```
SQL>
SQL> set linesize 100
SQL> set pagesize 200
SQL> set echo on
SQL> set feedback on
SQL>
SQL> /* Implement SQL script solution1.sql that performs the following actions.
*/
SQL>
SQL> /* (1) First the script connects to a database server as a user SYSTEM and list the following
columns from
SQL>       a dynamic performance view V$INSTANCE:
SQL>       INSTANCE_NAME,
SQL>       HOST_NAME,
SQL>       STARTUP_TIME,
SQL>       DATABASE_STATUS.
SQL>
SQL>       To connect to a database server within SQL script as a user xyz007 with a password
password insert
SQL>       the following line into the script.
SQL>
SQL>       connect xyz007/password
*/
SQL>
SQL> connect SYSTEM/oracle
Connected.
SQL>
SQL> column HOST_NAME format A30
SQL>
SQL> SELECT INSTANCE_NAME,
  2         HOST_NAME,
  3         STARTUP_TIME,
  4         DATABASE_STATUS
  5  FROM V$INSTANCE;

INSTANCE_NAME    HOST_NAME                      STARTUP_T DATABASE_STATUS
---------------- ------------------------------ --------- -----------------
db               localhost.localdomain          26-JUN-22 ACTIVE

1 row selected.

SQL>
SQL> /* (2) Next, the script connects as a user tpchr and processes ANALYZE TABLE statement to
load into
SQL>       a data dictionary statistical information related to the relational tables and indexes
implementing
SQL>       a sample database tpchr created earlier
*/
SQL>
SQL> connect tpchr/oracle
Connected.
SQL>
SQL> ANALYZE TABLE NATION COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE REGION COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE PART COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE SUPPLIER COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE PARTSUPP COMPUTE STATISTICS;
```

```
Table analyzed.

SQL> ANALYZE TABLE CUSTOMER COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE ORDERS COMPUTE STATISTICS;

Table analyzed.

SQL> ANALYZE TABLE LINEITEM COMPUTE STATISTICS;

Table analyzed.

SQL>
SQL> /* (3) Next, the script connects as a user sys.
*/
SQL>
SQL> connect sys/oracle as sysdba
Connected.
SQL>
SQL> /* (3) Next, while still connected as a user sys the script retrieves and lists the following
SQL>       information from a data dictionary.
SQL>       (i) The current timestamp obtained from an application of a function systimestamp.
*/
SQL>
SQL> SELECT systimestamp
  2  FROM DUAL;

SYSTIMESTAMP
---------------------------------------
26-JUN-22 02.24.02.684229000 AM -04:00

1 row selected.

SQL>
SQL> /* (ii) The names of relational tables, that belong to tpchr sample database together with
the total
SQL>        number of rows, total number of data blocks, total number of extents and the total
number of
SQL>        bytes occupied by each table. Display your results in the following format.
SQL>        table-name total-rows total-blocks total-extents total-bytes
*/
SQL>
SQL> column SEGMENT_NAME format A30
SQL>
SQL> SELECT DBA_SEGMENTS.SEGMENT_NAME, DBA_TABLES.NUM_ROWS, DBA_SEGMENTS.BLOCKS,
DBA_SEGMENTS.EXTENTS, DBA_SEGMENTS.BYTES
  2  FROM DBA_SEGMENTS JOIN DBA_TABLES
  3                       ON DBA_SEGMENTS.SEGMENT_NAME = DBA_TABLES.TABLE_NAME
  4  WHERE DBA_TABLES.TABLE_NAME IN ('NATION','REGION',
'PART','SUPPLIER','PARTSUPP','CUSTOMER','ORDERS','LINEITEM') AND
  5        UPPER(DBA_TABLES.OWNER) = 'TPCHR';

SEGMENT_NAME                   NUM_ROWS     BLOCKS    EXTENTS      BYTES
------------------------------ ---------- ---------- ---------- ----------
REGION                                  5         32          1     262144
NATION                                 25         32          1     262144
PART                                60000       1088         34    8912896
SUPPLIER                             3000         96          3     786432
PARTSUPP                           240000       5024        157   41156608
CUSTOMER                            45000       1056         33    8650752
ORDERS                             450000       7296        228   59768832
LINEITEM                          1800093      32576       1018  266862592

8 rows selected.

SQL>
SQL> /* (iii) The names of indexes on primary keys automatically created by the system when
```

```
processing
SQL>         CREATE INDEX statements together with total number of data blocks, total number of
extents and
SQL>         the total number of bytes occupied by each index. Display your results in the
following format.
SQL>          index-name total-blocks total-extents total-bytes
*/
SQL>
SQL> SELECT DBA_SEGMENTS.SEGMENT_NAME, DBA_SEGMENTS.BLOCKS, DBA_SEGMENTS.EXTENTS,
DBA_SEGMENTS.BYTES
  2  FROM DBA_SEGMENTS
  3  WHERE DBA_SEGMENTS.SEGMENT_NAME IN
('REGION_PKEY','NATION_PKEY','PART_PEKEY','SUPPLIER_PKEY','PARTSUPP_PKEY','CUSTOMER_PKEY','ORDERS_
PKEY','LINEITEM_PKEY') AND
  4        UPPER(DBA_SEGMENTS.OWNER) = 'TPCHR';

SEGMENT_NAME                        BLOCKS    EXTENTS      BYTES
------------------------------ ---------- ---------- ----------
REGION_PKEY                            32          1     262144
NATION_PKEY                            32          1     262144
PART_PEKEY                            160          5    1310720
SUPPLIER_PKEY                          32          1     262144
PARTSUPP_PKEY                         896         28    7340032
CUSTOMER_PKEY                         128          4    1048576
ORDERS_PKEY                           928         29    7602176
LINEITEM_PKEY                        4256        133   34865152

8 rows selected.

SQL>
SQL> spool off
```