CSE2004 –DATABASE SYSTEMS –J COMPONENT PROJECT WORK REPORT

# GARBAGE MANAGEMENT SYSTEM

PROJECT REVIEW REPORT

Submitted by

**V.Pragadeesh Dharsha (15BCE0707)**

**Vineet Nalawade (15BCE0284)**

Prepared for

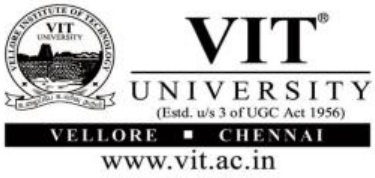
**DATABASE SYSTEMS (CSE2004) – PROJECT COMPONENT**

Submitted To

Prof. Sharmila Banu K

Slot: D1

**School of Computer Science and Engineering**



## ABSTRACT

This report is a complete study made by us for our project on ‘Garbage Management System’. This includes the database for the topic we have taken. The tables where properly normalized and reduced the redundancy. This table can be used to properly organize the garbage brought into the dumping area. It also holds the information on the recyclable garbage and where it is taken to.

## Table of Contents:

## 1.Introduction

This project was done because of the improper use of waste. Since there is a lot of garbage being produced everyday by all and maximum of it are not recycled properly and being thrown everywhere. This leads to foul smell in that place and a major reason for communicable disease. So a proper management and a proper track of all the wastes can reduce the risk of above mentioned problems.

## 2. Project SCOPE

Garbage Management System aims the following processes

a) Track down the information of the waste in the dumping area.
b) Recycle most non-biodegradable materials which comes to the dumping area.
c) Reduce the expense for many recycling factories by cutting down their searching cost for the materials.
d) Implementation of this is few areas can help in reducing the garbage even further

## 3.KEY CONTACTS AND STAKEHOLDERS

| NAME | REG. NUMBER | PHONE NUMBER |
|---|---|---|
| Pragadeesh Dharsha. V | 15BCE0707 | 9944292262 |
| Vineet Nalawade | 15BCE0284 | 8793007005 |

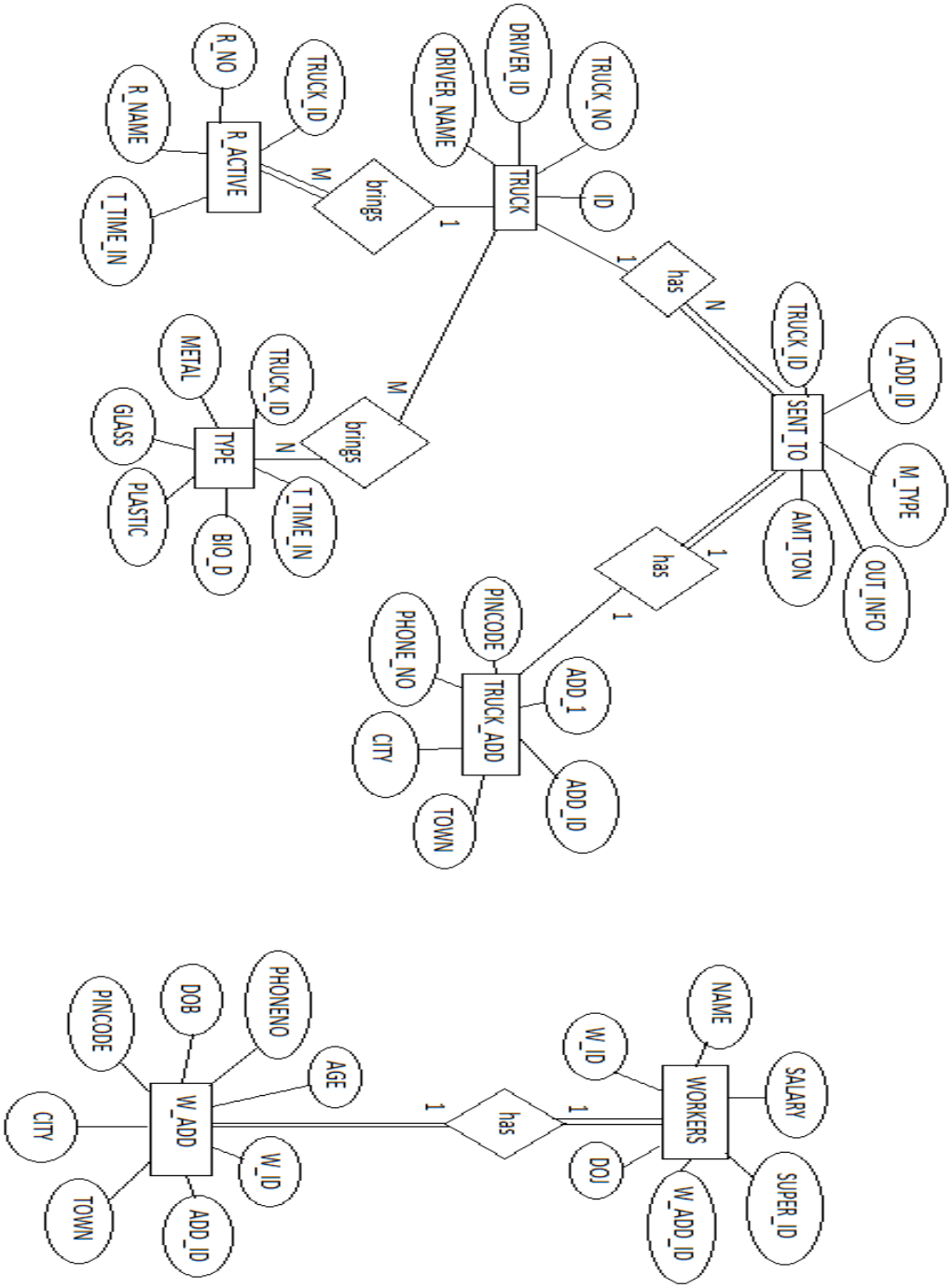
## 4. PROJECT RESOURCE REQUIREMENTS

- Knowledge of ER diagrams
- Knowledge on SQL queries

## 5. Software Resource Requirements

SQL DEVELOPER ORACLE

# 6. ER Diagram


TRUCK
DRIVER_ID
DRIVER_NAME
TRUCK_NO
ID
R_ACTIVE
R_NO
R_NAME
T_TIME_IN
TRUCK_ID
brings
M
1
has
1
N
SENT_TO
TRUCK_ID
T_ADD_ID
M_TYPE
AMT_TON
OUT_INFO
has
1
1
TRUCK_ADD
PINCODE
PHONE_NO
CITY
TOWN
ADD_1
ADD_ID
M
brings
N
TYPE
METAL
GLASS
PLASTIC
BIO_D
TRUCK_ID
T_TIME_IN
has
1
1
WORKERS
NAME
W_ID
SALARY
DOJ
SUPER_ID
W_ADD_ID
W_ADD
PINCODE
CITY
TOWN
DOB
PHONENO
AGE
W_ID
ADD_ID

## 7. Table and Constrains

TRUCK:

| Name | Null? | Type |
|---|---|---|
| ID | NOT NULL | NUMBER PRIMARY KEY |
| TRUCK_NO | NOT NULL | VARCHAR2(10) |
| DRIVER_NAME | NOT NULL | VARCHAR2(20) |
| DRIVER_ID | NOT NULL | VARCHAR2(3) |

GARBAGE:

| Name | Null? | Type |
|---|---|---|
| TRUCK_ID | NOT NULL | NUMBER FOREIGN KEY |
| ADD_ID | NOT NULL | NUMBER FOREIGN KEY |
| IN_INFO | NOT NULL | TIMESTAMP (6) WITH LOCAL TIME ZONE |

TYPE:

| Name | Null? | Type |
|---|---|---|
| TRUCK_ID | NOT NULL | NUMBER FOREIGN KEY |
| METAL | | NUMBER |
| GLASS | | NUMBER |
| BIO_D | | NUMBER |
| PLASTIC | | NUMBER |
| T_TIME_IN | | TIMESTAMP(6) WITH LOCAL TIME ZONE |

R_ACTIVE:

| Name | Null? | Type |
|---|---|---|
| R_NO | NOT NULL | NUMBER |
| R_NAME | | VARCHAR2(25) |
| T_TIME_IN | NOT NULL | TIMESTAMP (6) WITH LOCAL TIME ZONE |
| TRUCK_ID | NOT NULL | NUMBER |

SENT_TO:

| Name | Null? | Type |
|---|---|---|
| TRUCK_ID | NOT NULL | NUMBER |
| T_ADD_ID | NOT NULL | NUMBER |
| M_TYPE | NOT NULL | VARCHAR2(15) |
| AMT_TON | NOT NULL | NUMBER |
| OUT_INFO | NOT NULL | TIMESTAMP(6) WITH LOCAL TIME ZONE |

TRUCK_ADD:

| Name | Null? | Type |
|---|---|---|
| ADD_ID | NOT NULL | NUMBER |
| ADD_1 | NOT NULL | VARCHAR2(100) |
| TOWN | NOT NULL | VARCHAR2(50) |
| CITY | NOT NULL | VARCHAR2(15) |
| PINCODE | NOT NULL | NUMBER(6) |
| PHONE_NO | NOT NULL | NUMBER(10) |

WORKERS:

| Name | Null? | Type |
|---|---|---|
| W_ID | NOT NULL | NUMBER |
| SALARY | NOT NULL | NUMBER |
| SUPER_ID | | NUMBER |
| W_ADD_ID | NOT NULL | NUMBER |
| DOJ | NOT NULL | DATE |
| NAME | NOT NULL | NVARCHAR2(50) |

W_ADD:

| Name | Null? | Type |
|---|---|---|
| W_ID | NOT NULL | NUMBER |
| W_ADD1 | NOT NULL | VARCHAR2(50) |
| W_TOWN | NOT NULL | VARCHAR2(15) |
| W_CITY | NOT NULL | VARCHAR2(15) |
| W_PINCODE | NOT NULL | NUMBER(6) |
| DOB | NOT NULL | DATE |
| W_PHONE_NO | NOT NULL | NUMBER(10) |
| AGE | NOT NULL | NUMBER(2) |

## 8. LIST OF PROJECT SPECS ACCOMPLISHED

I. Multi-lingual data
II. Bulk Insert
III. Minimum 6 datatypes
IV. Demonstration of DUAL table usage
V. JOINS (Minimum FOUR types and multiple usage)
VI. SUBQUERIES (Minimum FOUR with nesting of 'FROM' and 'WHERE' clauses)
VII. GROUP BY, HAVING, ORDER BY
VIII. Effective Usage of SYSDATE

# 9. DESIGN FLAWS AND FIXES

**TRUCK** (ID, TRUCK_NO, DRIVER_NAME, DRIVER_ID, ADD_ID, ADD_1, TOWN, CITY, PINCODE, PHONENO)

**GARBAGE** (TRUCK_ID, ADD_ID)

**TYPE** (TRUCK_ID, METAL, GLASS, BIO_D, PLASTIC, R_NO, R_NAME , T_TIME_IN)

**SENT_TO** (TRUCK_ID, T_ADD_ID, M_TYPE, AMOUNT_TON, OUT_INFO)

**WORKERS** (W_ID, SALARY, SUPER_ID, W_ADD_ID, DOJ, W_ID, ADD_1, AGE, DOB, TOWN, CITY, PINCODE, PHONENO)

## After normalization:

**TRUCK** (ID, TRUCK_NO, DRIVER_NAME, DRIVER_ID)

**GARBAGE** (TRUCK_ID, ADD_ID)

**TYPE** (TRUCK_ID, METAL, GLASS, BIO_D, PLASTIC, T_TIME_IN)

**R_ACTIVE** (TRUCK_ID, R_NO, R_NAME, T_TIME_IN)

**SENT_TO** (TRUCK_ID, T_ADD_ID, M_TYPE, AMOUNT_TON, OUT_INFO)

**TRUCK_ADD** (ADD_ID, ADD_1, TOWN, CITY, PINCODE, PHONENO)

**WORKERS** (W_ID, SALARY, SUPER_ID, W_ADD_ID, DOJ)

**W_ADD** (W_ID, ADD_1, AGE, DOB, TOWN, CITY, PINCODE, PHONENO)

## 10. DATA FLAWS AND FIXES

- Changed data type for the attribute ‘name’ in workers table as ‘NVARCHAR2’ from ‘VARCHAR’ for addition of multi-lingual data.

- Size of the attribute ‘w_add1’ in the table W_add has been increased to fit the required amount of data.

## 11. COMPILATION ERRORS AND FIXES

1. Error: foreign key

```
SQL> alter table w_add add constraint w_add_fk (w_id) reference workers(w_id);
alter table w_add add constraint w_add_fk (w_id) reference workers(w_id)
                                                 *
ERROR at line 1:
ORA-00904: : invalid identifier
```

Fixed:

```
SQL> alter table w_add add constraint w_add_fk foreign key (w_id) references workers(w_id);

Table altered.
```

2. Error: having clause

```
SQL> select truck_id,sum(metal) from type group by truck_id
  2  where sum(metal)=40;
where sum(metal)=40
*
ERROR at line 2:
ORA-00933: SQL command not properly ended
```

Fixed

```
SQL> select truck_id,sum(metal) from type group by truck_id
  2  having sum(metal)=
  3  (select max(sum(metal)) from type group by truck_id);
```

3. Error: group by with rownum

```
SQL> select r_name,rownum from r_active group by r_name;
select r_name,rownum from r_active group by r_name
              *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

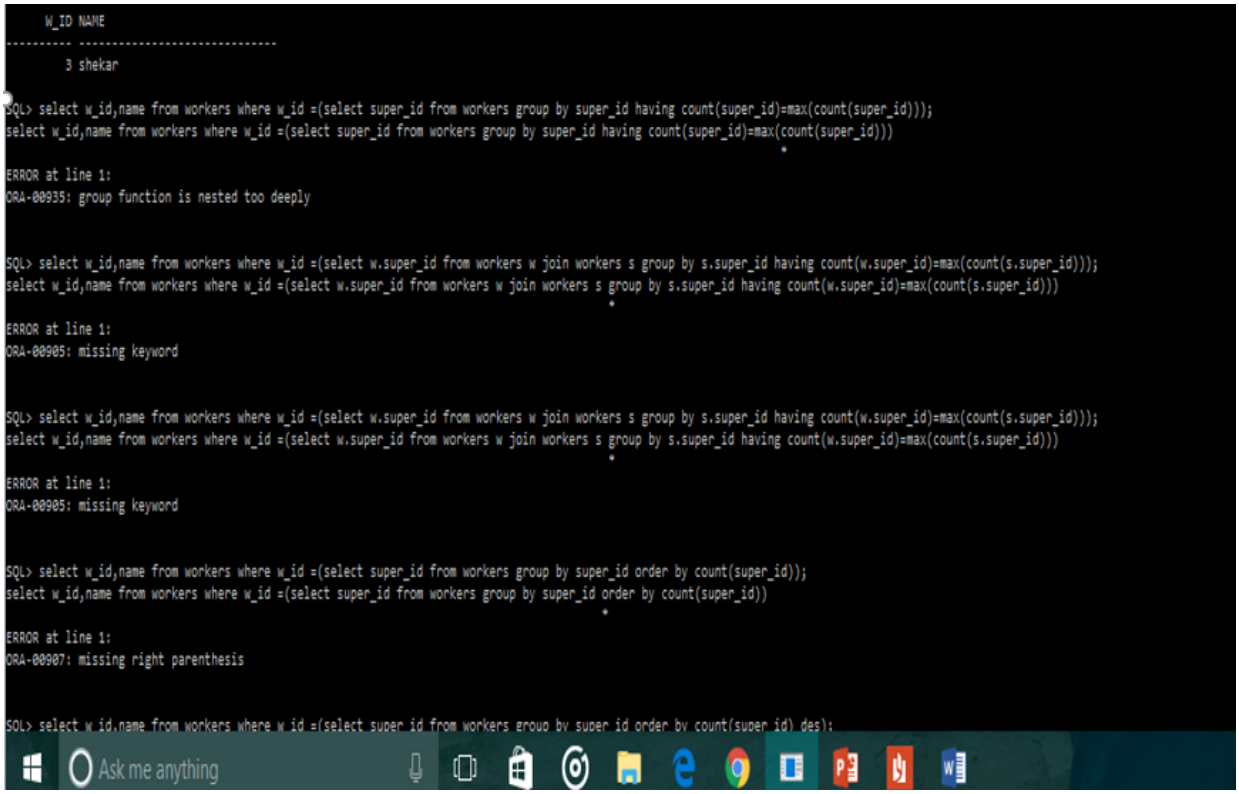
Fixed

```
SQL> with r_num as
  2  (
  3    select r_name,count(r_name) as ttl_dump,
  4    rank() over(order by count(r_name) desc) as row_num
  5    from r_active
  6    group by r_name
  7  )
  8  select  r_name,ttl_dump
  9  from r_num
 10  where row_num<3;
```

# 12. LIST OF REVIEW 2 QUERIES AND SOLUTION

Oracle SQLPLUS

```
10 rows selected.

SQL> select e.name as worker,s.name as supervisor from workers e join workers s on e.super_id=s.w_id;

WORKER                         SUPERVISOR
------------------------------ ------------------------------
shekar                         mareesan
chandran                       mareesan
surya                          ragu
karthi                         ragu
sai                            gokul
kathiresan                     gokul
swaroop                        gokul

7 rows selected.

SQL> select w_id,name from workers where w_id =(select super_id from workers group by super_id having max(count(super_id)));
select w_id,name from workers where w_id =(select super_id from workers group by super_id having max(count(super_id)))
                                                                                 *
ERROR at line 1:
ORA-00935: group function is nested too deeply


SQL> select w_id,name from workers where w_id =(select max(count(super_id)) from workers group by super_id);

      W_ID NAME
---------- ------------------------------
         3 shekar
```




i. Display the city which brings maximum metal at once

```
SELECT t.truck_id, metal, ta.city
FROM type t  JOIN  garbage g  ON  t.truck_id=g.truck_id
JOIN  truck_add ta  ON  g.add_id=ta.add_id
WHERE  metal=(SELECT MAX(metal) FROM type);
```

ii. Display the supervisor name with maximum number of workers under them

```
SELECT name  FROM workers  WHERE w_id=(
SELECT  super_id  FROM workers GROUP  BY super_id  HAVING
count(super_id)=( SELECT  MAX(count(super_id)) FROM  workers
GROUP BY super_id  HAVING  super_id!=0 ) AND  super_id!=0 );
```

# 13. LIST OF REVIEW 3 QUERIES AND SOLUTION

Tried with ma’am:

```
MAX(SUM(R_NO))
--------------
           95

SQL> select trunc(t_time_in),sum(r_no) from r_active where sum(r_no)=(
  2  select max(sum(r_no)) from r_active group by trunc(t_time_in) order by sum(r_no) desc;)
  3  ;
select trunc(t_time_in),sum(r_no) from r_active where sum(r_no)=(
                                                 *
ERROR at line 1:
ORA-00934: group function is not allowed here


SQL> select trunc(t_time_in) from r_active where sum(r_no)=(
  2  select max(sum(r_no)) from r_active group by trunc(t_time_in) order by sum(r_no) desc);
select trunc(t_time_in) from r_active where sum(r_no)=(
                                            *
ERROR at line 1:
ORA-00934: group function is not allowed here


SQL> select trunc(r.t_time_in) from r_active r s where r.sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) order by s.sum(r_no) desc);
select trunc(r.t_time_in) from r_active r s where r.sum(r_no)=(
                                          *
ERROR at line 1:
ORA-00933: SQL command not properly ended


SQL> select trunc(r.t_time_in) from r_active r where r.sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) order by s.sum(r_no) desc);
select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) order by s.sum(r_no) desc)
                                                                                *
ERROR at line 2:
ORA-00907: missing right parenthesis


SQL> select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) ;

MAX(SUM(S.R_NO))
----------------
              95

SQL> select trunc(r.t_time_in) from r_active r where r.sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) ;
select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in)
*
ERROR at line 2:
ORA-00921: unexpected end of SQL command


SQL> select trunc(r.t_time_in) from r_active r where r.sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) );
select trunc(r.t_time_in) from r_active r where r.sum(r_no)=(
                                               *
ERROR at line 1:
ORA-00904: "R"."SUM": invalid identifier
```

```
SQL> select rownum,sum(r_no) from r_active group by trunc(t_time_in) order by sum(r_no) desc;
select rownum,sum(r_no) from r_active group by trunc(t_time_in) order by sum(r_no) desc
       *
ERROR at line 1:
ORA-00979: not a GROUP BY expression


SQL> select sum(r_no) from r_active group by trunc(t_time_in) having rownum=1 order by sum(r_no) desc;
select sum(r_no) from r_active group by trunc(t_time_in) having rownum=1 order by sum(r_no) desc
                                                               *
ERROR at line 1:
ORA-00979: not a GROUP BY expression


SQL> select sum(r_no) from r_active group by trunc(t_time_in) having rownum=1 order by sum(r_no) desc;
select sum(r_no) from r_active group by trunc(t_time_in) having rownum=1 order by sum(r_no) desc
                                                               *
ERROR at line 1:
ORA-00979: not a GROUP BY expression


SQL> select trunc(t_time_in) from r_active group by trunc(t_time_in) order by sum(r_no) desc;

TRUNC(T_T
---------
06-NOV-16
08-NOV-16

SQL> select trunc(t_time_in) from r_active group by trunc(t_time_in) having sum(r_no)=max(sum(r_no));
select trunc(t_time_in) from r_active group by trunc(t_time_in) having sum(r_no)=max(sum(r_no))
                                                                              *
ERROR at line 1:
ORA-00935: group function is nested too deeply


SQL> select trunc(t_time_in) from r_active group by trunc(t_time_in) having sum(r_no)=95;

TRUNC(T_T
---------
06-NOV-16

SQL> select trunc(t_time_in) from r_active group by trunc(t_time_in) having sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) );

TRUNC(T_T
---------
06-NOV-16

SQL> select trunc(t_time_in),sum(r_no) from r_active group by trunc(t_time_in) having sum(r_no)=(
  2  select max(sum(s.r_no)) from r_active s group by trunc(s.t_time_in) );

TRUNC(T_T  SUM(R_NO)
--------- ----------
06-NOV-16         95
```

a) Display the date on which particular radioactive substance was dumped the most.

```
SELECT TRUNC(t_time_in)

FROM r_active

GROUP BY r_name, TRUNC(t_time_in)

HAVING  COUNT(r_name)=(

SELECT MAX(COUNT(r_name))

FROM r_active

GROUP BY  r_name,trunc(t_time_in)

HAVING r_name='uranium256')

AND r_name='uranium256';
```

```
SQL> select trunc(t_time_in)
  2  from r_active
  3  group by r_name,trunc(t_time_in)
  4  having count(r_name)=
  5  ( select max(count(r_name))
  6  from r_active
  7  group by r_name,trunc(t_time_in)
  8  having r_name='uranium256')
  9  and r_name='uranium256';

TRUNC(T_T
---------
06-NOV-16
```

b) Display the top two radioactive substance which was dumped the most using **'rownum'**

```
with r_num as
(  select r_name,count(r_name) as ttl_dump,
 rank() over(order by count(r_name) desc) as row_num
 from r_active
group by r_name  )
select  r_name,ttl_dump
from r_num
where row_num<3;end;
```

```
SQL> with r_num as
  2  (
  3    select r_name,count(r_name) as ttl_dump,
  4    rank() over(order by count(r_name) desc) as row_num
  5    from r_active
  6    group by r_name
  7  )
  8  select  r_name,ttl_dump
  9  from r_num
 10  where row_num<3;

R_NAME                    TTL_DUMP
------------------------- ----------
uranium256                         4
plutonium                          3
```

c) Display the top two radioactive substance which was dumped the most using **'cursors'**

```
declare
cursor c is select r_name,count(r_name) from r_active group by r_name
order by count(r_name) desc;
r_info c%rowtype;
x number:=0;
begin
open c;
 fetch c into r_info;
 dbms_output.put_line(r_info.r_name);
 fetch c into r_info;
dbms_output.put_line(r_info.r_name);
end;
/
```

```
SQL> declare
  2  cursor c is select r_name,count(r_name) from r_active group by r_name order by count(r_name) desc;
  3  r_info c%rowtype;
  4  x number:=0;
  5  begin
  6  open c;
  7    fetch c into r_info;
  8    dbms_output.put_line(r_info.r_name);
  9    fetch c into r_info;
 10    dbms_output.put_line(r_info.r_name);
 11  end;
 12  /
uranium256
plutonium

PL/SQL procedure successfully completed.
```

## 14. PL/SQL SP/FUNCTION/CURSOR/TRIGGER AND SOLUTION

1. Display the date where maximum number of vehicles came for dumping

```
SELECT  COUNT(TRUNC(t_time_in)) AS max_vec_in,
TRUNC(t_time_in) AS date_in
FROM    type      GROUP BY          TRUNC(t_time_in)
HAVING    COUNT(TRUNC(t_time_in))=  (
SELECT  MAX(COUNT(TRUNC(t_time_in))  )
FROM  type  GROUP BY  trunc(t_time_in));
```

2. Display the current date and time

```
SELECT TO_TIMESTAMP(SYSDATE) FROM dual;
```

3. Display the city name which took more amount of glass from the garbage

```
SELECT city,m_type,amt_ton
FROM    sent_to  JOIN  truck_add  ON  t_add_id=add_id
WHERE amt_ton=(
SELECT  MAX(amt_ton) FROM sent_to  WHERE  m_type='glass' );
```

4. Display the date of upcoming day and end of this month

```
SELECT  SYSDATE, TO_CHAR (sysdate,'day') day,
NEXT_DAY(sysdate,'friday') friday,
LAST_DAY  (sysdate) END_OF_MONTH  FROM dual;
```

5. Display the worker id and their name who has more than four months of experience

```
SELECT  w_id,name, TRUNC(MONTHS_BETWEEN(sysdate,doj)/30)  AS
no_of_months_worked  FROM  workers WHERE
TRUNC(MONTHS_BETWEEN(sysdate,doj)/30)  > 4;
```

6. Display the city which brings maximum metal at once

```
SELECT t.truck_id, metal, ta.city
FROM type t   JOIN   garbage g  ON  t.truck_id=g.truck_id
JOIN  truck_add ta  ON  g.add_id=ta.add_id
WHERE  metal=(SELECT MAX(metal) FROM type);
```

7. Display the supervisor name with maximum number of workers under them

```
SELECT name   FROM workers  WHERE w_id=(
SELECT  super_id  FROM workers GROUP  BY super_id  HAVING
count(super_id)=(  SELECT  MAX(count(super_id))  FROM  workers
GROUP BY super_id  HAVING  super_id!=0 ) AND  super_id!=0 );
```

8. Display the name of all the employees using cursor

```
DECLARE
cursor c is SELECT name NAME workers;
BEGIN
for x in c loop
dbms_output.put_line(x.name);
end loop;
end;
/
```

9. Display the date after two months

```
SELECT sysdate, ADD_MONTHS(sysdate,2) after_2_months
FROM dual;
```

10. Display the truck id's which left the place within 24 hours

```
SELECT t.truck_id
FROM type t  JOIN  sent_to s  ON  t.truck_id=s.truck_id
WHERE MONTHS_BETWEEN(out_info,t_time_in)/365 =0;
```

11. Display the day, month and year in different language

```
SELECT
TO_CHAR(sysdate,'Day MONTH,yyyy','nls_date_language=spanish')
FROM dual;
```

12. Display the date which had maximum number of radioactive dumpings

```
SELECT TRUNC(t_time_in), SUM(count_no)
FROM  r_active
GROUP BY TRUNC(t_time_in)
HAVING SUM(count_no)=(
SELECT MAX(SUM(count_no))  FROM  r_active
GROUP BY TRUNC(t_time_in) );
```

## 15. LIST OF RESOURCES REFERRED WITH SNAPSHOT OR TEXT

- ✓ https://www.youtube.com/watch?v=LeALKsu1MiY
- ✓ https://www.youtube.com/watch?v=H18UWBoHhHY
- ✓ https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions193.htm

**Examples**

The following example converts a character string to a timestamp. The character string is not in the default `TIMESTAMP` format, so the format mask must be specified:

```
SELECT TO_TIMESTAMP ('10-Sep-02 14:10:10.123000', 'DD-Mon-RR HH24:MI:SS.FF')
   FROM DUAL;

TO_TIMESTAMP('10-SEP-0214:10:10.123000','DD-MON-RRHH24:MI:SS.FF')
---------------------------------------------------------------------------
10-SEP-02 02.10.10.123000000 PM
```

- ✓ http://www.orafaq.com/wiki/NVARCHAR2

```
SQL> INSERT into nvarchar2_test values (to_nchar('働 (どう はたら'))
```

```
1 row created.
```

```
SQL> SELECT to_nchar(col1) from nvarchar2_test
```

```
TO_NCHAR(COL1)
--------------
働 (どう はたら
```

- ✓ https://www.youtube.com/watch?v=_0PUOgyqEYI