# PROJECT

**NE**
Yoo Seung-Chan
Choi Young-Jin

**SE**
Park Soon-ho

**DB**
Kang Da-Yeon
Jeong Min-Ji

# 프로젝트 일정

| | 1W | 2W | 3W | 4W | 5W | 6W | 7W | 8W | 9W |
|---|---|---|---|---|---|---|---|---|---|
| **전체** | 주제 선정 및 계획서 작성 | 자료수집 | | 기술 구현 | | | 구성 실습 및 보완 | | 최종 발표 준비 |
| **네트워크** | Topology | RoutingTable | | 네트워크 구축 | | | 암호화 방화벽 | | |
| **서버** | 아키텍쳐 구상 | 운영체제 설치 개발환경 구성 | | 리눅스 윈도우 서버 구축 | | | 기술 구현 | 수정 보완 | |
| **데이터베이스** | DB 디자인 | DB 설계 | | DB 구현 | | | 백업 및 보안 설정 | | |

# INFRA BUILD

## Network

Topology

GNS3

## Server

APM+Tomcat
Backup
Cloud

CentOS  v7.5
Apache  v2.4.6
PHP  v7.0
MariaDB  v10.4
Tomcat  v8.5
OpenJDK  v11.0.7
Windows 2016 R2
AWS cloud

## DataBase

Oracle
MariaDB

CentOS  v7.5
MariaDB v10.4
Oracle

# 군대 NETWORK

Armed forces

1. UNIQUE

1. 망분리

1. 보안

* Kang Da-Yeon AND Jeong Min-Ji *
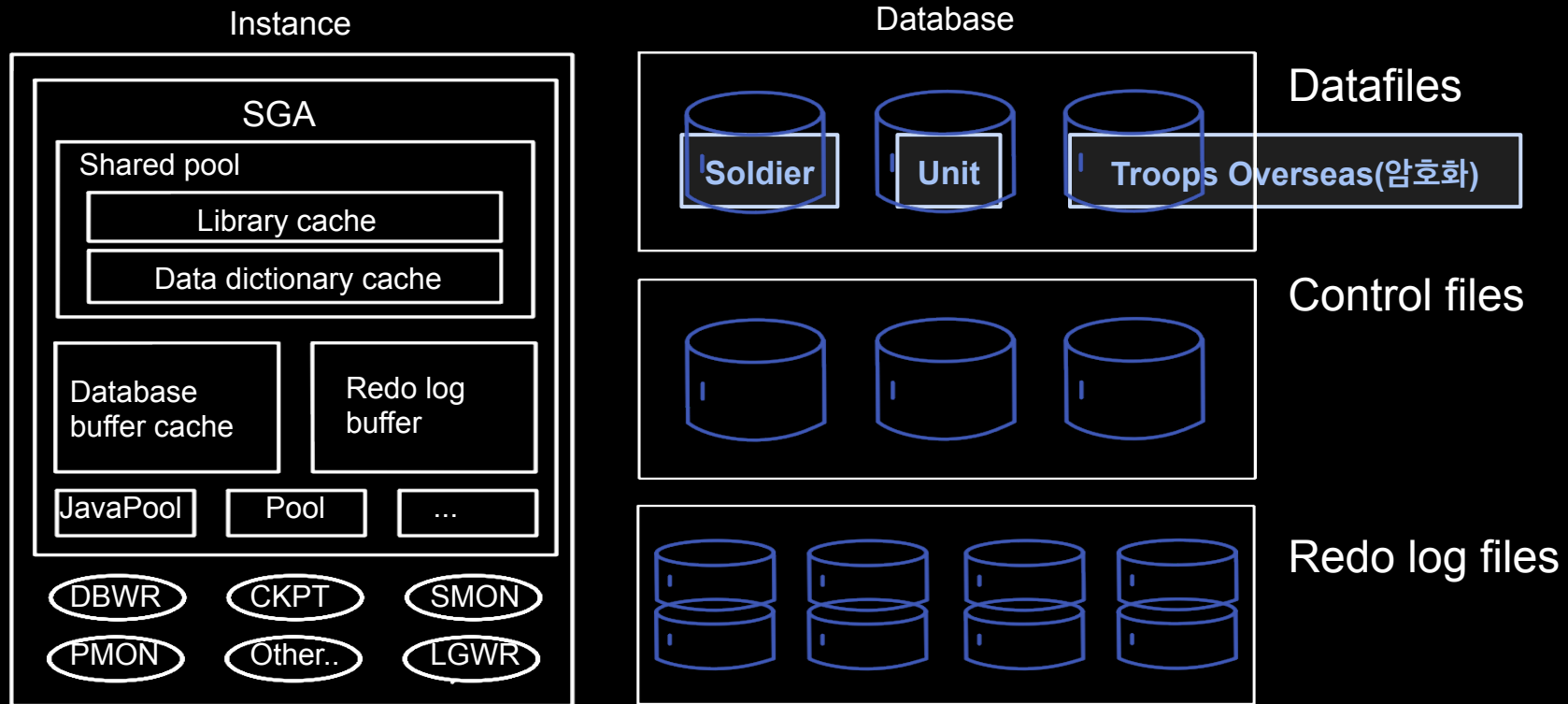
Thre
niT

D

SINCE ——— 2020

B

# DB

## Contents

1. 저장영역 구성 ( Database Design )
2. User 관리 및 보안
3. Schema 설계 ( logical Design, physical Design)
4. Schema 구현
5. Backup 전략
6. Database 보안 구현 ( Tablespace 단위, 데이터, backup)
7. 모니터링
8. 스크립트 파일

# Database Design

## Multiplexing & Circulation

Instance

Database

### SGA

#### Shared pool

Library cache

Data dictionary cache

Database buffer cache

Redo log buffer

JavaPool

Pool

...

DBWR

CKPT

SMON

PMON

Other..

LGWR

Datafiles

**Soldier**

**Unit**

**Troops Overseas(암호화)**

Control files

Redo log files

# Database Design

## Tablespace 구성 결과

# Database Design

## Controlfile Redologfile 구성 결과

```
SQL> select name from v$controlfile;

NAME
----------------------------------------------------------------
/u01/app/oracle/oradata/ORCL/controlfile/o1_mf_hdyt6n5j_.ctl
/u01/app/oracle/fast_recovery_area/orcl/ORCL/controlfile/o1_mf_hdyt6n68_.ctl
/u01/app/oracle/oradata/ORCL/datafile/control03.ctl
```

```
SQL> select group#, member
  2  from v$logfile
  3  order by group#;

    GROUP# MEMBER
---------- ---------------------------------------------------------
         1 /u01/app/oracle/fast_recovery_area/orcl/ORCL/onlinelog/o1_mf_1_hdyt6q61_.log
         1 /u01/app/oracle/oradata/ORCL/onlinelog/o1_mf_1_hdyt6pcw_.log
         2 /u01/app/oracle/oradata/ORCL/onlinelog/o1_mf_2_hdyt6pdx_.log
         2 /u01/app/oracle/fast_recovery_area/orcl/ORCL/onlinelog/o1_mf_2_hdyt6q65_.log
         3 /u01/app/oracle/fast_recovery_area/orcl/ORCL/onlinelog/o1_mf_3_hdyt6q63_.log
         3 /u01/app/oracle/oradata/ORCL/onlinelog/o1_mf_3_hdyt6pfd_.log
         4 /u01/app/oracle/oradata/ORCL/onlinelog/redo4a.log
         4 /u01/app/oracle/fast_recovery_area/orcl/ORCL/onlinelog/redo4b.log

8 rows selected.
```

```
SQL> select group#, members
  2  from v$log;

    GROUP#    MEMBERS
---------- ----------
         1          2
         2          2
         3          2
         4          2
```

# User 관리 및 보안

## Create User

| | ID / PASSWD | DEFAULT TABLESPACE | TEMPORARY TABLESPACE | PROFILE |
|---|---|---|---|---|
| 관리자 | userdba/ userdba | users(200M) | temp | |
| 개발자 | emp/ emp | users(200M) | temp | |
| USER | admin/ admin | users | temp | adminuser |

관리자

CREATE USER userdba
IDENTIFIED BY userdba
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA 200M ON users
QUOTA 200M ON tsoldier

개발자

CREATE USER emp
IDENTIFIED BY emp
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA 200M ON users

USER

CREATE USER admin
IDENTIFIED BY admin
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp

# User 관리 및 보안

## Create Profile
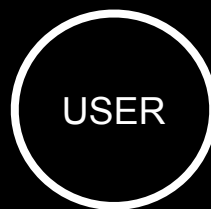
USER

일반 USER PROFILE

CREATE PROFILE adminuser limit
failed_login_attempt 3
password_lock_time unlimited
password_life_time 120
password_reuse_time 30
password_grace_time 5
password_verify_function ora12c_verify_function

ALTER USER admin PROFILE adminuser



| 비밀번호 | |
|---|---|
| | DEFAULT |
| 잠금(만기일 이후) | 5 |
| 잠금 일 수 | UNLIMITED |
| 복잡성 함수 | ORA12C_VERIFY_FUNCTION |
| 보관할 비밀번호 수 | DEFAULT |
| 보관 일 수 | 30 |
| 비밀번호 만기 시간(일) | 120 |
| 잠금 전 로그인 시도 실패 횟수 | 3 |

# User 관리 및 보안

## 롤 & 권한

개발자

| 롤<br>권한 | CONNECT | SELECT VIEW | RESOURCE | DBA | DBATABLE |
|---|---|---|---|---|---|
| 관리자 | O | | O | O | |
| 개발자 | O | | O | | O |
| USER | O | O | | | |

**롤 보기: DBATABLE**

계정 요약

롤 이름    DBATABLE
인증    NONE

세부정보

권한 및 롤 | 객체 권한

객체 권한

권한 부여    취소

| 스키마 | 객체 유형 | 객체 이름 | 권한 부여자 | 권한 |
|---|---|---|---|---|
| USERDBA | TABLE | FIREARM | USERDBA | DELETE |
| USERDBA | TABLE | FIREARM | USERDBA | UPDATE |
| USERDBA | TABLE | FIREARM | USERDBA | SELECT |
| USERDBA | TABLE | FIREARM | USERDBA | INSERT |
| USERDBA | TABLE | LOCATION | USERDBA | DELETE |
| USERDBA | TABLE | LOCATION | USERDBA | UPDATE |
| USERDBA | TABLE | LOCATION | USERDBA | SELECT |
| USERDBA | TABLE | LOCATION | USERDBA | INSERT |

# Schema 설계

## Logical Design
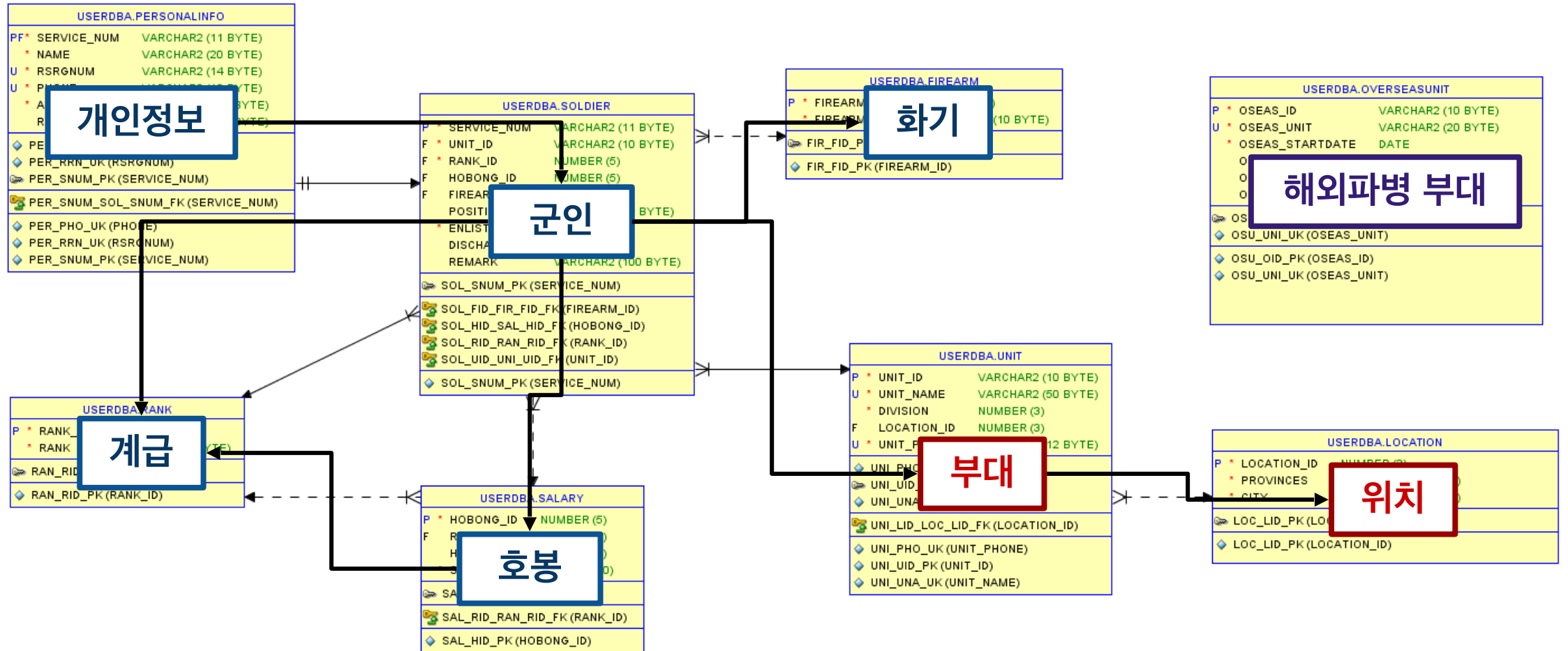
# Schema 설계

## Physical Design

### Table <SOLDIER>

| Column name | 군번 service_num | 부대ID unit_id | 계급ID rank_id | 호봉ID hobong_id | 총기번호 firearm_id | 직책 position | 입대일 enlistment | 제대일 discharge | 비고 remark |
|---|---|---|---|---|---|---|---|---|---|
| Key Type | PK | FK | FK | FK | FK | | | | |
| Null/Unique | | NN | NN | | | | NN | | |
| Check | | | | | | CK | | | |
| PK Table | | UNIT | RANK | HOBONG | FIREARM | | | | |
| PK Column | | unit_id | rank_id | hobong_id | firearm_id | | | | |
| Data Type | varchar2 | varchar2 | number | number | number | varchar2 | date | date | varchar2 |
| Length | 11 | 10 | 5 | 5 | 6 | 20 | | | 100 |

Position 의 Check 제약조건
'commander','commanding officer','Headquarters','divisional commander','battalion commander','platoon leader','squad leader'

# Schema 설계

## Physical Design

### Table <PERSONALINTO>

| Column name | 군번<br>**service_num** | 이름<br>**name** | 주민번호<br>**rsrgnum** | 전화번호<br>**phone** | 주소<br>**address** | 비고<br>**remark** |
|---|---|---|---|---|---|---|
| Key Type | PK / FK | | | | | |
| Null/Unique | | NN | NN / UK | NN / UK | NN | |
| Check | | | '_____-_____' | '___-%-____' | | |
| PK Table | SOLDIER | | | | | |
| PK Column | service_num | | | | | |
| Data Type | varchar2 | varchar2 | varchar2 | varchar2 | varchar2 | varchar2 |
| Length | 11 | 20 | 14 | 13 | 250 | 100 |

### Table <RANK>

| Column name | 계급ID<br>**rank_id** | 계급명<br>**rank** |
|---|---|---|
| Key Type | PK | |
| Null/Unique | | NN |
| Check | | |
| PK Table | | |
| PK Column | | |
| Data Type | number | varchar2 |
| Length | 5 | 30 |

# Schema 설계

## Physical Design

### Table <FIREARM>

| Column name | 총기번호<br>firearm_id | 장비명<br>firearm_name |
|---|---|---|
| Key Type | PK | |
| Null/Unique | | NN |
| Check | | |
| PK Table | | |
| PK Column | | |
| Data Type | number | varchar2 |
| Length | 6 | 10 |

### Table <SALARY>

| Column name | 호봉ID<br>hobong_id | 계급ID<br>rank_id | 호봉<br>hobong | 금액<br>salary |
|---|---|---|---|---|
| Key Type | PK | FK | | |
| Null/Unique | | | | NN |
| Check | | | | |
| PK Table | | RANK | | |
| PK Column | | rank_id | | |
| Data Type | number | number | number | number |
| Length | 5 | 5 | 2 | 10 |

# Schema 설계

## Physical Design

Table <UNIT>

| Column name | 부대ID<br>unit_id | 부대명<br>unit_name | 사단명<br>division | 위치id<br>location_id | 부대번호<br>unit_phone |
|---|---|---|---|---|---|
| Key Type | PK | | | FK | |
| Null/Unique | | NN / UK | NN | | NN / UK |
| Check | | | | | '%-%-%' |
| PK Table | | | | LOCATION | |
| PK Column | | | | location_id | |
| Data Type | varchar2 | varchar2 | number | number | varchar2 |
| Length | 10 | 50 | 5 | 3 | 20 |

Table <LOCATION>

| Column name | 위치<br>location_id | 광역시, 도<br>provinces | 시, 군, 구<br>city |
|---|---|---|---|
| Key Type | PK | | |
| Null/Unique | | NN | NN |
| Check | | | |
| PK Table | | | |
| PK Column | | | |
| Data Type | varchar2 | varchar2 | varchar2 |
| Length | 3 | 10 | 10 |

# Schema 설계

## Physical Design

Table <OVERSEASUNIT>

| Column name | 파병부대ID<br>oseas_id | 파병부대명<br>oseas_unit | 파견시작일<br>oseas_startdate | 파견예상종료일<br>oseas_enddate | 파병국<br>oseas_country | 인원수<br>oseas_troops |
|---|---|---|---|---|---|---|
| Key Type | PK | | | | | |
| Null/Unique | | NN, UK | NN | | | |
| Check | | | | | | |
| PK Table | | | | | | |
| PK Column | | | | | | |
| Data Type | varchar2 | varchar2 | date | date | varchar2 | number |
| Length | 10 | 10 | | | 50 | 10 |

# Schema 구현

## Table 생성 및 조회

### Table <SOLDIER>

```
 1  CREATE TABLE soldier
 2  (
 3      service_num VARCHAR2(11) CONSTRAINT sol_snum_pk PRIMARY KEY,
 4      unit_id VARCHAR2(10) CONSTRAINT Sol_Uid_nn NOT NULL,
 5      rank_id NUMBER(5) CONSTRAINT sol_rid_nn NOT NULL,
 6      hobong_id NUMBER(5),
 7      firearm_id NUMBER(6),
 8      position VARCHAR2(20) CONSTRAINT sol_pos_ck CHECK ( position IN
 9  ('commander','commanding officer','Headquarters','divisional commander','battalion commander','pla
toon leader','squad leader')),
10      enlistment DATE  CONSTRAINT sol_enl_nn   NOT NULL,
11      discharge DATE,
12      remark VARCHAR2(100),
13  CONSTRAINT sol_uid_uni_uid_fk FOREIGN KEY (Unit_id) REFERENCES Unit (Unit_id),
14      CONSTRAINT sol_rid_ran_rid_fk FOREIGN KEY (rank_id) REFERENCES Rank (rank_id),
15      CONSTRAINT sol_fid_fir_fid_fk FOREIGN KEY (firearm_id) REFERENCES Firearm (firearm_id),
16  CONSTRAINT sol_hid_sal_hid_fk FOREIGN KEY (hobong_id) REFERENCES Salary (hobong_id)
17* ) tablespace tsoldier
```

```
SQL> select * from soldier;

SERVICE_NUM UNIT_ID       RANK_ID  HOBONG_ID FIREARM_ID POSITION            ENLISTMEN DISCHARGE REMARK
----------- ----------    -------- --------- ---------- -------------------- --------- --------- ----------
19-12345678 30-3333333      11125     22282     142860 squad leader         10-APR-19
19-12345679 30-3333333      11125     22282     142861                      08-MAY-19
19-12345680 53-5555558      11125     22282     142862 squad leader         29-MAY-19
19-12345681 53-5555558      11126     22283     142863                      29-AUG-19
19-12345682 30-3333333      11126     22283     180970                      23-OCT-19
19-12345683 53-5555558      11126     22283     195286                      13-NOV-19
19-12345684 30-3333333      11126     22283     198559                      29-NOV-19
20-12345685 30-3333333      11127     22284     306171                      15-JAN-20
20-12345686 30-3333333      11127     22284     461703                      04-FEB-20
20-12345687 53-5555558      11127     22284     478192                      27-MAR-20
20-12345688 53-5555558      11127     22284     506714                      09-APR-20


SERVICE_NUM UNIT_ID       RANK_ID  HOBONG_ID FIREARM_ID POSITION            ENLISTMEN DISCHARGE REMARK
----------- ----------    -------- --------- ---------- -------------------- --------- --------- ----------
20-12345689 30-3333333      11128     22285     639239                      11-JUN-20
20-12345690 30-3333333      11128     22285     626834                      29-JUN-20
20-12345691 53-5555558      11128     22285     656825                      02-JUL-20
99-2589655  30-3333333      11116     22241     908324 battalion commander  07-JAN-99
00-6589557  30-3333335      11116     22240     917763 battalion commander  10-SEP-00
01-6985455  53-5555552      11116     22239     946806 battalion commander  06-OCT-01
00-9875556  53-5555555      11116     22240     982600 battalion commander  30-DEC-00
99-8954585  53-5555558      11116     22241     995862 battalion commander  25-NOV-99
```

# Schema 구현

## Table 생성 및 조회

### Table <PERSONALINFO>

```
 1  CREATE TABLE personalinfo
 2  (
 3      service_num VARCHAR2(11) CONSTRAINT per_snum_pk PRIMARY KEY,
 4      name VARCHAR2(20) CONSTRAINT per_name_nn NOT NULL,
 5      rsrgnum VARCHAR2(14) CONSTRAINT per_rrn_nn NOT NULL
 6                          CONSTRAINT per_rrn_uk UNIQUE
 7                          CONSTRAINT per_rrn_ck CHECK (rsrgnum LIKE '_____-_____'),
 8      phone VARCHAR2(13) CONSTRAINT per_pho_nn NOT NULL
 9                          CONSTRAINT per_pho_uk UNIQUE
10                          CONSTRAINT per_pho_ck CHECK (phone LIKE '___-%-____'),
11      address VARCHAR2(250) CONSTRAINT per_add_nn NOT NULL,
12      remark VARCHAR2(100),
13      CONSTRAINT per_snum_sol_snum_fk FOREIGN KEY (service_num) REFERENCES Soldier (service_num)
14* ) tablespace tsoldier
```

```
SQL> select * from personalinfo;

SERVICE_NUM NAME              RSRGNUM        PHONE         ADDRESS                                          REMARK
----------- -------------     -------------  ------------- -------------                                    -------
19-12345678 OhJunil           980421-1324632 010-4143-1350 Gangnam-daero, Gangnam-gu, Seoul
19-12345679 LeeSabu           981025-1439403 010-4937-7482 Gayangnam-ro, Dong-gu, Daejeon
19-12345680 LeeJaejeong       981202-1605724 010-9400-8834 Baesan-ro, Iksan-si, Jeollakuk-do
19-12345681 KimMin            990408-1961245 010-3939-3939 Galmaesunhwan-ro, Guri-si, Gyeonggi-do
19-12345682 LeeJinho          990909-1065720 010-3949-3847 Sinseon-ro, Nam-gu, Busan
19-12345683 LeeGeuun          991012-1823580 010-9878-9845 Gayangnam-ro, Dong-gu, Daejeon
19-12345684 KimYongha         990329-1430823 010-2398-1029 Suyeong-ro, Suyeong-gu, Busan
20-12345685 KimYongjae        990525-1101272 010-6758-3849 Galmaesunhwan-ro, Guri-si, Gyeonggi-do
20-12345686 LeeMingyu         991015-1422463 010-6231-1344 Jungang-daero, Dong-gu, Busan
20-12345687 Parkhun           980810-1861854 010-7112-6162 Cheongpa-rp, Jung-gu, Seoul
20-12345688 KimYunhui         991125-1862481 010-8433-1143 Taheran-ro, Gangnam-gu, Seoul

SERVICE_NUM NAME              RSRGNUM        PHONE         ADDRESS                                          REMARK
----------- -------------     -------------  ------------- -------------                                    -------
20-12345689 KangTaeseop       990101-1030429 010-9139-2023 Nari-ro, Sejong-si
20-12345690 SonGonju          000505-3823082 010-6119-1452 Jaeban-ro, Haeundae-gu, Busan
20-12345691 JeongMin          000113-3785637 010-5912-9345 Hwaji-ro, Busanjin-gu, Busan
99-2589655  OhSejae           781126-1987465 010-7492-5123 Gunnam-ro, Yeoncheon-eup, Yeoncheon-gun, Gyeonggi-do
00-6589557  KimYusin          790603-1839932 010-8693-0243 Munhwa-ro, Yeoncheon-eup, Yeoncheon-gun, Gyeonggi-do
01-6985455  KimMinsu          801213-1039847 010-2934-3545 Mangmibeonyeing-ro, 38beon-gil, Suyeong-gu, Busan
00-9875556  HanSeok           790330-1263746 010-2339-5237 Suyeong-ro, Suyeong-gu, Busan
```

# Schema 구현

## Table 생성 및 조회

### Table <RANK>

```
1   CREATE TABLE Rank
2   (
3       rank_id    NUMBER(5) CONSTRAINT ran_rid_pk PRIMARY KEY,
4       rank       VARCHAR2(30) CONSTRAINT ran_ran_nn NOT NULL
5* ) tablespace tsoldier
```

```
SQL> select * from rank;

  RANK_ID RANK
---------- ------------------------------
    11111 general
    11112 lieutenant general
    11113 major general
    11114 brigadier general
    11115 colonel
    11116 lieutenant colonel
    11117 major
    11118 captain
    11119 first lieutenant
    11120 second lieutenant
    11121 warrant officer

  RANK_ID RANK
---------- ------------------------------
    11122 master sergeant
    11123 sergeant
    11124 staff sergeant
    11125 first class
    11126 corporal
    11127 private first class
    11128 private second class

18 rows selected.
```

### Table <FIREARM>

```
SQL> CREATE TABLE Firearm
  2  ( firearm_id NUMBER(6) CONSTRAINT fir_fid_pk PRIMARY KEY,
  3    firearm_name VARCHAR2(10) CONSTRAINT fir_fna_nn NOT NULL ) tablespace tsoldier;
```

```
SQL> select * from firearm;

FIREARM_ID FIREARM_NA
---------- ----------
    142860 K1
    142861 K1
    142862 K1
    142863 K1
    180970 K1
    195286 K1
    198559 K1
    306171 K1
    461703 K1
    478192 K2
    506714 K2

FIREARM_ID FIREARM_NA
---------- ----------
    639239 K2
    626834 K2
    656825 K2
    908324 K2
    917763 K2
    946806 K2
    982600 K2
    995862 K2
```

# Schema 구현

## Table 생성 및 조회

### Table <SALARY>

```
1   CREATE TABLE salary
2   (
3       hobong_id NUMBER(5) CONSTRAINT sal_hid_pk PRIMARY KEY,
4       rank_id NUMBER(5),
5       hobong NUMBER(2),
6       salary NUMBER(10) CONSTRAINT sal_sal_nn NOT NULL,
7       CONSTRAINT sal_rid_ran_rid_fk FOREIGN KEY (rank_id) REFERENCES Rank (rank_id)
8* ) tablespace tsoldier
```

```
SQL> select * from salary;

HOBONG_ID    RANK_ID      HOBONG      SALARY
---------- ---------- ---------- ----------
    22222      11111                8431600
    22223      11112                8281300
    22224      11113          1     5346400
    22225      11113          2     5478600
    22226      11113          3     5610800
    22227      11113          4     5743000
    22228      11113          5     5875200
    22229      11114          1     5043700
    22230      11114          2     5174700
    22231      11114          3     5305700
    22232      11114          4     5436700

HOBONG_ID    RANK_ID      HOBONG      SALARY
---------- ---------- ---------- ----------
    22233      11114          5     5567700
    22234      11115          1     4094500
    22235      11115          2     4232200
    22236      11115          3     4369900
    22237      11115          4     4507600
    22238      11115          5     4645300
    22239      11116          1     3599200
    22240      11116          2     3736800
    22241      11116          3     3874400
    22242      11116          4     4012000
    22243      11116          5     4149600
```

### Table <UNIT>

```
1   CREATE TABLE unit
2   (
3       unit_id VARCHAR2(10) CONSTRAINT uni_uid_pk PRIMARY KEY,
4       unit_name VARCHAR2(50) CONSTRAINT uni_una_nn NOT NULL
5                             CONSTRAINT uni_una_uk UNIQUE,
6       division NUMBER(3) CONSTRAINT uni_div_nn  NOT NULL,
7       location_id NUMBER(3),
8       unit_phone VARCHAR2(12) CONSTRAINT uni_pho_nn NOT NULL
9                             CONSTRAINT uni_pho_uk UNIQUE
10                            CONSTRAINT uni_pho_ck CHECK ( unit_phone LIKE '%-%-%'),
11      CONSTRAINT uni_lid_loc_lid_fk FOREIGN KEY (location_id) REFERENCES location (loca
tion_id)
12* ) tablespace tunit
```

```
SQL> select * from unit;

UNIT_ID    UNIT_NAME                         DIVISION LOCATION_ID UNIT_PHONE
---------- ------------------------------- ---------- ----------- ------------
30-3333333 30_Signal battalion                    30         260 031-123-5865
30-3333334 30_Air defense battalion               30         260 031-123-5866
30-3333335 30_Maintenance battalion               30         260 031-123-5867
30-3333336 30_Military police                     30         260 031-123-5868
30-3333337 30_Military band                       30         260 031-123-5869
30-3333338 30_Medical corps                       30         260 031-123-5870
30-3333339 30_Reconnaissance battalion            30         260 031-123-5871
53-5555551 53_Air defense battalion               53         262 051-571-9935
53-5555552 53_Maintenance battalion               53         262 051-571-9936
53-5555553 53_Military police                     53         262 051-571-9937
53-5555554 53_Military band                       53         262 051-571-9938

UNIT_ID    UNIT_NAME                         DIVISION LOCATION_ID UNIT_PHONE
---------- ------------------------------- ---------- ----------- ------------
53-5555555 53_Medical corps                       53         262 051-571-9939
53-5555556 53_Reconnaissance battalion            53         262 051-571-9940
53-5555558 53_Signal battalion                    53         262 051-571-9942
```

# Schema 구현

## Table 생성 및 조회

### Table <LOCATION>

```
1   CREATE TABLE location
2   (
3       location_id NUMBER(3) CONSTRAINT loc_lid_pk PRIMARY KEY,
4       provinces VARCHAR2(10) CONSTRAINT loc_pro_nn NOT NULL,
5       city VARCHAR2(10) CONSTRAINT loc_cit_nn NOT NULL
6* ) tablespace tunit
```

```
SQL> select * from location;

LOCATION_ID PROVINCES   CITY
----------- ----------  ----------
        259 Gyeonggido  Gapyeong
        260 Gyeonggido  Yeoncheon
        261 Gyeonggido  Yangju
        262 Busan       Haeundaegu
        263 Busan       Gangseogu
```

### Table <OVERSEASUNIT>

```
1   CREATE TABLE overseasunit
2   (
3       oseas_id VARCHAR2(10) CONSTRAINT osu_oid_pk PRIMARY KEY,
4       oseas_unit VARCHAR2(20) CONSTRAINT osu_uni_nn NOT NULL
5                              CONSTRAINT osu_uni_uk unique,
6       oseas_startdate DATE CONSTRAINT osu_stat_nn NOT NULL,
7       oseas_enddate DATE,
8       oseas_country VARCHAR2(50)
9* ) tablespace tto
```

```
SQL> ALTER TABLE overseasunit
  2  ADD ( oseas_troops number(10));

Table altered.
```

```
SQL> select * from overseasunit;

OSEAS_ID    OSEAS_UNIT          OSEAS_STA OSEAS_END OSEAS_COUNTRY              OSEAS_TROOPS
---------- ------------------   --------- --------- -------------------------  ------------
OS-157878  17_Arc unit          25-NOV-17 10-JUL-18 United Arab Emirates               140
OS-157898  18_Arc unit          28-JUL-18 11-MAR-19 United Arab Emirates               125
OS-186988  15_Hanbit unit       23-FEB-15 25-OCT-16 Republic of South Sudan            275
OS-186989  16_Hanbit unit       25-OCT-16 28-JUN-17 Republic of South Sudan            269
OS-254785  04_Zaytun division   12-AUG-04 24-JUN-05 Iraq                              3550
OS-254786  05_Zaytun division   21-FEB-05 11-DEC-05 Iraq                              3459
OS-487988  13_Araw contingent   27-DEC-13 23-DEC-14 Philippines                        520
```

# Schema 구현

## Sequence

### Rank Table의 rank_id

```
SQL> create sequence ran_ranid_seq
  2  increment by 1
  3  start with 11111
  4  nomaxvalue
  5  nocache
  6  nocycle;

Sequence created.
```

```
SQL> select * from rank;

   RANK_ID RANK
---------- --------------------
     11111 general
     11112 lieutenant general
     11113 major general
     11114 brigadier general
     11115 colonel
     11116 lieutenant colonel
     11117 major
     11118 captain
     11119 first lieutenant
     11120 second lieutenant
     11121 warrant officer
```

### Salary 테이블의 hobong_id

```
SQL> create sequence sal_salid_seq
  2  increment by 1
  3  start with 22222
  4  nomaxvalue
  5  nocache
  6  nocycle;

Sequence created.
```

```
SQL> select * from salary;

 HOBONG_ID    RANK_ID     HOBONG     SALARY
---------- ---------- ---------- ----------
     22222      11111              8431600
     22223      11112              8281300
     22224      11113          1   5346400
     22225      11113          2   5478600
     22226      11113          3   5610800
     22227      11113          4   5743000
     22228      11113          5   5875200
     22229      11114          1   5043700
     22230      11114          2   5174700
     22231      11114          3   5305700
     22232      11114          4   5436700
```

### Location 테이블의 Location_id

```
SQL> create sequence loc_locid_seq
  2  increment by 1
  3  start with 259
  4  nomaxvalue
  5  nocache
  6  nocycle;

Sequence created.
```

```
SQL> select * from location;

LOCATION_ID PROVINCES  CITY
----------- ---------- ----------
        259 Gyeonggido Gapyeong
        260 Gyeonggido Yeoncheon
        261 Gyeonggido Yangju
        262 Busan      Haeundaegu
        263 Busan      Gangseogu
```

# Schema 구현

## View

### 부대별 대대장 정보조회

```
1   CREATE VIEW v_commander (service_num,position,name,phone,unit,rank)
2   AS select s.service_num,s.position,p.name,p.phone,s.unit,s.rank
3   from v_soldierinfo s join personalinfo p on s.service_num=p.service_num
4*  AND s.position IS NOT NULL
```

```
SQL> select * from v_commander;

SERVICE_NUM  POSITION            NAME        PHONE          UNIT                    RANK
-----------  ------------------  ----------  -------------  ----------------------  ------------------
19-12345678  squad leader        OhJunil     010-4143-1350  30_Signal battalion     first class
19-12345680  squad leader        LeeJaejeong 010-9400-8834  53_Signal battalion     first class
99-2589655   battalion commander OhSejae     010-7492-5123  30_Signal battalion     lieutenant colonel
00-6589557   battalion commander KimYusin    010-8693-0243  30_Maintenance battalion lieutenant colonel
01-6985455   battalion commander KimMinsu    010-2934-3545  53_Maintenance battalion lieutenant colonel
00-9875556   battalion commander HanSeok     010-2339-5237  53_Medical corps        lieutenant colonel
99-8954585   battalion commander YuGwansun   010-7997-2844  53_Signal battalion     lieutenant colonel
```

### 부대별 병사조회

```
1   create view v_solunit(unit_name, division, service_num, name)
2   as select u.unit_name, u.division, s.service_num, p.name
3   from unit u JOIN soldier s on u.unit_id = s.unit_id
4*  join personalinfo p on s.service_num=p.service_num
```

```
SQL> select * from v_solunit order by unit_name;

UNIT_NAME                     DIVISION  SERVICE_NUM  NAME
----------------------------  --------  -----------  ----------------
30_Maintenance battalion      30        00-6589557   KimYusin
30_Signal battalion           30        19-12345679  LeeSabu
30_Signal battalion           30        19-12345678  OhJunil
30_Signal battalion           30        99-2589655   OhSejae
30_Signal battalion           30        19-12345682  LeeJinho
30_Signal battalion           30        20-12345689  KangTaeseop
30_Signal battalion           30        19-12345684  KimYongha
30_Signal battalion           30        20-12345685  KimYongjae
30_Signal battalion           30        20-12345686  LeeMingyu
30_Signal battalion           30        20-12345690  SonGonju
```

# Schema 구현

## View

### 개인정보 조회

```
1  CREATE VIEW v_soldierinfo(service_num,position,unit,rank)
2  as select s.service_num , s.position , u.unit_name , r.rank
3* from soldier s join unit u on s.unit_id=u.unit_id join rank r on s.rank_id=r.rank_id
```

```
SQL> select * from v_solname;

SERVICE_NUM NAME                    PHONE         ADDRESS                                          REMARK
----------- ----------------------- ------------- ------------------------------------------------ ------
19-12345678 OhJunil                 010-4143-1350 Gangnam-daero, Gangnam-gu, Seoul
19-12345679 LeeSabu                 010-4937-7482 Gayangnam-ro, Dong-gu, Daejeon
19-12345680 LeeJaejeong             010-9400-8834 Baesan-ro, Iksan-si, Jeollakuk-do
19-12345681 KimMin                  010-3939-3939 Galmaesunhwan-ro, Guri-si, Gyeonggi-do
19-12345682 LeeJinho                010-3949-3847 Sinseon-ro, Nam-gu, Busan
19-12345683 LeeGeuun                010-9878-9845 Gayangnam-ro, Dong-gu, Daejeon
19-12345684 KimYongha               010-2398-1029 Suyeong-ro, Suyeong-gu, Busan
20-12345685 KimYongjae              010-6758-3849 Galmaesunhwan-ro, Guri-si, Gyeonggi-do
20-12345686 LeeMingyu               010-6231-1344 Jungang-daero, Dong-gu, Busan
20-12345687 Parkhun                 010-7112-6162 Cheongpa-rp, Jung-gu, Seoul
20-12345688 KimYunhui               010-8433-1143 Taheran-ro, Gangnam-gu, Seoul

SERVICE_NUM NAME                    PHONE         ADDRESS                                          REMARK
----------- ----------------------- ------------- ------------------------------------------------ ------
20-12345689 KangTaeseop             010-9139-2023 Nari-ro, Sejong-si
20-12345690 SonGonju                010-6119-1452 Jaeban-ro, Haeundae-gu, Busan
20-12345691 JeongMin                010-5912-9345 Hwaji-ro, Busanjin-gu, Busan
```

### 파병부대 예상 기간 조회

```
1  CREATE VIEW v_osunit ( id, unit, period)
2  as select oseas_id, oseas_unit, oseas_enddate-oseas_startdate
3  from overseasunit
4* where oseas_enddate is not null
```

```
SQL> select * from v_osunit;

ID         UNIT                      PERIOD
---------- ------------------------- ----------
OS-157878  17_Arc unit                  227
OS-157898  18_Arc unit                  226
OS-186988  15_Hanbit unit               610
OS-186989  16_Hanbit unit               246
OS-254785  04_Zaytun division           316
OS-254786  05_Zaytun division           293
OS-487988  13_Araw contingent           361

7 rows selected.
```
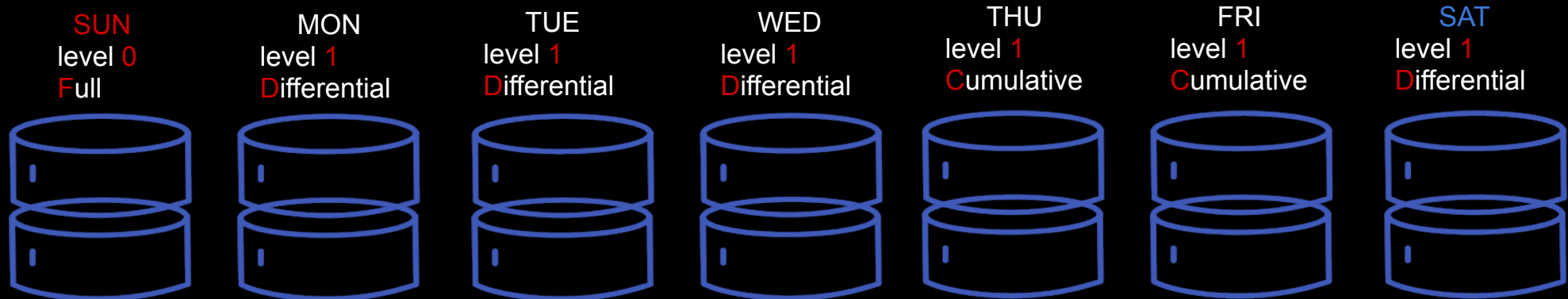
# Backup 전략

## Backup 계획 및 스케줄

### Backup 계획

| 백업도구 | RMAN |
|---|---|
| 백업전략 | 전체 DB 백업 |
| 백업모드 | Incremental 백업 |
| 백업파일 유형 | Backup Set |

### Backup 스케줄 (매일 오전 2시)

| SUN | MON | TUE | WED | THU | FRI | SAT |
|---|---|---|---|---|---|---|
| level 0 | level 1 | level 1 | level 1 | level 1 | level 1 | level 1 |
| Full | Differential | Differential | Differential | Cumulative | Cumulative | Differential |

# Backup 전략

## Backup 전략

### Backup 위치 설정

```
SQL> show parameter db_recovery_file_dest

NAME                             TYPE        VALUE
-------------------------------- ----------- ------------------------------
db_recovery_file_dest            string      /u01/app/oracle/fast_recovery_
                                             area/orcl
db_recovery_file_dest_size       big integer 8016M
```

### Backup 유형 설정

```
RMAN> show device type;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name ORCL are:
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
```

### controlfile 자동 백업 기능 설정

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

```
RMAN> show all;

RMAN configuration parameters for database with db_unique_name ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u01/app/oracle/product/12.2.0/dbhome_1/dbs/snapcf_orcl.f'; # default
```

# Backup 전략

## Backup 전략 & 모드

### 전체 DB Backup



### Incremental Backup

# Database 보안 구현

## 전자지갑

### 전자지갑 parameter 설정

$ORACLE_HOME/network/admin/sqlnet.ora

```
ENCRYPTION_WALLET_LOCATION=
( SOURCE=
        (METHOD=FILE)
        (METHOD_DATA=
                (DIRECTORY=/u01/app/oracle/admin/orcl/wallet)))

~
```

### 전자지갑 생성

```
SQL> alter system set encryption key identified by army_2020;

System altered.
```

army_2020 암호를
사용한 전자지갑 생성

# Database 보안 구현

## Tablespace

### Encrypted_tto Tablespace

```
SQL> ed
Wrote file afiedt.buf

  1  create tablespace encrypted_tto
  2  datafile '/u01/app/oracle/oradata/ORCL/datafile/entto01.dbf' size 50M
  3  autoextend on next 50M
  4* encryption default storage(encrypt)
SQL> /

Tablespace created.
```

```
SQL> select tablespace_name, encrypted
  2  from dba_tablespaces;

TABLESPACE_NAME                 ENC
------------------------------- ---
SYSTEM                          NO
SYSAUX                          NO
UNDOTBS1                        NO
TEMP                            NO
USERS                           NO
TUNIT                           NO
TSOLDIER                        NO
ENCRYPTED_TTO                   YES
```

Tablespace
암호화 및 조회

```
SQL> alter system set encryption wallet close identified by army_2020;

System altered.
```

Wallet close 후 조회

```
SQL> select * from overseasunit;
select * from overseasunit
              *
ERROR at line 1:
ORA-28365: wallet is not open
```

# Database 보안 구현

## Column

```
SQL> alter table personalinfo
  2  modify ( rsrgnum encrypt no salt);

Table altered.
```

```
SQL> alter system set encryption wallet close identified by army_2020;

System altered.
```

특정 column
암호화 및 조회

후 조회

Wallet close 후 조회

```
SQL> desc personalinfo;
 Name                                      Null?    Type
 ---------------------------------------- -------- -----------------------
 SERVICE_NUM                               NOT NULL VARCHAR2(11)
 NAME                                      NOT NULL VARCHAR2(20)
 RSRGNUM                                   NOT NULL VARCHAR2(14) ENCRYPT
 PHONE                                     NOT NULL VARCHAR2(13)
 ADDRESS                                   NOT NULL VARCHAR2(250)
 REMARK                                             VARCHAR2(100)
```

```
SQL> select * from user_encrypted_columns;

TABLE_NAME        COLUMN_NAME      ENCRYPTION_ALG          SAL INTEGRITY_AL
----------------- ---------------- ----------------------- --- ------------
PERSONALINFO      RSRGNUM          AES 192 bits key         NO  SHA-1
```

```
SQL> select * from personalinfo;
select * from personalinfo
                          *
ERROR at line 1:
ORA-28365: wallet is not open
```

# Database 보안 구현

## 암호화 백업

RMAN 이용 암호화된 백업기능 설정

```
RMAN> configure encryption for database on;

new RMAN configuration parameters:
CONFIGURE ENCRYPTION FOR DATABASE ON;
new RMAN configuration parameters are successfully stored

RMAN> configure encryption algorithm 'AES192';

new RMAN configuration parameters:
CONFIGURE ENCRYPTION ALGORITHM 'AES192';
new RMAN configuration parameters are successfully stored
```

```
SQL> select algorithm_name, is_default
  2  from v$rman_encryption_algorithms;

ALGORITHM_NAME                                          IS_
---------------------------------------------------- ---
AES128                                                  YES
AES192                                                  NO
AES256                                                  NO
```

=> 백업시 지원되는 암호화 알고리즘 조회

AES192 암호화 알고리즘 설정

```
RMAN> configure encryption algorithm 'AES192';

using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE ENCRYPTION ALGORITHM 'AES192';
new RMAN configuration parameters:
CONFIGURE ENCRYPTION ALGORITHM 'AES192';
new RMAN configuration parameters are successfully stored
```

# 모니터링

## 파라메터 값 조회

```
SQL> show parameter shared_pool_size

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
shared_pool_size                big integer 0
```

```
SQL> show parameter sga_target

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
sga_target                      big integer 0
```

```
SQL> show parameter db_cache_size

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
db_cache_size                   big integer 0
```

```
SQL> show parameter log_buffer

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
log_buffer                      big integer 7624K
```

```
SQL> show parameter pga_aggregate_target

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
pga_aggregate_target            big integer 0
```

```
SQL> show parameter memory_target

NAME                            TYPE        VALUE
------------------------------- ----------- ----------
memory_target                   big integer 1520M
```

# 모니터링

## 저장영역 모니터링

### Datafiles

```
SQL> select name, ts#, status, bytes
  2  from v$datafile;

NAME                                           TS# STATUS        BYTES
---------------------------------------- ---------- --------- ----------
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_system   0 SYSTEM    838860800
_hdyt40hy_.dbf

/u01/app/oracle/oradata/ORCL/datafile/o1_mf_tsoldi   7 ONLINE    104857600
er.dbf

/u01/app/oracle/oradata/ORCL/datafile/o1_mf_sysaux   1 ONLINE    524288000
_hdyt53pw_.dbf

/u01/app/oracle/oradata/ORCL/datafile/o1_mf_undotb   2 ONLINE     73400320
s1_hdyt5lv6_.dbf

NAME                                           TS# STATUS        BYTES
---------------------------------------- ---------- --------- ----------

/u01/app/oracle/oradata/ORCL/datafile/o1_mf_tunit.   6 ONLINE    104857600
dbf

/u01/app/oracle/oradata/ORCL/datafile/o1_mf_users_   4 ONLINE      5242880
hdyt5n0g_.dbf

/u01/app/oracle/oradata/ORCL/datafile/entto01.dbf    9 ONLINE     52428800
```

### Controlfiles

```
SQL> select name, block_size, file_size_blks
  2  from v$controlfile;

NAME                                          BLOCK_SIZE FILE_SIZE_BLKS
---------------------------------------- --------------- --------------
/u01/app/oracle/oradata/ORCL/controlfile/o1_mf_hdy   16384           646
t6n5j_.ctl

/u01/app/oracle/fast_recovery_area/orcl/ORCL/contr   16384           646
olfile/o1_mf_hdyt6n68_.ctl

/u01/app/oracle/oradata/ORCL/datafile/control03.ct   16384           646
l
```

### Redologfile

```
SQL> select group#, status, members, archived, bytes, sequence#, first_change#
  2  from v$log;

    GROUP# STATUS           MEMBERS ARC       BYTES  SEQUENCE# FIRST_CHANGE#
---------- ---------------- ------- --- ---------- ---------- -------------
         1 INACTIVE               2 YES  209715200          5      1561925
         2 CURRENT                2 NO   209715200          6      1584652
         3 INACTIVE               2 YES  209715200          3      1561788
         4 INACTIVE               2 YES  104857600          4      1561832
```

# 모니터링

## 메모리 모니터링

### SGA 조회

```
SQL> select * from v$sgainfo;

NAME                                   BYTES RES     CON_ID
------------------------------- ------------- ---     ----------
Fixed SGA Size                       8621184 No            0
Redo Buffers                         8155136 No            0
Buffer Cache Size                  654311424 Yes           0
In-Memory Area Size                        0 No            0
Shared Pool Size                   251658240 Yes           0
Large Pool Size                     16777216 Yes           0
Java Pool Size                      16777216 Yes           0
Streams Pool Size                          0 Yes           0
Shared IO Pool Size                 50331648 Yes           0
Data Transfer Cache Size                   0 Yes           0
Granule Size                        16777216 No            0

NAME                                   BYTES RES     CON_ID
------------------------------- ------------- ---     ----------
Maximum SGA Size                  1593835520 No            0
Startup overhead in Shared Pool    193551392 No            0
Free SGA Memory Available          637534208               0
```

### PGA 메모리사용량 조회

```
SQL> select program, pga_used_mem, pga_alloc_mem, pga_max_mem
  2   from v$process
  3   order by 1;

PROGRAM                        PGA_USED_MEM PGA_ALLOC_MEM PGA_MAX_MEM

PSEUDO                                    0            0            0
oracle@localhost.localdomain (AQPC)  1575028      2027932      2027932
oracle@localhost.localdomain (ARC0) 18043548     19394972     19394972
oracle@localhost.localdomain (ARC1) 18043548     19394972     19394972
oracle@localhost.localdomain (ARC2) 18043548     19394972     19394972
oracle@localhost.localdomain (ARC3) 18043548     19394972     19394972
oracle@localhost.localdomain (CJQ0)  3400948      9433468      9433468
oracle@localhost.localdomain (CKPT)  1152036      1603188      1603188
oracle@localhost.localdomain (CLMN)  1118108      1257924      1257924
oracle@localhost.localdomain (D000)  1410124      1492724      1501124
oracle@localhost.localdomain (DBRM)  1371604      1634716      1634716

PROGRAM                        PGA_USED_MEM PGA_ALLOC_MEM PGA_MAX_MEM

oracle@localhost.localdomain (DBW0)  7832020      9031756      9048812
oracle@localhost.localdomain (DIA0)  4029268      4173564      4173564
oracle@localhost.localdomain (DIAG)  1095236      1241500      1241500
oracle@localhost.localdomain (GEN0)  1137844      1241468      1241468
oracle@localhost.localdomain (GEN1)  1099340      1241500      1241500
oracle@localhost.localdomain (LG00)   940460      1042372      1042372
oracle@localhost.localdomain (LG01)   940460      1042372      1042372
oracle@localhost.localdomain (LGWR)  1309844      1774156      1774156
oracle@localhost.localdomain (LREG)  1593596      2159004      2159004
oracle@localhost.localdomain (MMAN)  1098340      1241500      1241500
oracle@localhost.localdomain (MMNL)  1124932      1438108      1438108
```

# 모니터링

## DB에 접속된 세션 모니터링

```
SQL> select sid, serial#, username, user#, status
  2  from v$session
  3  where username is not null;

       SID    SERIAL# USERNAME             USER# STATUS
---------- ---------- ---------- ---------- --------
         3      51106 SYS                      0 ACTIVE
         8      28251 USERDBA                106 INACTIVE
        17      64133 SYS                      0 INACTIVE
       146      28704 USERDBA                106 INACTIVE
       272      22702 SYS                      0 ACTIVE
       398      15517 USERDBA                106 INACTIVE
```

# 모니터링

## EM을 통한 DB전체 성능 모니터링

# 스크립트

치환변수를 사용하여 데이터 조작시 편리한 스크립트 파일

1) Insert 문

```
SQL> @/home/oracle/sql/Insertsoldier.sql
Enter value for s_num:
Enter value for u_id:
Enter value for r_id:
Enter value for h_id:
Enter value for f_id:
Enter value for position:
Enter value for startdate:
Enter value for enddate:
Enter value for remark:
```

실행

```
Insert into soldier
values('&s_num','&u_id',&r_id,&h_id,&f_id,'&position','&startdate','&enddate','&remark')

Insert into personalinfo(service_num,name,rsrgnum,phone,address)
values('&s_num','&name','&rsrg','&phone','&add')

Insert into salary
values (sal_salid_seq.NEXTVAL,&rank_id,&hobong,&salary)

Insert into rank
values (ran_ranid_seq.NEXTVAL, '&rank')

Insert into firearm
values (&f_id,'&f_name')

Insert into unit
values ('&u_id', '&u_name', &division, &l_id, '&u_phone')

Insert into location
values (loc_locid_seq.NEXTVAL, '&rovinces', '&city')

Insert into overseasunit(oseas_id, oseas_unit,oseas_startdate,oseas_enddate,oseas_country,oseas_troops)
values('&ose_id', '&ose_unit','&ose_startdate' ,'&ose_enddate' ,'&oseas_Country','&oseas_troops')
```

# 스크립트

치환변수를 사용하여 데이터 조작시 편리한 스크립트 파일

2) Select 문

```
SELECT * FROM v_solname
WHERE lower(name) = lower('&soldiername')

SQL> @/home/oracle/sql/Selectvsolname.sql
Enter value for soldiername:
```

실행

```
SQL> @/home/oracle/sql/Selectvsolname.sql
Enter value for soldiername: Kimmin
old    2: WHERE lower(name) = lower('&soldiername')
new    2: WHERE lower(name) = lower('Kimmin')

SERVICE_NUM NAME          PHONE         ADDRESS                    REMAR
----------- ------------- ------------- -------------------------- -----
19-12345681 KimMin        010-3939-3939 Galmaesunhwan-ro, Gu
                                        ri-si, Gyeonggi-do
```

```
SELECT * FROM v_solunit
WHERE lower(unit_name) = lower('&unit_name')

SQL> @/home/oracle/sql/Selectvsolunit.sql
Enter value for unit_name:
```

실행

```
SQL> @/home/oracle/sql/Selectvsolunit.sql
Enter value for unit_name: 53_signal battalion
old    2: WHERE lower(unit_name) = lower('&unit_name')
new    2: WHERE lower(unit_name) = lower('53_signal battalion')

UNIT_NAME            DIVISION SERVICE_NUM NAME
------------------- --------- ----------- ----------------
53_Signal battalion       53 19-12345680 LeeJaejeong
53_Signal battalion       53 19-12345681 KimMin
53_Signal battalion       53 19-12345683 LeeGeuun
53_Signal battalion       53 20-12345687 Parkhun
53_Signal battalion       53 20-12345688 KimYunhui
53_Signal battalion       53 20-12345691 JeongMin
53_Signal battalion       53 99-8954585  YuGwansun
```

# 스크립트

치환변수를 사용하여 데이터 조작시 편리한 스크립트 파일

3) 테이블에 설정된 제약조건 확인

```
SELECt c1.column_name, c1.constraint_name, c2.constraint_type, c2.search_condition,c2.r_constraint_name
FROm user_cons_columns c1 JOIN user_constraints c2
ON c1.constraint_name = c2.constraint_name
AND lower(c1.table_name) = lower('&t_name');
```

실행

```
SQL> @/home/oracle/sql/cons.sql
Enter value for t_name: personalinfo
old   4: AND lower(c1.table_name) = lower('&t_name')
new   4: AND lower(c1.table_name) = lower('personalinfo')

COLUMN_NAME      CONSTRAINT_NAME           C SEARCH_CONDITION              R_CONSTRAINT_NAME
---------------  -----------------------   - --------------------------   -------------------------
NAME             PER_NAME_NN               C "NAME" IS NOT NULL
RSRGNUM          PER_RRN_NN                C "RSRGNUM" IS NOT NULL
PHONE            PER_PHO_NN                C "PHONE" IS NOT NULL
ADDRESS          PER_ADD_NN                C "ADDRESS" IS NOT NULL
RSRGNUM          PER_RRN_CK                C rsrgnum LIKE '_____-_____'
PHONE            PER_PHO_CK                C phone LIKE '___-%-____'
SERVICE_NUM      PER_SNUM_PK               P
RSRGNUM          PER_RRN_UK                U
PHONE            PER_PHO_UK                U
SERVICE_NUM      PER_SNUM_SOL_SNUM_FK      R                              SOL_SNUM_PK
```

# 스크립트

DB에 접속된 세션 정보 및 세션들의 LOCK충돌 모니터링 할 수 있는 스크립트 파일

1) DB에 접속된 세션 정보 모니터링 스크립트 파일

```
select sid, serial#, username, user#, status
from v$session
where username is not null
```

실행 →

```
SQL> @/home/oracle/sql/session_info.sql

      SID    SERIAL# USERNAME           USER# STATUS
---------- ---------- ---------- ---------- ----------
        3      51106 SYS                    0 ACTIVE
      259       5752 SYS                    0 ACTIVE
```

2) 세션들의 LOCK충돌 모니터링 할 수 있는 스크립트 파일

```
select sid, serial#, username, status, blocking_session
from v$session
where sid in (select blocking_session from v$session)
```

실행 →

```
SQL> @/home/oracle/sql/session_lock.sql

no rows selected
```

# 스크립트

저장영역(T/S)을 모니터링 할 수 있는 스크립트 파일

```sql
select tablespace_name, status, contents, logging, extent_management,
       allocation_type, segment_space_management
from dba_tablespaces
```

실행

```
SQL> @/home/oracle/sql/ts.sql

TABLESPACE STATUS      CONTENTS            LOGGING   EXTENT_MAN ALLOCATIO SEGMEN
---------- ---------   ------------------- --------- ---------- --------- ------
SYSTEM     ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    MANUAL
SYSAUX     ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
UNDOTBS1   ONLINE      UNDO                LOGGING   LOCAL      SYSTEM    MANUAL
TEMP       ONLINE      TEMPORARY           NOLOGGING LOCAL      UNIFORM   MANUAL
USERS      ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
TUNIT      ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
TSOLDIER   ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
TTO        ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
ENCRYPTED_ ONLINE      PERMANENT           LOGGING   LOCAL      SYSTEM    AUTO
TTO
```

# 스크립트

## 그 외 DB관리 시 필요한 추가 스크립트 파일 구성하기

1) datafile 조회

```
select ts.name tablespace, d.name datafile
from v$tablespace ts join v$datafile d
on (ts.ts# = d.ts#)
```

실행 →

```
SQL> @/home/oracle/sql/sel_data.sql
TABLESPACE              DATAFILE
--------------------    --------------------------------------------
SYSTEM                  /u01/app/oracle/oradata/ORCL/datafile/o1_mf_system
                        _hdyt40hy_.dbf

TSOLDIER                /u01/app/oracle/oradata/ORCL/datafile/o1_mf_tsoldi
                        er.dbf

SYSAUX                  /u01/app/oracle/oradata/ORCL/datafile/o1_mf_sysaux
                        _hdyt53pw_.dbf

UNDOTBS1                /u01/app/oracle/oradata/ORCL/datafile/o1_mf_undotb
                        s1_hdyt5lv6_.dbf

TABLESPACE              DATAFILE
--------------------    --------------------------------------------
TUNIT                   /u01/app/oracle/oradata/ORCL/datafile/o1_mf_tunit.
                        dbf

USERS                   /u01/app/oracle/oradata/ORCL/datafile/o1_mf_users_
                        hdyt5n0g_.dbf

TTO                     /u01/app/oracle/oradata/ORCL/datafile/o1_mf_tto.db
                        f

ENCRYPTED_TTO           /u01/app/oracle/oradata/ORCL/datafile/entto01.dbf
```

# 스크립트

그 외 DB관리 시 필요한 추가 스크립트 파일 구성하기

2) Controlfile 조회

```
select name
from v$controlfile
```

실행 →

```
SQL> @/home/oracle/sql/sel_con.sql

NAME
--------------------------------------------------------------------
/u01/app/oracle/oradata/ORCL/controlfile/o1_mf_hdyt6n5j_.ctl
/u01/app/oracle/fast_recovery_area/orcl/ORCL/controlfile/o1_mf_hdyt6n68_.ctl
/u01/app/oracle/oradata/ORCL/datafile/control03.ctl
```

3) Redo log file 조회

```
select group#, members
from v$log
```

실행 →

```
SQL> @/home/oracle/sql/sel_log.sql

    GROUP#    MEMBERS
---------- ----------
         1          2
         2          2
         3          2
         4          2
```

# Thank You ..