds

Run By	RU.ALEXANDER.SMIRNOV@GMAIL.COM
Parsing Schema	FURIOUSTEABAG
Script Started	Friday, October 9, 2020
	7 seconds ago
Elapsed time	190 seconds
Statements Processed	44
Successful	41
With Errors	3
Download	