

—
마이그레이션
—

Tibero

전환 가이드



Copyright © 2024 TmaxTibero. All Rights Reserved

Copyright Notice

Copyright © 2024 TIBERO Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로 258 번길 29, 티맥스수내타워 우)13595

Website

www.tmaxtibero.com

Restricted Rights Legend

All TIBERO Software (Tibero®) and documents are protected by copyright laws and international convention. TIBERO software and documents are made available under the terms of the TIBERO License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TIBERO Co., Ltd.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TIBERO Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TIBERO의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2 차적 저작물 작성 등의 행위를 하여서는 안 됩니다.

Trademarks

Tibero® is a registered trademark of TIBERO Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TIBERO Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

안내서 정보

안내서 제목: Tibero 전환 가이드

발행일: 2024-02-14

소프트웨어 버전: Tibero7 FS02

안내서 버전: 1.0

제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0	2024.02.14	최초 제정	작성자:윤준수

목록

제 1장 개요.....	4
1.1. 전환 전 고려사항.....	4
1.2. 전환 방법.....	7
1.3. 전환 방법 적용.....	11
제 2장 Oracle 전환	13
2.1. 사전 점검.....	13
2.2. 전환 수행.....	27
2.3. 애플리케이션 전환.....	53
Appendix. 데이터 타입.....	56

제 1장 개요

본 장에서는 전환 전 고려사항과 전환 방법에 대해서 설명한다.

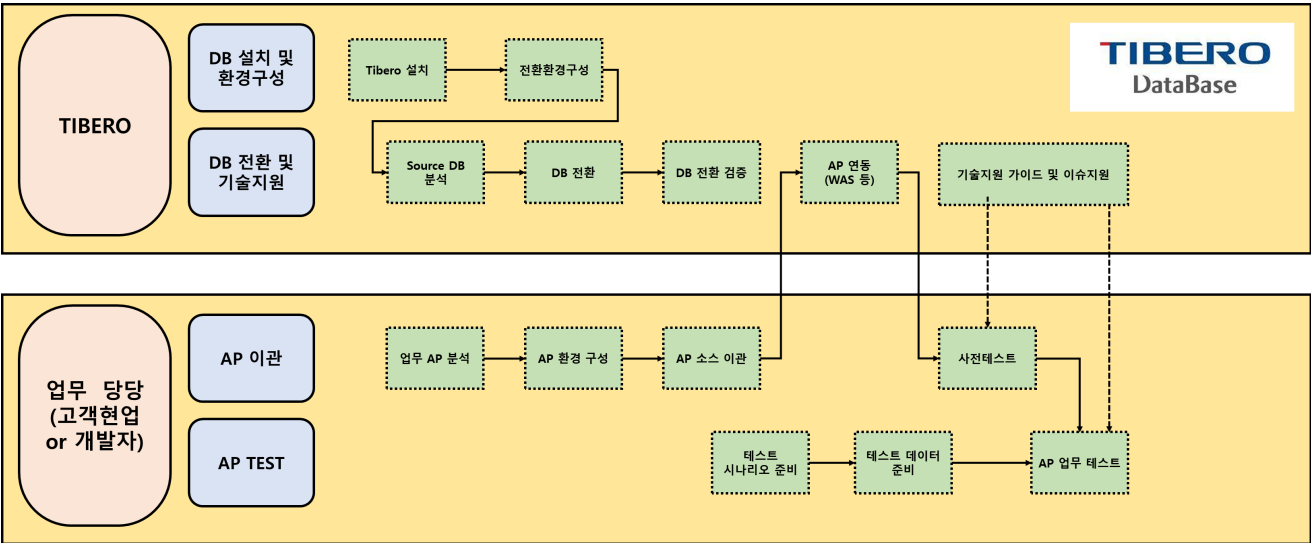
1.1. 전환 전 고려사항

본 절에서는 전환 전 고려사항에 대해서 설명한다.

전환 수행 절차

다음은 DB 전환 수행 절차이다. 전환을 위해서는 DB 전환팀과 AP 전환 및 테스트팀이 필요하다.

[그림1.1] 전환 수행 절차



전환 난이도

다른 DB에서 Tiberio DB로 전환할 때 상황에 따라 난이도가 구분되며 난이도에 따라 전환 방법이 달라질 수 있다.

다음은 각 난이도에 대한 상황의 설명이다.

난이도	상황
쉬움	몇 개의 Schema(=사용자)로 구성되어 있다. Object들 사이에 의존성이 복잡하게 구성되어 있지 않다. 실제 사용하는 Object들로 구성되어 있으며 모두 VALID 상태이다.

난이도	상황
어려움	다수의 Schema(=사용자)로 구성되어 있다. Object들 사이에 의존성이 복잡하게 구성되어 있다. Object들이 사용되지 않는 것들이 많으며 INVALID 상태이다.

전환 전 권장사항

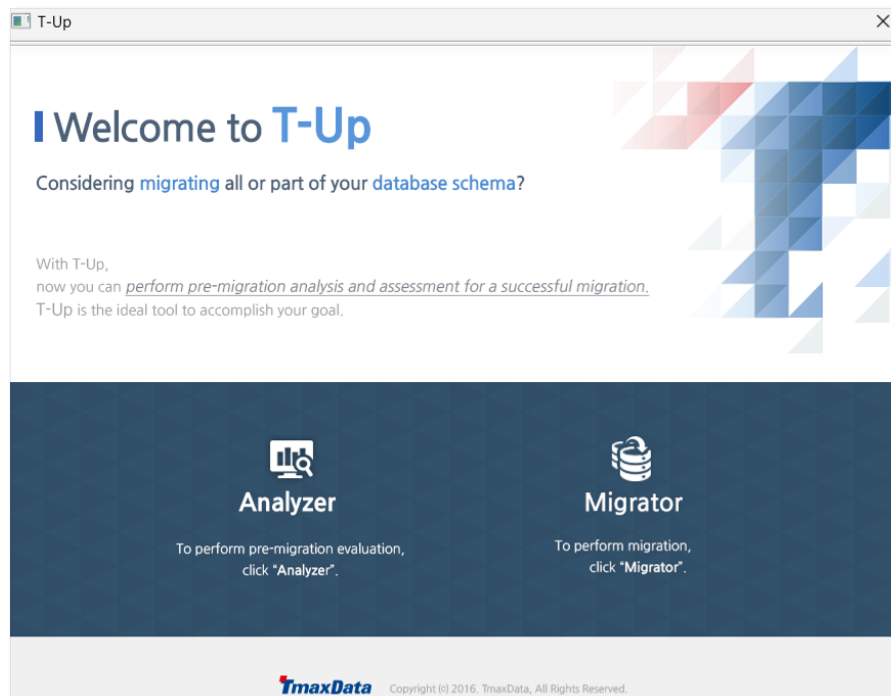
전환 검증 시간을 줄이기 위하여 전환을 시작하기 전 고객사의 DB 담당자는 Source DB의 INVALID한 Object를 정리하는 것이 좋다.

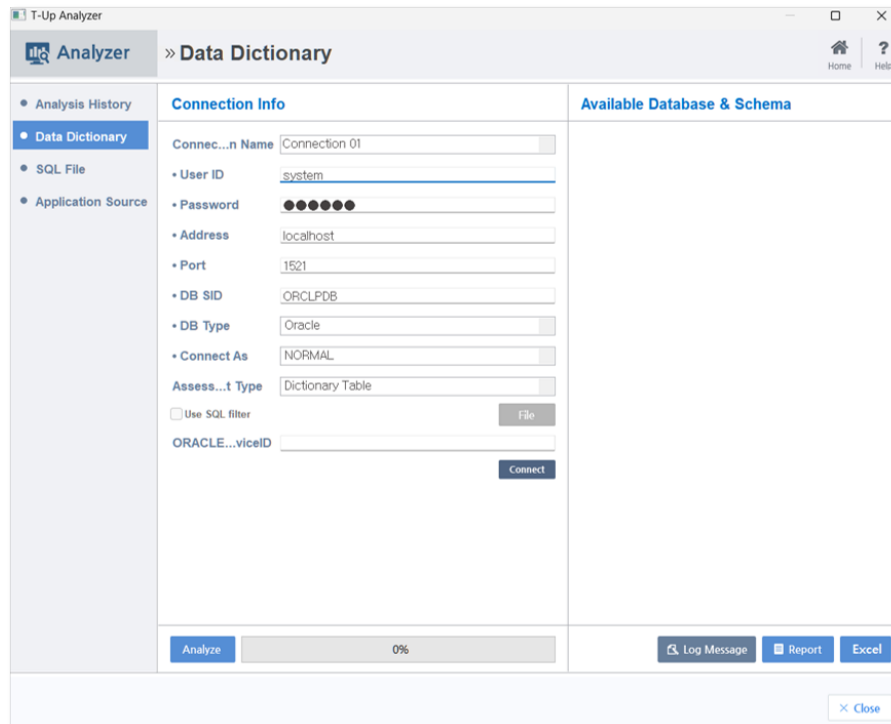
(Source DB에서 에러가 발생하면 Tibero에서도 에러가 발생한다.)

전환 전 분석

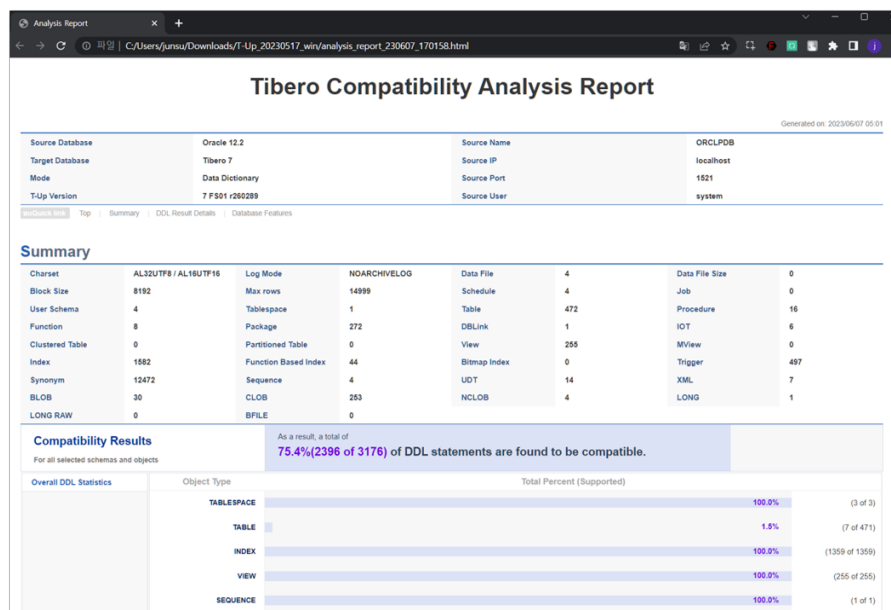
T-UP의 Analysis 기능을 이용해 전환 전 호환성 검증을 진행 한다.

[그림1.2] T-UP Analysis Oracle





[그림1.3] T-UP Analysis Report



Analysis Report

DDL Result Details

Compatibility Results for DDL statements

Unsupported Analyzed Objects

TABLE	Source	Line	Col	Result	Error Message
FUNCTION	Dictionary Table	1	48	Unsupported	syntax error, unexpected ID, expecting K_CMLTYPE or K_LOB or K_NESTED or K_XMLTYPE
PROCEDURE	Assessed Statement				
PACKAGE	CREATE TABLE "APEX_050000"."WWW_FLOW_CHARSETS" SHARING=METADATA				
PACKAGE BODY	("ID" NUMBER, "DISPLAY_NAME" VARCHAR2(4000) NOT NULL ENABLE, "IANA_CHARSET" VARCHAR2(255) NOT NULL ENABLE, "DB_CHARSET" VARCHAR2(255) NOT NULL ENABLE, CONSTRAINT "WWW_FLOW_CHARSETS_PK" PRIMARY KEY ("ID") USING INDEX PCTFREE 10 INTRANS 2 MAXTRANS 255 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT) TABLESPACE "SYSAUX" ENABLE) PCTFREE 10 PCTUSED 40 INTRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT) TABLESPACE "SYSAUX" 				
TYPE BODY	Dictionary Table	1	53	Unsupported	syntax error, unexpected ID, expecting K_CMLTYPE or K_LOB or K_NESTED or K_XMLTYPE
	Assessed Statement				
	CREATE TABLE "APEX_050000"."WWW_FLOW_RANDOM_IMAGES" SHARING=METADATA				
	("ID" NUMBER, "IMAGE_NAME" VARCHAR2(4000) NOT NULL ENABLE, "IMAGE_CODE" VARCHAR2(10) NOT NULL ENABLE, "BLOB_CONTENT" BLOB NOT NULL ENABLE, CONSTRAINT "WWW_FLOW_RANDOM_IMAGES_PK" PRIMARY KEY ("ID") USING INDEX PCTFREE 10 INTRANS 2 MAXTRANS 255 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 				

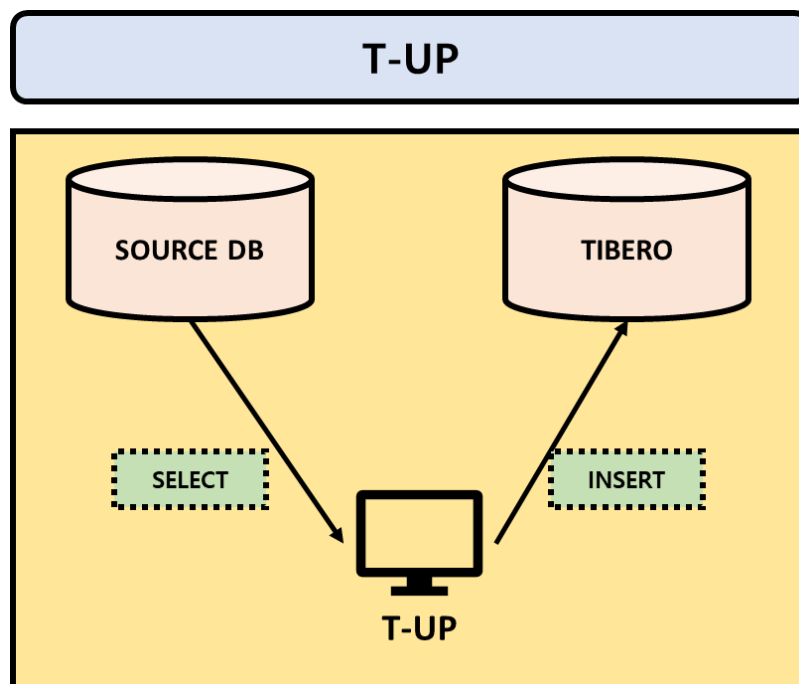
1.2 전환 방법

본 절에서는 DB를 전환할 때 사용하는 방법들에 대해서 설명한다.

T-UP

다음은 T-UP의 데이터 전환 과정을 보여주는 그림이다.

[그림2.1 T-UP 마이그레이션 예시]



- 특징

- 내부적으로 JAVA의 JDBC Driver를 이용하여 Source DB와 Tiberio에 접속한다. 단, Source DB와 Tiberio JDBC Driver가 필요하고 JDK 또는 JRE가 미리 설치되어 있어야 한다.
- Source DB가 이기종 RDBMS라면 Source DB의 JDBC Driver도 필요하다.
- Source DB > T-UP > Tiberio 방향으로 네트워크를 총 2번 거쳐 데이터가 이동한다. T-UP을 기점으로 Source DB와 Tiberio의 네트워크를 거쳐 데이터 이동을 하므로 대용량으로 이관할 경우 불리할 수 있다.
- 전체, 사용자, 테이블 모두 가능하고 GUI 형태로 보여주므로 조작하기 쉽다.
- 중간에 문제가 발생해도 멈추지 않으며 전환이 모두 완료된 후에 문제를 처리한다.

- 사전 작업 (선택사항)

- Tablespace 내의 데이터 파일 경로 및 이름을 변경하려면 생성문 추출 후 미리 수동으로 생성한다.
- Schema(=사용자)에 대해 Default Tablespace를 변경하려면 생성문 추출 후 미리 수동으로 생성한다.

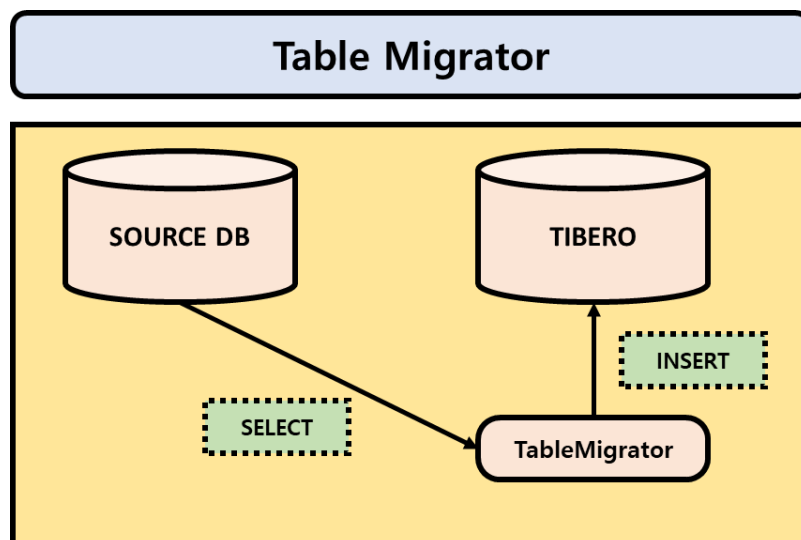
참고

- T-UP 사용법은 “Tiberio 유틸리티 안내서”의 “T-UP” 을 참고한다.

Table Migrator

다음은 Table Migrator의 데이터 전환 과정을 보여주는 그림이다.

[그림 2.2 Table Migrator 과정]



- 특징

- Command 기반으로 내부적으로 JAVA의 JDBC Driver를 이용하여 Source DB와 Tiberio에 접속한다. 단, Source DB와 Tiberio JDBC Driver가 필요하고 JDK 또는 JRE가 미리 설치되어 있어야 한다.
- 별도의 컴퓨터(클라이언트)가 필요하지 않고 Source DB 서버 또는 Tiberio 서버에서 기동한다.

- 중간 서버가 필요하지 않으므로 네트워크를 1번 거쳐서 데이터가 이동한다.
- 1개의 테이블 데이터만 이관 가능하고 대용량 테이블을 이관할 경우 유리하다.
- 조건절 또는 파티션 테이블 사용이 가능하다.

- **사전 작업 (필수사항)**

- Table Migrator를 사용하여 DB를 전환하기 전에 이관할 Table을 미리 생성해야 한다.

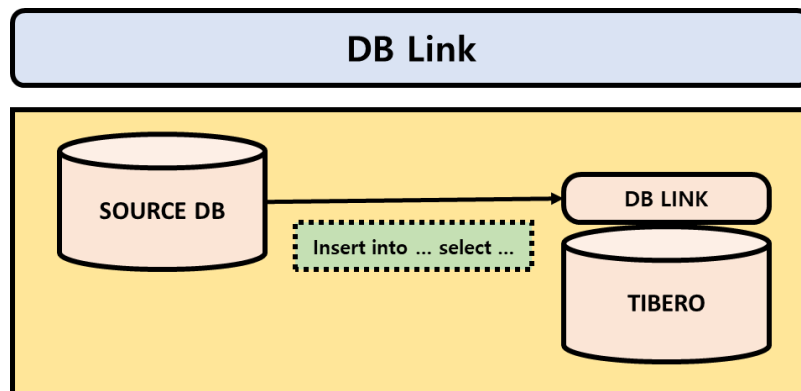
참고

- Table Migrator 가이드가 필요할 경우 (주)TmaxTibero 기술지원팀에 요청한다.

DB Link

다음은 DB Link의 데이터 전환 과정을 보여주는 그림이다.

[그림 2.3 DB Link 과정]



- **특징**

- Source DB 또는 Tibero에 DB Link Object를 생성한다.
- 중간 서버가 필요하지 않으므로 네트워크를 1번 거쳐서 데이터가 이동한다.
- 1개의 테이블 데이터만 이관 가능하고 대용량 테이블을 이관할 경우 유리하다.

- **사전 작업 (필수사항)**

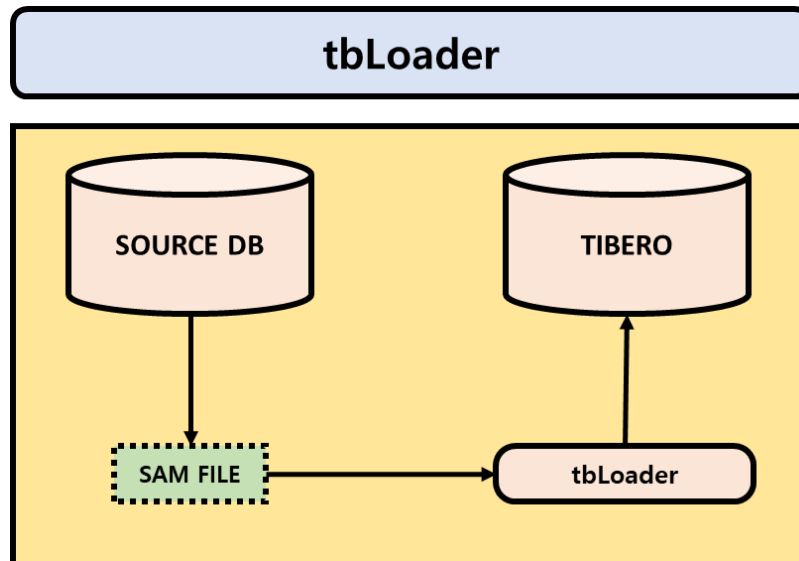
- DB Link를 사용하여 DB를 전환하기 전에 DB Link 구성 작업을 진행해야 한다. 만약, Source DB가 Oracle일 경우 Tibero To Oracle 구성을 하는 것이 제약사항이 적다.

참고

- DB Link 구성 방법은 제공되는 기술문서 중 "DB Link 가이드"를 참고한다.

다음은 tbLoader의 데이터 전환 과정을 보여주는 그림이다.

[그림 2.4 tbLoader 과정]



- 특징

- 데이터만 SAM 파일로 만들어서 한 번에 적재할 수 있다.
- 대용량 데이터를 이관할 경우 유리하다.

- 사전작업 (필수사항)

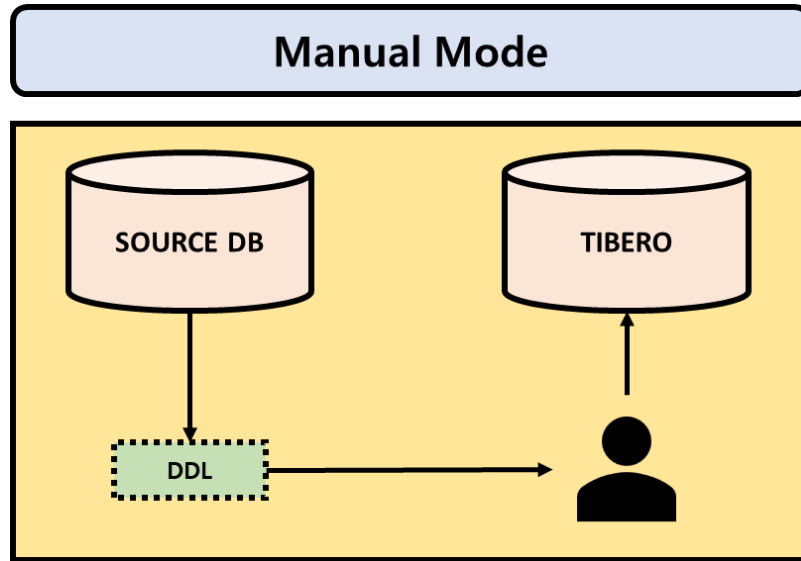
- tbLoader를 사용하여 DB를 전환하기 전에 Source DB에서 미리 텍스트 파일(SAM 파일)을 만들어서 Tiberο 서버로 복사해야 한다.

참고

- tbLoader 사용법은 "Tiberο 유틸리티 안내서"의 "제5장 tbLoader"를 참고한다.

다음은 수동이관으로 데이터를 전환하는 과정을 보여주는 그림이다.

[그림 2.5 수동이관 과정]



- 특징

- Source DB에 클라이언트 Tool을 이용하여 스크립트를 추출한 후 Tibero에 수동으로 적용하는 방식이다.
- 데이터의 경우 수동으로 옮기기가 불가능하므로, T-UP과 같은 Tool을 이용하여 이관한다.
- 중간에 문제가 발생하면 바로 문제 처리를 한 후 다음을 진행할 수 있다.
- Shell 스크립트와 연동하면 자동화할 수 있다.

- 사전작업(필수사항)

- 수동이관을 사용하기 전에 클라이언트 Tool을 이용하여 미리 DDL 생성문을 추출해야 한다.

참고

- 수동이관 사용법은 본 가이드 문서를 참고한다.

1.3. 전환 방법 적용

다음은 전환 방법 적용할 때 고려사항이다.

- 전환을 수행할 때 한가지 방법으로 수행하는 것이 아니라 여러 가지 방법을 혼합하여 사용할 수 있다.
- 고객사에서 실제 운영을 전환할 때 최소한의 Down Time이 필요하다. (해당 고객사에서 각 전환 방법의 테스트 및 시간 측정을 통해 최소한의 전환 시간을 확보해야한다.)

각 전환 방법의 적용 시기는 다음과 같다.

전환 방법	적용 시기
T-UP	GUI 기반으로 간편하게 전환을 할 경우(전체, 사용자 모드)와 수동 이관할 때 데이터 이관 용도로 사용한다.

전환 방법	적용 시기
Table Migrator	DB Link 구성이 어렵고 대용량 테이블 데이터를 이관할 때 사용한다.
DB Link	Table Migrator 사용이 어렵고 대용량 테이블 데이터를 이관할 때 사용한다.
tbLoader	Source DB에서 SAM 파일 생성이 가능할 경우 사용한다.
수동 이관	전환 난이도가 어려워 다른 전환 방법을 적용하기가 힘든 경우와 동일한 전환을 특정 기간에 걸쳐 반복적으로 수행해야 할 경우 사용한다. (자동화 스크립트를 이용하면 2차 이상의 반복적인 수행일 경우에 간편하다.)

제 2장 Oracle 전환

본 장에서는 Oracle에서 Tibero DB로 전환하는 절차 및 방법을 설명한다.

2.1. 사전 점검

본 절에서는 Oracle에서 Tibero DB로 전환하기 전 점검사항에 대해서 설명한다. 사전 점검은 전환하기 전 Oracle의 특성을 파악하고 Tibero에 적용해야 하는 값들을 확인하는 중요한 절차이다.

! 주의

- 본 절에서 설명하는 SQL문(10g, 11g에서 수행 가능한 쿼리)은 Oracle 버전에 따라서 일부 달라질 수 있음을 유의한다.

2.1.1. 기본 항목

Oracle의 sys 또는 system 등의 DBA 권한을 가진 사용자로 접속하여 아래 정보를 확인한다.

항목	설명
버전 정보	Oracle 9i, 10g, 11g는 전환 사례가 많으므로 그 이외 버전에서 전환할 경우에 대해서는 검토가 필요하다.
전환 문자 집합	Oracle에서 사용 중인 문자 집합에 대응되는 Tibero 문자 집합으로 선택한다.
전환 사용자 선택	Oracle을 설치할 때 자동 생성되는 기본 사용자의 경우는 제외한다. 전환 사용자에 대해서 고객사 담당자 확인이 필요하다.
전환 크기	전환 시간에 가장 큰 영향을 미치는 요소이다. 그 외 DISK I/O, NETWORK I/O 등에 따라서 고객사의 전환 시간이 많이 달라질 수 있다.
전환 검증항목	전환이 완료된 후에 비교해 보는 정보이다.

버전 정보

- 전환 Oracle 버전을 확인한다.

```
select * from v$version;
```

전환 문자 집합

- Oracle과 Tiberog간 대응되는 문자 집합을 사용한다.

```
$ tbboot -C
```

Available character set list

Charset name	Equivalent Oracle Charset name
AR8ISO8859P6	AR8ISO8859P6
AR8MSWIN1256	AR8MSWIN1256
ASCII	US7ASCII
CL8ISO8859P5	CL8ISO8859P5
CL8KOI8R	CL8KOI8R
CL8MSWIN1251	CL8MSWIN1251
EE8ISO8859P2	EE8ISO8859P2
EL8ISO8859P7	EL8ISO8859P7
EL8MSWIN1253	EL8MSWIN1253
EUCKR	KO16KSC5601
EUCTW	ZHT32EUC
GB18030	GB18030
GBK	ZHS16GBK
IW8ISO8859P8	IW8ISO8859P8
JA16EUC	JA16EUC
JA16EUCTILDE	JA16EUCTILDE
JA16SJIS	
JA16SJISTILDE	
MSWIN949	KO16MSWIN949
RU8PC866	RU8PC866
SJIS	JA16SJIS
SJISTILDE	JA16SJISTILDE
TH8TISASCII	TH8TISASCII
UTF16	AL16UTF16
UTF8	UTF8
VN8VN3	VN8VN3
WE8ISO8859P1	WE8ISO8859P1
WE8ISO8859P15	WE8ISO8859P15
WE8ISO8859P9	WE8ISO8859P9
WE8MSWIN1252	WE8MSWIN1252
ZHT16BIG5	ZHT16BIG5
ZHT16HKSCS	ZHT16HKSCS
ZHT16MSWIN950	ZHT16MSWIN950

Available nls_date_lang set list

```
AMERICAN
BRAZILIAN PORTUGUESE
JAPANESE
KOREAN
RUSSIAN
SIMPLIFIED CHINESE
THAI
TRADITIONAL CHINESE
```

전환 사용자(=Schema) 선택

Oracle의 'sys' 또는 'system' 계정 등의 DBA 권한이 있는 사용자로 접속한 후 조회한다.

- 사용자 조회

다음은 전체 사용자 목록이다.

```
select username, account_status, default_tablespace, temporary_tablespace
from dba_users order by oracle;
```

- Open된 사용자 조회

다음은 현재 Open 사용자를 조회하는 쿼리이다.

```
set linesize 120
set pagesize 100
select username, default_tablespace, temporary_tablespace
from dba_users where account_status='OPEN' order by username;
```

- Oracle 기본 사용자

아래와 같은 사용자 이름의 경우 Oracle을 설치할 때 자동으로 생성되는 사용자로 전환할 때 제외한다.

(Oracle 버전에 따라 사용자명이 일부 다를 수 있다.)

User Name	Description
ANONYMOUS	Enables HTTP access to Oracle XML DB.
APEX_050100	The account that owns the Oracle Application Express schema and metadata.
APEX_PUBLIC_USER	The minimally privileged account used for Oracle Application Express configuration with Oracle Application Express Listener or Oracle HTTP Server and mod_plsql.
APPQOSSYS	Used for storing and managing all data and metadata required by Oracle Quality of Service Management.
AUDSYS	The account where the unified audit data trail resides.
AURORA\$ORB\$UNAUTHENTICATED	Used for users who do not authenticate in Aurora/ORB
AWR_STAGE	Used to load data into the AWR from a dump file
CSMIG	User for Database Character Set Migration Utility

User Name	Description
CTXSYS	The Oracle Text account.
DBSFUSER	The account used to run the DBMS_SFW_ACL_ADMIN package.
DBSNMP	The account used by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database.
DEMO	User for Oracle Data Browser Demonstration (last version 9.2)
DIP	The account used by the Directory Integration Platform (DIP) to synchronize the changes in Oracle Internet Directory with the applications in the database.
DMSYS	Data Mining user
DSSYS	Oracle Dynamic Services and Syndication Server
DVF	The account owned by Database Vault that contains public functions to retrieve the Database Vault Factor values.
DVSYS	There are two roles associated with this account. The Database Vault owner role manages the Database Vault roles and configurations. The Database Vault Account Manager is used to manage database user accounts. Note: Part of Oracle Database Vault user interface text is stored in database tables in the DVSYS schema. By default, only the English language is loaded into these tables. You can use the DVSYS.DBMS_MACADM.ADD-NLS_DATA procedure to add other languages to Oracle Database Vault.
EXFSYS	User to hold the dictionary, APIs for the Expression Filter
FLows_FILES	The account owns the Oracle Application Express uploaded files.
GGSYS	The internal account used by Oracle GoldenGate. It should not be unlocked or used for a database login.
GSMADMIN_INTERNAL	The internal account that owns the Global Data Services schema. It should not be unlocked or used for a database login.
GSMCATUSER	The account used by Global Service Manager to connect to the Global Data Services catalog.
GSMUSER	The account used by Global Service Manager to connect to the database.
HR (Human Resources)	
OE (Order Entry)	
SH (Sales History)	Training/ demonstration users containing the popular EMPLOYEES and DEPARTMENTS tables
LBACSYS	The Oracle Label Security administrator account. Starting with Oracle Database 18c, the LBACSYS user account is created as a schema-only account.
MDDATA	The schema used by Oracle Spatial and Graph for storing geocoder and router data.
MDSYS	The Oracle Spatial and Graph administrator account.
ORACLE_OCM	This account contains the instrumentation for configuration collection used by the Oracle Configuration Manager.
ORDPLUGINS	Object Relational Data (ORD) User used by Time Series, etc.
ORDSYS	Object Relational Data (ORD) User used by Time Series, etc.

User Name	Description
OUTLN	The account that supports plan stability. Plan stability enables you to maintain the same execution plans for the same SQL statements. OUTLN acts as a role to centrally manage metadata associated with stored outlines.
PERFSTAT	Oracle Statistics Package (STATSPACK) that supersedes UTLBSTAT/UTLESTAT
REMOTE_SCHEDULER_AGENT	The account to disable remote jobs on a database. This account is created during the remote scheduler agent configuration. You can disable the capability of a database to run remote jobs by dropping this user.
SCOTT	
ADAMS	
JONES	
CLARK	
BLAKE	Training/ demonstration users containing the popular EMP and DEPT tables
SYS	The account used to perform database administration tasks.
SYS\$UMF	The account used to administer Remote Management Framework, including the remote Automatic Workload Repository (AWR).
SYSBACKUP	The account used to perform backup and recovery tasks.
SYSDG	The account used to administer and monitor Oracle Data Guard.
SYSKM	The account used to perform encryption key management.
SYSRAC	The account used to administer Oracle Real Application Clusters (RAC).
SYSTEM	Another account used to perform database administration tasks.
TRACESVR	Oracle Trace server
TSMSYS	User for Transparent Session Migration (TSM) a Grid feature
WMSYS	The account used to store the metadata information for Oracle Workspace Manager.
XDB	The account used for storing Oracle XML DB data and metadata.
XS\$NULL	The internal account that represents the absence of a database schema user in a session, and indicates an application user session is in use. XS\$NULL cannot be authenticated to a database, nor can it own any database schema objects, or possess any database privileges.

💡 출처

[Database Installation Guide](#)

전환 크기

전환 대상 사용자에게 속하는 Table, Index가 저장된 Tablespace만 확인한다.

아래 예제는 Tablespace 전체에 대하여 확인하는 예제이다.

- 전체 Tablespace 크기와 사용량 확인

```
select x.a tablespace_name
      , sum(x.b)/1024 "tot_size(mb)"
      , sum(x.c)/1024 "used_size(mb)"
      , sum(x.c)/sum(x.b)*100 rate
from (
    select
        b.tablespace_name a
        ,sum(bytes)/1024 b
        ,0 c
    from dba_data_files b
    group by b.tablespace_name
    union
    select
        d.tablespace_name,
        0,
        sum(bytes)/1024
    from dba_segments d
    group by d.tablespace_name
)x
group by x.a;
```

- 전체 합계

```
select sum(x.b)/1024 "tot(mb)", sum(x.c)/1024 "used(mb)"
from (
    select sum(bytes)/1024 b, 0 c
    from dba_data_files b
    union
    select 0, sum(bytes)/1024
    from dba_segments d
) x;
```

전환 검증 항목

전환 대상 사용자에게 대해당하는 정보만 확인한다.

- Object 상태별 개수 확인

```
select owner, object_type, status, count(*)
from dba_objects
where owner in (select username from dba_users where account_status='OPEN' )
group by owner, object_type, status
order by owner, object_type, status;
```

참고

- 실제 dba_object에는 삭제된 Object까지 포함(\$BIN~ 형식)되어 있어 정확한 Count를 위해서는 dba_tables, dba_index 같은 정보를 보는 것이 좋다.
- INVALID한 Object에 대해서 고객센터 DB 담당자에게 정리를 요청하는 것이 전환시간을 단축할 수 있다.

• Object 개수 확인

전환 완료 후 Oracle과 Object 개수를 비교할 때 검증용으로 사용한다.

```
WITH mig_user AS
(
    SELECT username
    FROM dba_users
    WHERE account_status='OPEN'
    AND username NOT IN('SYS', 'SYSTEM', 'WMSYS', 'EXFSYS', 'XDB',
        'ORDSYS', 'MDSYS', 'SYSMAN')
)
SELECT
    O.OWNER
    ,NVL(TBL.TABLE_CNT, 0) AS "TABLE_CNT"
    ,O.VIEW_CNT
    ,O.DBLINK_CNT
    ,O.SEQ_CNT
    ,O.PKG_CNT
    ,O.PKGBODY_CNT
    ,O.PROC_CNT
    ,O.FUNC_CNT
    ,O.TYPE_CNT
    ,O.TYPEBODY_CNT
    ,O.LIB_CNT
    ,TRI.TRIGGER_CNT
    ,NVL(SYN.SYN_CNT,0) AS "SYN_CNT"
    ,NVL(PUB_SYN.PUB_SYN_CNT,0) AS "PUB_SYN_CNT"
    ,NVL(IDX.IDX_CNT, 0) AS "IDX_CNT"
    ,NVL(G.GRANT_CNT, 0) AS "GRANT_CNT"
    ,NVL(LOB.LOB_CNT, 0) AS "LOB_CNT"
    ,NVL(JOB.JOB_CNT, 0) AS "JOB_CNT"
FROM (
    SELECT
        T.OWNER
        ,SUM( DBLINK_CNT ) AS "DBLINK_CNT"
        ,SUM( VIEW_CNT ) AS "VIEW_CNT"
        ,SUM( SEQ_CNT ) AS "SEQ_CNT"
        ,SUM( PKG_CNT ) AS "PKG_CNT"
```

```

        ,SUM( PKGBODY_CNT ) AS "PKGBODY_CNT"
        ,SUM( PROC_CNT ) AS "PROC_CNT"
        ,SUM( FUNC_CNT ) AS "FUNC_CNT"
        ,SUM( TYPE_CNT ) AS "TYPE_CNT"
        ,SUM( TYPEBODY_CNT ) AS "TYPEBODY_CNT"
        ,SUM( LIB_CNT ) AS "LIB_CNT"
FROM (
        SELECT
                OWNER
                ,CASE WHEN OBJECT_TYPE = 'DATABASE LINK' THEN COUNT(1) ELSE 0 END
AS "DBLINK_CNT"
                ,CASE WHEN OBJECT_TYPE = 'VIEW' THEN COUNT(1) ELSE 0 END AS
"VIEW_CNT"
                ,CASE WHEN OBJECT_TYPE = 'SEQUENCE' THEN COUNT(1) ELSE 0 END AS
"SEQ_CNT"
                ,CASE WHEN OBJECT_TYPE = 'PACKAGE' THEN COUNT(1) ELSE 0 END AS
"PKG_CNT"
                ,CASE WHEN OBJECT_TYPE = 'PACKAGE BODY' THEN COUNT(1) ELSE 0 END
AS "PKGBODY_CNT"
                ,CASE WHEN OBJECT_TYPE = 'PROCEDURE' THEN COUNT(1) ELSE 0 END AS
"PROC_CNT"
                ,CASE WHEN OBJECT_TYPE = 'FUNCTION' THEN COUNT(1) ELSE 0 END AS
"FUNC_CNT"
                ,CASE WHEN OBJECT_TYPE = 'TYPE' THEN COUNT(1) ELSE 0 END AS
"TYPE_CNT"
                ,CASE WHEN OBJECT_TYPE = 'TYPE BODY' THEN COUNT(1) ELSE 0 END AS
"TYPEBODY_CNT"
                ,CASE WHEN OBJECT_TYPE = 'LIBRARY' THEN COUNT(1) ELSE 0 END AS
"LIB_CNT"
        FROM dba_objects
        WHERE owner in (select username from mig_user)
        GROUP BY owner, OBJECT_TYPE
) T
GROUP BY T.OWNER
) O
LEFT JOIN (
        SELECT OWNER, COUNT(1) AS TABLE_CNT
        FROM DBA_TABLES
        WHERE owner in (select username from mig_user)
        GROUP BY OWNER
) TBL ON O.OWNER = TBL.OWNER
LEFT JOIN (
        SELECT OWNER, COUNT(1) AS GRANT_CNT
        FROM DBA_TAB_PRIVS
        WHERE owner in (select username from mig_user)
        AND TABLE_NAME NOT LIKE 'BIN$%'
        GROUP BY OWNER
) G ON O.OWNER = G.OWNER
LEFT JOIN (
        SELECT OWNER, COUNT(1) AS IDX_CNT
        FROM DBA_INDEXES
        WHERE owner in (select username from mig_user)
        GROUP BY OWNER

```

```

) IDX ON O.OWNER = IDX.OWNER
LEFT JOIN (
    SELECT OWNER, COUNT(1) AS LOB_CNT
    FROM DBA_LOBS
    WHERE owner in (select username from mig_user)
    GROUP BY OWNER
) LOB ON O.OWNER = LOB.OWNER
LEFT JOIN (
    SELECT SCHEMA_USER, COUNT(1) AS JOB_CNT
    FROM DBA_JOBS
    WHERE SCHEMA_USER in (select username from mig_user)
    GROUP BY SCHEMA_USER
) JOB ON O.OWNER = JOB.SCHEMA_USER
LEFT JOIN (
    SELECT OWNER, COUNT(1) AS TRIGGER_CNT
    FROM DBA_TRIGGERS
    WHERE OWNER in (select username from mig_user)
    AND TRIGGER_NAME NOT LIKE 'BIN$%'
    GROUP BY OWNER
) TRI ON O.OWNER = TRI.OWNER
LEFT JOIN (
    SELECT OWNER, COUNT(1) AS "SYN_CNT"
    FROM DBA_SYNONYMS
    WHERE OWNER in (select username from mig_user)
    AND OWNER != 'PUBLIC'
    GROUP BY OWNER
) SYN ON O.OWNER = SYN.OWNER
LEFT JOIN (
    SELECT TABLE_OWNER as "OWNER", COUNT(1) AS "PUB_SYN_CNT"
    FROM DBA_SYNONYMS
    WHERE TABLE_OWNER in (select username from mig_user)
    AND OWNER = 'PUBLIC'
    GROUP BY TABLE_OWNER
) PUB_SYN ON O.OWNER = PUB_SYN.OWNER
ORDER BY O.OWNER;

```

- Constraint 개수 확인

```

set linesize 150
col owner format a30

select owner
,constraint_type
,count( constraint_name ) as "con_cnt"
from dba_constraints
where owner in (select username from dba_users where account_status='OPEN' )
group by owner, constraint_type
order by owner ,constraint_type;

```

2.1.2. DB 생성 관련 항목

Oracle의 sys 또는 system 등의 DBA 권한을 가진 사용자로 접속하여 아래의 정보를 확인하고, Tibero 설치 과정 중 CREATE DATABASE를 수행할 경우 아래의 정보를 참고하여 DB를 생성한다.

다음은 DB 생성관련 항목에 대한 설명이다.

항목	설명
문자 집합	Oracle과 대응되는 문자 집합을 선택한다.
초기 파라미터	세션 개수 및 메모리 설정 등을 확인한다.
Redo 구성	Group과 Member의 개수 또는 크기 등을 확인한다.
Undo와 Temp 구성	Undo와 Temp Tablespace 크기를 확인한다.
Log 모드	Tibero 미지원 사항으로 일반 Index로 변경을 고려한다.

문자 집합

문자 집합 정보를 확인한다.

- Oracle

```
set linesize 150
set pagesize 100
column value$ format a100

select
    name,
    value$
from
    sys.props$
where
    name like '%CHARACTERSET'
order by
    name;
```

- Tibero

```
select * from _vt_nls_character_set;
```

확인 결과 항목에 대한 설명은 다음과 같다.

항목	설명
NLS_CHARACTERSET	DB 문자 집합
NLS_NCHAR_CHARACTERSET	National 문자 집합

다음은 Oracle과 Tiberio의 문자 집합 대응 정보이다.

Oracle	Tibero
US7ASCII	ASCII
KO16KSC5601	EUCKR or MSWIN949
KO16MSWIN949	MSWIN949
UTF8, AL32UTF8	UTF8
JA16SJIS[TILDE]	JA16SJIS[TILDE]
JA16EUC[TILDE]	JA16EUC[TILDE]
ZHS16GBK	GBK
VN8VN3	VN8VN3

참고

- US7ASCII 캐릭터 셋에 한글이 들어있는 경우 목적에 따라 MSWIN949로 할 수도 있다. 단, 애플리케이션의 인코딩과 디코딩 부분의 변경이 필요할 수 있다.
- KO16KSC5601, KO16MSWIN949, JA16SJIS에서 UTF8, AL32UTF8로 변경할 경우 해당 테이블 컬럼 크기를 1.5~2배 정도 늘린 후에 데이터 전환을 해야한다.

National 문자 집합

Oracle	Tibero
UTF8	UTF8
AL16UTF16(기본값)	UTF16(기본값)

참고

- Oracle의 National 캐릭터 셋이 UTF8이고 Tibero가 UTF16이라면 관련 타입의 글자가 깨져서 전환되므로 동일한 설정이 필요하다.

초기 파라미터

초기 파라미터 정보를 확인한다.

```
show parameter <파라미터명>
```

다음은 Oracle에서 전환할 때 확인해야 할 초기 파라미터이다.

메모리 관련 SGA 세부항목 크기에 대해 확인하려면 "V\$SGA"로 조회하여 확인한다.

파라미터	설명
SESSIONS	세션 수를 설정한다.
OPEN_CURSORS	세션당 열 수 있는 Cursor의 개수이다.
UNDO_RETENTION	Undo 보존 시간을 설정한다.
SGA_MAX_SIZE	사용 가능한 SGA 전체 크기를 확인한다.
SGA_TARGET	SGA의 목표 크기를 설정한다.
PGA_AGGREGATE_TARGET	사용 가능한 PGA 전체 크기를 확인한다.
MEMORY_TARGET	사용 가능한 전체 메모리 크기를 설정한다. (Oracle 11g용)

Redo 로그 구성

Tibero에서 Redo 로그를 생성할 경우 참고하며 파일 크기를 동일하거나 더 크게 생성해 준다.

다음은 Redo를 확인하는 예이다.

```
set linesize 150

select * from v$logfile;
select group#, bytes/1024/1024 as "size(mb)", members, archived from v$log;
```

Undo와 Temp 구성

Tibero에서 Undo와 Temp를 생성할 경우 참고하며 파일 크기를 동일하거나 더 크게 생성해준다.

다음은 Undo Tablespace를 확인하는 예이다.

```
show parameters undo_tablespace;
set linesize 150
col tablespace_name format a40
col file_name format a50
select tablespace_name, file_name, bytes/1024/1024 "size(mb)"
,maxbytes/1024/1024 "maxsize(mb)", autoextensible
from dba_data_files
where tablespace_name = 'UNDOTBS1' -- show parameters undo_tablespace;의 결과값 입력
```

다음은 Temp Tablespace 확인 예이다.

```
col file_name format a50
select tablespace_name, file_name, bytes/1024/1024 "size mb", autoextensible
from dba_temp_files;
```

Log 모드

다음은 Log 모드를 확인하는 예이다.

만약, ARCHIVE LOG 모드라면 전환 시간 단축을 위해 전환이 모두완료된 후에 모드를 변경한다.

```
select log_mode from v$database;
```

2.1.3. 추가 Object 확인

다음은 추가로 확인해야 할 Object에 대한 설명이다.

항목	설명
DB Link	전환 사용자의 DB Link 정보와 데이터가 존재하는 DB의 종류를 조사한다.
tnsnames.ora	DB Link의 경우 사용하는 Host 정보를 확인한다.
Job	전환 사용자 또는 전체를 수행하는 Job이 없는지 확인하고, 전환할 경우 수행되어 데이터가 변경되지 않는지 확인한다.
Cluster Object	Tibero 미지원 Object로 일반 Table 형태로 변경을 고려한다.
XML Type	Table 생성문 column부분 type의 sys.xmltype을 xmltype으로 변경이 필요하다.

항목	설명
Bit Map Index	Tibero 미지원 사항으로 일반 Index로 변경을 고려한다.

- **DB Link 확인**

```
set linesize 150
col owner format a15
col db_link format a20
col username format a20
col host format a50
select * from dba_db_links;
```

- **tnsnames.ora 파일 확인**

```
cat $ORACLE_HOME/network/admin/tnsnames.ora
```

- **Job 확인**

```
set linesize 200
select job, schema_user, broken, instance from dba_jobs;
set linesize 150
col interval format a50
select job, schema_user, next_date, broken, interval from dba_jobs;
```

- **Cluster Object 확인**

```
SELECT *
FROM DBA_OBJECTS
WHERE OBJECT_TYPE LIKE '%CLUSTER%'
AND OWNER NOT IN ('SYS');
```

- **XML Type 확인**

```
set linesize 200
SELECT owner, table_name, column_name, data_type
FROM DBA_TAB_COLS
WHERE DATA_TYPE LIKE '%XML%'
AND OWNER NOT IN ('SYS');
```

- **Bit Map Index 확인**

```
SELECT *
FROM DBA_INDEXES
WHERE INDEX_TYPE LIKE '%BITMAP%';
```

2.2. 전환 수행

본 절에서는 실제 전환을 수행하는 방법에 대해서 설명한다.

다음은 전환을 수행하는 과정이다.

- [1. 전환 사전 준비](#)
- [2. 전환 DDL 스크립트 추출](#)
- [3. 전환 DDL 스크립트 보정](#)
- [4. 전환](#)
- [5. 전환 검증](#)

각 과정에 대한 상세한 설명은 해당 절의 내용을 참고한다.

2.2.1. 전환 사전 준비

전환할 때 Object를 생성하는 기본순서는 아래와 같다. 4~8번의 경우는 고객사의 Object들의 의존성에 따라 변동성이 있으므로 고객사의 상황에 맞게 변경한다.

순 서	Object	내용
1	Tablespace	-
2	User	-
3	Table	-
4	Synonym	생성할 때 Source Object가 존재하지 않아도 문제되지 않지만 추후 조회할 때 자동으로 VALID하므로 먼저 만들어주는 것이 좋다.
5	View	의존성 때문에 에러가 발생할 수 있다.
6	Sequence	-
7	Data Migration	
8	psm(=pl/sql)	T-UP 또는 Table Migrator 등의 Tool을 사용하여 데이터를 전환한다.
9	Index	package, procedure, function이 해당하며 의존성 때문에 에러가 발생할 수 있다.
10	Constraint	-

순 서	Object	내용
11	Trigger	-
12	Grant	-
13	기타(C, Java external procedure, Job 등)	-

2.2.2. 전환 DDL 스크립트 추출

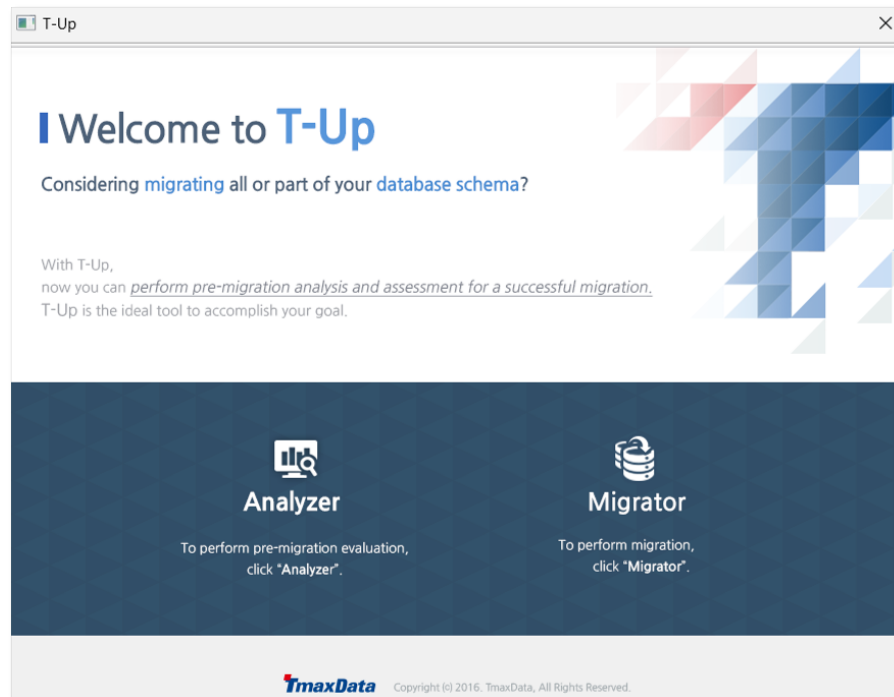
Oracle의 생성문(=DDL)을 추출하기 위하여 T-UP DDL export 기능을 사용 (스크립트를 추출할 수 있는 다른 Tool을 사용해도 무방)한다.

데이터베이스 주요 오브젝트 DDL 스크립트 추출

Tablespace, User, Table 생성문(=DDL)을 추출한다.

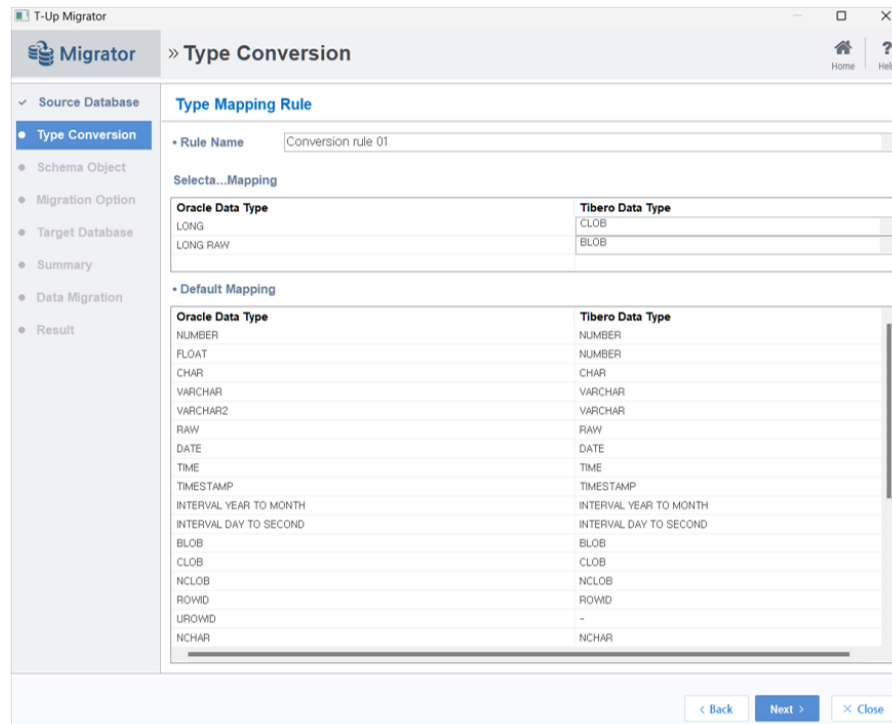
1. T-UP > Migrator

- “Migrator” 선택



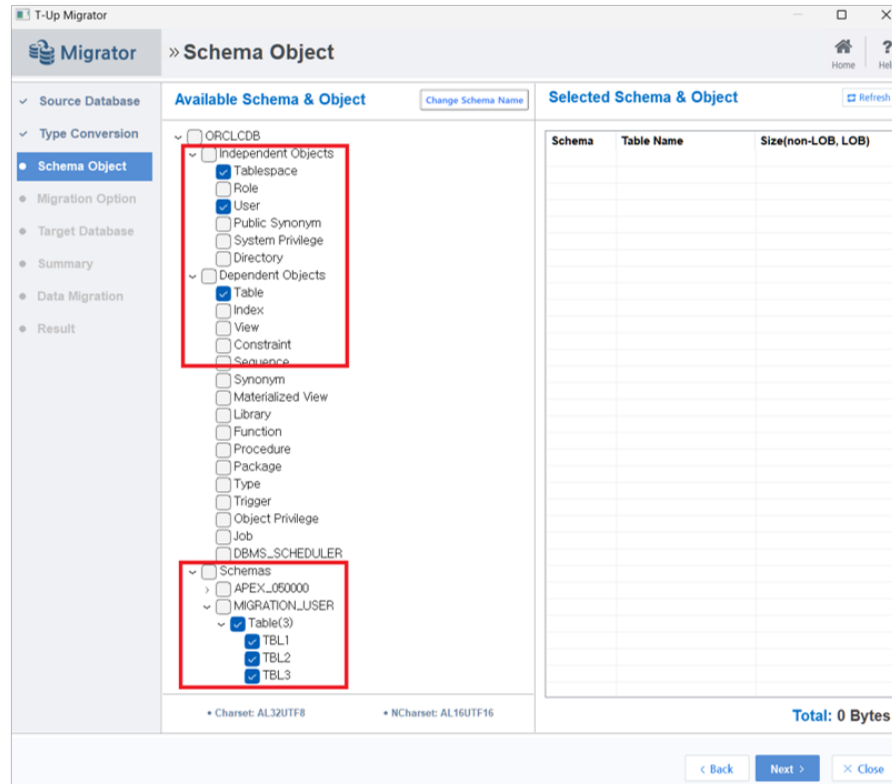
2. Source Database > Type Conversion

- ORACLE과 대응되는 데이터 타입 선택



3. Schema Object

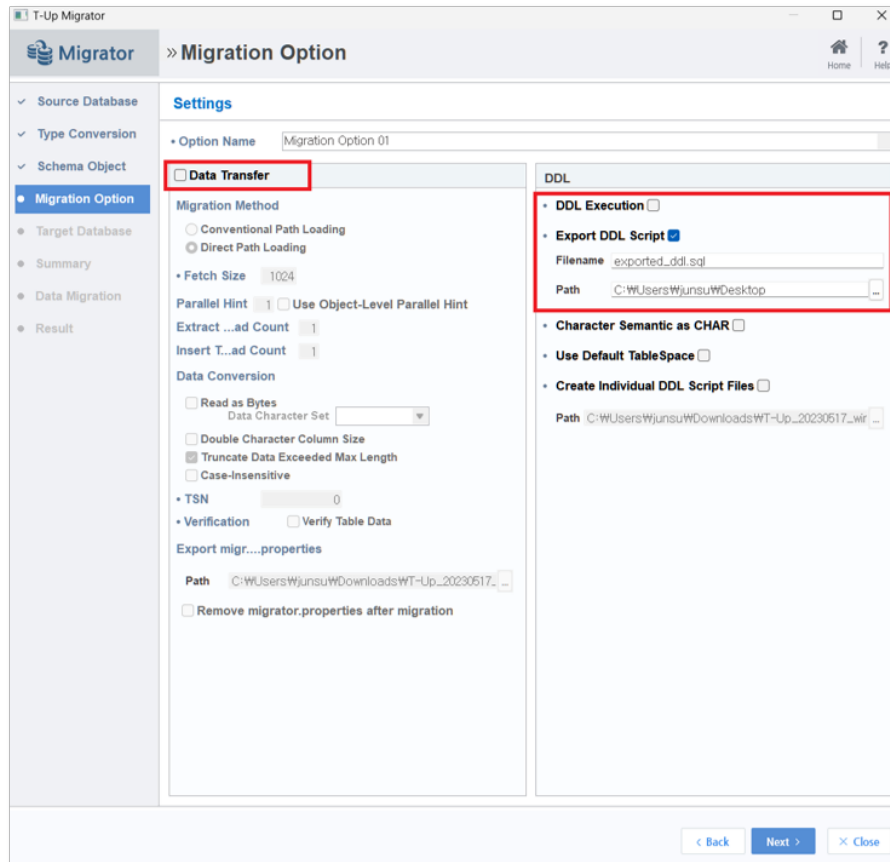
- "Independent Objects" Tablespace, User선택
- "Dependent Objects" Table 선택
- "Schema" Table 선택



4. Migration Option

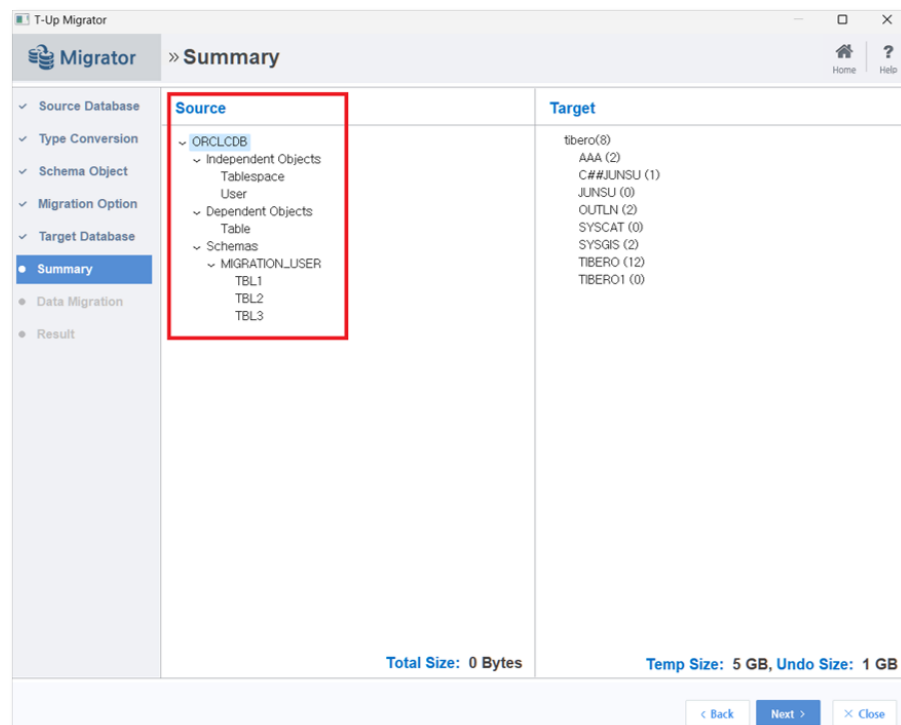
- "Data Transfer", "DDL Execution" 비활성

- “Export DDL Script” 선택
 - “Filename”과 “Path” 지정
- 스키마별로 파일을 분리하고 싶다면, “Create Individual DDL Script Files” 선택한다.



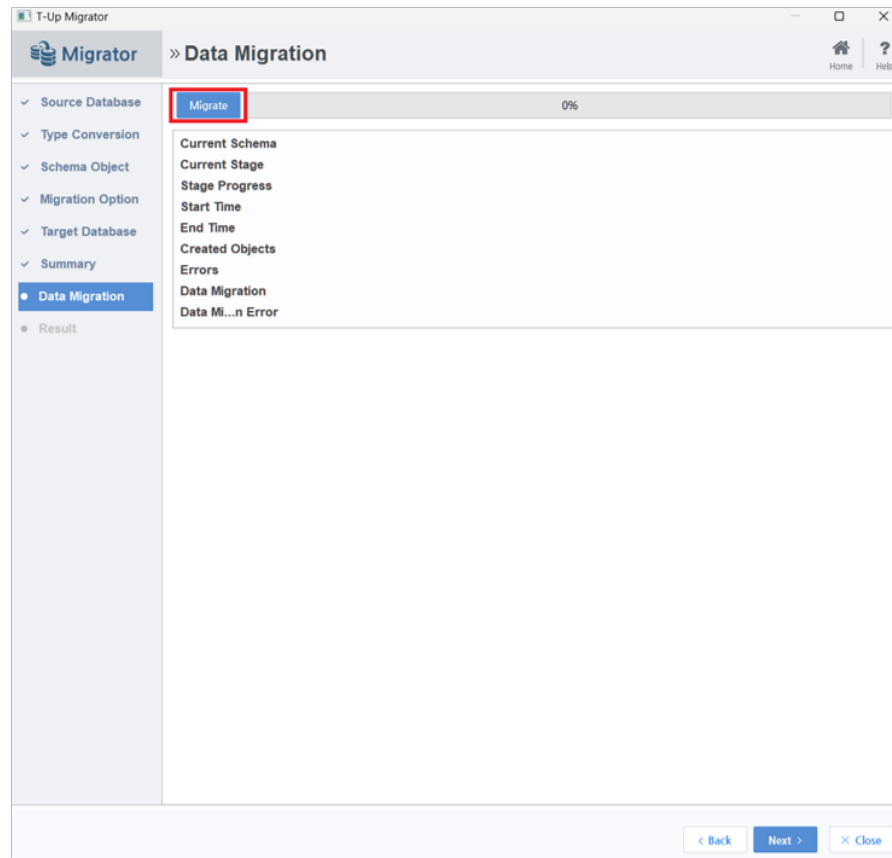
5. Target Database > Summary

- 추출 대상을 확인한다.



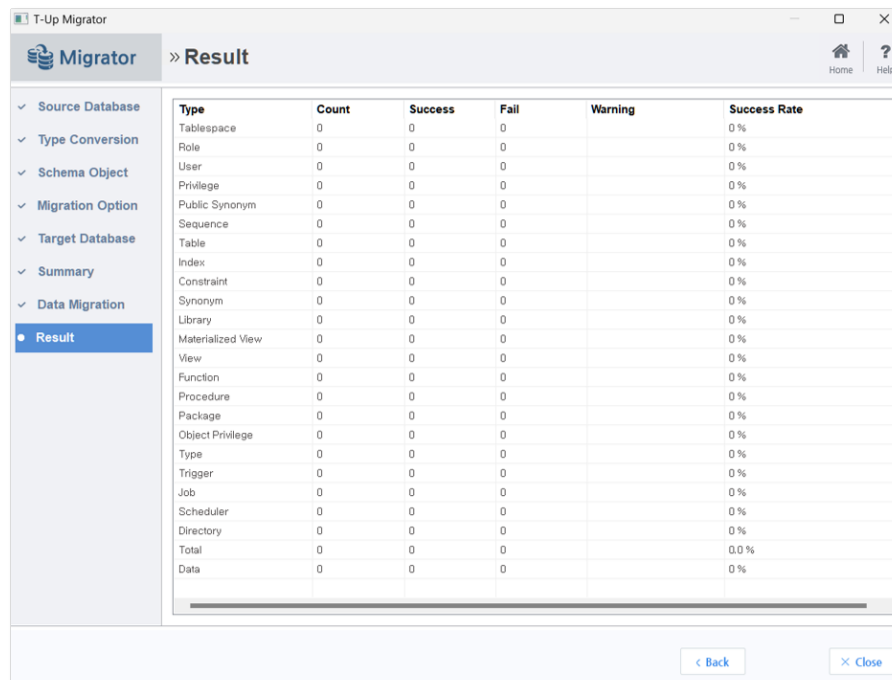
6. Data Migration

- “Migrate” 수행



7. Result > “close”

- “Result” 단계는 Execution 정보가 출력되며 DDL Export 정보는 출력되지 않음



8. 파일 확인

- DDL문이 정상적으로 출력됐는지 확인

```

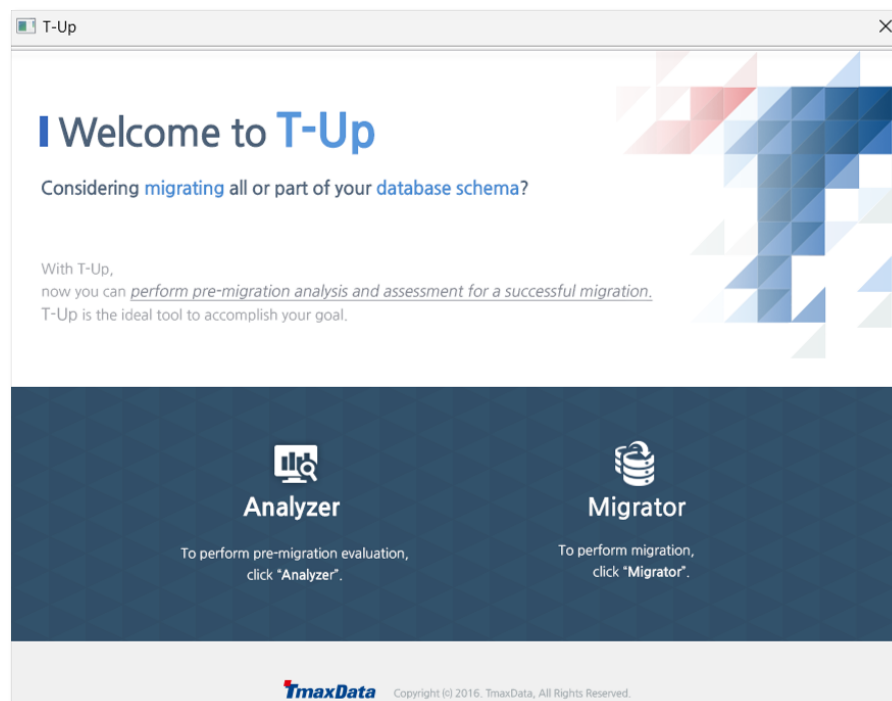
CREATE TABLESPACE "USERS" DATAFILE
  'USERS01.dtf' SIZE 6291456 AUTOEXTEND ON NEXT 1310720 MAXSIZE 34359721984,
  'USERS02.dtf' SIZE 1288490189
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
/
CREATE USER "MIGRATION_USER" IDENTIFIED BY tiberio
DEFAULT TABLESPACE "USERS"
/
ALTER SESSION SET CURRENT_SCHEMA = tiberio
/
ALTER SESSION SET CURRENT_SCHEMA = "MIGRATION_USER"
/
CREATE TABLE "MIGRATION_USER"."TBL1" (
  "COLUMN1" VARCHAR2(100 BYTE),
  "COLUMN2" VARCHAR2(100 BYTE),
  "COLUMN3" VARCHAR2(100 BYTE)
)
TABLESPACE "USERS"
PCTFREE 10
PCTUSED 0
INITRANS 1
MAXTRANS 255
LOGGING
  
```

데이터베이스 기타 오브젝트 DDL 스크립트 추출

Tablespace, User, Table을 제외한 나머지 Object(=DDL)를 추출한다.

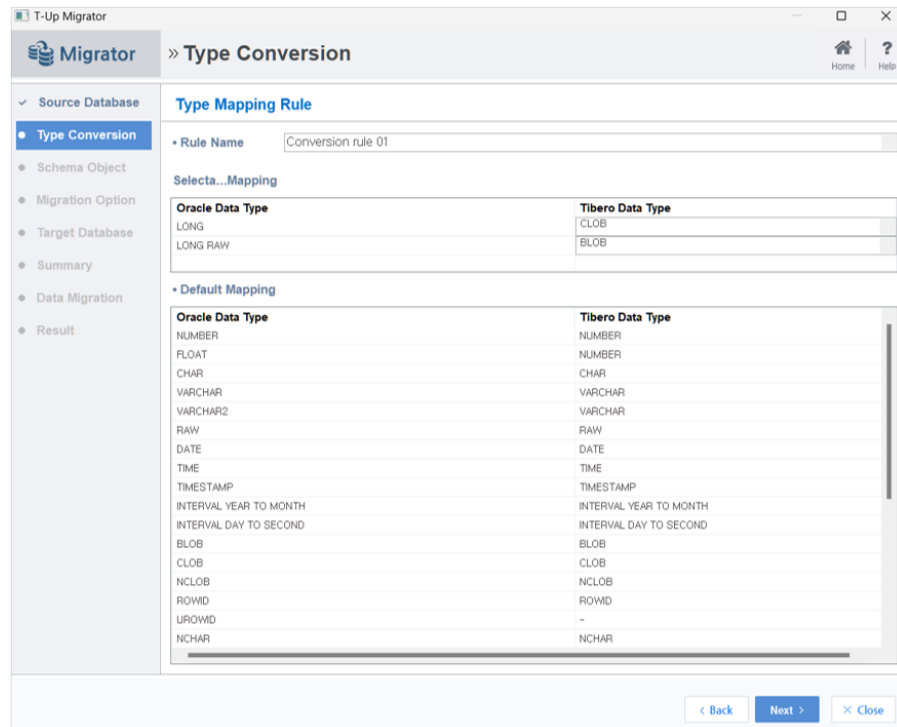
1. T-UP > Migrator

- “Migrator” 선택



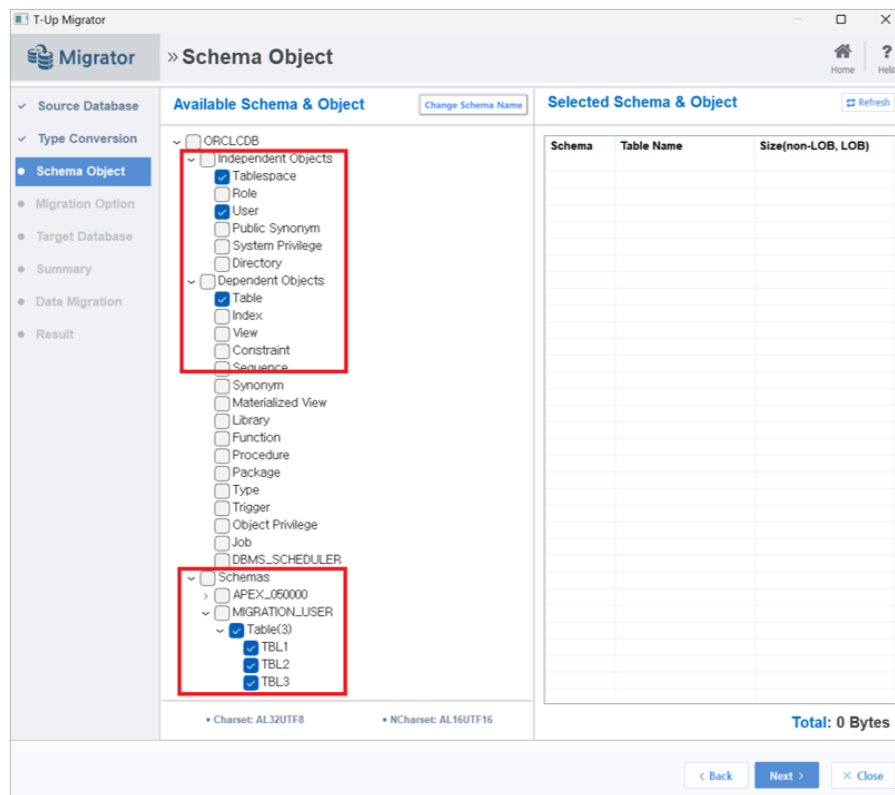
2. Source Database > Type Conversion

- ORACLE과 대응되는 데이터 타입 선택, Table 포함되지 않아 무시해도 된다.



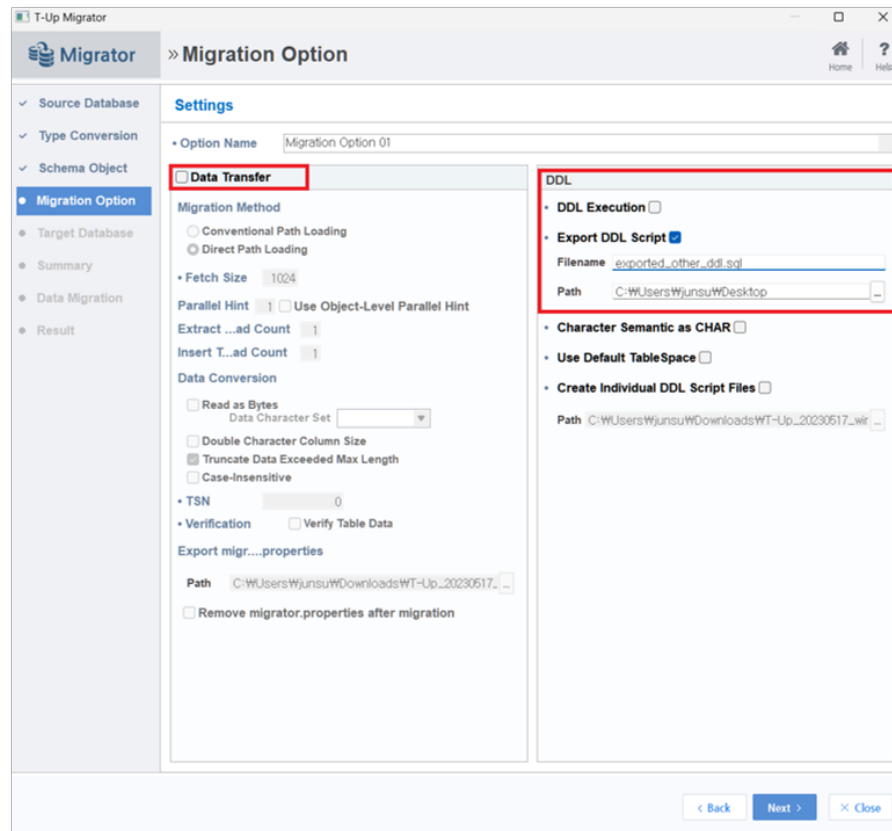
3. Schema Object

- “Dependent Objects” 추출 오브젝트 선택
- “Schema” Table 선택, 의존성을 위해 테이블을 선택



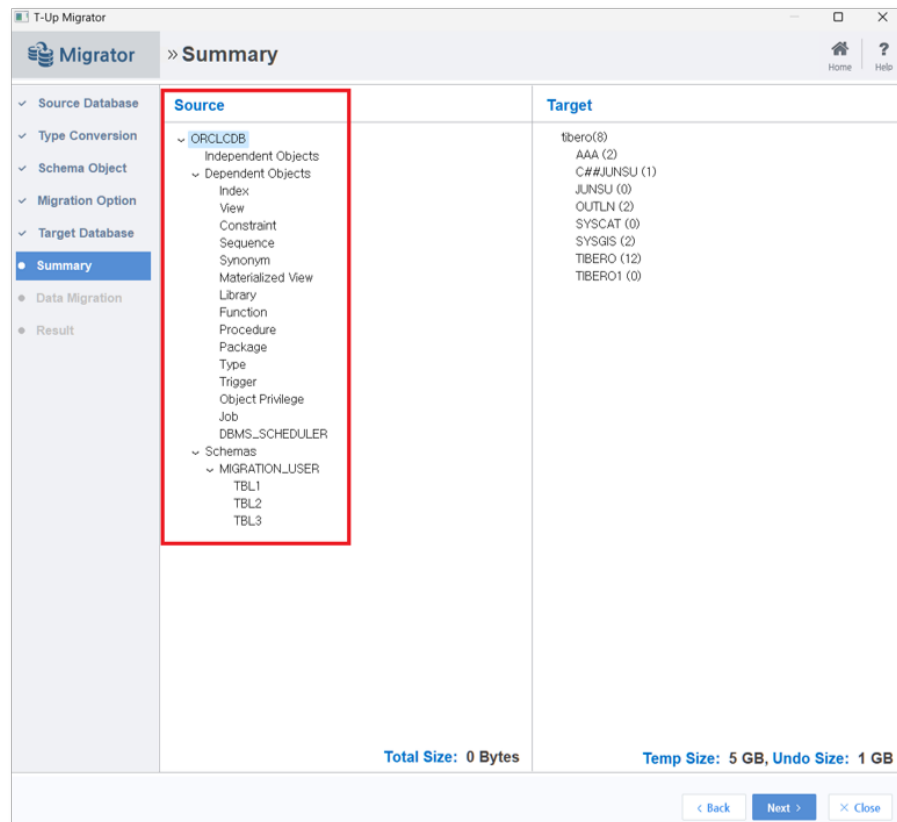
4. Migration Option

- "Data Transfer", "DDL Execution" 비활성
- "Export DDL Script" 선택
 - "Filename"과 "Path" 지정
- 스키마 및 오브젝트 별로 파일을 분리하고 싶다면, "Create Individual DDL Script Files" 선택한다.



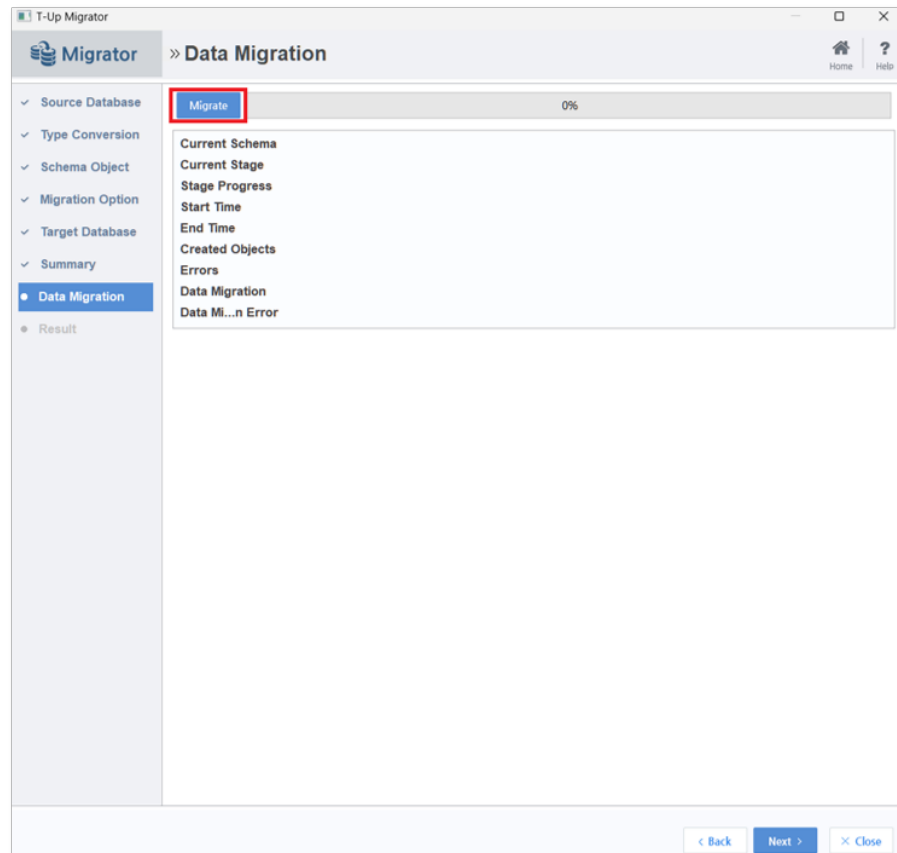
5. Target Database > Sumamry

- 추출 대상을 확인한다.



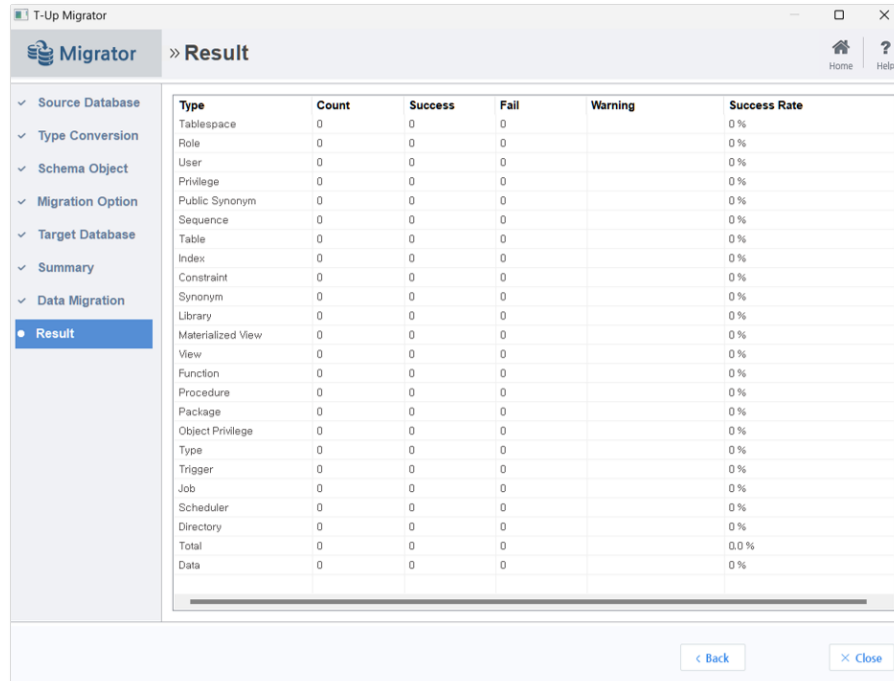
6. Data Migration

- “Migrate” 수행



7. Result > "close"

- "Result" 단계는 Execution 정보가 출력되며 DDL Export 정보는 출력되지 않음



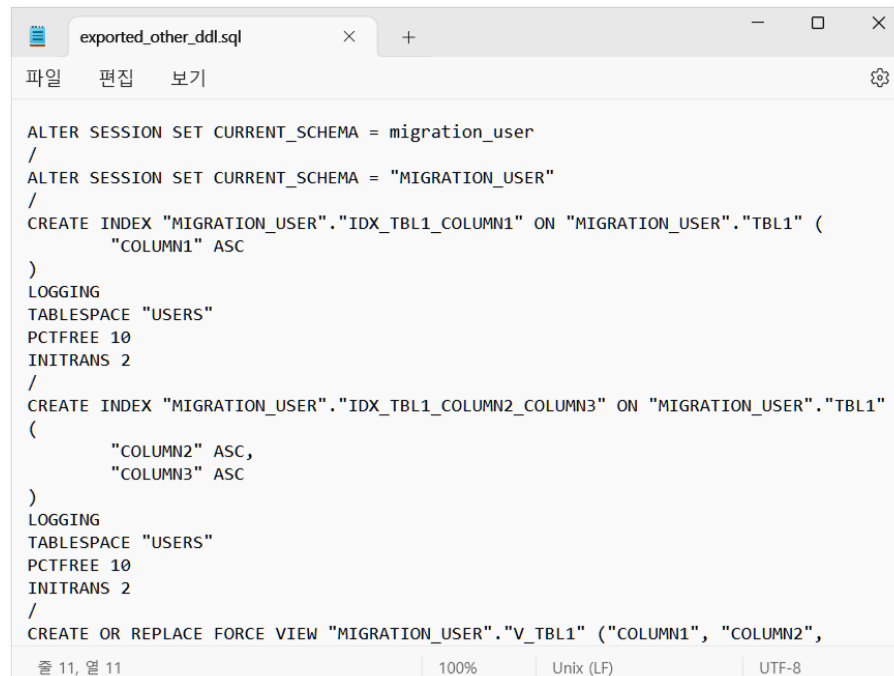
The screenshot shows the 'Result' window of T-Up Migrator. On the left is a sidebar with navigation items: Source Database, Type Conversion, Schema Object, Migration Option, Target Database, Summary, Data Migration, and Result (selected). The main area displays a table with migration statistics.

Type	Count	Success	Fail	Warning	Success Rate
Tablespace	0	0	0		0 %
Role	0	0	0		0 %
User	0	0	0		0 %
Privilege	0	0	0		0 %
Public Synonym	0	0	0		0 %
Sequence	0	0	0		0 %
Table	0	0	0		0 %
Index	0	0	0		0 %
Constraint	0	0	0		0 %
Synonym	0	0	0		0 %
Library	0	0	0		0 %
Materialized View	0	0	0		0 %
View	0	0	0		0 %
Function	0	0	0		0 %
Procedure	0	0	0		0 %
Package	0	0	0		0 %
Object Privilege	0	0	0		0 %
Type	0	0	0		0 %
Trigger	0	0	0		0 %
Job	0	0	0		0 %
Scheduler	0	0	0		0 %
Directory	0	0	0		0 %
Total	0	0	0		0.0 %
Data	0	0	0		0 %

At the bottom right, there are buttons for '< Back' and '× Close'.

8. 파일 확인

- DDL문이 정상적으로 출력됐는지 확인



The screenshot shows a text editor window titled 'exported_other_ddl.sql'. The editor contains the following SQL script:

```
ALTER SESSION SET CURRENT_SCHEMA = migration_user
/
ALTER SESSION SET CURRENT_SCHEMA = "MIGRATION_USER"
/
CREATE INDEX "MIGRATION_USER"."IDX_TBL1_COLUMN1" ON "MIGRATION_USER"."TBL1" (
    "COLUMN1" ASC
)
LOGGING
TABLESPACE "USERS"
PCTFREE 10
INITRANS 2
/
CREATE INDEX "MIGRATION_USER"."IDX_TBL1_COLUMN2_COLUMN3" ON "MIGRATION_USER"."TBL1"
(
    "COLUMN2" ASC,
    "COLUMN3" ASC
)
LOGGING
TABLESPACE "USERS"
PCTFREE 10
INITRANS 2
/
CREATE OR REPLACE FORCE VIEW "MIGRATION_USER"."V_TBL1" ("COLUMN1", "COLUMN2",
```

The status bar at the bottom indicates '줄 11, 열 11', '100%', 'Unix (LF)', and 'UTF-8'.

2.2.3. 전환 DDL 스크립트 보정

추출한 스크립트를 Tiberio에서 실행할 수 있게 정리와 변경을 한다.

• **Tablespace, User, Table DDL 스크립트 보정**

- Tablespace, User, Table 스크립트 파일명은 export_ddl.sql 이다.

• **Tablespace DDL**

- Tibero DB를 생성하였다면 System, Undo, 기본 Temp Tablespace는 이미 존재하므로 생성하지 않는다. 단, 기본 Temp Tablespace 이외의 Temp Tablespace는 전환해야 한다. (예 : TEMP1)
- Oracle 전용 Tablespace는 제외한다. (예 : SYSAUX)
- 데이터 파일의 경로, 파일명, Size 조정이 필요할 경우 수정한다.
- Tibero에서 미지원되는 사항은 주석처리 한다.

다음은 Tablespace 스크립트의 주석처리 항목에 대한 설명이다.

항목	설명
BLOCKSIZE 4K	Tibero는 Tablespace마다 Block Size 지정이 가능하지 않으므로 주석처리한다. 초기화 파라미터에서 DB를 생성할 때 고정되며 보통 8KB를 사용한다.
SEGMENT SPACE MANAGEMENT MANUAL	Tibero는 AUTO 부분만 지원하므로 MANUAL 부분을 AUTO로 변경한다.
FLASHBACK ON	Tibero는 파라미터를 통해서 전체모드로 FLASHBACK 기능 지원을 하므로 주석처리 한다.
TABLESPACE GROUP	TABLESPACE GROUP이 지원되지 않으므로 주석처리 한다. 하나의 Tablespace에 여러 데이터파일로 분리하는 것을 검토한다.

• **User DDL**

- 사용자별 패스워드를 확인한다. (Oracle DB 관리자에게 문의한다.)
- 사용자별 패스워드는 기본 tiber로 설정된다.
- 전환 대상 사용자만 적용한다. Oracle을 설치할 때 자동으로 생성되는 사용자는 전화이 필요하지 않다.
- Tibero에서 미지원되는 사항은 주석처리 한다.

항목	설명
GRANT UNLIMITED TABLESPACE TO USER_NAME	Tibero는 기본으로 UNLIMITED이며 해당 문법이 지원되지 않으므로 주석 처리 한다.
ALTER USER USERNAME QUOTA UNLIMITED ON USERNAME	사용자별 QUOTA 기능이 지원되지 않으므로 주석 처리 한다.

- **Table DDL**

- Oracle에서 전환되는 문자 집합으로 인해 Active Value 에러가 발생할 수 있다.
- 컬럼 타입 부분에서 에러가 발생할 경우 “Appendix C. 데이터 타입”의 "Oracle"을 참고한다.

데이터베이스 기타 오브젝트 DDL 스크립트

- **Tablespace, User, Table을 제외한 나머지 Object DDL 스크립트 보정**

- 추출 파일명은 export_other_ddl.sql 이다.

- **기타 오브젝트 리스트**

- Index
- View
- Constraint
- Sequence
- Synonym
- Materialized View
- Library
- Function
- Procedure
- Package
- Type
- Trigger
- Object Privilege
- Job
- DBMS_SCHEDULER
- External Procedure
 - 기타 오브젝트 DDL에 포함되지 않아, Oracle을 참조하여 생성한다.

2.2.4. 전환

데이터베이스 주요 오브젝트 DDL 스크립트

- Tiberio 데이터베이스 서버에 export_ddl.sql 파일을 업로드 한다.
 - 컬럼 타입 부분에서 에러가 발생할 경우 “Appendix C. 데이터 타입”의 "Oracle"을 참고한다.

- export_ddl.sql 수행

```
$ tbsql sys/tibero

tbsQL

Connected to Tibero.

SQL> spool export_ddl.log
spooling is started.

SQL> @export_ddl.sql
-- spool 파일을 통해 오류 확인

SQL> spool off
spooling is stopped: export_ddl.log
```

데이터 전환 수행

데이터 전환 작업 수행 시 다양한 도구를 통해 전환을 수행한다.

- **T-UP**
 - 클라이언트 PC에서 T-UP을 통해 전환을 수행한다.
 - Oracle → T-UP(클라이언트 PC) → Tibero
- **Table Migrator**
 - Tibero에서 Table Migrator를 통해 전환을 수행한다.
 - Oracle → Tibero
- **DB Link**
 - Tibero에서 Oracle DB Link를 통해 전환을 수행한다.
 - Oracle → Tibero

자세한 도구 사용법은 각 도구의 매뉴얼을 참고한다.

데이터베이스 기타 오브젝트 DDL 스크립트

- Tibero 데이터베이스 서버에 export_other_ddl.sql 파일을 업로드 한다.

- 사용자가 많거나 전환을 반복적으로 수행할 경우 Shell(또는 Batch) 스크립트를 작성하여 진행할 것을 권장한다.
- 전환을 반복적으로 수행할 경우 1차 전환을 할 때 의존성과 예외상황을 정리해 놓고 2차, 3차 전환을 수행할 때 스크립트에 사전작업을 수행하여 에러가 발생하지 않게 한다.
- Shell 프로그램을 사용하여 Object들이 자동으로 생성되도록 하고 다음과 같은 방법으로 검증한다.
- 모든 사용자에게 대해서 생성이 완료된 후 Oracle과 개수를 비교하여 누락된 부분이 없는지 확인한다
- 만약, 정상적으로 Object가 생성되지 않았다면 원인을 찾아 해결한 후 문제가 발생한 Object 수동 생성 및 다음 단계로 진행한다.

• export_other_ddl.sql 수행

```
$ tbsql sys/tibero

tbsql

Connected to Tibero.

SQL> spool export_other_ddl.log
Spooling is started.

SQL> @export_other_ddl.sql
-- spool 파일을 통해 오류 확인

SQL> spool off
Spooling is stopped: export_other_ddl.log
```

• Synonym

Synonym에서 참조하는 원본의 Object가 존재하지 않아도 생성할 때 문제가 되지 않는다. 이후에 원본 Object가 생성되었다면 사용할 때 자동으로 VALID하게 된다.

• View

생성할 때 의존성, Object 권한으로 인해 에러가 발생할 수 있다. 처음 생성할 때 컴파일 에러가 발생하더라도 모든 Object를 생성 후 "Object 상태변경"을 참고하여 VALID 수행 작업을 한다.

다음은 의존성을 확인하는 예이다.

```
SELECT OWNER, NAME, REFERENCED_OWNER, REFERENCED_NAME, REFERENCED_TYPE
FROM DBA_DEPENDENCIES
WHERE TYPE='VIEW'
AND REFERENCED_TYPE NOT IN ('TABLE', 'SYNONYM')
AND REFERENCED_NAME NOT IN ('DUAL')
AND REFERENCED_NAME NOT LIKE 'DBA_%'
AND OWNER NOT IN ('SYS', 'ORANGE', 'MDSYS', 'EXFSYS', 'XDB', 'SYSMAN', 'DBSNMP',
'SYSTEM' )
AND REFERENCED_OWNER NOT IN ('SYS', 'ORDSYS', 'WMSYS', 'MDSYS', 'EXFSYS', 'XDB',
'SYSMAN', 'SYSTEM', 'OLAPSYS', 'APEX_030200', 'CTXSYS')
ORDER BY 1, 2;
```

- **Sequence**

최종 전환 작업을 수행할 때에는 Oracle의 Current Value가 변경되지 않게 애플리케이션을 종료한 후 진행한다.

- **PSM(=PL/SQL)**

PSM을 생성할 때 다음과 같은 사항을 고려한다.

- 생성할 때 의존성, Object 권한으로 인해 에러가 발생할 수 있다. 처음 생성할 때 컴파일 에러가 발생 하더라도 모든 Object를 생성 후 "Object 상태 변경"을 참고하여 VALID 수행 작업을 한다.

- 해당 Object에 대한 권한이 없어 "TBR-8053: Not authorized" 에러가 발생할 수 있다. 이때, Object의 권한을 부여한 후 재생성하면 된다.

- PSM 내의 쿼리 부분에서 에러가 발생할 수 있다. 자세한 내용은 "애플리케이션 쿼리 수정"을 참고하여 수정한다.

- Object 타입의 경우 Tibero는 지원되지 않으므로 Package 내의 타입을 선언하여 우회한다.

다음은 Object 타입에 대하여 Package 내의 타입으로 우회 생성하는 방법이다.

- **Oracle**

```
CREATE OR REPLACE TYPE DOCUROLE_ID_OBJ IS OBJECT(DOCUROLE_ID VARCHAR2(6));  
/  
CREATE OR REPLACE TYPE DOCUROLE_ID_TAB IS TABLE OF DOCUROLE_ID_OBJ;  
/  
CREATE OR REPLACE FUNCTION USER_DOCUROLESET(PARAM IN VARCHAR2)  
RETURN DOCUROLE_ID_TAB  
IS  
L_DOCUROLE_ID_TAB DOCUROLE_ID_TAB := DOCUROLE_ID_TAB();  
BEGIN  
L_DOCUROLE_ID_TAB.EXTEND;  
L_DOCUROLE_ID_TAB(1).DOCUROLE_ID := 'test';  
RETURN L_DOCUROLE_ID_TAB;  
END;  
/
```

- **Tibero**

```
CREATE OR REPLACE PACKAGE "PK_TYPES" AS  
TYPE DOCUROLE_ID_OBJ IS RECORD(DOCUROLE_ID VARCHAR2(6));  
TYPE DOCUROLE_ID_TAB IS TABLE OF DOCUROLE_ID_OBJ;  
END;  
/  
CREATE OR REPLACE FUNCTION USER_DOCUROLESET(PARAM IN VARCHAR2)  
RETURN PK_TYPES.DOCUROLE_ID_TAB
```

```

IS
L_DOCUROLE_ID_TAB PK_TYPES.DOCUROLE_ID_TAB := PK_TYPES.DOCUROLE_ID_TAB();
N INTEGER := 0;
BEGIN
L_DOCUROLE_ID_TAB.EXTEND;
L_DOCUROLE_ID_TAB(1).DOCUROLE_ID := 'test';
RETURN L_DOCUROLE_ID_TAB;
END;
/

```

• Index

다음은 Index 생성 이후 Oracle과 Tiberio의 Index 개수 비교에 대한 설명이다.

상황	설명
Tibero가 더 적을 경우	Constraint가 생성되면서 자동으로 Index가 생성되는 것도 고려해야 하므로 Constraint 생성 이후에 비교한다.
Tibero가 더 많을 경우	Table내의 LOB Column에 대응하는 Index가 자동생성된 부분이 아닌지 확인한다. (Oracle 11g r2에서도 동일하게 자동생성되며 Oracle 과거버전의 경우 생성 안하는 경우가 있다.)

• Constraint

전환 검증항목"의 "Constraint 개수 확인"에서 확인한 항목과 비교한다.

• Trigger

Oracle에서 Disable된 Trigger를 전환할 때 Syntax 에러가 발생하므로 아래와 같이 처리한다.

- Disable 구문 부분을 주석처리 한다.
- Tibero에서 Disable 처리한다.

```
alter trigger trigger_name disable;
```

• Grant

다음은 Tibero의 Grant 미지원 사항에 대한 설명으로, Grant Count 계산할 때 제외시킨다.

항목	설명
COMMIT REFRESH	Mview를 생성할 때 생성 구문으로 지원하며 권한으로는 지원하지 않는다.
QUERY REWRITE	Mview를 생성할 때 생성 구문으로 지원하며 권한으로는 지원하지 않는다.
DEBUG	PL/SQL DEBUGGING 권한으로 현재 Tibero는 지원하지 않는다.
FLASHBACK	파라미터 형태로 지원하며 권한으로는 지원하지 않는다.

위와 같은 권한이 정상적인 권한과 함께 부여될 경우 모두 부여되지 않으므로 미지원 권한을 찾아서 삭제한 이후에 재수행이 필요하다.

- 기타

기타 External Procedure, Job 부분은 Oracle을 참조하여 생성한다.

Object 상태 변경

생성할 때 의존성, Object 권한으로 인해 Object 상태가 INVALID할 수 있으므로 아래와 같이 처리한다.

1. Object 확인

- 상태별 개수 확인

```
col owner format a20
set pages 500
select owner, object_type, status, count(*)
from dba_objects
group by owner, object_type, status
order by owner, object_type, status;
```

- INVALID Object 확인

```
set linesize 120
col object_name format A30
col owner format a20
select owner, object_name, object_type, status
from dba_objects
where status='INVALID';
```

2. 재컴파일 수행

'sys' 계정으로 접속하여 아래와 같이 특정 스키마의 INVALID된 오브젝트 재컴파일을 수행한다.

```
$ tbsql sys/tibero

tbsQL

Connected to Tibero.

SQL> exec utl_recomp.recomp_serial(schema=>'MIGRATION_USER');

PSM completed.
```

3. Object별 재컴파일 및 에러 확인

해당 사용자로 접속하여 Object별로 재컴파일 및 에러 내용을 확인한다.

- View 재컴파일

```
select * from view_name where rownum < 1;
```

- PSM(=PL/SQL) 재컴파일

```
alter function function_name compile;  
alter procedure procedure_name compile;
```

- PSM(=PL/SQL) 에러 확인

```
show errors
```

- Index 재컴파일

```
alter index index_name rebuild;
```

2.2.5. 전환 검증

전환이 완료된 후 DB 측에서 확인하는 사항으로 크게 Object 개수와 데이터 검증이 있다.

Object 개수

- Table

```
set linesize 150  
column owner format a30  
  
select  
    owner,  
    count(table_name) as "table_cnt"  
from  
    dba_tables  
where
```

```

owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
owner
order by
owner;

```

- **View**

```

set linesize 150
column owner format a30

select
owner,
count(view_name) as "view_cnt"
from
dba_views
where
owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
owner
order by
owner;

```

- **Synonym**

```

set linesize 150
column owner format a30

select
owner,
count(1) as "synonym_cnt"
from
dba_synonyms
where
owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
owner
order by owner;

```

- **Sequence**

```

set linesize 150
column sequence_owner format a30

select
sequence_owner,
count( sequence_name ) as "sequence_cnt"

```

```

from
    dba_sequences
where
    sequence_owner not in
('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    sequence_owner
order by
    sequence_owner;

```

- **Package**

```

set linesize 150
column owner format a30

select
    owner,
    count( name ) as "package_cnt"
from
    dba_source
where
    type = 'PACKAGE' and
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;

```

- **Package Body**

```

set linesize 150
column owner format a30

select
    owner,
    count( name ) as "package_body_cnt"
from
    dba_source
where
    type = 'PACKAGE BODY' and
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;

```

- **Function**

```

set linesize 150
column owner format a30

select
    owner,
    count( name ) as "function_cnt"
from
    dba_source
where
    type = 'FUNCTION' and
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;

```

- Procedure

```

set linesize 150
column owner format a30

select
    owner,
    count( name ) as "procuedure_cnt"
from
    dba_source
where
    type = 'PROCEDURE' and
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;

```

- Index

```

set linesize 150
column owner format a30

select
    owner,
    count( index_name ) as "index_cnt"
from
    dba_indexes
where
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by

```

```
owner;
```

- **Constraint**

```
set linesize 150
column owner format a30

select
    owner,
    constraint_type,
    count( constraint_name ) as "constraint_cnt"
from
    dba_constraints
where
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner, constraint_type
order by
    owner ,constraint_type;
```

- **Grant**

```
set linesize 150
column col format a30

select
    owner,
    count(1) as "grant_cnt"
from
    dba_tab_privs
where
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;
```

- **Trigger**

```
set linesize 150
column owner format a30

select
    owner,
    count(trigger_name) as "trigger_cnt"
from
    dba_triggers
```

```

where
    owner not in ('SYS','SYSGIS','SYSCAT','PUBLIC','OUTLN','TIBERO','TIBERO1')
group by
    owner
order by
    owner;

```

데이터 검증

- Oracle DB Link

- Count 건수 비교

```

select 'tbl1' as table_name, count(*) as oracle_count, count(*) - (select
count(*) from tbl1 where rownum = 1) as diff from tbl1@ora;

```

- Minus 결과 비교

```

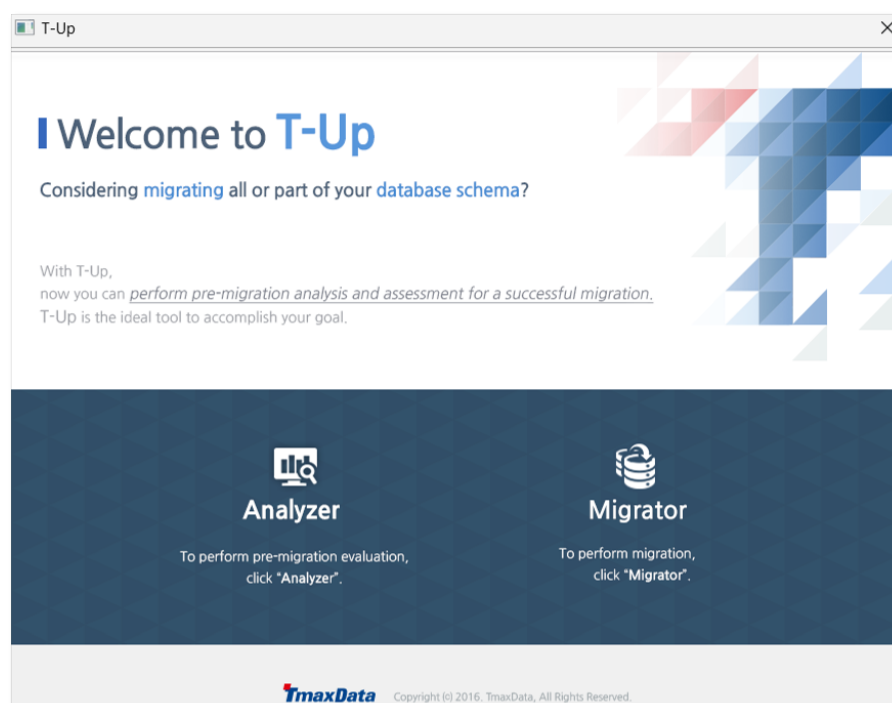
select * from tbl1@ora minus
select * from tbl1;

```

- T-UP Data Transfer Verification

1. T-UP > Migrator

- “Migrator” 선택



2. Source Database > Type Conversion

- ORACLE과 대응되는 데이터 타입 선택

The screenshot shows the 'Type Conversion' window in T-Up Migrator. The left sidebar has 'Type Conversion' selected. The main area is titled 'Type Mapping Rule' and shows a 'Rule Name' of 'Conversion rule 01'. Below this, there are two sections: 'Selecta...Mapping' and 'Default Mapping'. Each section contains two columns: 'Oracle Data Type' and 'Tibero Data Type'.

Oracle Data Type	Tibero Data Type
LONG	CLOB
LONG RAW	BLOB

Oracle Data Type	Tibero Data Type
NUMBER	NUMBER
FLOAT	NUMBER
CHAR	CHAR
VARCHAR	VARCHAR
VARCHAR2	VARCHAR
RAW	RAW
DATE	DATE
TIME	TIME
TIMESTAMP	TIMESTAMP
INTERVAL YEAR TO MONTH	INTERVAL YEAR TO MONTH
INTERVAL DAY TO SECOND	INTERVAL DAY TO SECOND
BLOB	BLOB
CLOB	CLOB
NCLOB	NCLOB
ROWID	ROWID
UROWID	-
NCHAR	NCHAR

At the bottom right, there are buttons for '< Back', 'Next >', and 'X Close'.

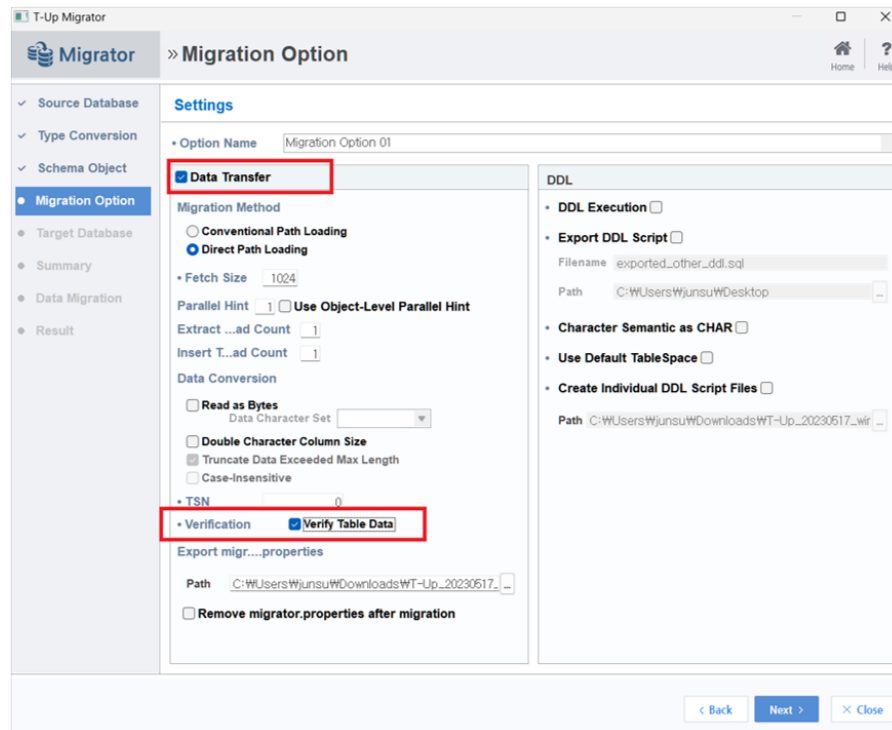
3. Schema Object

- "Dependent Objects" Table 선택
- "Schemas" 이관 대상 Schema 및 Table 선택

This screenshot is identical to the one above, showing the 'Type Conversion' window in T-Up Migrator with the same mapping rule configuration.

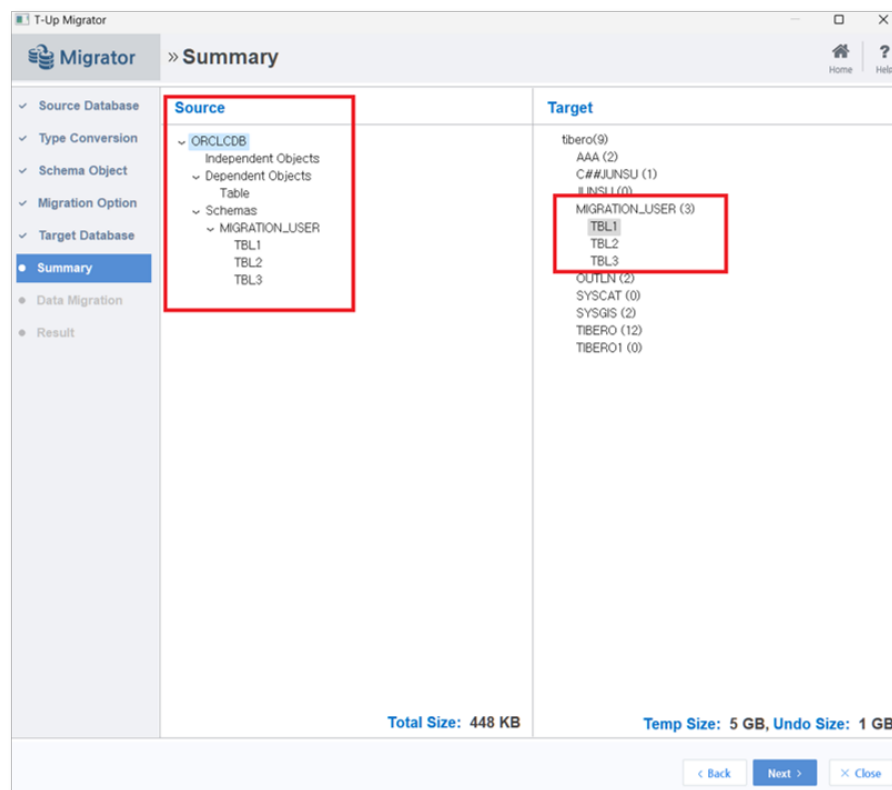
4. Migration Option

- "Data Transfer" 활성화
- "Verification" 선택



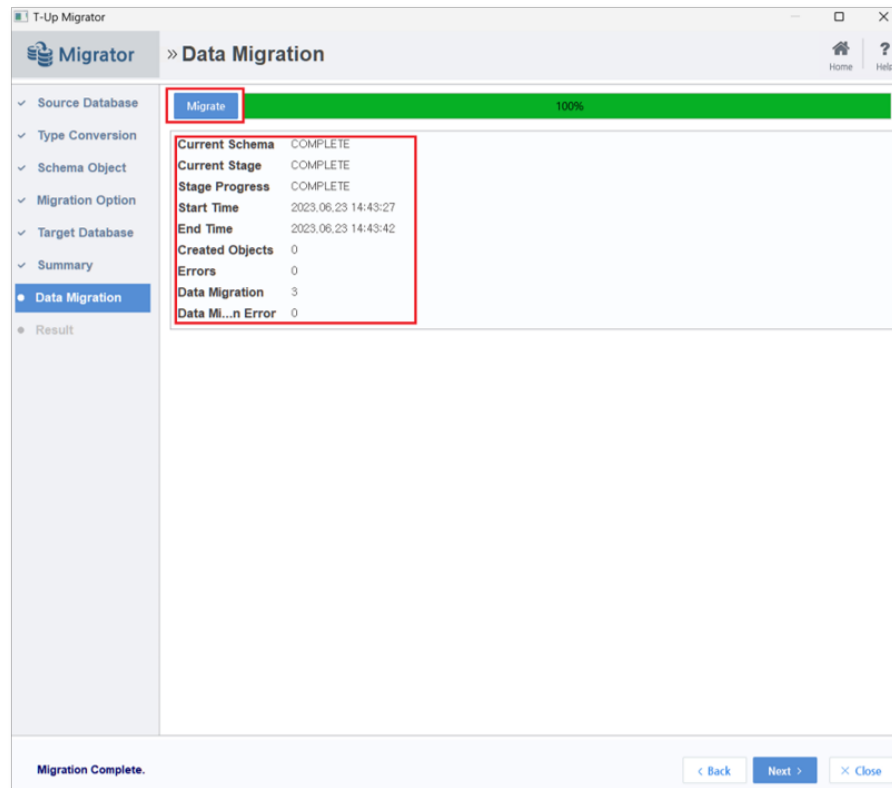
5. Target Database > Summary

- Source와 Target 대응 Schema 및 Table 확인



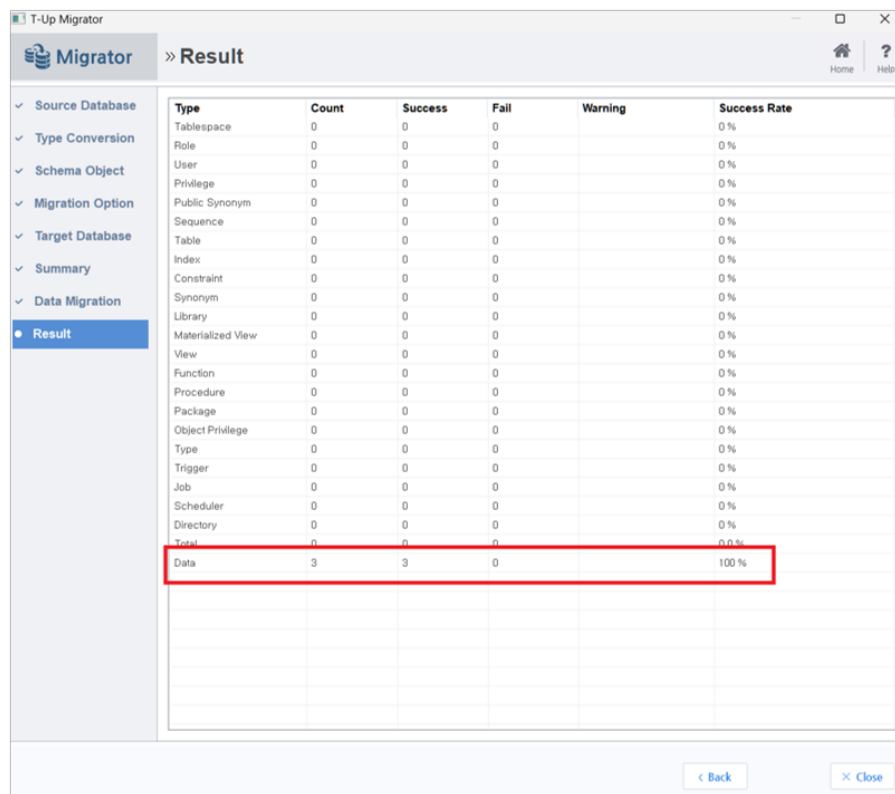
6. Data Migration

- "Migrate" 수행

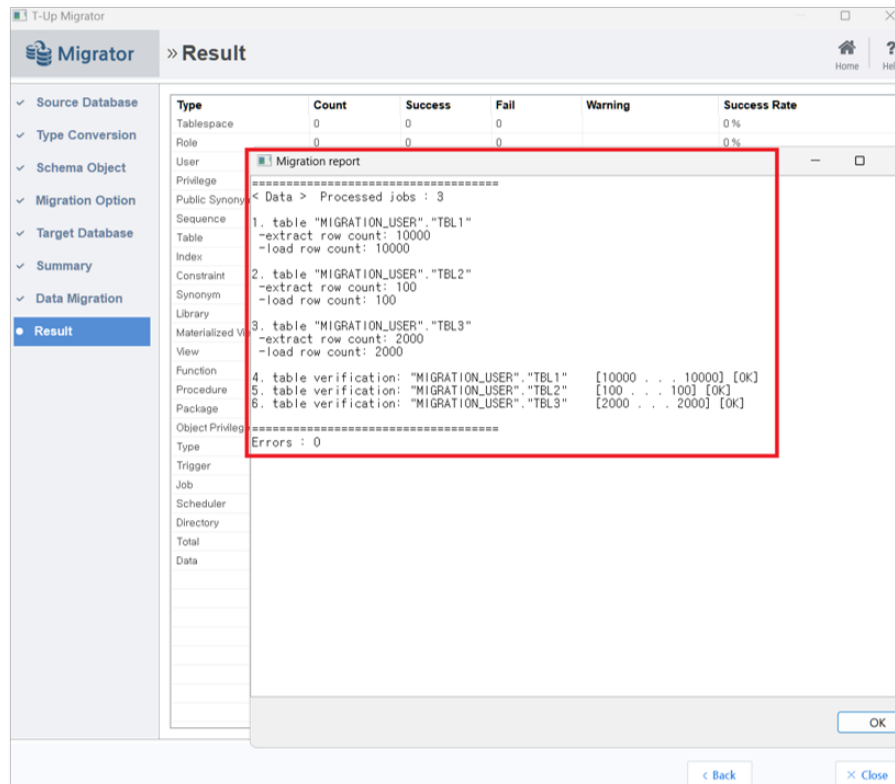


7. Result

- Type: Data의 지정 Table 성공률 확인



■ Type: Data 클릭 시 Verification 결과 확인



2.3. 애플리케이션 전환

DB 전환이 완료된 후 애플리케이션 전환을 수행한다. 크게 인터페이스 수정 및 애플리케이션 쿼리 수정이 필요할 수 있다.

인터페이스 수정

- JDBC

\$TB_HOME/client/lib/jar 폴더에 tibero6-jdbc.jar 드라이버를 애플리케이션의 드라이버 관리 폴더에 복사한다.

- Oracle

```

String jdbc_url = "jdbc:oracle:thin:@127.0.0.1:1521:ORCL";
String user = "oracle_username";
String passwd = "oracle_passwd";
Class.forName("oracle.jdbc.driver.OracleDriver");
  
```

- **Tibero**

```
String jdbc_url = "jdbc:tibero:thin:@127.0.0.1:8629:tibero";
String user = "tibero_username";
String passwd = "tibero_passwd";
Class.forName("com.tmax.tibero.jdbc.TbDriver");
```

- **ODBC**

Tibero ODBC를 설치한다.

- **Oracle**

```
DRIVER={Microsoft ODBC for Oracle}; server=(DESCRIPTION=
(AADDRESS=(PROTOCOL=TCP)(HOST=127.0.0.1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=utf8)));UID=scott;PWD=tiger;
```

- **Tibero**

```
DRIVER={Tibero 6 ODBC Driver};SERVER=127.0.0.1;PORT=8629
;DB=t6;UID=dbtech;PWD=dbtech;
```

- **OLE DB**

Tibero OLE DB를 설치한다.

- **Oracle**

```
Provider=msdaora;Data Source=MyDB;User Id=myUsername
;Password=myPassword;
```

- **Tibero**

```
Provider=tbprov.Tbprov.6;Data Source=MyDB;User ID=dbtech;Password=dbtech
;Updatable Cursor=True;OLE DB Services=-2
```

애플리케이션 쿼리 수정

애플리케이션 쿼리 수정은 필요한 경우에만 사용하며 다음과 같은 상황에서 필요하다.

- **Syntax Error가 발생할 경우 1**

상황	SQL을 작성할 때 From절에 있는 Table 또는 서브 쿼리절에 Alias를 주지 않고 Join 등을 실행할 때 중복된 Column이 존재할 경우 Syntax Error가 발생한다.
대응 방법	Alias를 주어서 중복되지 않게 처리한다.

- Syntax Error가 발생할 경우 2

상황	Union All 같은 쿼리를 사용하여 결과를 보려는 경우 실제 상위 쿼리는 특정 Column을 number type으로 하고, 하위 쿼리는 varchar type으로 하면 Syntax Error가 발생한다.
대응 방법	같은 Column절에서는 동일한 Type을 사용한다.

- Syntax Error가 발생할 경우 3

상황	PSM(=PL/SQL) 내부에 같은 이름의 변수를 선언할 경우 Syntax Error가 발생한다.
대응 방법	최종 선언된 변수 이름을 사용하고 앞 부분에 선언한 같은 이름의 변수를 삭제한다.

- 결과가 정렬된 상태로 출력되지 않을 경우

상황	Oracle은 orderby를 주지 않아도 결과가 정렬된 형태로 나오는 경우가 있는 반면에 Tibero는 정렬된 상태로 출력되지 않을 수 있다. 단, 대용량 테이블에서 Index 등을 강제로 태워서 일부 값만 보고자 하는 쿼리 구현의 경우는 Tibero 역시 order by를 사용하지 않아도 정렬되어 출력된다.
대응 방법	정렬된 결과를 꼭 얻어야 한다면 order by를 명시해준다.

- PSM(=PL/SQL) 작성 후 컴파일을 수행할 때 Error가 발생하는 경우

상황	PSM(=PL/SQL)을 작성한 후 컴파일을 수행할 때 내부 SQL의 Object에 대하여 실제 Object, Column 등이 존재하는지 체크해서 유효하지 않다면 Tibero는 Error를 발생한다. 반면에 Oracle은 위와 같은 상황에서 컴파일을 성공했다더라도 runtime때 Error가 발생하는 차이가 있다.
대응 방법	실제 존재하는 Object를 사용하고 정확한 Column Name을 사용한다.

Appendix. 데이터 타입

본 장에서는 Tibero와 다른 DBMS의 컬럼 데이터 타입을 비교하여 설명한다.

ORACLE

다음은 Oracle과 Tibero의 데이터 타입 비교이다.

Oracle	Tibero
CHAR	CHAR
VARCHAR	VARCHAR
VARCHAR2	VARCHAR2(내부적으로는 VARCHAR로 생성하지만 Oracle의 VARCHAR2와 동작방식이 동일하다)
NCHAR	NCHAR
NVARCHAR	NVARCHAR
NVARCHAR2	NVARCHAR2(내부적으로는 NVARCHAR로 생성하지만 Oracle의 NVARCHAR2와 동작방식이 동일하다)
BLOB	BLOB
CLOB	CLOB
NCLOB	NCLOB
LONG	LONG
NUMBER	NUMBER
NUMERIC	NUMERIC(내부적으로 NUMBER를 생성함)
INT	INT(내부적으로 NUMBER를 생성함)
DECIMAL	DECIMAL(내부적으로 NUMBER를 생성함)
DOUBLE	NUMBER
FLOAT	FLOAT(내부적으로 NUMBER를 생성함)
REAL	REAL(내부적으로 NUMBER를 생성함)
SMALLINT	SMALLINT(내부적으로 NUMBER를 생성함)
BINARY_FLOAT	NUMBER
BINARY_DOUBLE	NUMBER
DATE	DATE

Oracle	Tibero
TIMESTAMP	TIMESTAMP
BFILE	TIBERO BFILE 지원하지 않음, BLOB으로 우회
RAW	RAW
LONG RAW	LONG RAW
XMLTYPE	XMLTYPE(생성문법이 일부 다르므로 문법 변경이 필요함, Oracle : SYS.XMLTYPE, Tibero : XMLTYPE)
GEOMETRY	GEOMETRY

MS-SQL

다음은 MS-SQL과 Tibero의 데이터 타입 비교이다.

MS-SQL	Tibero
VARCHAR(MAX)	CLOB
NVARCHAR(MAX)	NCLOB
VARBINARY(MAX)	BLOB
UNIQUEIDENTIFIER	VARCHAR
SQL_VARIANT LONG	RAW
BIGINT	NUMBER
NTEXT	CLOB

Sybase

다음은 Sybase와 Tibero의 데이터 타입 비교이다.

Sybase	Tibero
TINYINT NUMBER	NUMBER
SMALLINT NUMBER	NUMBER
INT NUMBER	NUMBER

Sybase	Tibero
NUMERIC(P,S)	NUMBER(n,n)
DECIMAL(P,S)	NUMBER(n,n)
FLOAT(PRECISION)	NUMBER
DOUBLE(PRECISION)	NUMBER
REAL NUMBER	NUMBER
SMALLMONEY	NUMBER(10,4)
MONEY	NUMBER(19,4)
SMALLDATETIME	DATE 또는 TIMESTAMP
DATETIME	DATE 또는 TIMESTAMP
CHAR(N)	CHAR(N)
VARCHAR(N)	VARCHAR(N)
UNICHAR(N)	NCHAR(N)
UNIVARCHAR(N)	NVARCHAR(N)
NCHAR(N)	NCHAR(N)
NVARCHAR(N)	NVARCHAR(N)
BINARY(N)	CLOB
VARBINARY(N)	BLOB
BIT	NUMBER(1) 또는 CHAR(1)
TEXT CLOB	CLOB
IMAGE	BLOB