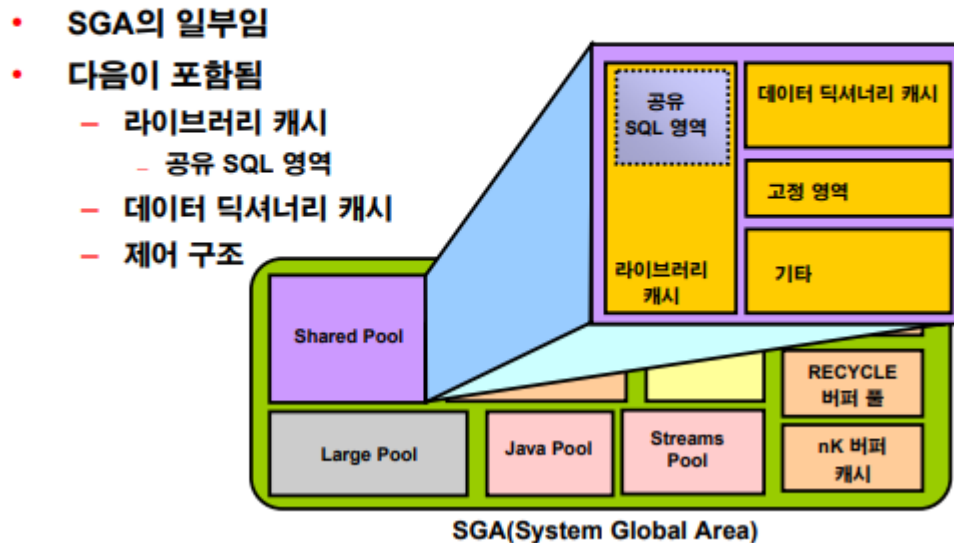


6. Shared Pool



SQL을 실행하면 parsing을 하는데 parsing된 데이터를 shared pool에 올려 재사용하므로 다시 parsing하지 않는다. 이 parsing이 cpu를 많이 사용하는

※ select 문의 처리 과정 3가지

- 유저 프로세서

```
select empno, ename, sal
from emp
where ename = 'SCOTT';
```

- 서버 프로세서

1. parsing 수행

=> 문법 검사 (syntax check) : SQL이 문법적으로 문제가 없는지

=> 의미 검사 (semantic check): emp 테이블이 db에 있는지

parsing 결과물 3가지

1. SQL 문장

2. 실행 계획

3. Parse tree (실행 가능한 코드)

⇒ 위의 3가지 결과물을 다음번에 똑같은 문장이 들어오면 parsing 과정을 shared pool의 library cache에 올린다.

2. 실행 (execute)

- 옵티마이저가 생성해준 실행 계획으로 실행 데이터를 검색하는 작업

3. 패치 (fetch)

서버 프로세서 =====> 유저 프로세서
결과 데이터

실습 주제: 똑같은 sql이 들어와야 parsing 과정을 생략할 수 있는데 똑같은

1. 대소문자 구분

- SELECT ENAME, SAL FROM EMP WHERE EMPNO=7788;
- select ename, sal from emp where empno=7788;

2. 공백, 들여쓰기 구분

- select empno, ename sal from emp;
- select empno, ename, sal from emp;

3. Literal SQL 구분

- select empno, ename, sal from emp where empno = 7788;
- select empno, ename, sal from emp where empno = 7902;

실습 1. dbeaver로 오라클에 접속한다.

실습 2. 아래의 sql을 실행한다.

```
select empno, ename, sal from emp where empno = 7788;
```

실습 3. 위의 sql이 shared pool에 있는지 확인한다.

```
select sql_id, child_number, sql_text
from v$sql
where sql_text like '%emp%' AND sql_text NOT LIKE '%v$sql%'
ORDER BY last_load_time DESC;
```

실습 4. 위의 sql에 공백을 좀 더 넣어서 실행한다.

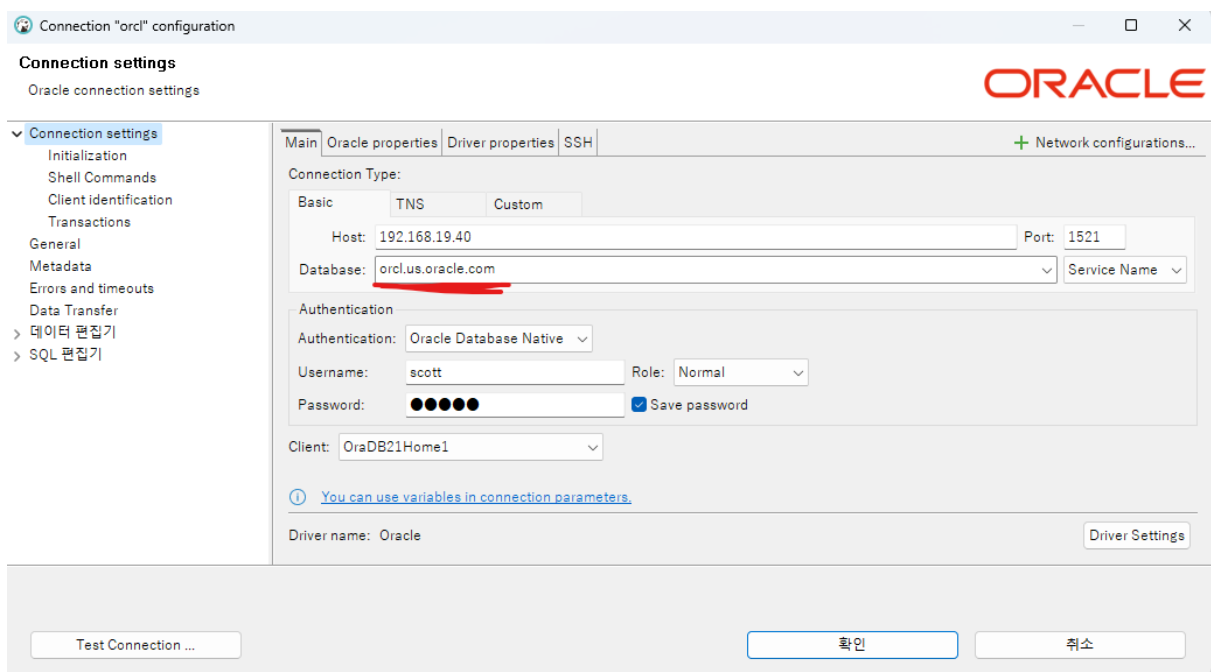
```
select empno, ename, sal from emp where empno = 7788;
```

실습 5. 위의 sql이 shared pool에 있는 확인한다.

```
select sql_id, child_number, sql_text
from v$sql
where sql_text like '%emp%' AND sql_text NOT LIKE '%v$sql%'
order by last_load_time desc;
```

dbeaver 연결 방법

```
[orcl:~]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
192.168.19.40 edydr1p0.us.oracle.com edydr1p0 localhost.localdomain localhost
```



문제 1. 포트 포워딩 말고 static ip로 dbeaver로 접속하기
vi /etc/hosts

- 리스너의 상태 확인하기
lsnrctl status

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
192.168.19.40 edydr1p0.us.oracle.com edydr1p0 localhost.localdomain localhost
```

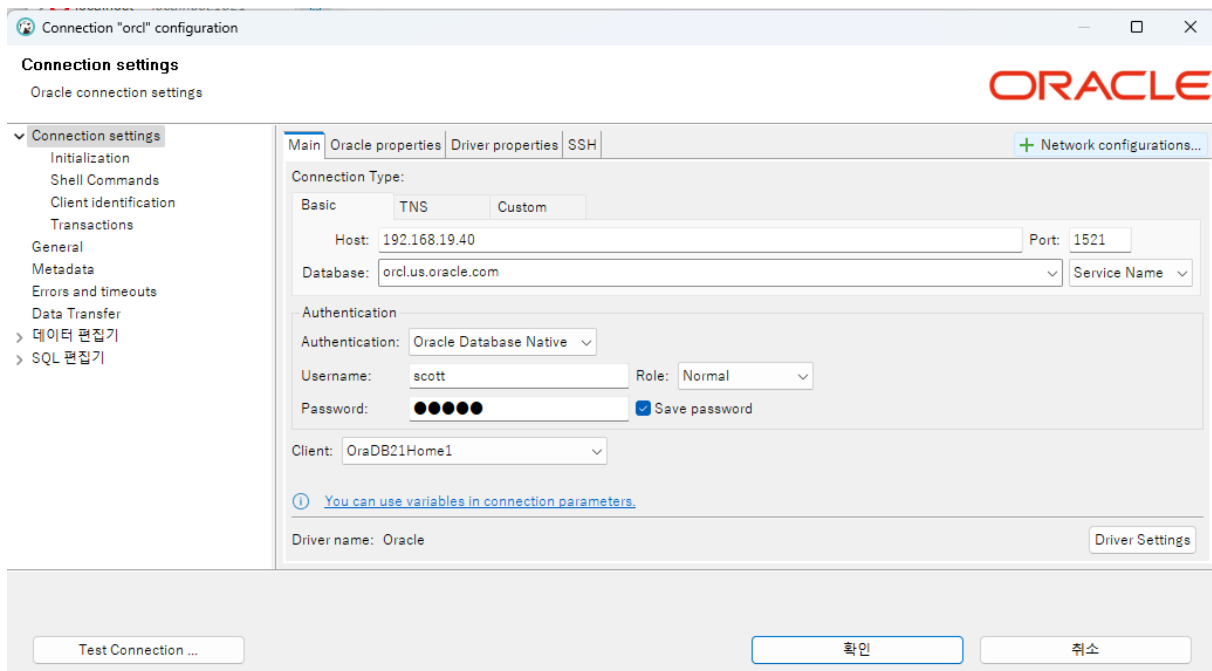
```
[orcl:~]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
192.168.19.40 edydr1p0.us.oracle.com edydr1p0 localhost.localdomain localhost
```

```
[orcl:~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 29-JAN-2024 16:48:18

Copyright (c) 1991, 2009, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                29-JAN-2024 15:47:28
Uptime                    0 days 1 hr. 0 min. 50 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Log File         /u01/app/oracle/diag/tnslsnr/edydr1p0/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edydr1p0.us.oracle.com)(PORT=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM", status READY, has 1 handler(s) for this service...
Service "orcl.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
Service "orclXDB.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
The command completed successfully
```



The screenshot shows the 'Connection "orcl" configuration' dialog box in Oracle SQL Developer. The 'Main' tab is selected, showing the following configuration:

- Connection Type:** Basic (selected), TNS, Custom
- Host:** 192.168.19.40
- Port:** 1521
- Database:** orcl.us.oracle.com
- Service Name:** (empty)
- Authentication:** Oracle Database Native (selected)
- Username:** scott
- Role:** Normal (selected)
- Password:** (masked with dots)
- Save password:** (checked)
- Client:** OraDB21Home1
- Driver name:** Oracle

At the bottom, there are buttons for 'Test Connection ...', '확인' (OK), and '취소' (Cancel).

문제 2. 똑같은 SQL이 shared pool에서 공유되는지 테스트 하기

1. 대소문자 구분

```
SELECT ENAME, SAL, mgr, empno FROM EMP WHERE EMPNO=7788;
select ename, sal,mgr, empno from emp where empno=7788;
```

2. 공백, 들여쓰기 구분

```
select ename, sal,mgr, empno from emp where empno=7788;
select          ename, sal,mgr, empno from emp where empno=
```

3. Literal SQL 구분

```
select empno, ename, sal from emp where empno = 7788;
select empno, ename, sal from emp where empno = 7902;
```

```
select sql_id, child_number, sql_text
from v$sql
where sql_text like '%emp%' AND sql_text NOT LIKE '%v$sql%'
ORDER BY last_load_time DESC;
```

1. 대소문자 구분

```
2vv0ny35ruml6|      0|select ename, sal,mgr, empno from emp where empno=7788
7ryjfdyvt73rq|      0|SELECT ENAME, SAL, mgr, empno FROM EMP WHERE EMPNO=7788
```

2. 공백, 들여쓰기 구분

```
bqgl7q7yhpbd|      0|select          ename, sal,mgr, empno from emp where empno=7788
2vv0ny35ruml6|      0|select ename, sal,mgr, empno from emp where empno=7788
```

3. Literal SQL 구분

```
aszg5bfcbytwm|      0|select empno, ename, sal from emp where empno = 7902
bqgl7q7yhpbd|      0|select          ename, sal,mgr, empno from emp where empno=7788
2vv0ny35ruml6|      0|select ename, sal,mgr, empno from emp where empno=7788
```