CORE isimli bir package oluşturulacak. Paket altında oluşturulacak function ve procedure ler aşağıdaki gibidir.

- **schema_size(pschema,ptype)**(ptype ile dışarıdan parametre alıp ona göre sonuç dönecek. Ptype boş gelirse schemanın toplam boyutunu getirecek. Ptype Table gelirse schemaya ait tabloların toplam boyutunu getirecek. Index gelirse schemaya ait indexlerin toplam boyutlarını getirecek. Mview gelirse materialized view toplam boyutunu getirecek) Function
- **schema_size_detail(pschema,ptype)**(ptype ile dışarıdan parametre alıp ona göre sonuç dönecek. Ptype Table gelirse schemaya ait Boyutu 1GB ve üstü tabloları getirecek. Index gelirse schemaya ait boyutu 1GB ve üstü olan indexleri getirecek. Mview gelirse materialized viewleri ve boyutlarını getirecek) Function
- **table_size(pschema,ptable,ptype)**(Schemaya ait tabloların boyutlarını getirecek. ptype ile içeriye bir parametre alınacak. Ptype boş gelirse tablo+blob+index boyutu toplanıp yazılacak. Bu parametreye All (gelirse table, blob, index ayrı ayrı tümünü getirip yazacak. Table gelirse sadece tablo boyutu, index gelirse sadece o tabloya ait index boyutu ve blob gelirse sadece o tabloya ait blob saha boyutunu yazacak.) Function
- **tbs_free(ptbs)**(Tablespace in boş kalan alanını ve son 7 günlük büyümesini getirecek. Boş alanı getirirken geçtiğimiz günlerde yapılan gerçek boş alanı bulma kısmı da kullanılacak.) Function
- **tbs_cnt**(Tablespacelerin boş kalan boyutlarını, bir tabloya yazacak. Bunun için oluşturulacak tablo aşağıdadır.) Procedure

tbs_log

ID number,
tablespace_Name varchar2(50),
tablespace_id number,
total_size number,
free_size number,
available_size number,
control_date date default sysdate

SCHEMA_SIZE

```
CREATE OR REPLACE FUNCTION DATADK.schema_size (pschema     VARCHAR2,
                                               ptype      VARCHAR2)
   RETURN varchar2
   IS
   total_schema varchar2(15);
   total_table  varchar2(15);
   total_index  varchar2(15);
   total_mview  varchar2(15);
BEGIN

   IF ptype IS NULL
   THEN
        SELECT SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_SCHEMA into total_schema
          FROM dba_segments
         WHERE owner = pschema;
      return total_schema || ' GB';

   ELSIF (ptype = 'TABLE')
      THEN
           SELECT  SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_TABLE into total_table
             FROM dba_segments
            WHERE owner = pschema AND segment_type = ptype;
            return total_table || ' GB';
```

```
      ELSIF (ptype  = 'INDEX')
         THEN
              SELECT SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_TABLE into total_index
                FROM dba_segments
               WHERE owner = pschema AND segment_type = ptype;
            return total_index ||' GB';


         ELSIF (ptype = 'MATERIALIZED VIEW')
            THEN
            select SUM (bytes) / 1024 / 1024 / 1024  as SIZE_MVİEW into total_mview
            from dba_mviews a, dba_segments b
            where a.owner = pschema and b.owner = pschema
            and a.mview_name = b.segment_name and segment_type = ptype;
           return total_mview ||' GB';
            IF total_mview IS NULL
            THEN
            return 0;
            ELSE
                return total_mview;
            END IF;
       END IF;
          END schema_size;
```

```
select datadk.schema_size('DATADK', NULL) from dual;
```

DATADK.SCHEMA_SIZE('DATADK',NULL)
.0023193359375 GB

```
 select datadk.schema_size('DATADK', 'TABLE') from dual;
```

DATADK.SCHEMA_SIZE('DATADK','TABLE')
.00177001953125 GB

```
select datadk.schema_size('DATADK', 'INDEX') from dual;
```

DATADK.SCHEMA_SIZE('DATADK','INDEX')
.00042724609375 GB

Materialized view oluşturma:

```
CREATE MATERIALIZED VIEW mview BUILD IMMEDIATE REFRESH FORCE ON DEMAND AS
SELECT * FROM datadk.iller;
```

```
select datadk.schema_size('DATADK', 'MATERIALIZED VIEW') from dual;
```

DATADK.SCHEMA_SIZE('DATADK','MATERIALIZEDVIEW')
.00006103515625 GB

SCHEMA_SIZE_DETAIL

```
CREATE OR REPLACE FUNCTION DATADK.schema_size_detail (pschema     VARCHAR2 ,
                                                      ptype       VARCHAR2 )
   RETURN varchar2
is
    total_value     VARCHAR2 (500);
   total            varchar2(5000);
   CURSOR c1
```

```
   IS
        SELECT owner,
             segment_name,
             segment_type,
            bytes
             FROM dba_segments
        WHERE owner = pschema AND segment_type = 'TABLE';

  CURSOR c2
   IS
     SELECT  owner,
             segment_name,
             segment_type,
            bytes
           FROM dba_segments
           WHERE owner = pschema AND segment_type = 'INDEX';

  cursor c3
  IS
   SELECT ds.owner,
             ds.segment_name,
             ds.segment_type,
             ds.bytes,
             mv.mview_name
        FROM dba_segments ds
             LEFT OUTER JOIN dba_mviews mv ON ds.segment_name = mv.mview_name
       WHERE     ds.owner = pschema
             AND ds.segment_type = 'TABLE'
             AND mv.mview_name = ds.segment_name;

BEGIN
   IF ptype = 'TABLE'
      THEN
       FOR i IN c1
     LOOP
           SELECT bytes/1024/1024/1024
           INTO total_value
             FROM dba_segments
            WHERE bytes> 1073741824 AND owner = pschema AND segment_type=ptype AND
segment_name=i.segment_name;
         total:= total || CHR(13) ||i.segment_name ||' tablosunun boyutu: ' || total_value ||
' GB';
    END LOOP;

      ELSIF (ptype = 'INDEX')
      THEN
      FOR i IN c2
     LOOP
         SELECT bytes/1024/1024/1024
           INTO total_value
             FROM dba_segments
            WHERE bytes> 1073741824 AND owner = pschema AND segment_type=ptype AND
segment_name=i.segment_name;
         total:= total || CHR(13) || i.segment_name ||' indexinin boyutu: ' ||total_value ||
' GB';

       END loop;

      ELSIF (ptype = 'MATERIALIZED VIEW')
      THEN
      FOR i IN c3
     LOOP

            select bytes/1024/1024/1024 into total_value from dba_segments ds,dba_mviews mv
where ds.segment_name=mv.mview_name and ds.segment_type='TABLE'
            and ds.owner=UPPER(pschema)
            and mv.mview_name=i.segment_name and bytes > 1073741824 order by bytes desc;
            total:= total|| CHR(13) ||  i.segment_name || ' isimli MATERALİZED_VIEWin boyutu
:' || total_value||' GB';
        END loop
      END IF;
```

```
   RETURN total;
END schema_size_detail;
```

```
select datadk.schema_size_detail('DATADK', 'TABLE') from dual;
```

```
2   MVIEW tablosunun boyutu: .00006103515625 GB
3   PERSONEL_IZIN tablosunun boyutu: .00006103515625 GB
4   JOB_INSERT tablosunun boyutu: .00006103515625 GB
5   PERSONEL2 tablosunun boyutu: .00006103515625 GB
6   JOB_INSERT_080120 tablosunun boyutu: .00006103515625 GB
7   GEC_KALMA tablosunun boyutu: .00006103515625 GB
8   PER_IZIN_121119 tablosunun boyutu: .00006103515625 GB
9   TEST123 tablosunun boyutu: .0006103515625 GB
10  TABLESPACEADI tablosunun boyutu: .00006103515625 GB
11  DEPARTMANLAR tablosunun boyutu: .00006103515625 GB
12  URUNLER tablosunun boyutu: .00006103515625 GB
13  TEST1234 tablosunun boyutu: .00006103515625 GB
14  PER_MAAS_LOG tablosunun boyutu: .00006103515625 GB
15  ILLER tablosunun boyutu: .00006103515625 GB
16  KATEGORILER tablosunun boyutu: .00006103515625 GB
17  CALISANLAR tablosunun boyutu: .00006103515625 GB
18  PERSONEL tablosunun boyutu: .00006103515625 GB
19  PERSONEL_HAREKETLERI tablosunun boyutu: .00006103515625 GB
20  INDEX_CONTROL tablosunun boyutu: .00006103515625 GB
21  SUPPLIERS tablosunun boyutu: .00006103515625 GB
```

```
select datadk.schema_size_detail('DATADK', 'INDEX') from dual;
```

```
1
2   PERSONEL_PK indexinin boyutu: .00006103515625 GB
3   SYS_C007226 indexinin boyutu: .00006103515625 GB
4   CALISANLAR_CALIS_IDX indexinin boyutu: .00006103515625 GB
5   UNQ_CALISANLAR indexinin boyutu: .00006103515625 GB
6   INDX_CALISANLAR_FUNC_BASED indexinin boyutu: .00006103515625 GB
7   DENEME_INDEX indexinin boyutu: .00006103515625 GB
8   IDX_REVER indexinin boyutu: .00006103515625 GB
```

```
select datadk.schema_size_detail('DATADK', 'MATERIALIZED VIEW') from dual;
```

```
1
2   Girilen şemaya ait MVIEW isimli MATERALİZED_VIEWin boyutu :.00006103515625 GB
```

TABLE_SIZE

```
/* Formatted on 22.03.2020 13:28:30 (QP5 v5.256.13226.35510) */
CREATE OR REPLACE FUNCTION DATADK.table_size (pschema     VARCHAR2,
                                              ptable      VARCHAR2,
                                              ptype       VARCHAR2)
   RETURN VARCHAR2
IS
   p_total            VARCHAR2(1000) ;
   index_total_size   VARCHAR2 (1000);
   tbl_size           VARCHAR2 (1000);
   lob_size           VARCHAR2 (1000);
   total              VARCHAR2 (1000);

BEGIN
   IF ptype IS NULL
   THEN
     SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO index_total_size
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                                FROM dba_indexes
                               WHERE     table_OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));
```

```
      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO tbl_size
        FROM DBA_SEGMENTS
       WHERE     SEGMENT_NAME = UPPER (ptable)
             AND SEGMENT_TYPE = 'TABLE'
             AND owner = UPPER (pschema);

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO lob_size
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                              FROM dba_lobs
                             WHERE     OWNER = UPPER (pschema)
                                   AND table_name = UPPER (ptable));


             total := total || CHR (13) || pschema || ' şemasının '|| ptable ||' tablosuna
ait TABLE, INDEX ve LOB toplam boyutu : ' ||
             to_char(index_total_size + tbl_size +lob_size)  || 'GB';
     END IF;

   IF ptype = 'ALL'
   THEN
      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO index_total_size
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                              FROM dba_indexes
                             WHERE     table_OWNER = UPPER (pschema)
                                   AND table_name = UPPER (ptable));

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO tbl_size
        FROM DBA_SEGMENTS
       WHERE     SEGMENT_NAME = UPPER (ptable)
             AND SEGMENT_TYPE = 'TABLE'
             AND owner = UPPER (pschema);

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO lob_size
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                              FROM dba_lobs
                             WHERE     OWNER = UPPER (pschema)
                                   AND table_name = UPPER (ptable));

      total :=
            total
         || CHR (13)
         ||CHR (13)  || pschema || (' şemasının ' || ptable || ' tablosuna ait');
      total := total || CHR (13) || ('INDEX boyutu : ' || index_total_size || 'GB');
      total := total || CHR (13) || ('BLOB boyutu : ' || lob_size || 'GB');
      total := total || CHR (13) || ('TABLO boyutu : ' || tbl_size || 'GB');
   END IF;



   IF ptype = 'TABLE'
   THEN
      SELECT SUM (BYTES) / 1024 / 1024 / 1024
        INTO p_total
        FROM DBA_SEGMENTS
       WHERE SEGMENT_NAME = UPPER (ptable) AND owner = UPPER (pschema);
      total := total || CHR (13)  || ( pschema || ' şemasının ' || ptable  || ' tablosuna ait
TABLO boyutu: ' || p_total || ' GB') ;
   END IF;

   IF ptype = 'INDEX'
   THEN
      SELECT SUM (bytes) / 1024 / 1024 / 1024
        INTO p_total
```

```
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                                FROM dba_indexes
                               WHERE     table_OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      total := total || CHR (13) || pschema || (' şemasının '  || ptable  || ' tablosuna ait
INDEX boyutu: ' || p_total || 'GB');
   END IF;



   IF ptype = 'LOBSEGMENT'
   THEN
      SELECT NVL (SUM (bytes) / 1024 / 1024 / 1024, 0)
        INTO p_total
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                                FROM dba_lobs
                               WHERE     OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      total := total|| CHR (13)|| (pschema || ' şemasının ' || ptable || ' tablosuna ait LOB
SEGMENT boyutu: '  || p_total  || ' GB');
   END IF;

   RETURN total;
END table_size;
/
```

```
select datadk.table_size('DATADK','ILLER',NULL) from dual;
```

```
1
2   DATADK şemasının ILLER tablosuna ait TABLE, INDEX ve LOB toplam boyutu : .000244140625GB
```

```
select datadk.table_size('DATADK','ILLER', 'ALL') from dual;
```

```
3   DATADK şemasının ILLER tablosuna ait
4   INDEX boyutu : .00012207031250GB
5   BLOB boyutu : .00006103515625GB
6   TABLO boyutu : .00006103515625GB
```

```
select datadk.table_size('DATADK','PERSONEL', 'LOBSEGMENT') from dual;
```

```
1
2   DATADK şemasının PERSONEL tablosuna ait LOB SEGMENT boyutu: 0 GB
```

```
select datadk.table_size('DATADK','ILLER', 'TABLE') from dual;
```

```
1
2   DATADK şemasının ILLER tablosuna ait TABLO boyutu: .00006103515625 GB
```

```
select datadk.table_size('DATADK','ILLER', 'INDEX') from dual;
```

```
1
2       DATADK şemasının ILLER tablosuna ait INDEX boyutu: .0001220703125GB
```

BLOB ALAN EKLENMESI:

```
alter table ILLER add x_blob blob;
insert into DATADK.iller(il_id, il_adi,x_blob) values(09,'Aydın',hextoraw('453d7ggga34'));
```

TBS_FREE

```
CREATE OR REPLACE FUNCTION DATADK.tbs_free(ptbs VARCHAR2)

RETURN VARCHAR2
IS
growth varchar2(1000);
free varchar2(1000);
 cursor cr1 IS select thedate, (mbsize - prev_mbsize) diff
                    from (select thedate, mbsize, lag(mbsize, 1) over(order by r) prev_mbsize
                    from (select rownum r, thedate, mbsize
                    from (select trunc(thedate) thedate, max(mbsize) mbsize
                    from (select to_date(to_char(ss.begin_interval_time -1,'YYYY-MON-DD
HH24:MI:SS'),'YYYY-MON-DD HH24:MI:SS') thedate,
                    round((us.tablespace_usedsize * bs.value)/1024/1024,2) mbsize
                    from dba_hist_tbspc_space_usage us,v$tablespace ts,dba_hist_snapshot
ss,v$parameter bs
                    where us.snap_id = ss.snap_id
                    and us.tablespace_id = ts.ts#
                    and ts.name =ptbs
                    and bs.name = 'db_block_size'
                    and ss.begin_interval_time > sysdate-7
               and ss.begin_interval_time < trunc(sysdate))
                    group by trunc(thedate)
                    order by trunc(thedate))));

cursor cr2 IS select q1.tablespace_name,q1.usable,q2.tfree from(SELECT
           substr(A.tablespace_name,1,14) tablespace_name,

sum(trunc(decode(A.autoextensible,'YES',A.MAXSIZE-A.bytes+b.free,'NO',b.free)/1024/1024))
usable
    FROM  (
         SELECT file_id, file_name,
                tablespace_name,
                autoextensible,
                bytes,
                decode(autoextensible,'YES',maxbytes,bytes) maxsize
         FROM   dba_data_files
         GROUP BY file_id, file_name,
                  tablespace_name,
                  autoextensible,
                  bytes,
                  decode(autoextensible,'YES',maxbytes,bytes)
         ) a,
         (SELECT file_id,
                 tablespace_name,
                 sum(bytes) free
          FROM   dba_free_space
          GROUP BY file_id,
                    tablespace_name
          ) b
          ,
          (select file# from v$datafile where creation_time>sysdate-100) c
    WHERE a.file_id=b.file_id(+)
    AND A.tablespace_name=b.tablespace_name(+) and c.file#=b.file_id
    group by a.tablespace_name
    ORDER BY A.tablespace_name ASC) q1 inner join
    (select tspace,tfree
```

```
      from (SELECT tspace,
                   sayi,
                   ttoplam,
                   tsonboyut,
                   ((ttoplam - tsonboyut) + tfree) tfree
              FROM (SELECT t.tablespace_name tspace,
                           COUNT(*) sayi,
                           SUM(t.maxbytes / 1024 / 1024) ttoplam,
                           SUM(t.user_bytes / 1024 / 1024) tsonboyut,
                           (SELECT SUM(bytes / 1024 / 1024)
                              FROM dba_free_space dfs
                             WHERE dfs.tablespace_name = t.tablespace_name) tfree
                      FROM dba_data_files t, dba_tablespaces s
                     WHERE t.tablespace_name = s.tablespace_name
                         AND t.tablespace_name=ptbs
                      GROUP BY t.tablespace_name))
    order by 1
    ) q2 on q1.tablespace_name=q2.tspace;

  BEGIN
  for i in cr1
      loop
                growth:= growth || CHR(13) || (' Tarih : ' || i.thedate || '  Boyut :  ' ||
i.diff||'MB');

          end loop;
       for i in cr2
        loop
                 free := free || CHR (13) || (ptbs ||  ' tablespaceinin  boş alanı:' ||
i.tfree ||'GB' || ' kullanılabilir alanı:'|| i.usable ||'GB' );

        end loop;

   return free||growth;

  END tbs_free;
```

```
select datadk.tbs_free('DATAIDX') from dual;
```

```
DATAIDX tablespaceinin  boş alanı:16260.75GB kullanılabilir alanı:8076GB
 Tarih : 16-MAR-20  Boyut :  MB
 Tarih : 17-MAR-20  Boyut :  0MB
 Tarih : 18-MAR-20  Boyut :  0MB
 Tarih : 19-MAR-20  Boyut :  0MB
```

TBS_CNT

```
    CREATE OR REPLACE PROCEDURE DATADK.TBS_CNT
    IS

CURSOR cr
   IS
      SELECT q1.tablespace_name,
             q1.usable,
             q2.tfree,
             tid,
             ttoplam
        FROM ( (  SELECT SUBSTR (A.tablespace_name, 1, 14) tablespace_name,
                         SUM (
```

```
                          TRUNC ( DECODE (A.autoextensible,'YES', A.MAXSIZE - A.bytes +
b.free,'NO', b.free) / 1024/ 1024)) usable
                  FROM (  SELECT file_id,
                                 file_name,
                                 tablespace_name,
                                 autoextensible,
                                 bytes,
                                 DECODE (autoextensible,
                                         'YES', maxbytes,
                                         bytes)
                                    maxsize
                            FROM dba_data_files
                        GROUP BY file_id,
                                 file_name,
                                 tablespace_name,
                                 autoextensible,
                                 bytes,
                                 DECODE (autoextensible,
                                         'YES', maxbytes,
                                         bytes)) a,
                        (  SELECT file_id, tablespace_name, SUM (bytes) free
                             FROM dba_free_space
                         GROUP BY file_id, tablespace_name) b,
                        (SELECT file#
                           FROM v$datafile
                          WHERE creation_time > SYSDATE - 100) c
                WHERE     a.file_id = b.file_id(+)
                      AND A.tablespace_name = b.tablespace_name(+)
                      AND c.file# = b.file_id
              GROUP BY a.tablespace_name
              ORDER BY A.tablespace_name ASC) q1
            INNER JOIN
            (  SELECT tspace,
                      tfree,
                      tid,
                      ttoplam
                 FROM (SELECT tspace,
                              sayi,
                              ttoplam,
                              tsonboyut,
                              tid,
                              ( (ttoplam - tsonboyut) + tfree) tfree
                         FROM (  SELECT t.tablespace_name tspace,
                                        tt.ts# tid,
                                        COUNT (*) sayi,
                                        SUM (t.maxbytes / 1024 / 1024) ttoplam,
                                        SUM (t.user_bytes / 1024 / 1024)
                                           tsonboyut,
                                        (SELECT SUM (bytes / 1024 / 1024)
                                           FROM dba_free_space dfs
                                          WHERE dfs.tablespace_name =
                                                   t.tablespace_name)
                                           tfree
                                   FROM dba_data_files t,
                                        dba_tablespaces s,
                                        v$tablespace tt
                                  WHERE     t.tablespace_name =
                                               s.tablespace_name
                                        AND t.tablespace_name = tt.name
                               GROUP BY t.tablespace_name, tt.ts#))
             ORDER BY 1) q2
               ON q1.tablespace_name = q2.tspace);
    BEGIN

    for i in cr
     loop

    INSERT INTO DATADK.TBS_LOG
(tablespace_name,tablespace_id,free_size,total_size,available_size,control_date) VALUES
(i.tablespace_name,i.tid,i.tfree,i.ttoplam,i.usable,sysdate);
```

```
       commit;
      end loop;

     END;
 /

BEGIN
DATADK.TBS_CNT;
END;
/
```

TBS_LOG

```
CREATE TABLE DATADK.TBS_LOG
(
  ID                NUMBER,
  TABLESPACE_NAME   VARCHAR2(50),
  TABLESPACE_ID     NUMBER,
  TOTAL_SIZE        NUMBER,
  FREE_SIZE         NUMBER,
  AVAILABLE_SIZE    NUMBER,
  CONTROL_DATE      DATE)
```

```
SELECT * FROM TBS_LOG;
```

ID	TABLESPACE_N...	TABLESPACE_ID	TOTAL_SIZE	FREE_SIZE	AVAILABLE_SIZE	CONTROL_DATE
	ADSS	5	8192	8190,6875	8189	21.03.2020 19:20:03
	DATA	7	8192	8188,3125	8187	21.03.2020 19:20:03
	DATAIDX	6	16384	16260,75	8076	21.03.2020 19:20:03
	DENEME	8	8192	8192	8191	21.03.2020 19:20:03
	SYSAUX	1	32767,984375	32582,734375	32581	21.03.2020 19:20:03
	SYSTEM	0	32767,984375	32422,171875	32421	21.03.2020 19:20:03
	TABLESPACE	9	24576	24438	8071	21.03.2020 19:20:03
	TBS	10	8192	8192	8191	21.03.2020 19:20:03
	UNDOTBS1	2	32767,984375	32748,734375	32747	21.03.2020 19:20:03
▶	USERS	4	32767,984375	32767,546875	32766	21.03.2020 19:20:03

CORE İSİMLİ PACKAGE OLUŞTURULMASI:

SPEC

```
CREATE OR REPLACE PACKAGE DATADK.CORE
AS
 FUNCTION schema_size (pschema IN VARCHAR2,ptype IN VARCHAR2)
 RETURN VARCHAR2;
 FUNCTION schema_size_detail (pschema IN VARCHAR2,ptype IN VARCHAR2)
 RETURN VARCHAR2;
 FUNCTION table_size(pschema IN VARCHAR2,ptable IN VARCHAR2,ptype IN VARCHAR2)
 RETURN VARCHAR2;
 FUNCTION TBS_FREE(ptbs IN VARCHAR2)
 RETURN VARCHAR2;
 PROCEDURE TBS_CNT ;
END;
```

BODY:

```
CREATE OR REPLACE PACKAGE BODY DATADK.CORE
AS
   FUNCTION schema_size (pschema    VARCHAR2,ptype VARCHAR2)
   RETURN varchar2
   IS
   total_schema varchar2(15);
   total_table  varchar2(15);
   total_index  varchar2(15);
   total_mview  varchar2(15);
BEGIN

   IF ptype IS NULL
   THEN
        SELECT SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_SCHEMA into total_schema
          FROM dba_segments
         WHERE owner = pschema;
      return total_schema || ' GB';

   ELSIF (ptype = 'TABLE')
      THEN
           SELECT  SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_TABLE into total_table
             FROM dba_segments
            WHERE owner = pschema AND segment_type = ptype;
            return total_table || ' GB';

      ELSIF (ptype  = 'INDEX')
         THEN
              SELECT SUM (bytes) / 1024 / 1024 / 1024 AS SIZE_TABLE into total_index
                FROM dba_segments
               WHERE owner = pschema AND segment_type = ptype;
            return total_index ||' GB';


         ELSIF (ptype = 'MATERIALIZED VIEW')
            THEN
            select SUM (bytes) / 1024 / 1024 / 1024  as SIZE_MVİEW into total_mview
            from dba_mviews a, dba_segments b
            where a.owner = pschema and b.owner = pschema
            and a.mview_name = b.segment_name and segment_type = ptype;
           return total_mview ||' GB';
            IF total_mview IS NULL
            THEN
            return 0;
            ELSE
                return total_mview;
            END IF;
       END IF;
          END schema_size;


FUNCTION DATADK.schema_size_detail (pschema    VARCHAR2 ,
                                                ptype      VARCHAR2 )
   RETURN varchar2
is

   total_value    VARCHAR2 (500);
   total           varchar2(5000);

   CURSOR c1
   IS
        SELECT owner,
             segment_name,
             segment_type,
            bytes
             FROM dba_segments
        WHERE owner = pschema AND segment_type = 'TABLE';

  CURSOR c2
   IS
```

```
     SELECT  owner,
             segment_name,
             segment_type,
            bytes
           FROM dba_segments
           WHERE owner = pschema AND segment_type = 'INDEX';

  cursor c3
  IS
   SELECT ds.owner,
             ds.segment_name,
             ds.segment_type,
             ds.bytes,
             mv.mview_name
        FROM dba_segments ds
             LEFT OUTER JOIN dba_mviews mv ON ds.segment_name = mv.mview_name
       WHERE     ds.owner = pschema
             AND ds.segment_type = 'TABLE'
             AND mv.mview_name = ds.segment_name;

BEGIN
   IF ptype = 'TABLE'
      THEN
       FOR i IN c1
     LOOP
           SELECT bytes/1024/1024/1024
           INTO total_value
             FROM dba_segments
            WHERE bytes> 1073741824 AND owner = pschema AND segment_type=ptype AND
segment_name=i.segment_name;
         total:= total || CHR(13) ||i.segment_name ||' tablosunun boyutu: ' || total_value ||
' GB';
    END LOOP;

      ELSIF (ptype = 'INDEX')
      THEN
      FOR i IN c2
     LOOP
         SELECT bytes/1024/1024/1024
           INTO total_value
             FROM dba_segments
            WHERE bytes> 1073741824 AND owner = pschema AND segment_type=ptype AND
segment_name=i.segment_name;
         total:= total || CHR(13) || i.segment_name ||' indexinin boyutu: ' ||total_value ||
' GB';



        END loop;

      ELSIF (ptype = 'MATERIALIZED VIEW')
      THEN
      FOR i IN c3
     LOOP

            select bytes/1024/1024/1024 into total_value from dba_segments ds,dba_mviews mv
where ds.segment_name=mv.mview_name and ds.segment_type='TABLE'
            and ds.owner=UPPER(pschema)
            and mv.mview_name=i.segment_name and bytes > 1073741824 order by bytes desc;
            total:= total|| CHR(13) ||  i.segment_name || ' isimli MATERALİZED_VIEWin boyutu
:' || total_value||' GB';
        END loop;


      END IF;

   RETURN total;
END schema_size_detail;
```

```
FUNCTION DATADK.table_size (pschema    VARCHAR2,
                                        ptable     VARCHAR2,
                                        ptype      VARCHAR2)
   RETURN VARCHAR2
IS
   p_total            VARCHAR2(1000) ;
   index_total_size   VARCHAR2 (1000);
   tbl_size           VARCHAR2 (1000);
   lob_size           VARCHAR2 (1000);
   total              VARCHAR2 (1000);

BEGIN
   IF ptype IS NULL
   THEN
     SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO index_total_size
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                                FROM dba_indexes
                               WHERE     table_OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO tbl_size
        FROM DBA_SEGMENTS
       WHERE     SEGMENT_NAME = UPPER (ptable)
             AND SEGMENT_TYPE = 'TABLE'
             AND owner = UPPER (pschema);

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO lob_size
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                                FROM dba_lobs
                               WHERE     OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));


             total := total || CHR (13) || pschema || ' şemasının '|| ptable ||' tablosuna
ait TABLE, INDEX ve LOB toplam boyutu : ' ||
             to_char(index_total_size + tbl_size +lob_size)  || 'GB';
     END IF;

   IF ptype = 'ALL'
   THEN
      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO index_total_size
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                                FROM dba_indexes
                               WHERE     table_OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO tbl_size
        FROM DBA_SEGMENTS
       WHERE     SEGMENT_NAME = UPPER (ptable)
             AND SEGMENT_TYPE = 'TABLE'
             AND owner = UPPER (pschema);

      SELECT NVL (SUM (BYTES) / 1024 / 1024 / 1024, 0)
        INTO lob_size
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                                FROM dba_lobs
                               WHERE     OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      total :=
            total
         || CHR (13)
```

```
         ||CHR (13)  || pschema || (' şemasının ' || ptable || ' tablosuna ait');
      total := total || CHR (13) || ('INDEX boyutu : ' || index_total_size || 'GB');
      total := total || CHR (13) || ('BLOB boyutu : ' || lob_size || 'GB');
      total := total || CHR (13) || ('TABLO boyutu : ' || tbl_size || 'GB');
   END IF;



   IF ptype = 'TABLE'
   THEN
      SELECT SUM (BYTES) / 1024 / 1024 / 1024
        INTO p_total
        FROM DBA_SEGMENTS
       WHERE SEGMENT_NAME = UPPER (ptable) AND owner = UPPER (pschema);
      total := total || CHR (13)  || ( pschema || ' şemasının ' || ptable  || ' tablosuna ait
TABLO boyutu: ' || p_total || ' GB') ;
   END IF;

   IF ptype = 'INDEX'
   THEN
      SELECT SUM (bytes) / 1024 / 1024 / 1024
        INTO p_total
        FROM dba_segments
       WHERE segment_name IN (SELECT index_name
                                FROM dba_indexes
                               WHERE     table_OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      total := total || CHR (13) || pschema || (' şemasının '  || ptable  || ' tablosuna ait
INDEX boyutu: ' || p_total || 'GB');
   END IF;



   IF ptype = 'LOBSEGMENT'
   THEN
      SELECT NVL (SUM (bytes) / 1024 / 1024 / 1024, 0)
        INTO p_total
        FROM dba_segments
       WHERE segment_name IN (SELECT segment_name
                                FROM dba_lobs
                               WHERE     OWNER = UPPER (pschema)
                                     AND table_name = UPPER (ptable));

      total := total|| CHR (13)|| (pschema || ' şemasının ' || ptable || ' tablosuna ait LOB
SEGMENT boyutu: '  || p_total  || ' GB');
   END IF;

   RETURN total;
END table_size;
/

 FUNCTION tbs_free(ptbs VARCHAR2)

RETURN VARCHAR2
IS
growth varchar2(1000);
free varchar2(1000);
 cursor cr1 IS select thedate, (mbsize - prev_mbsize) diff
                    from (select thedate, mbsize, lag(mbsize, 1) over(order by r) prev_mbsize
                    from (select rownum r, thedate, mbsize
                    from (select trunc(thedate) thedate, max(mbsize) mbsize
                    from (select to_date(to_char(ss.begin_interval_time -1,'YYYY-MON-DD
HH24:MI:SS'),'YYYY-MON-DD HH24:MI:SS') thedate,
                    round((us.tablespace_usedsize * bs.value)/1024/1024,2) mbsize
                    from dba_hist_tbspc_space_usage us,v$tablespace ts,dba_hist_snapshot
ss,v$parameter bs
                    where us.snap_id = ss.snap_id
                    and us.tablespace_id = ts.ts#
                    and ts.name =ptbs
                    and bs.name = 'db block size'
```

```
                    and ss.begin_interval_time > sysdate-7
                and ss.begin_interval_time < trunc(sysdate))
                    group by trunc(thedate)
                    order by trunc(thedate))));

cursor cr2 IS select q1.tablespace_name,q1.usable,q2.tfree from(SELECT
           substr(A.tablespace_name,1,14) tablespace_name,

sum(trunc(decode(A.autoextensible,'YES',A.MAXSIZE-A.bytes+b.free,'NO',b.free)/1024/1024))
usable
    FROM  (
         SELECT file_id, file_name,
                tablespace_name,
                autoextensible,
                bytes,
                decode(autoextensible,'YES',maxbytes,bytes) maxsize
         FROM   dba_data_files
         GROUP BY file_id, file_name,
                  tablespace_name,
                  autoextensible,
                  bytes,
                  decode(autoextensible,'YES',maxbytes,bytes)
         ) a,
         (SELECT file_id,
                 tablespace_name,
                 sum(bytes) free
          FROM   dba_free_space
          GROUP BY file_id,
                   tablespace_name
          ) b
          ,
          (select file# from v$datafile where creation_time>sysdate-100) c
    WHERE a.file_id=b.file_id(+)
    AND A.tablespace_name=b.tablespace_name(+) and c.file#=b.file_id
    group by a.tablespace_name
    ORDER BY A.tablespace_name ASC) q1 inner join
    (select tspace,tfree
      from (SELECT tspace,
                   sayi,
                   ttoplam,
                   tsonboyut,
                   ((ttoplam - tsonboyut) + tfree) tfree
              FROM (SELECT t.tablespace_name tspace,
                           COUNT(*) sayi,
                           SUM(t.maxbytes / 1024 / 1024) ttoplam,
                           SUM(t.user_bytes / 1024 / 1024) tsonboyut,
                           (SELECT SUM(bytes / 1024 / 1024)
                              FROM dba_free_space dfs
                             WHERE dfs.tablespace_name = t.tablespace_name) tfree
                      FROM dba_data_files t, dba_tablespaces s
                     WHERE t.tablespace_name = s.tablespace_name
                         AND t.tablespace_name=ptbs
                      GROUP BY t.tablespace_name))
    order by 1
    ) q2 on q1.tablespace_name=q2.tspace;

  BEGIN
  for i in cr1
      loop
                growth:= growth || CHR(13) || (' Tarih : ' || i.thedate || '  Boyut :  ' ||
i.diff||'MB');

          end loop;
       for i in cr2
        loop
                 free := free || CHR (13) || (ptbs ||  ' tablespaceinin  boş alanı:' ||
i.tfree ||'GB' || ' kullanılabilir alanı:'|| i.usable ||'GB' );

        end loop;
```

```
  return free||growth;

  END tbs_free;

  PROCEDURE TBS_CNT
    IS

CURSOR cr
   IS
      SELECT q1.tablespace_name,
             q1.usable,
             q2.tfree,
             tid,
             ttoplam
        FROM ( (  SELECT SUBSTR (A.tablespace_name, 1, 14) tablespace_name,
                         SUM (
                            TRUNC ( DECODE (A.autoextensible,'YES', A.MAXSIZE - A.bytes +
b.free,'NO', b.free) / 1024/ 1024)) usable
                    FROM (  SELECT file_id,
                                   file_name,
                                   tablespace_name,
                                   autoextensible,
                                   bytes,
                                   DECODE (autoextensible,
                                           'YES', maxbytes,
                                           bytes)
                                      maxsize
                              FROM dba_data_files
                          GROUP BY file_id,
                                   file_name,
                                   tablespace_name,
                                   autoextensible,
                                   bytes,
                                   DECODE (autoextensible,
                                           'YES', maxbytes,
                                           bytes)) a,
                         (  SELECT file_id, tablespace_name, SUM (bytes) free
                              FROM dba_free_space
                          GROUP BY file_id, tablespace_name) b,
                         (SELECT file#
                            FROM v$datafile
                           WHERE creation_time > SYSDATE - 100) c
                   WHERE     a.file_id = b.file_id(+)
                         AND A.tablespace_name = b.tablespace_name(+)
                         AND c.file# = b.file_id
                GROUP BY a.tablespace_name
                ORDER BY A.tablespace_name ASC) q1
             INNER JOIN
             (  SELECT tspace,
                       tfree,
                       tid,
                       ttoplam
                  FROM (SELECT tspace,
                               sayi,
                               ttoplam,
                               tsonboyut,
                               tid,
                               ( (ttoplam - tsonboyut) + tfree) tfree
                          FROM (  SELECT t.tablespace_name tspace,
                                         tt.ts# tid,
                                         COUNT (*) sayi,
                                         SUM (t.maxbytes / 1024 / 1024) ttoplam,
                                         SUM (t.user_bytes / 1024 / 1024)
                                            tsonboyut,
                                         (SELECT SUM (bytes / 1024 / 1024)
                                            FROM dba_free_space dfs
                                           WHERE dfs.tablespace_name =
                                                    t.tablespace_name)
                                            tfree
                                    FROM dba_data_files t,
                                         dba_tablespaces s,
```

```
                                        v$tablespace tt
                                  WHERE     t.tablespace_name =
                                              s.tablespace_name
                                        AND t.tablespace_name = tt.name
                                GROUP BY t.tablespace_name, tt.ts#))
                ORDER BY 1) q2
                  ON q1.tablespace_name = q2.tspace);
     BEGIN
          for i in cr
      loop
     INSERT INTO DATADK.TBS_LOG
(tablespace_name,tablespace_id,free_size,total_size,available_size,control_date) VALUES
(i.tablespace_name,i.tid,i.tfree,i.ttoplam,i.usable,sysdate);
      commit;
      end loop;
       END;
     END CORE;
 /
```

PACKAGE KULLANARAK FONKSİYON VE PROCEDURE ÇALIŞTIRMAK İÇİN;

```
SELECT CORE.SCHEMA_SIZE('DATADK','TABLE') FROM DUAL; (Function çalıştırmak)
```

CORE.SCHEMA_SIZE('DATADK','TABLE')
.0018310546875 GB

```
SELECT CORE.TABLE_SIZE('DATADK','ILLER','INDEX') FROM DUAL;
```

```
1
2       DATADK şemasının ILLER  tablosuna ait INDEX boyutu:.00006103515625
```

```
SELECT CORE.SCHEMA_SIZE_DETAIL('DATADK','INDEX') FROM DUAL;
SELECT CORE.TBS_FREE('DATAIDX') FROM DUAL;

EXEC CORE.TBS_CNT;   (Procedure çalıştırmak)
```

```
PL/SQL procedure successfully completed
```

```
CREATE OR REPLACE FUNCTION DATADK.rman(ptype VARCHAR2)
RETURN VARCHAR2
IS
        total  varchar2(5000);

      cursor  cr1 is
      select (to_char(start_time, 'dd.mm.yyyy hh24:mi:ss')) as
start_time,(to_char(end_time, 'dd.mm.yyyy hh24:mi:ss')) as end_time,status,
     time_taken_display as duration_time,
    (((input_bytes / 1024) / 1024) / 1024) AS INPUT_SIZE, (((output_bytes /
1024) / 1024) / 1024) AS OUTPUT_SIZE from v$rman_backup_job_details
    where trunc(start_time) >=(SELECT TRUNC(MAX(START_TIME)-7) FROM
v$rman_backup_job_details) and rownum<10;

     cursor cr2 is
         select (to_char(start_time, 'dd.mm.yyyy hh24:mi:ss')) as
start_time,(to_char(end_time, 'dd.mm.yyyy hh24:mi:ss')) as end_time,status,
        time_taken_display as duration_time,
    (((input_bytes / 1024) / 1024) / 1024) AS INPUT_SIZE, (((output_bytes /
1024) / 1024) / 1024) AS OUTPUT_SIZE from v$rman_backup_job_details
     where status !='COMPLETED' and trunc(start_time) >=(SELECT
TRUNC(MAX(START_TIME)-7) FROM v$rman_backup_job_details) and rownum<10;
    BEGIN

          IF ptype = 'ALL'
          THEN
           FOR i IN cr1
         LOOP
         total := total ||  CHR(13)  ;

         total:= total || CHR(13) ||   ' başlangıç tarihi: ' ||
i.start_time ;
         total := total ||  CHR(13) ||' bitiş tarihi: ' || i.end_time;
         total := total ||  CHR(13) ||   ' geçen süre: ' || i.duration_time
;
         total := total ||  CHR(13) ||   ' input size: ' || i.INPUT_SIZE ;
         total := total ||  CHR(13) ||   ' output size: ' || i.OUTPUT_SIZE
;
         total := total ||  CHR(13) ;
        END LOOP;

        elsif  ptype = 'FAIL'
       THEN
         for i in cr2
          loop
           total := total ||  CHR(13)  ;
          total:= total || CHR(13) ||   ' başlangıç tarihi: ' ||
i.start_time ;
           total := total ||  CHR(13) ||' bitiş tarihi: ' || i.end_time;
```

```
          total := total ||  CHR(13) ||    ' geçen süre: ' ||
i.duration_time ;
          total := total ||  CHR(13) ||    ' input size: ' || i.INPUT_SIZE ;
          total := total ||  CHR(13) ||    ' output size: ' || i.OUTPUT_SIZE
;
           total := total ||  CHR(13)  ;
         end loop;

       END IF;

       RETURN total;

    END rman;

select datadk.rman('ALL') from dual;

select datadk.rman('FAIL') from dual;
```