

3. **SPFILEXE.ORA**

xe.\_\_db\_cache\_size=1107296256

xe.\_\_inmemory\_ext\_roarea=0

xe.\_\_inmemory\_ext\_rwarea=0

xe.\_\_java\_pool\_size=16777216

xe.\_\_large\_pool\_size=33554432

xe.\_\_oracle\_base='C:\app\MokoPC\product\18.0.0'#ORACLE\_BASE set from environment

xe.\_\_pga\_aggregate\_target=536870912

xe.\_\_sga\_target=1610612736

xe.\_\_shared\_io\_pool\_size=83886080

xe.\_\_shared\_pool\_size=352321536

xe.\_\_streams\_pool\_size=0

\*.audit\_file\_dest='C:\app\MokoPC\product\18.0.0\admin\XE\adump'

\*.audit\_trail=CC"~'db'

\*.compatible='18.0.0'

\*.control\_files='C:\app\MokoPC\product\18.0.0\oradata\XE\control01.ctl','C:\app\MokoPC\product\18.0.0\oradata\XE\control02.ctl'

\*.db\_block\_size=8192

\*.db\_name='XE'

\*.diagnostic\_dest='C:\app\MokoPC\product\18.0.0'

\*.dispatchers='(PROTOCOL=TCP) (SERVICE=XEXDB)'

\*.enable\_pluggable\_database=true

\*.local\_listener='LISTENER\_XE'

\*.nls\_language='RUSSIAN'

\*.nls\_territory='RUSSIA'

\*.open\_cursors=300

\*.pga\_aggregate\_target=512m

\*.processes=320

\*.remote\_login\_passwordfCC"v+ile='EXCLUSIVE'

\*.sga\_target=1536m

\*.undo\_tablespace='UNDOTBS1'

**Listener.ora**

# listener.ora Network Configuration File: C:\app\MokoPC\product\18.0.0\dbhomeXE\NETWORK\ADMIN\listener.ora

# Generated by Oracle configuration tools.

DEFAULT\_SERVICE\_LISTENER = XE

SID\_LIST\_LISTENER =

(SID\_LIST =

(SID\_DESC =

(SID\_NAME = CLRExtProc)

(ORACLE\_HOME = C:\app\MokoPC\product\18.0.0\dbhomeXE)

(PROGRAM = extproc)

(ENVS = "EXTPROC\_DLLS=ONLY:C:\app\MokoPC\product\18.0.0\dbhomeXE\bin\oraclr18.dll")

)

)

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = DESKTOP-UDOC0KF)(PORT = 1521))

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

)

**Tnsnames.ora**

# tnsnames.ora Network Configuration File: C:\app\MokoPC\product\18.0.0\dbhomeXE\NETWORK\ADMIN\tnsnames.ora

# Generated by Oracle configuration tools.

XE =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = DESKTOP-UDOC0KF)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = XE)

)

)

LISTENER\_XE =

(ADDRESS = (PROTOCOL = TCP)(HOST = DESKTOP-UDOC0KF)(PORT = 1521))

ORACLR\_CONNECTION\_DATA =

(DESCRIPTION =

(ADDRESS\_LIST =

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

(CONNECT\_DATA =

(SID = CLRExtProc)

(PRESENTATION = RO)

)

)

**Sqlnet.ora**

# sqlnet.ora Network Configuration File: C:\app\MokoPC\product\18.0.0\dbhomeXE\NETWORK\ADMIN\sqlnet.ora

# Generated by Oracle configuration tools.

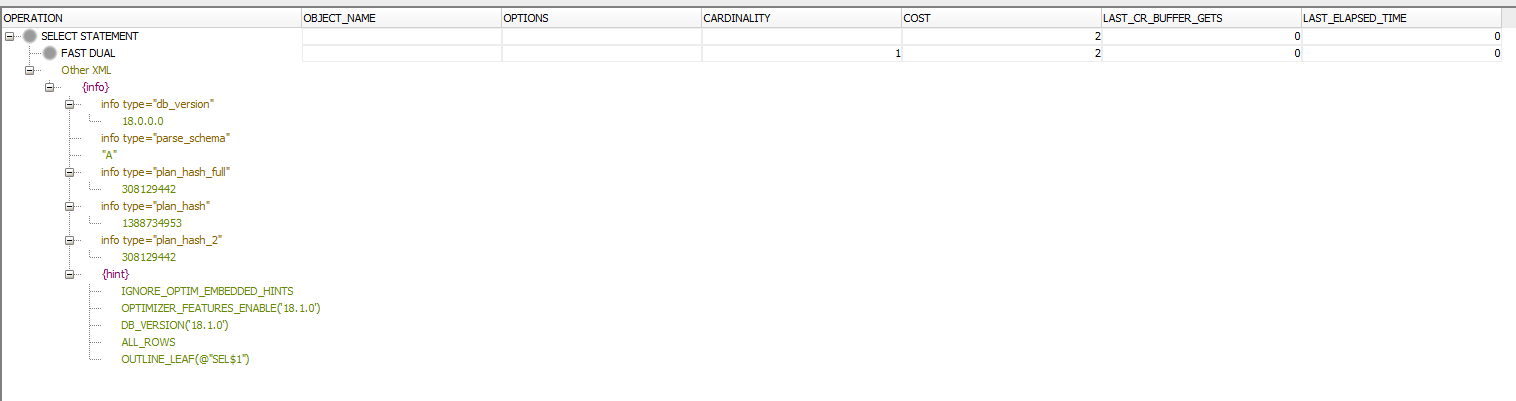
# This file is actually generated by netca. But if customers choose to

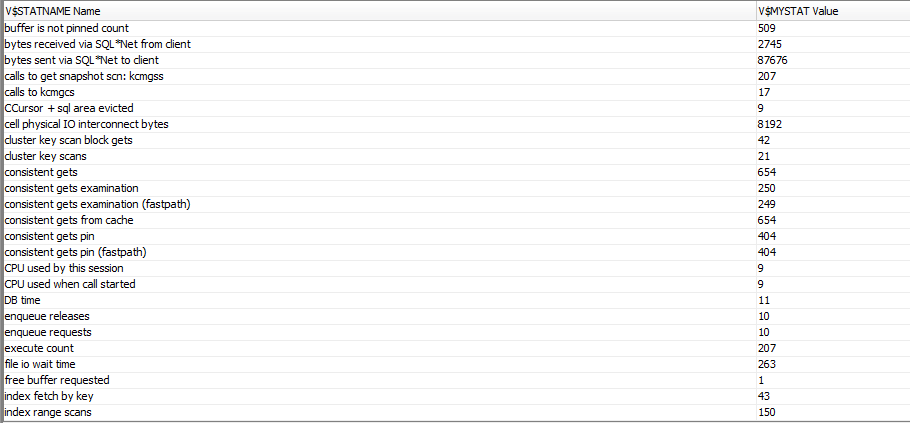
# install "Software Only", this file wont exist and without the native

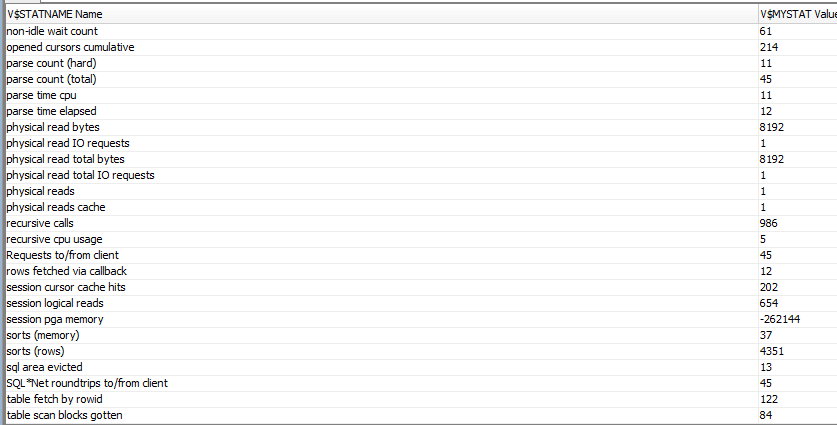
# authentication, they will not be able to connect to the database on NT.

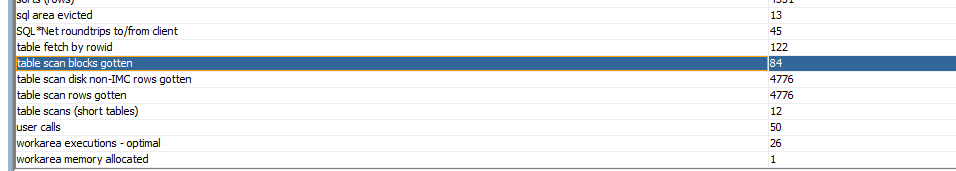
SQLNET.AUTHENTICATION\_SERVICES= (NTS)

NAMES.DIRECTORY\_PATH= (TNSNAMES, EZCONNECT)

4. 



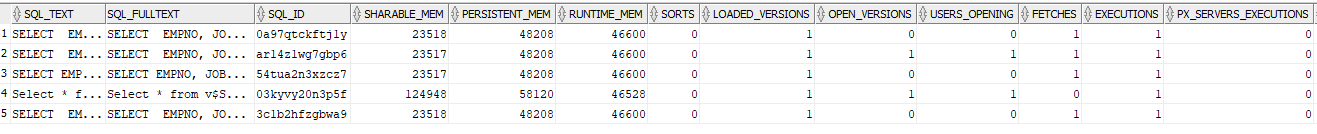


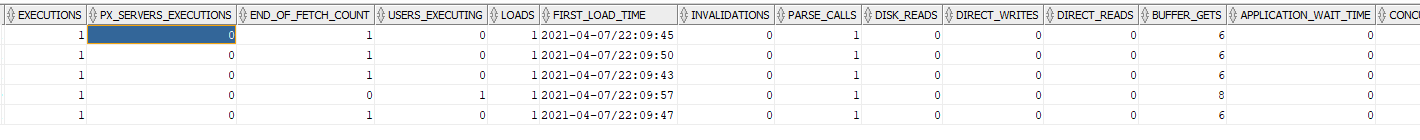


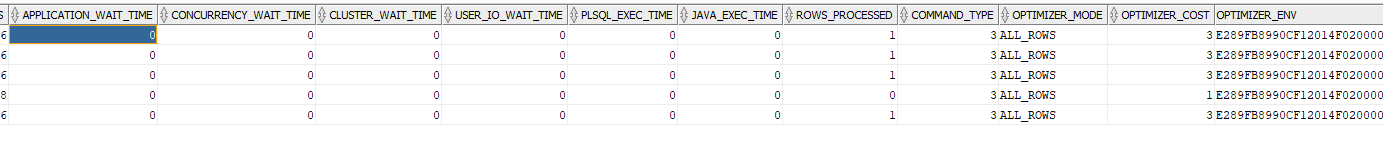
4.

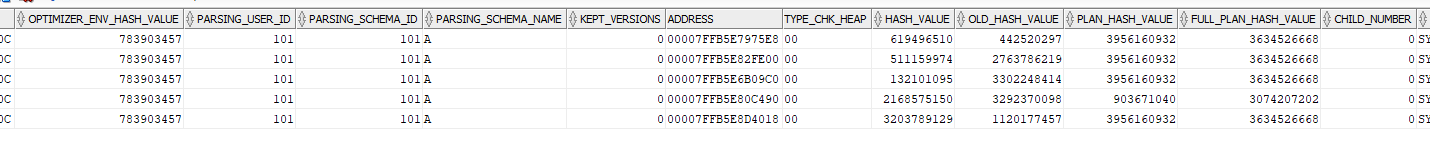


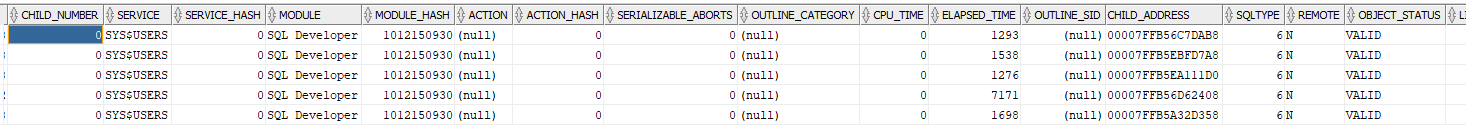
5.

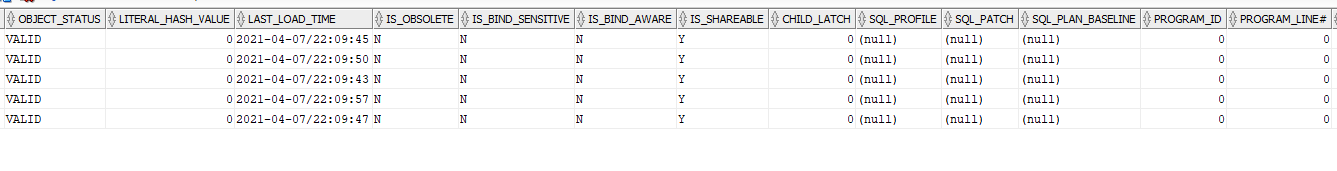


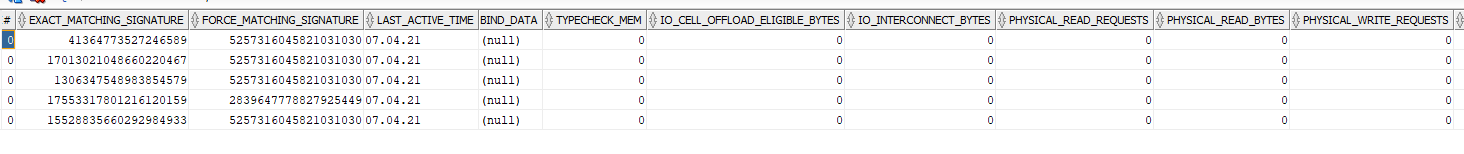


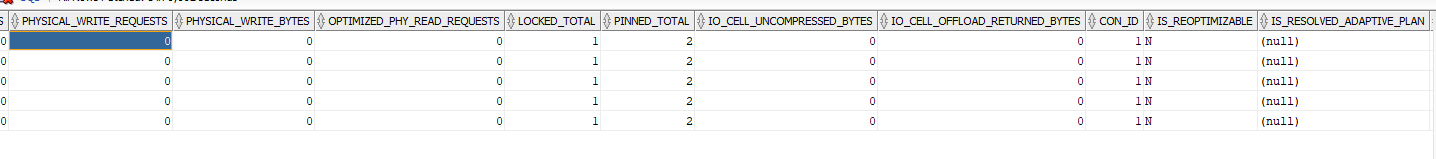


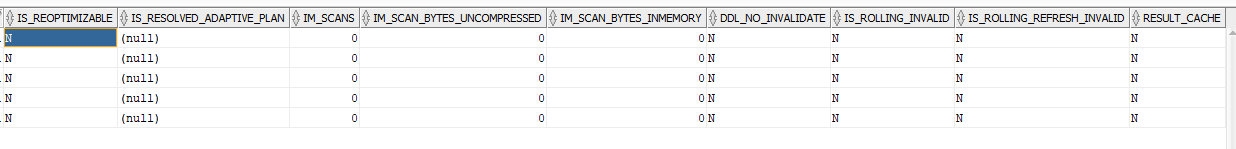




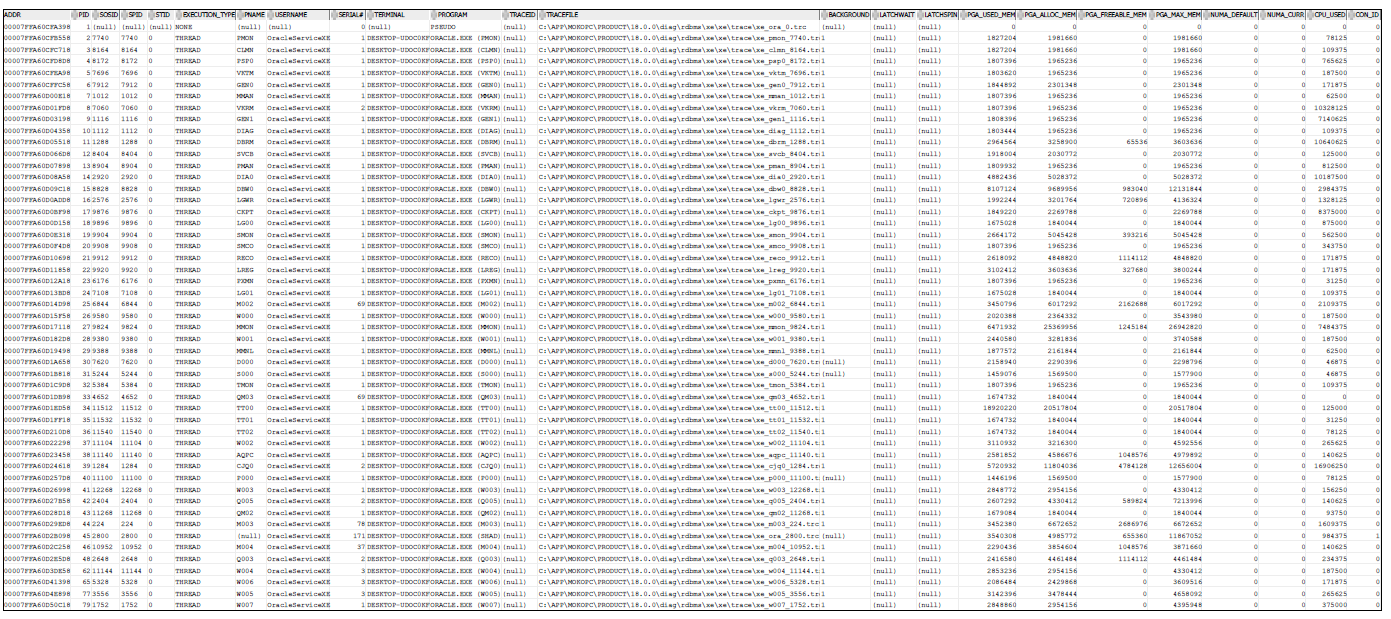








6.



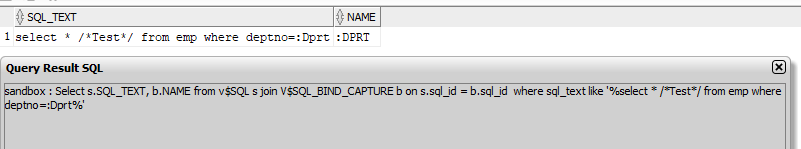
Вопросы к лекции 2

1. Существует четыре основные фазы выполнения SQL выражения: parse, bind, execute, fetch.

2. Разбор происходит следующим образом:

* Сервер проверяет выражение на правильность семантики и синтаксиса.
* Определяет имеются ли у вызывающего пользователя права на выполнение выражения.
* Ищутся совпадения выполняемого кода с уже разобранным кодом, хранящимся в библиотечном кеше.
* Выделяется частную SQL область для обработки выражения

3. Select s.SQL\_TEXT, b.NAME from v$SQL s join V$SQL\_BIND\_CAPTURE b on s.sql\_id = b.sql\_id where sql\_text like '%select \* /\*Test\*/ from emp where deptno=:Dprt%';



4. DML выражения выполняются в две фазы: Parse в процессе которой разбирается выражение и фаза Execute, в которую для DML включены операции захвата данных и сортировки.

Процесс выполнения DML выражения:

1. Если данные и блоки отката для изменяемых в процессе выполнения DML данных еще не помещены в буферный кэш, серверный процесс считывает их с диска в буферный кэш. Серверный процесс при этом блокирует строки, которые будут модифицированы.
2. Серверный процесс затем применяет изменения данных, запрашиваемые в DML к данным, прочитанным в буферный кэш, и записывает данные в буфер отката. Изменения записываются в redo log buffer **перед тем как** данные будут изменены в буферном кэше. Эта технология применения изменений называется "упреждающее логирование" (write-ahead logging)
3. Буферы отката содержат значения данных перед модификацией. Буферы отката используются для хранения снимка данных до изменения и таким образом изменения, внесенные DML операциями, могут быть отменены в случае необходимости. В буферы данных записываются изменения данных.
4. Пользователь получает ответ от операции DML (сколько строк изменено в процессе выполнения операции)

5. Когда запускается COMMIT, выполняются следующие шаги

* Серверный процесс помещает запись о commit c SCN в redo log buffer.
* Фоновый процесс LGWR записывает исходные данные в redo log файлы. Это гарантирует, что в случае отказа экземпляра, данные об изменениях произведенных транзакцией не будут потеряны.
* Серверный процесс отправляет сообщение об окончании транзакции пользовательскому процессу.