SQL 프로그래밍

금융연수원

Contents

01 DBMS 설치 및 환경구성
02 SELECT 기본 문장 작성
03 WHERE 절을 이용한 행 제한
04 ORDER BY 절을 이용한 정렬
05 단일 행 함수 활용
06 GROUP BY 절을 이용한 그룹 생성
07 Join 문장 작성
08 Subquery 활용
09 집합 연산자 활용
10 테이블 생성 및 관리
11 DML 활용
12 트랜잭션 이해 및 관리
13 제약조건 이해
14 기타 객체 관리

01. Database 설치 및 환경 구성

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

1. Oracle Database Server 구축

Oracle Database Software 는 크게 두 가지 버전이 있습니다. Enterprise/Standard Edition 과 Express Edition 입니다. Enterprise/Standard Edition 은 기업용으로 비용을 지불해야하는 제품이고 실제 운영되는 서버들이 이 제품으로 구축됩니다. 구매 옵션에 따라 모든 기능을 활용할 수 있으나 학습용으로 설치하기엔 제품이 무겁습니다. 때문에 저희는 Express Edition 을 설치하겠습니다. Express Edition 은 무료로 제공되는 제품이며 기능이나 CPU, 메모리, 저장 공간에는 제약이 있으나 개인이 사용하기에는 충분한 기능과 리소스를 제공합니다.

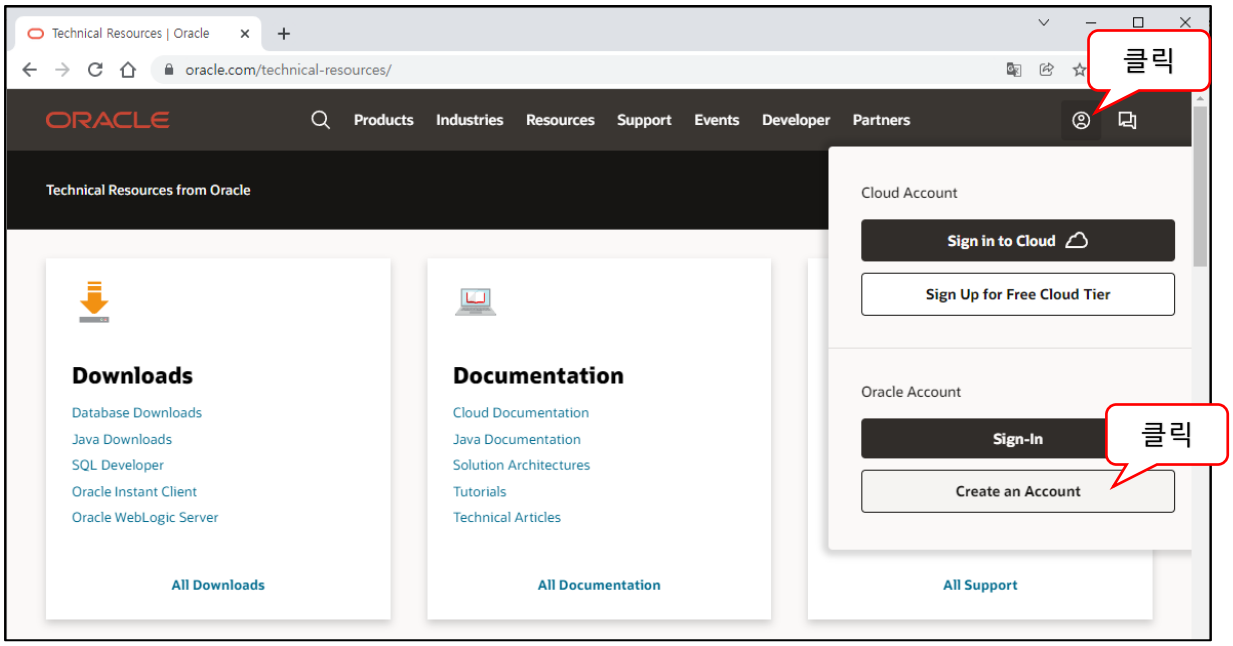
Oracle Database Express Edition 21c 설치 전 확인사항

- 지원 OS : Windows (64bit)
- 메모리 : 4GB 이상 메모리 사용 권장
- 저장공간: 15GB 이상 저장 공간 필요
- PC 이름 및 로그인 유저: 한글 이름 사용 불가 (Microsoft 계정일 경우 로컬 계정으로 변경)

위의 내용이 이상 없다면 우리는 Windows(64bit) 환경에서 Oracle Database Express Edition 21c 를 설치하고 구성하겠습니다.

Oracle Database Express Edition 21c 다운로드

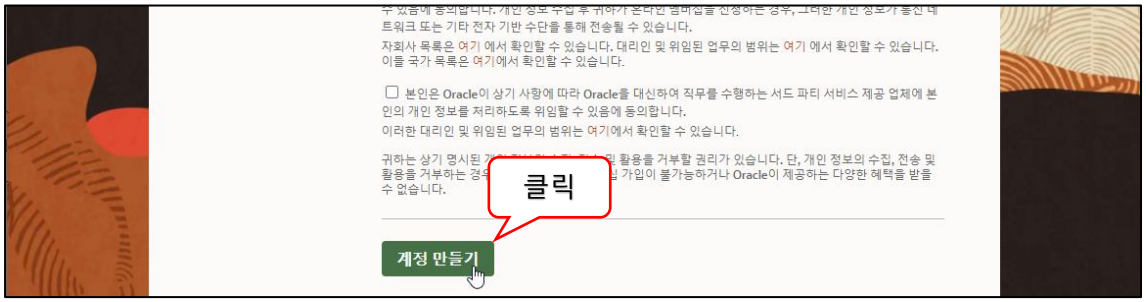
http://otn.oracle.com 사이트에 접속하고 Oracle 계정이 없다면 무료 가입을 먼저 진행합니다. 오른쪽 상단의 View Accounts 를 클릭하고 Create an Account 를 클릭합니다. (이미 계정을 가지고 있다면 다음 단계로 넘어가세요.)



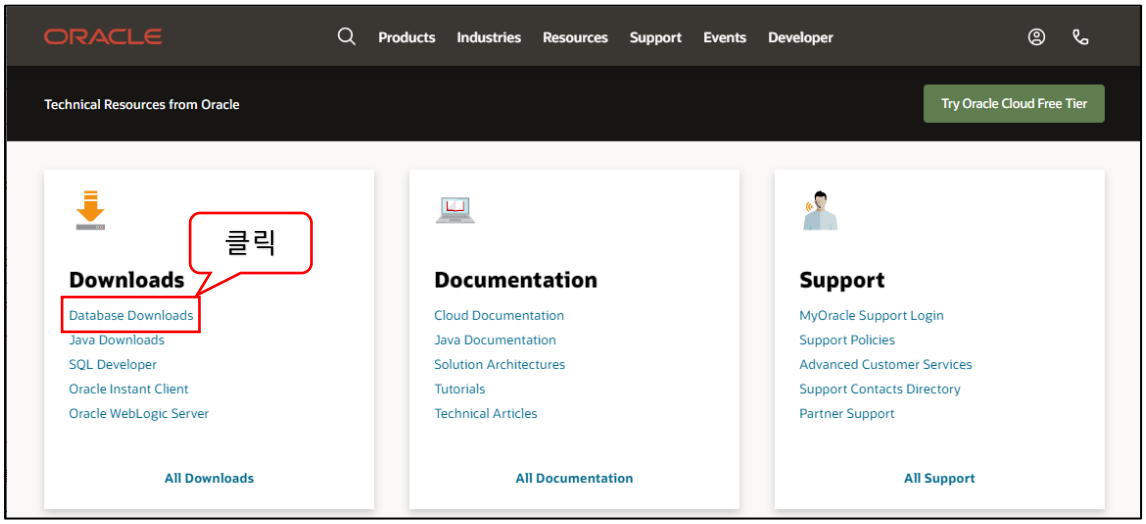
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

일반적인 사이트 가입과 다르지 않습니다. 필요한 정보를 기입하고 화면 아래의 계정 만들기를 클릭합니다.

. . .

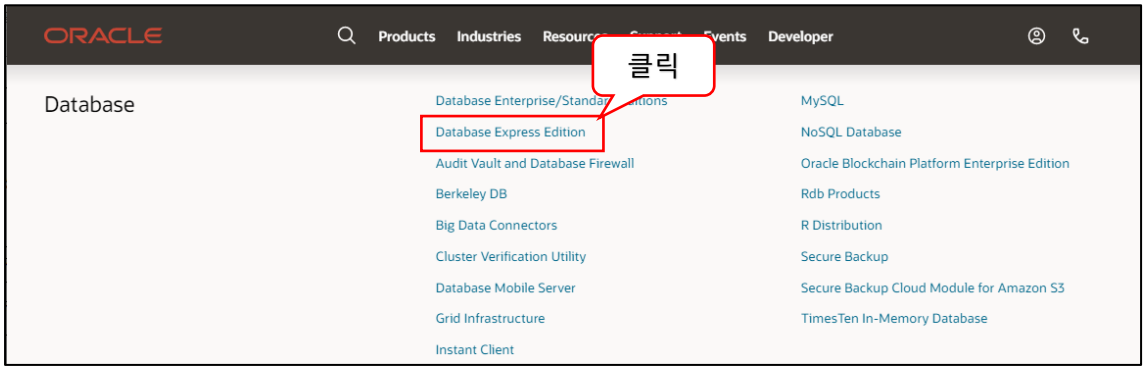


다시 처음 화면으로 돌아와서 Database Downloads 링크를 클릭합니다. (http://otn.oracle.com)

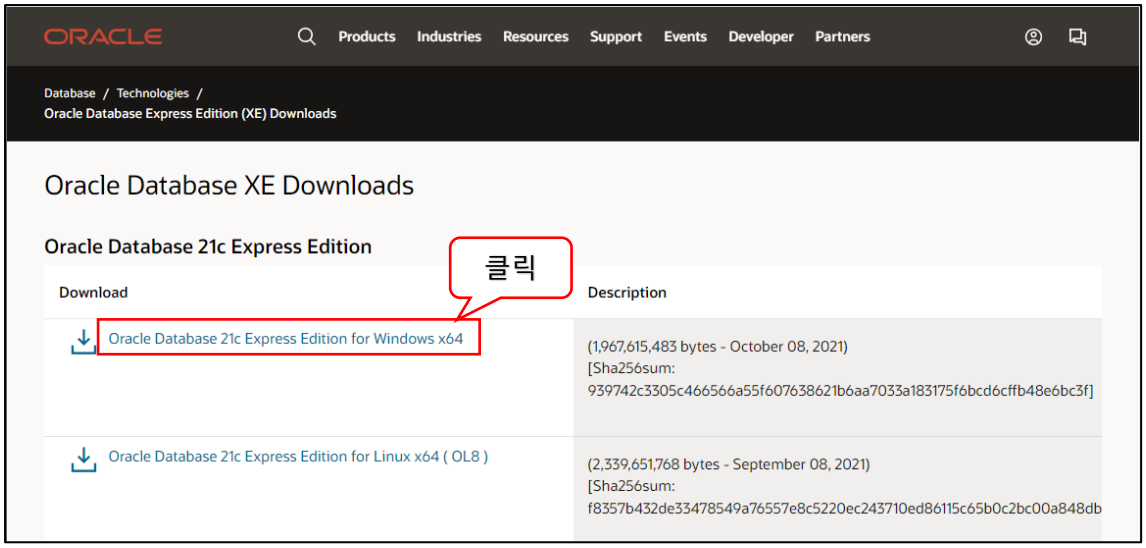


Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

목록 중에 Database Express Edition 링크를 클릭합니다.



Oracle Database 21c Express Edition for Windows x64 링크를 클릭합니다.

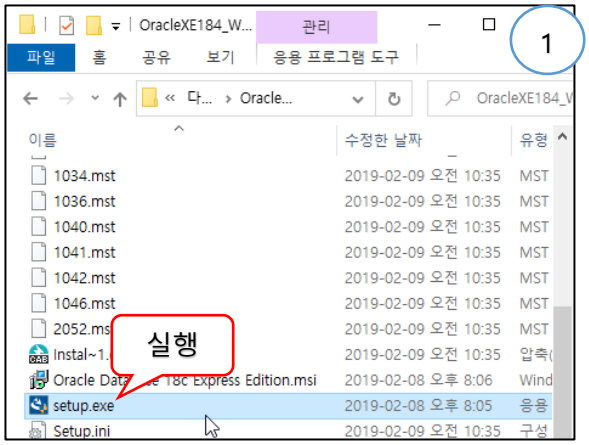


이제 다운로드가 시작되면 네트워크 상황에 따라 시간이 소요될 수 있습니다.

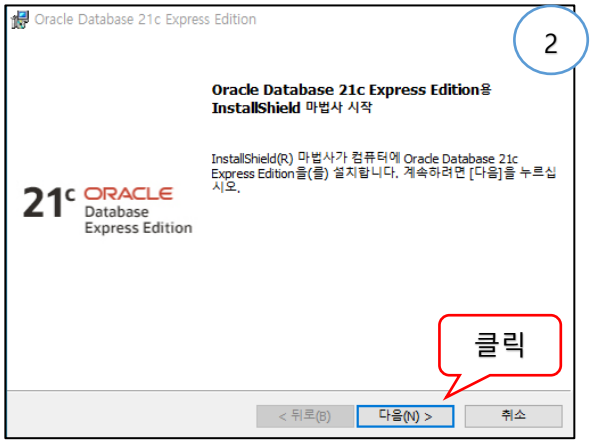
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Oracle Database Express Edition 21c 설치 및 Database 구성

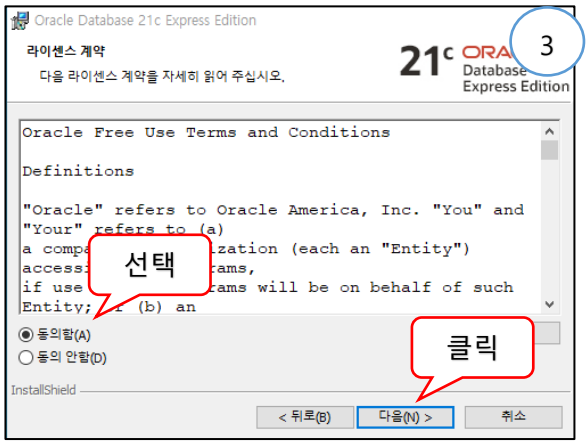
다운받은 설치 파일의 압축을 해제하고 현재 사용중인 로컬 PC 에 설치합니다. 만약 Express Edition 21c 보다 이전 버전의 설치파일이 있다면 동일한 방법으로 설치를 진행하기 때문에 아래 내용대로 설치를 진행해도 됩니다.



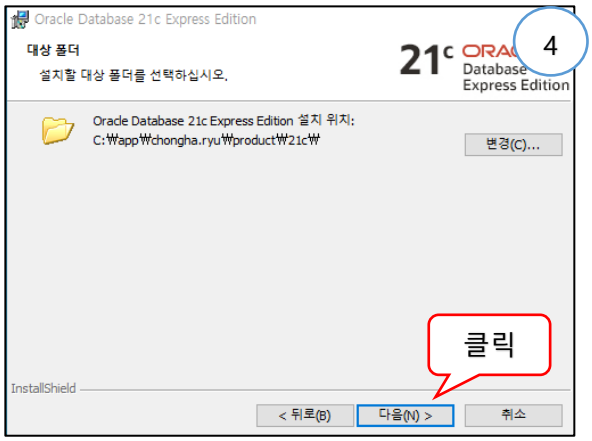
다운받은 파일을 압축해제하고, setup.exe 파일을 실행합니다.



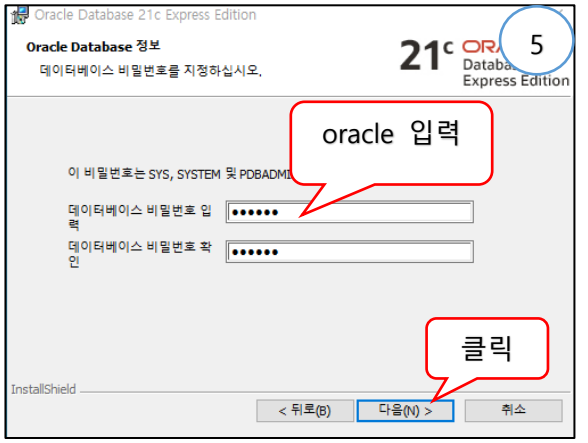
다음을 클릭합니다.



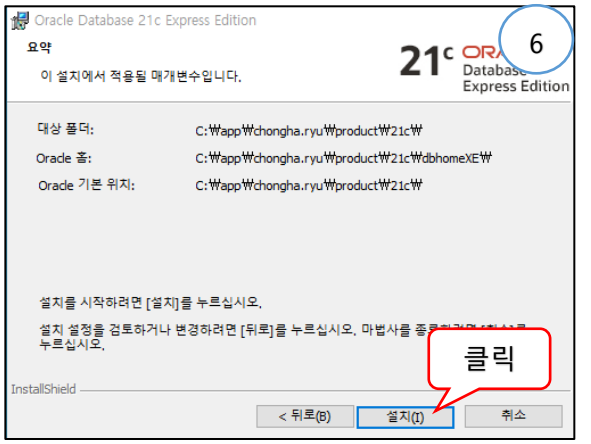
라이센스는 동의함을 선택하고 다음을 클릭합니다.



가급적 기본 경로에 설치를 진행하며, 필요하다면 경로는 변경할 수 있습니다.

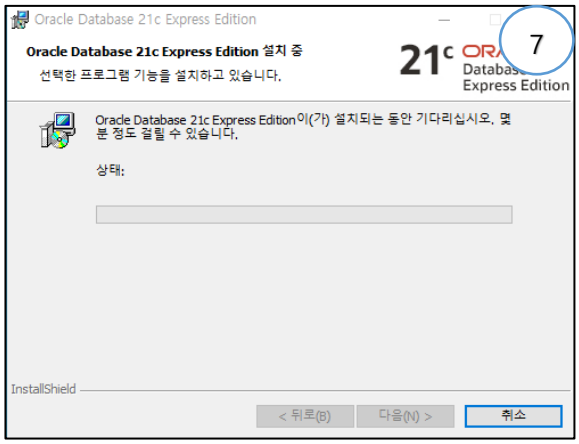


oracle (대소문자 구분)을 입력하고 다음을 클릭합니다.

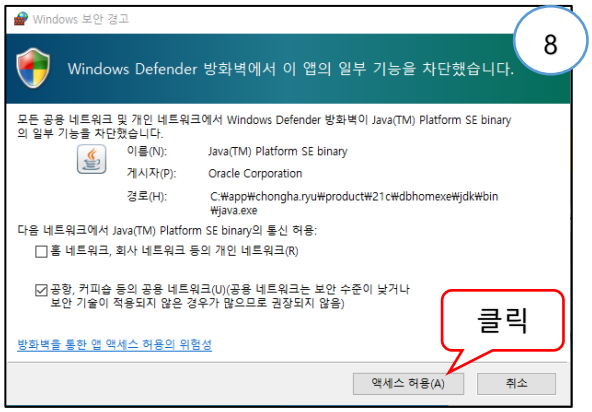


이상 없다면 설치를 클릭합니다.

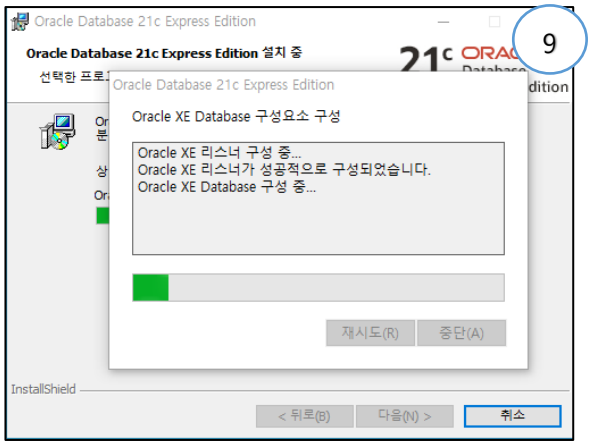
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.



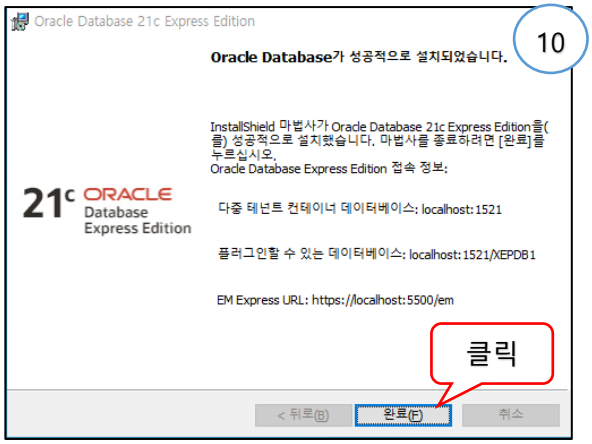
Oracle DB Software 를 설치하고 있습니다.



설치 진행도중 보안 경고가 나오면 액세스 허용을 클릭합니다.



DB Software 가 설치된 이후 Database 생성이 시작됩니다.



설치가 완료됐습니다. 완료 버튼을 클릭하여 설치를 마무리합니다.

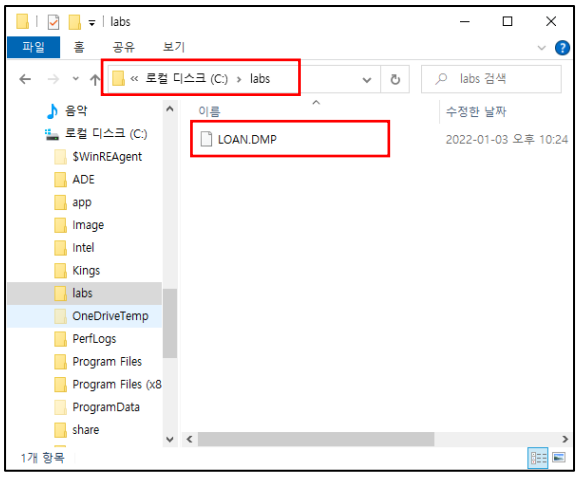
※ 마지막 화면에서 플러그인할 수 있는 데이터베이스 이름을 확인합니다. 추후 DB 접속 시 사용할 정보이기 때문에 화면과 다르다면 따로 메모가 필요합니다. (상황에 따라 1522 포트나 다른 번호가 보일 수 있음)

만약 Oracle Database Express Edition 11g 버전을 설치하고 있다면 보안 경고를 제외한 모든 화면이 동일합니다. 그리고 설치된 Oracle Database Server 를 삭제하기전까지는 언제든지 실습을 하실 수 있습니다.

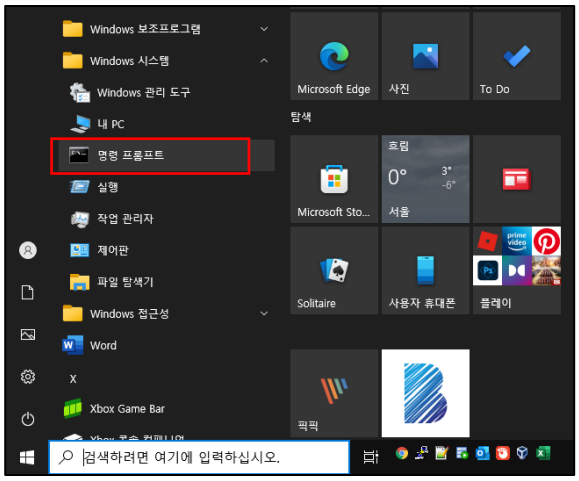
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

실습 계정 Import

강사가 제공한 덤프파일(LOAN.DMP)을 C:\labs 폴더에 복사합니다. 폴더가 없다면 생성하고, 경우에 따라 제공되는 덤프파일 이름은 다를 수 있습니다.

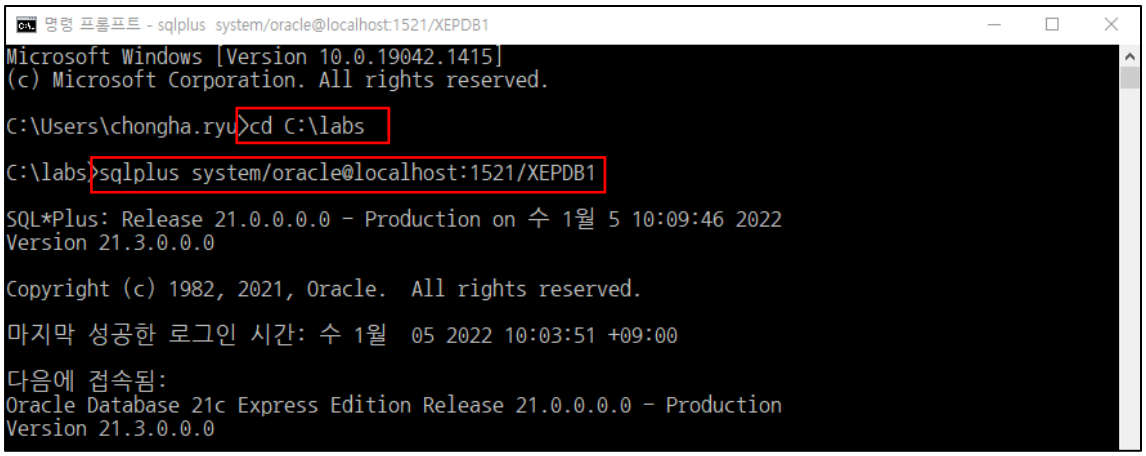


C:\labs\LOAN.DMP 파일 복사



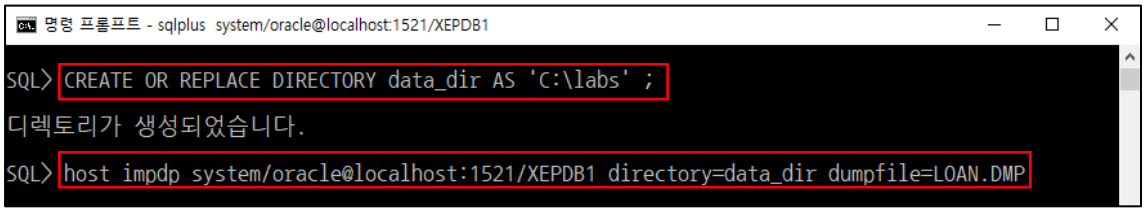
명령 프롬프트 실행

덤프파일이 있는 경로로 이동하여 sqlplus 를 실행합니다. 이때 접속 정보는 DB 설치 시 확인했던 정보를 이용하며 1522 나 다른 번호를 사용할 수도 있습니다.




```
cd C:\labs
sqlplus system/oracle@localhost:1521/XEPDB1
```

디렉토리 객체를 생성하고 덤프파일을 임포트 합니다.




```
CREATE OR REPLACE DIRECTORY data_dir AS 'C:\labs' ;
host impdp system/oracle@localhost:1521/XEPDB1 directory=data_dir dumpfile=LOAN.DMP
```


Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

임포트가 정상적으로 진행된다면 다음과 같은 메시지가 출력됩니다. 마지막 SQL 프롬프트가 보일 때까지 대기하고, 임포트가 종료되면 EXIT 합니다.

```
Import: Release 21.0.0.0.0 - Production on 수 1월 5 10:11:09 2022
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle and/or its affiliates.  All rights reserved.

접속 대상: Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
마스터 테이블 "SYSTEM"."SYS_IMPORT_FULL_01"이(가) 성공적으로 로드됨/로드 취소됨
"SYSTEM"."SYS_IMPORT_FULL_01" 시작 중: system/********@localhost:1521/XEPDB1 directory=data_dir
dumpfile=LOAN.DMP
객체 유형 SCHEMA_EXPORT/USER 처리 중
객체 유형 SCHEMA_EXPORT/SYSTEM_GRANT 처리 중
객체 유형 SCHEMA_EXPORT/ROLE_GRANT 처리 중
객체 유형 SCHEMA_EXPORT/DEFAULT_ROLE 처리 중
객체 유형 SCHEMA_EXPORT/TABLESPACE_QUOTA 처리 중
객체 유형 SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA 처리 중
객체 유형 SCHEMA_EXPORT/SEQUENCE/SEQUENCE 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/TABLE 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/TABLE_DATA 처리 중
. . "LOAN"."TBACK_PLAN"                          3.963 MB   94008행이 임포트됨
. . "LOAN"."TBACK"                               1.291 MB   33792행이 임포트됨
. . "LOAN"."TDLQ"                                1.213 MB   52762행이 임포트됨
. . "LOAN"."TACCT"                               117.4 KB    1473행이 임포트됨
. . "LOAN"."TCREDIT"                             86.99 KB    2687행이 임포트됨
. . "LOAN"."TIDCK"                               71.39 KB    1500행이 임포트됨
```

...

```
. . "LOAN"."TAGENCY"                             5.617 KB       7행이 임포트됨
. . "LOAN"."REGIONS"                             5.546 KB       4행이 임포트됨
객체 유형 SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/COMMENT 처리 중
객체 유형 SCHEMA_EXPORT/VIEW/VIEW 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/INDEX/INDEX 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT 처리 중
객체 유형 SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS 처리 중
"SYSTEM"."SYS_IMPORT_FULL_01" 작업이 수 1월 5 10:12:24 2022 elapsed 0 00:01:12에서 성공적으로 완
료됨


SQL> exit
```

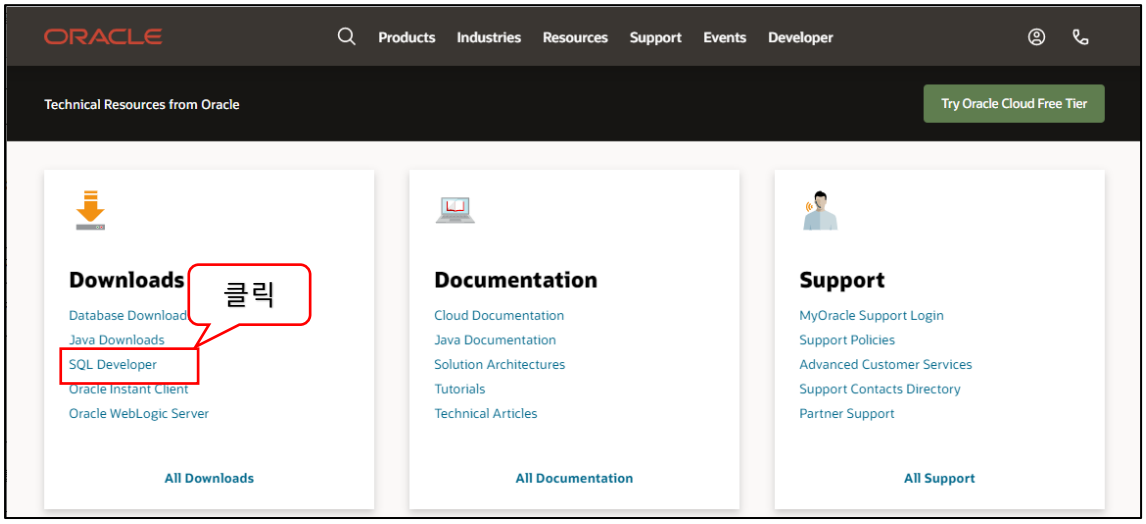
데이터 임포트가 모두 끝났습니다. sqlplus 에서도 실습을 진행할 수 있지만 작업이 불편할 수 있어서 SQL Developer 를 준비하겠습니다.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

SQL Developer 구성 (Client Tool)

앞에서 설치한 것은 Oracle Database Server 입니다. Server 제품이 설치되면 CLI(Command Line Interface)를 이용한 실습도 가능하지만 불편한 것이 사실이고, 실제 업무에서도 GUI (Graphical User Interface) 환경의 툴을 많이 사용합니다. 여러 제품들이 있지만 무료로 사용할 수 있는 SQL Developer 를 이용하겠습니다.

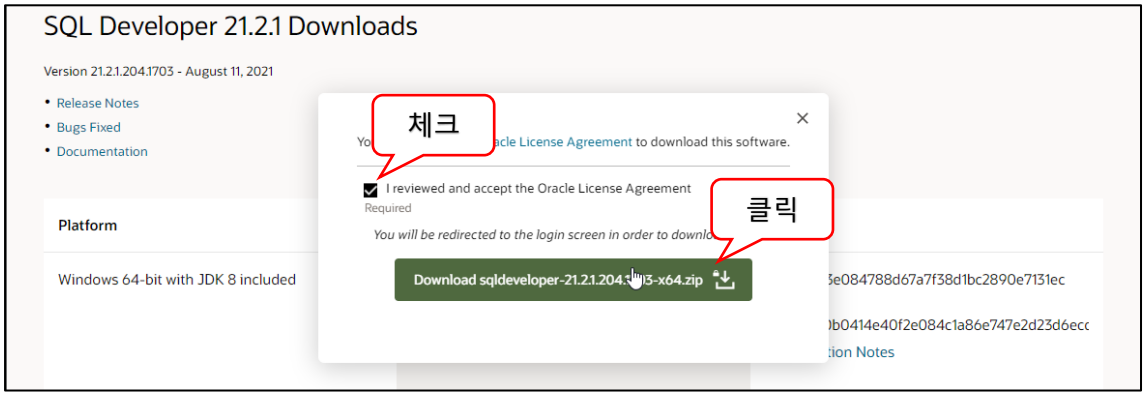
우선 필요한 제품을 다운받아야 하기 때문에 http://otn.oracle.com 사이트에 다시 접속합니다. 그리고 Downloads 항목에 SQL Developer 링크를 클릭합니다.



최신 버전의 SQL Developer 가 다양한 플랫폼 별로 제공되며 여기서는 Windows 64-bit with JDK8 included 를 다운받겠습니다. 만약 사용하는 OS 가 다른 플랫폼을 이용중이라면 해당 플랫폼에 맞는 제품을 다운 받으면 됩니다. 단, Windows 64bit 용을 제외한 나머지 제품은 JDK 가 포함되지 않아서 JDK 설치를 먼저 진행하고 SQL Developer 를 설치해야 합니다. 여기서는 Windows 64bit 환경에 설치를 진행하기 때문에 JDK 설치는 필요하지 않습니다. (교재의 버전보다 더 최신 버전을 다운 받아도 됩니다.)

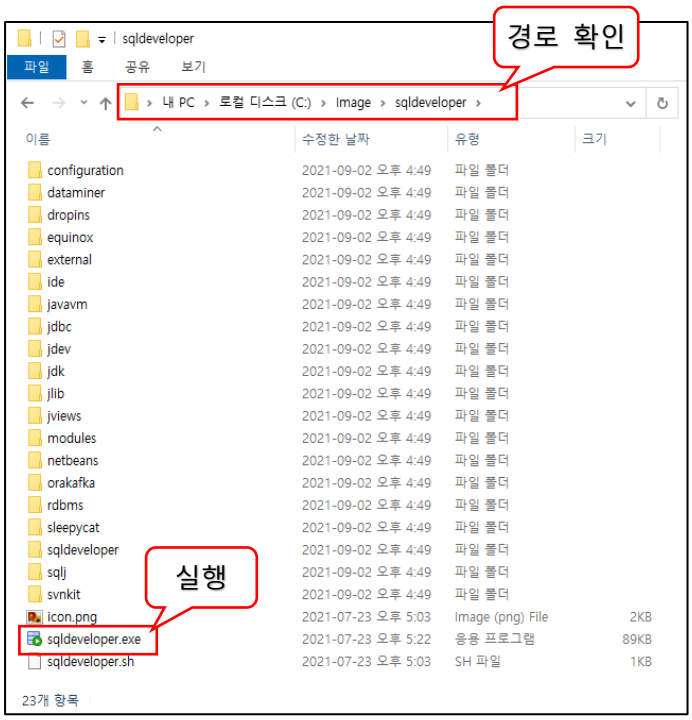
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

I reviewed and accept the Oracle License Agreement 를 체크하고, Download 버튼을 클릭합니다.



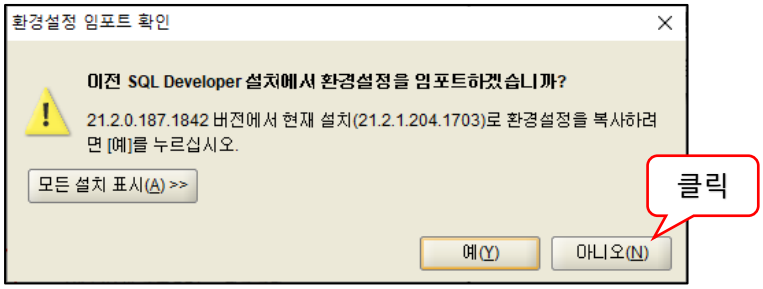
※ Oracle 사이트에 로그인이 필요하면 앞서 생성한 계정으로 로그인을 합니다.

다운받은 파일은 한글이 포함되지 않은 경로에 압축을 해제하고, 폴더안의 sqldeveloper.exe 파일을 실행합니다.



SQL Developer 는 Install 버전으로 제공되지 않고, 압축을 해제 후 실행하면 되는 제품입니다. 그리고 실행 도중 추가적인 환경 설정 파일들이 C 드라이브에 생성되는데, 간혹 한글이 섞인 경로를 이용할 때 정상적인 실행이 안되는 경우가 있습니다. 그리고 Windows 유저명이 한글을 사용할 때도 문제가 발생할 수 있으니 SQL Developer 가 실행되지 않는다면 압축 해제 경로 및 Windows 의 유저명을 영문으로 변경하고 다시 실행해 보세요.

SQL Developer 를 처음 실행하면 이전 환경설정을 임포트 여부를 물어봅니다. 우리는 이전 버전이 없으므로 '아니오'를 클릭합니다.

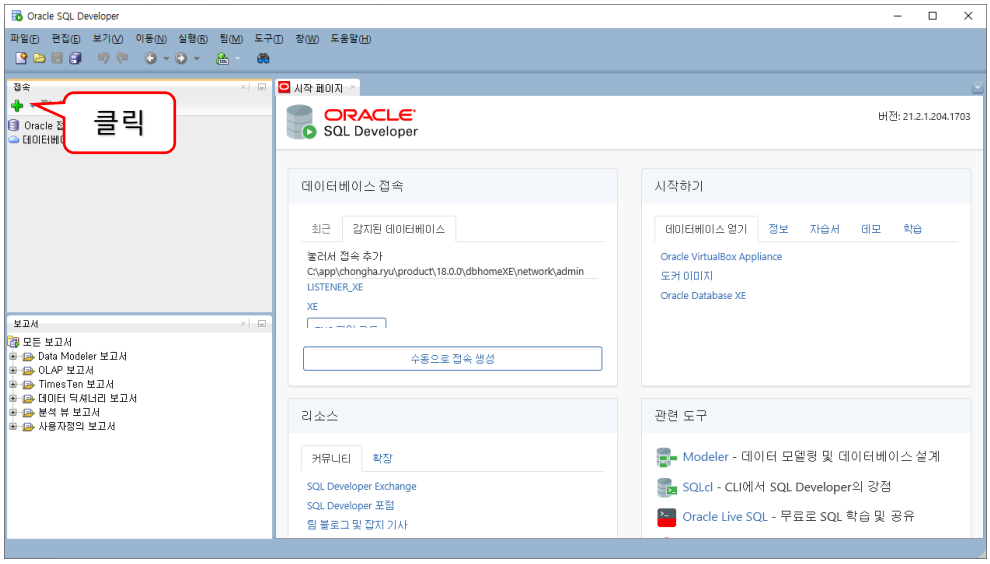


실행된 SQL Developer 가 보이나요? 그렇다면 DB 에 접속해서 실습 환경을 구성하겠습니다.

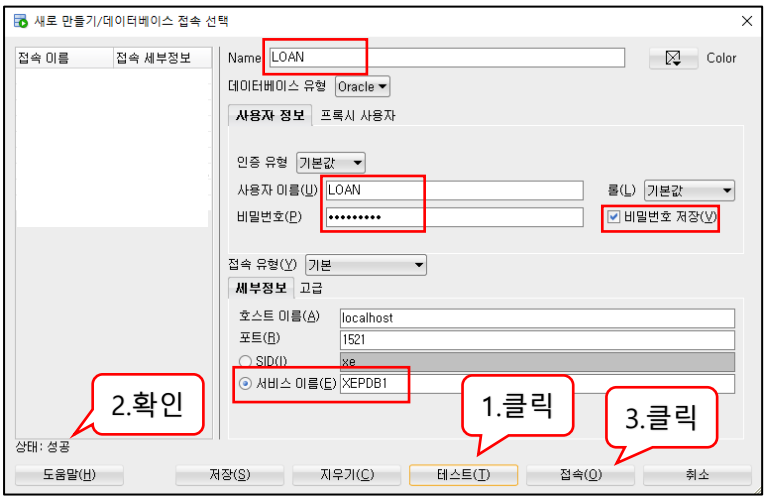
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Database 접속

이제 데이터베이스에 접속해서 실습 환경을 구성하겠습니다. 우선 데이터베이스에 관리자 계정으로 접속하기 위해 접속 정보를 생성합니다. 왼쪽 상단의 ✚ 버튼을 클릭합니다.



표시된 정보를 입력하고, 테스트 버튼을 클릭해서 상태가 '성공'이면 접속 버튼을 클릭합니다.



항 목	입 력 값
Name	LOAN
사용자 이름	LOAN
비밀번호	oracle_4U
비밀번호 저장	체크
호스트 이름	localhost
포트	1521
서비스 이름	XEPDB1

포트번호는 앞서 확인했던 번호를 이용합니다.

※ 비밀번호가 oracle 이 아닌 경우 DB 설치 시 지정한 비밀번호를 입력하면 되며, 접속 버튼이 보이지 않는다면 현재 열린 창이 작아서 그렇습니다. 우측 하단의 모서리를 늘려서 창 크기를 크게 조절하시면 숨어있던 접속 버튼이 나옵니다.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Oracle Cloud DB 접속

SQL Developer를 이용해서 Oracle Cloud DB에 접속할 수 있습니다.

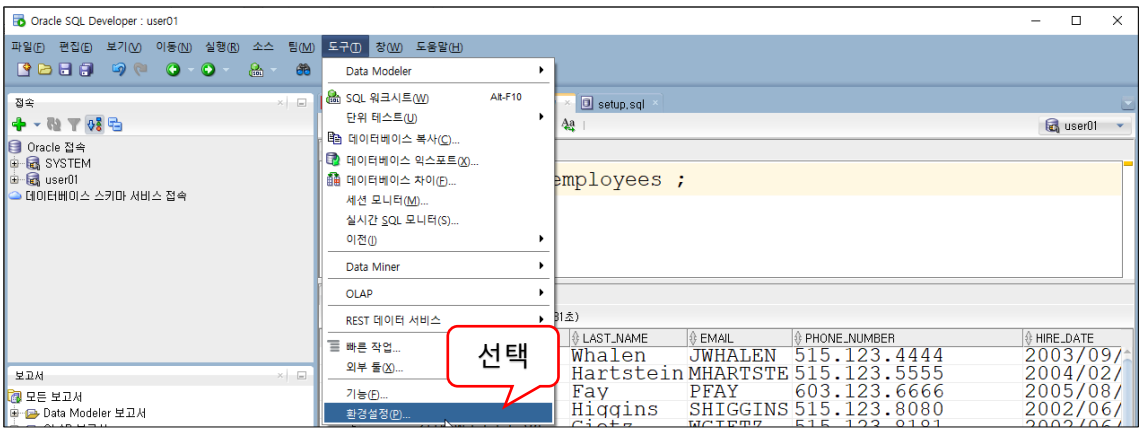
SQL Developer를 실행하고 ✚ 아이콘을 클릭합니다.

강사가 제공하는 사용자 이름과 비밀번호를 입력하고, 접속 유형을 클라우드 전자 지갑으로 변경합니다. 그리고 구성 파일은 제공받은 전자 지갑 파일을 선택해주면 됩니다. (압축 파일을 그대로 선택합니다.) 테스트 클릭 후 상태가 성공이면 접속을 클릭합니다.

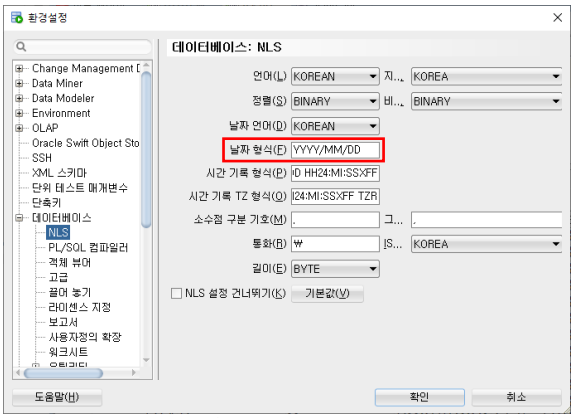
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

SQL Developer 환경 설정

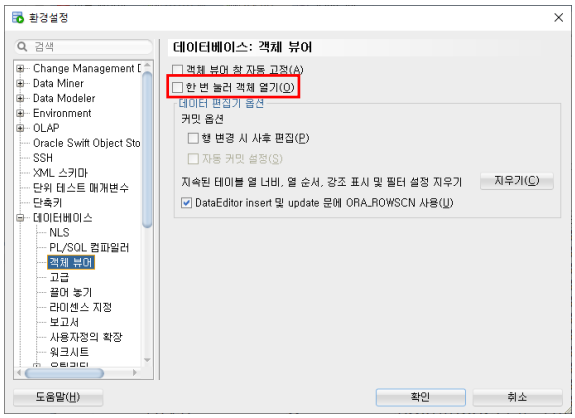
SQL Developer 의 도구 - 환경설정을 클릭합니다.



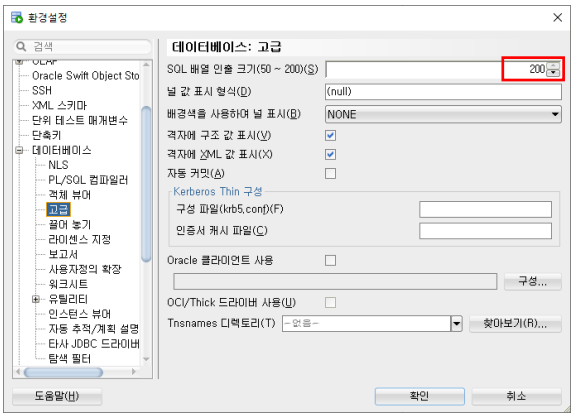
데이터베이스 항목을 확장하고 NLS, 객체 뷰어, 고급 설정을 아래와 같이 진행합니다.



NLS: 날짜 형식 => YYYY/MM/DD



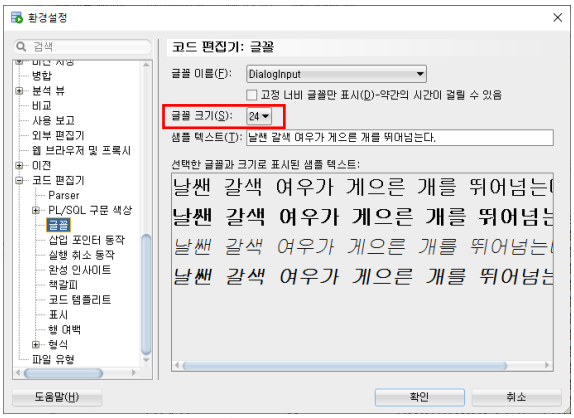
객체 뷰어: 한 번 눌러 객체 열기 체크 해제



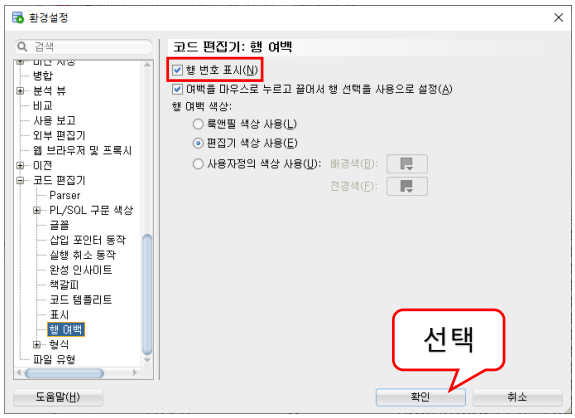
고급: SQL 배열 인출 크기 => 200

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

코드 편집기를 선택하고 글꼴, 행 여백 설정을 아래와 같이 진행합니다.



글꼴: 글꼴 크기 => 18 ~ 24 사이 지정



행 여백: 행 번호 표시 체크

모든 설정이 마무리됐으면 확인 버튼을 클릭합니다. 사용하던 SQL Developer 를 종료하고 다시 실행해야 할 때는 SQL Developer 를 압축 해제했던 폴더로 이동해서 sqldeveloper.exe 파일을 다시 실행하면 됩니다.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

2 SQL SELECT 기본 문장 작성

기본 SELECT 명령문

```
SELECT  *|{[DISTINCT] column [alias],...}
FROM    table;
```

- SELECT절은 표시할 열을 식별합니다.
- FROM절은 테이블을 식별합니다.
- SQL 문은 대소문자를 구분하지 않습니다.
- SQL 문은 여러 줄에 작성될 수 있습니다.
- SQL 문은 세미콜론(;)으로 종료됩니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

기본 SELECT 명령문

```
SELECT *
FROM   tid ;
```

SQL | 인출된 모든 행: 100(0.004초)

	LNID	ID_TYP	BTHDAY	GENDER	SCORE	GRADE
1	10000	2	2012/08/25	(null)	(null)	AA
2	10001	2	2015/06/14	(null)	(null)	A
3	10002	1	1997/08/09	1	970	(null)
4	10003	2	2009/10/12	(null)	(null)	AA-
5	10004	1	1962/09/30	1	920	(null)

```
SELECT lnid, id_typ, bthday
FROM   tid ;
```

SQL | 인출된 모든 행: 100(0.005초)

	LNID	ID_TYP	BTHDAY
1	10000	2	2012/08/25
2	10001	2	2015/06/14
3	10002	1	1997/08/09
4	10003	2	2009/10/12
5	10004	1	1962/09/30
6	10005	2	1991/07/16

열 머리글: 대문자

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

산술식

•산술 연산자를 사용하여 숫자 및 날짜 데이터로 표현식을 작성합니다.

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기

```
SELECT lnid, ln_dt, ln_amt
       ,ln_amt / 1000000, (ln_amt*0.0325)/12
FROM   tacct ;
```

SQL | 인출된 모든 행: 212(0.007초)

	LNID	LN_DT	LN_AMT	LN_AMT/1000000	(LN_AMT*0.0325)/12
1	10000	2016/02/13	170000000	170	460416.6666666666666666666666666666667
2	10000	2016/02/13	170000000	170	460416.6666666666666666666666666666667
3	10001	2020/09/17	60000000	60	162500
4	10001	2020/09/17	60000000	60	162500
5	10002	2021/06/23	550000000	550	1489583.3333333333333333333333333333333

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

날짜 연산자 사용

```
SELECT lnid, ln_amt, exp_dt, exp_dt - 14
FROM   tacct ;
```

SQL | 200개의 행이 인출됨(0.005초)

	LNID	LN_AMT	EXP_DT	EXP_DT-14
1	10000	170000000	2035/02/12	2035/01/29
2	10000	170000000	2035/02/12	2035/01/29
3	10001	60000000	2028/09/16	2028/09/02
4	10001	60000000	2028/09/16	2028/09/02
5	10002	550000000	2046/06/22	2046/06/08
6	10002	550000000	2046/06/22	2046/06/08
7	10003	650000000	2034/10/24	2034/10/10
8	10003	650000000	2034/10/24	2034/10/10
9	10004	700000000	2047/09/19	2047/09/05
10	10004	390000000	2047/09/19	2047/09/05

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

연결 연산자

- 연결 연산자:
  - 두 개의 세로선(||)을 사용합니다.
  - 열이나 문자열을 다른 열에 연결합니다.
  - 연결된 결과는 문자로 처리됩니다.

```
SELECT lnid || ' 차주의 등급은 ' || grade || ' 입니다.'
FROM   tid
WHERE id_typ = '2' ;
```

SQL | 인출된 모든 행: 69(0.011초)

	LNID||'차주의등급은'||GRADE||'입니다.'
1	10000 차주의 등급은 AA 입니다.
2	10001 차주의 등급은 A 입니다.
3	10003 차주의 등급은 AA- 입니다.
4	10005 차주의 등급은 A+ 입니다.
5	10006 차주의 등급은 BBB- 입니다.
6	10008 차주의 등급은 BBB+ 입니다.
7	10009 차주의 등급은 BBB- 입니다.
8	10011 차주의 등급은 BBB+ 입니다.
9	10014 차주의 등급은 A- 입니다.
10	10015 차주의 등급은 A+ 입니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Null 값 정의

- Null은 값을 가지고 있지 않은 상태를 의미합니다.
- Null은 공백문자와 다릅니다.

```
SELECT *
  FROM tid ;
```

작업이 완료되었습니다.(0.053초)

LNID	ID_TYP	BTHDAY	GENDER	SCORE	GRAD
10000	2	2012/08/25			AA
10001	2	2015/06/14			A
10002	1	1997/08/09	1	970	
10003	2	2009/10/12			AA-
10004	1	1962/09/30	1	920	
10005	2	1991/07/16			A+
10006	2	2002/02/27			BBB-
10007	1	1997/01/19	2	930	
10008	2	1989/09/19			BBB+
10009	2	1983/02/22			BBB-
10010	1	1974/02/14	1	840	

F5 실행

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

열 alias 정의

- 열 alias:
  - 열 머리글의 이름을 바꿉니다.
  - 열 이름 바로 뒤에 나옵니다. 열 이름과 alias 사이에 선택 사항인 AS 키워드가 올 수도 있습니다.
  - 공백이나 특수 문자를 포함하거나 대소문자를 구분하는 경우 큰따옴표가 필요합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

열 alias 사용

```
SELECT lnid   AS 차주번호, id_typ AS 차주구분코드
      ,bthday AS 생일,  gender AS 성별
  FROM tid ;
```

SQL | 인출된 모든 행: 100(0.003초)

	차...	차주구분코드	생일	성별
1	10000	2	2012/08/25	(null)
2	10001	2	2015/06/14	(null)
3	10002	1	1997/08/09	1
4	10003	2	2009/10/12	(null)
5	10004	1	1962/09/30	1

```
SELECT lnact, lnact_seq, ln_amt,
       ln_amt/1000 "대출금(천원)"
  FROM tacct ;
```

SQL | 200개의 행이 인출됨(0.007초)

	LNACT	LNACT_SEQ	LN_AMT	대출금(천원)
1	200000	00	170000000	170000
2	200000	01	170000000	170000
3	200001	00	60000000	60000
4	200001	01	60000000	60000
5	200002	00	550000000	550000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

중복 행

- query 결과에는 중복 행을 포함한 모든 행이 표시됩니다.
- 중복행을 제거하려면 DISTINCT를 사용합니다.

1

```
SELECT grade
  FROM tid ;
```

SQL | 인출된 모든 행: 100(0.003초)

	GRADE
1	AA
2	A
3	(null)
4	AA-
5	(null)
6	A+
7	BBB-
8	(null)
9	BBB+
10	BBB-
11	(null)
12	BBB+
13	(null)

2

```
SELECT DISTINCT grade
  FROM tid ;
```

SQL | 인출된 모든 행: 13(0.003초)

	GRADE
1	AA
2	A
3	(null)
4	AA-
5	A+
6	BBB-
7	BBB+
8	A-
9	AA+
10	BBB
11	BB
12	BB-
13	BB+

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

3 WHERE 절을 이용한 행 제한

선택되는 행 제한

```
 SELECT  *|{[DISTINCT] column [alias],...}
 FROM    table
[WHERE logical expression(s)];
```

- WHERE 절을 사용하여 반환되는 행을 제한합니다.
- WHERE 절은 FROM 절 다음에 나옵니다.
- 조건식은 TRUE 또는 FALSE 평가가 가능하도록 정의합니다.
- 조건식은 하나 이상 사용할 수 있습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

조건 비교

- 문자열 및 날짜 값은 작은따옴표로 묶습니다.
- 문자 값은 대소문자를 구분하고 날짜 값은 형식을 구분합니다.

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_term, ln_amt
FROM tacct
WHERE ln_amt > 2000000000 ;
```

```
SELECT *
  FROM tid
 WHERE grade = 'A';
```

```
SELECT *
  FROM tid
 WHERE bthday = '1983/06/24';
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

비교 연산자

연산자	의미
=	같음
>	보다 큼
>=	보다 크거나 같음
<	보다 작음
<=	보다 작거나 같음
<>	같지 않음
BETWEEN ...AND...	두 값 사이(경계값 포함)
IN(set)	값 리스트 중 일치하는 값 검색
LIKE	일치하는 문자 패턴 검색
IS NULL	null 값인지 여부

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

BETWEEN 연산자를 사용하는 범위 조건

- BETWEEN 연산자를 사용하여 값의 범위를 기반으로 행을 표시합니다.

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_term, ln_amt
FROM tacct
WHERE ln_amt BETWEEN  1000000000 AND 2000000000 ;
```

SQL | 인출된 모든 행: 40(0.003초)

	LNACT	LNACT_SEQ	LNID	LN_DT	LN_TERM	LN_AMT
1	200009	00	10009	2016/04/14	192	1390000000
2	200009	01	10009	2016/04/14	192	1390000000
3	200010	00	10010	2020/03/25	324	1230000000
4	200016	00	10016	2019/05/13	360	1740000000
5	200016	01	10016	2019/05/13	360	1040000000
6	200017	01	10017	2018/08/26	264	1660000000
7	200019	00	10019	2013/11/26	216	1020000000
8	200019	01	10019	2013/11/26	216	1020000000
9	200026	00	10026	2015/02/06	204	1090000000
10	200026	01	10026	2015/02/06	204	1090000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

IN 연산자 사용

- IN 연산자를 사용하여 리스트의 값을 테스트합니다.

```
SELECT lnid, id_typ, bthday, grade
FROM tid
WHERE grade IN ('AA+','AA','AA-') ;
```

SQL | 인출된 모든 행: 21(0.002초)

	LNID	ID_TYP	BTHDAY	GRADE
1	10000	2	2012/08/25	AA
2	10041	2	1994/09/18	AA
3	10077	2	2009/09/19	AA
4	10083	2	2009/04/20	AA
5	10091	2	2005/08/26	AA
6	10097	2	2003/09/24	AA
7	10098	2	1983/09/08	AA
8	10099	2	1989/04/23	AA
9	10018	2	2012/08/17	AA+
10	10022	2	2010/04/21	AA+

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

LIKE 연산자를 사용하여 패턴 일치

- LIKE 연산자를 사용하여 대체 문자 검색을 수행합니다.
- 검색 조건에는 리터럴 문자나 숫자가 포함될 수 있습니다.
  - %는 0개 이상의 문자를 나타냅니다.
  - _은 한 문자를 나타냅니다.
- 두 개의 대체 문자(%, _)를 리터럴 문자와 결합할 수 있습니다.

```
SELECT *
FROM tcode
WHERE grade LIKE 'A%' ;
```

SQL | 인출된 모든 행: 7(0.003초)

	CODE	GRADE	GRADE_DESC
1	06	A	신용상태 양호
2	05	A+	신용상태 양호
3	07	A-	신용상태 양호
4	03	AA	신용상태 우수
5	02	AA+	신용상태 우수
6	04	AA-	신용상태 우수
7	01	AAA	최상 신용상태

```
SELECT *
FROM tcode
WHERE grade LIKE 'A_' ;
```

SQL | 인출된 모든 행: 3(0.002초)

	CODE	GRADE	GRADE_DESC
1	05	A+	신용상태 양호
2	07	A-	신용상태 양호
3	03	AA	신용상태 우수

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

NULL 조건 사용

•IS NULL 연산자로 null을 검색합니다.

```
SELECT lnact, lnact_seq, acct_typ, lmt_typ,
       lnid, ln_dt, ln_amt
FROM tacct
WHERE lmt_typ IS NULL ;
```

SQL | 인출된 모든 행: 137(0.004초)

	LNACT	LNACT_SEQ	ACCT_TYP	LMT_TYP	LNID	LN_DT	LN_AMT
1	200000	01	2	(null)	10000	2016/02/13	170000000
2	200001	01	2	(null)	10001	2020/09/17	60000000
3	200002	01	2	(null)	10002	2021/06/23	550000000
4	200003	01	2	(null)	10003	2018/10/25	650000000
5	200004	01	2	(null)	10004	2020/09/20	390000000
6	200005	01	2	(null)	10005	2016/11/06	680000000
7	200006	01	2	(null)	10006	2021/06/28	460000000
8	200007	00	1	(null)	10007	2021/03/13	20000000
9	200008	01	2	(null)	10008	2018/01/01	350000000
10	200009	01	2	(null)	10009	2016/04/14	1390000000
11	200010	01	2	(null)	10010	2020/03/25	630000000
12	200011	01	2	(null)	10011	2021/06/09	710000000
13	200012	01	2	(null)	10012	2019/10/29	670000000
14	200013	01	2	(null)	10013	2017/06/03	520000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

논리 연산자를 사용하여 조건 정의

연산자	의미
AND	*두* 구성 요소 조건이 모두 참인 경우 TRUE를 반환합니다.
OR	구성 요소 중 *하나*가 참인 경우 TRUE를 반환합니다.
NOT	조건이 거짓인 경우 TRUE를 반환합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

AND 연산자 사용

- AND 연산에서는 두 구성 요소 조건이 모두 참이어야 합니다.

```
SELECT lnact, lnact_seq, acct_typ, lmt_typ,
       lnid, ln_dt, ln_amt
FROM tacct
WHERE lmt_typ IS NULL
  AND ln_dt   >= '2022/01/01';
```

SQL | 인출된 모든 행: 8(0.003초)

	LNACT	LNACT_SEQ	ACCT_TYP	LMT_TYP	LNID	LN_DT	LN_AMT
1	200030	00	1	(null)	10030	2022/04/05	240000000
2	200051	00	1	(null)	10051	2022/03/16	280000000
3	200073	01	2	(null)	10073	2022/01/19	760000000
4	200079	01	2	(null)	10079	2022/02/19	1300000000
5	200088	01	2	(null)	10088	2022/01/26	1060000000
6	200011	02	2	(null)	10011	2022/02/24	180000000
7	200016	02	2	(null)	10016	2022/02/19	700000000
8	200035	02	2	(null)	10035	2022/02/15	80000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

OR 연산자 사용

- OR 연산에서는 두 구성 요소 조건 중 하나가 참이어야 합니다.

```
SELECT lnact, lnact_seq, acct_typ, prod_cd,
       lnid, ln_dt, ln_amt
FROM tacct
WHERE prod_cd = '101'
   OR prod_cd = '102' ;
```

SQL | 인출된 모든 행: 8(0.001초)

	LNACT	LNACT_SEQ	ACCT_TYP	PROD_CD	LNID	LN_DT	LN_AMT
1	200007	00	1	102	10007	2021/03/13	20000000
2	200037	00	1	102	10037	2018/11/18	120000000
3	200045	00	1	101	10045	2018/02/10	50000000
4	200051	00	1	101	10051	2022/03/16	280000000
5	200062	00	1	102	10062	2018/05/24	140000000
6	200071	00	1	101	10071	2019/11/27	90000000
7	200076	00	1	101	10076	2019/10/19	240000000
8	200090	00	1	101	10090	2020/10/12	210000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

NOT 연산자 사용

```
SELECT lnid, id_typ, bthday, grade
FROM tid
WHERE grade NOT IN ('AA+','AA','AA-') ;
```

SQL | 인출된 모든 행: 48(0.002초)

	LNID	ID_TYP	BTHDAY	GRADE
1	10001	2	2015/06/14	A
2	10005	2	1991/07/16	A+
3	10006	2	2002/02/27	BBB-
4	10008	2	1989/09/19	BBB+
5	10009	2	1983/02/22	BBB-
6	10011	2	1991/08/14	BBB+
7	10014	2	1990/10/18	A-
8	10015	2	1990/11/27	A+
9	10019	2	2006/02/25	BBB
10	10023	2	1994/06/19	BB
11	10024	2	1993/03/15	BB
12	10025	2	1986/11/29	A+
13	10026	2	2005/07/31	BBB+
14	10027	2	1995/04/22	BBB-
15	10028	2	2005/06/04	A+

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

우선 순위 규칙

```
SELECT lnact, lnact_seq, prod_cd, lnid, ln_dt, ln_amt
FROM tacct
WHERE ln_dt   >= '2021/01/01'
  AND prod_cd = '101'
   OR prod_cd = '102' ;
```

1

SQL | 인출된 모든 행: 4(0.002초)

	LNACT	LNACT_SEQ	PROD_CD	LNID	LN_DT	LN_AMT
1	200007	00	102	10007	2021/03/13	20000000
2	200037	00	102	10037	2018/11/18	120000000
3	200051	00	101	10051	2022/03/16	280000000
4	200062	00	102	10062	2018/05/24	140000000

```
SELECT lnact, lnact_seq, prod_cd, lnid, ln_dt, ln_amt
FROM tacct
WHERE ln_dt   >= '2021/01/01'
  AND (prod_cd = '101'
   OR  prod_cd = '102') ;
```

2

SQL | 인출된 모든 행: 2(0초)

	LNACT	LNACT_SEQ	PROD_CD	LNID	LN_DT	LN_AMT
1	200007	00	102	10007	2021/03/13	20000000
2	200051	00	101	10051	2022/03/16	280000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

4 ORDER BY절을 이용한 정렬

ORDER BY 절 사용

```
 SELECT  *|{[DISTINCT] column [alias],...}
 FROM    table
[WHERE logical expression(s)]
[ORDER BY col_name|col_alias|col_position [ASC|DESC, ....];
```

- ORDER BY 절을 사용하여 검색된 행을 정렬합니다.
- SELECT 명령문의 가장 마지막 절로 사용합니다.
- 하나 이상의 컬럼 이름, 별칭, 순서를 이용하여 정렬합니다.
- ASC : 오름차순, 기본값
- DESC: 내림차순

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

정렬

- 정렬 수행

```
SELECT *
FROM tid
ORDER BY bthday ;
```

SQL | 인출된 모든 행: 100(0.003초)

	LNID	ID_TYP	BTHDAY	GENDER	SCORE	GRADE
1	10012	1	1952/06/13	2	990	(null)
2	10066	1	1956/07/22	2	960	(null)
3	10053	1	1956/10/27	2	660	(null)
4	10039	1	1956/11/20	1	640	(null)
5	10013	1	1959/02/13	2	690	(null)

- 내림차순으로 정렬:

```
SELECT *
FROM tid
ORDER BY bthday DESC ;
```

SQL | 인출된 모든 행: 100(0.005초)

	LNID	ID_TYP	BTHDAY	GENDER	SCORE	GRADE
1	10062	2	2017/11/19	(null)	(null)	A-
2	10078	2	2017/10/01	(null)	(null)	A-
3	10088	2	2016/05/15	(null)	(null)	AA-
4	10001	2	2015/06/14	(null)	(null)	A
5	10079	2	2015/02/03	(null)	(null)	A

- 열 alias를 기준으로 정렬:

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_amt AS 대출금
FROM tacct
WHERE lmt_typ IS NULL
ORDER BY 대출금 ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

정렬

- 열의 위치를 사용하여 정렬

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_amt AS 대출금
FROM tacct
WHERE lmt_typ IS NULL
ORDER BY 3 ;
```

- 여러 열을 기준으로 정렬

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_amt AS 대출금
FROM tacct
WHERE lmt_typ IS NULL
ORDER BY 5, 4 DESC ;
```

SQL | 인출된 모든 행: 137(0.003초)

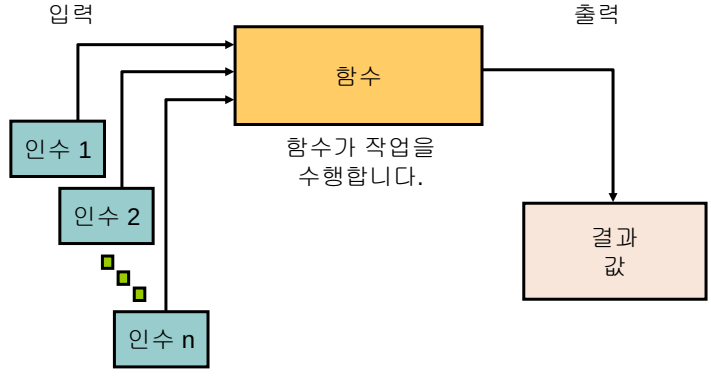
	LNACT	LNACT_SEQ	LNID	LN_DT	대출금
1	200082	00	10082	2020/04/05	19000000
2	200007	00	10007	2021/03/13	20000000
3	200047	01	10047	2015/06/09	20000000
4	200052	00	10052	2017/11/12	39000000
5	200072	00	10072	2018/11/27	50000000
6	200045	00	10045	2018/02/10	50000000
7	200001	01	10001	2020/09/17	60000000
8	200018	00	10018	2018/08/23	60000000
9	200029	00	10029	2018/01/04	60000000
10	200020	02	10020	2021/07/29	70000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

5 단일 행 함수 활용

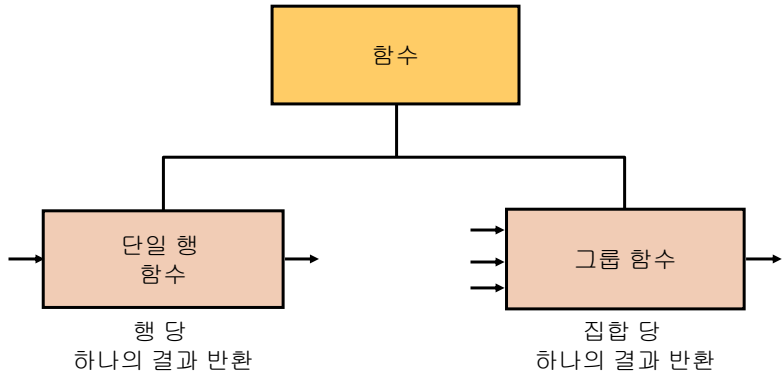
SQL 함수



ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

SQL 함수의 두 가지 유형



ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

단일 행 함수

- 단일 행 함수:
  - 데이터 항목을 조작합니다.
  - 인수를 사용하고 하나의 값을 반환합니다.
  - 반환되는 각 행에서 실행됩니다.
  - 행당 하나의 결과를 반환합니다.
  - 중첩될 수 있습니다.

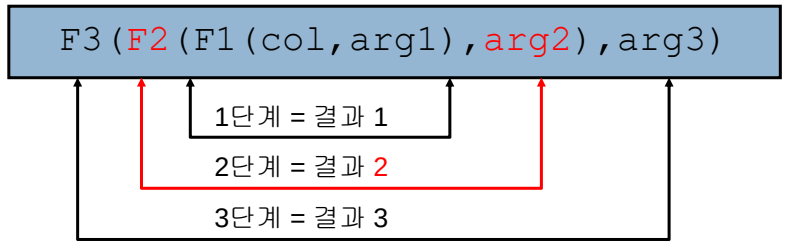
```
function_name [(arg1, arg2,...)]
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

함수 중첩

- 단일 행 함수는 어떠한 레벨로도 중첩될 수 있습니다.
- 중첩 함수는 가장 깊은 레벨에서 덜 깊은 레벨로 평가됩니다.



ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

문자 함수

- 다음은 문자열의 대소문자를 변환하는 함수입니다.

함수	결과
LOWER('SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP('SQL Course')	Sql Course

- 다음은 문자열을 조작하는 함수입니다.

함수	결과
SUBSTR('HelloWorld',1,5)	Hello
LENGTH('HelloWorld')	10
INSTR('HelloWorld', 'W')	6

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

숫자 함수

- ROUND: 지정된 소수점 자릿수로 값을 반올림합니다.
- TRUNC: 지정된 소수점 자릿수로 값을 truncate합니다.

함수	결과
ROUND(45.926, 2)	45.93
TRUNC(45.926, 2)	45.92

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

날짜 조작 함수

함수	결과
MONTHS_BETWEEN	두 날짜 간의 월 수
ADD_MONTHS	날짜에 월 추가
LAST_DAY	월의 마지막 날

함수	결과
MONTHS_BETWEEN ('2005/09/01','2004/01/11')	19.6774194
ADD_MONTHS ('2004/01/31',1)	'2004/02/29'
LAST_DAY ('2005/02/01')	'2005/02/28'

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

날짜에 TO_CHAR 함수 사용

```
TO_CHAR(date[,'format_model'])
```

- 형식 모델:
  - 작은따옴표로 묶어야 합니다.
  - 대소문자를 구분합니다.
  - 유효한 날짜 형식 요소를 포함할 수 있습니다.
  - 쉼표로 날짜 값과 구분됩니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

날짜 형식 모델의 요소

요소	결과
YYYY	숫자로 된 전체 연도
YEAR	영어 철자로 표기된 연도
MM	월의 2자리 값
MONTH	전체 월 이름
MON	월의 3자 약어
DY	요일의 3자 약어
DAY	요일의 전체 이름
DD	숫자 형식의 월간 일

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

날짜에 TO_CHAR 함수 사용

```
SELECT lnid, gender, bthday
       ,TO_CHAR(bthday,'MM/DD')
       ,TO_CHAR(bthday,'DD Month, YYYY')
FROM tid
WHERE id_typ = '1' ;
```

SQL | 인출된 모든 행: 31(0.002초)

	LNID	GENDER	BTHDAY	TO_CHAR(BTHDAY,'MM/DD')	TO_CHAR(BTHDAY,'DDMONTH,YYYY')
1	10002	1	1997/08/09	08/09	09 8월 , 1997
2	10004	1	1962/09/30	09/30	30 9월 , 1962
3	10007	2	1997/01/19	01/19	19 1월 , 1997
4	10010	1	1974/02/14	02/14	14 2월 , 1974
5	10012	2	1952/06/13	06/13	13 6월 , 1952
6	10013	2	1959/02/13	02/13	13 2월 , 1959
7	10016	1	1983/06/24	06/24	24 6월 , 1983
8	10017	2	1971/11/20	11/20	20 11월, 1971
9	10020	2	1995/12/24	12/24	24 12월, 1995
10	10031	2	1975/08/18	08/18	18 8월 , 1975

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

숫자에 TO_CHAR 함수 사용

```
TO_CHAR(number[, 'format_model'])
```

•다음은 TO_CHAR 함수와 함께 사용하여 숫자 값을 문자로 표시할 수 있는 몇 가지 형식 요소입니다.

요소	결과
9	숫자를 나타냄
0	0이 표시되도록 강제 적용
$	부동 달러 기호 배치
L	부동 로컬 통화 기호 사용
.	소수점 출력
,	천 단위 표시자로 쉼표 출력

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

숫자에 TO_CHAR 함수 사용

```
SELECT lnact, lnact_seq, lnid, ln_dt
      ,TO_CHAR(ln_amt,'L999,999,999,999')
      ,TO_CHAR(ln_amt,'L000,000,999,999')
FROM tacct
WHERE lmt_typ IS NULL ;
```

SQL | 인출된 모든 행: 137(0.005초)

	LNACT	LNACT_SEQ	LN_DT	TO_CHAR(LN_AMT,'L999,999,999,999')	TO_CHAR(LN_AMT,'L000,000,999,999')
1	200000	01	2016/02/13	₩170,000,000	₩000,170,000,000
2	200001	01	2020/09/17	₩60,000,000	₩000,060,000,000
3	200002	01	2021/06/23	₩550,000,000	₩000,550,000,000
4	200003	01	2018/10/25	₩650,000,000	₩000,650,000,000
5	200004	01	2020/09/20	₩390,000,000	₩000,390,000,000
6	200005	01	2016/11/06	₩680,000,000	₩000,680,000,000
7	200006	01	2021/06/28	₩460,000,000	₩000,460,000,000
8	200007	00	2021/03/13	₩20,000,000	₩000,020,000,000
9	200008	01	2018/01/01	₩350,000,000	₩000,350,000,000
10	200009	01	2016/04/14	₩1,390,000,000	₩001,390,000,000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

일반 함수: NVL

```
SELECT lnact, lnact_seq, acct_typ, lnid, ln_amt, rate
      ,TRUNC((ln_amt * rate) / 12)          AS 이자1
      ,TRUNC((ln_amt * NVL(rate,0)) / 12) AS 이자2
FROM tacct
WHERE branch = '10';
```

SQL | 인출된 모든 행: 4(0.002초)

	LNACT	LNACT_SEQ	ACCT_TYP	LNID	LN_AMT	RATE	이자1	이자2
1	200022	00	1	10022	140000000	0.0387	451500	451500
2	200041	00	2	10041	580000000	(null)	(null)	0
3	200041	01	2	10041	580000000	0.0393	1899500	1899500
4	200086	00	1	10086	170000000	0.0188	266333	266333

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

CASE 식

•IF-THEN-ELSE 문 작업을 수행하여 조건부 조회를 편리하게 수행하도록 합니다.

```
CASE expr WHEN comparison_expr1 THEN return_expr1
         [WHEN comparison_expr2 THEN return_expr2
          WHEN comparison_exprn THEN return_exprn
          ELSE else_expr]
END
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

CASE 식 사용

```
SELECT lnid, bthday, score, gender,
       CASE gender WHEN '1' THEN '남'
                   WHEN '2' THEN '여'
                   ELSE '법인'
       END AS 성별
FROM tid
WHERE id_typ = '1';
```

SQL | 인출된 모든 행: 31(0.003초)

	LNID	BTHDAY	SCORE	GENDER	성별
1	10002	1997/08/09	97	0 1	남
2	10004	1962/09/30	92	0 1	남
3	10007	1997/01/19	93	0 2	여
4	10010	1974/02/14	84	0 1	남
5	10012	1952/06/13	99	0 2	여
6	10013	1959/02/13	69	0 2	여
7	10016	1983/06/24	99	0 1	남
8	10017	1971/11/20	66	0 2	여
9	10020	1995/12/24	73	0 2	여
10	10031	1975/08/18	92	0 2	여
11	10035	1977/11/13	99	0 2	여
12	10039	1956/11/20	64	0 1	남
13	10040	1993/05/24	65	0 2	여

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

검색된 CASE 표현식

```
CASE
   WHEN condition1 THEN use_expression1
   WHEN condition2 THEN use_expression2
   WHEN condition3 THEN use_expression3
   ELSE default_use_expression
END
```

```
SELECT lnact, lnact_seq, ln_amt, rate, rate_typ,
       CASE WHEN rate_typ = '1'
                  THEN TRUNC((ln_amt * rate)/12)
            WHEN rate_typ = '2'
                  THEN TRUNC((ln_amt * (rate+0.01))/12)
            ELSE 0
       END AS 이자
FROM tacct
WHERE lmt_typ IS NULL ;
```

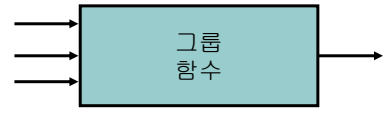
ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

6 GROUP BY 절을 이용한 그룹 생성

그룹 함수 유형

- AVG
- COUNT
- MAX
- MIN
- SUM



ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

AVG 및 SUM 함수 사용

•숫자 데이터에 대해 AVG 및 SUM 함수를 사용할 수 있습니다.

```
SELECT SUM(ln_amt), AVG(ln_amt),
       MAX(ln_amt), MIN(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL  ;
```

SQL | 인출된 모든 행: 1(0.001초)

	SUM(LN_AMT)	AVG(LN_AMT)	MAX(LN_AMT)	MIN(LN_AMT)
1	74128000000	541080291.9708029197080291970803	2740000000	19000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

MIN 및 MAX 함수 사용

• 숫자, 문자 및 날짜 데이터 유형에 대해 MIN 및 MAX 함수를 사용할 수 있습니다.

```
SELECT MAX(ln_dt), MIN(ln_dt)
FROM tacct
WHERE lmt_typ IS NULL  ;
```

SQL | 인출된 모든 행: 1(0.003초)

	MAX(LN_DT)	MIN(LN_DT)
1	2022/04/05	2012/05/26

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

COUNT 함수 사용

•COUNT(*)는 테이블의 행 수를 반환합니다.

1
```
SELECT COUNT(*)
FROM tid ;
```

SQL | 인출된 모든 행: 1(0.001초)

	COUNT(*)
1	100

•COUNT(*expr*)은 *expr*에 대해 null이 아닌 값을 가진 행의 수를 반환합니다.

2
```
SELECT COUNT(gender)
FROM tid ;
```

SQL | 인출된 모든 행: 1(0.002초)

	COUNT(GENDER)
1	31

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

DISTINCT 키워드 사용

• COUNT(DISTINCT expr)은 *expr*의 null이 아닌 구분 값의 수를 반환합니다.

```
SELECT COUNT(DISTINCT gender)
FROM tid ;
```

SQL | 인출된 모든 행: 1(0.002초)

	COUNT(DISTINCTGENDER)
1	2

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

데이터 그룹 생성: GROUP BY 절 구문

- GROUP BY 절을 사용하여 테이블의 행을 더 작은 그룹으로 나눌 수 있습니다.

```
SELECT     column, group_function(column)
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[ORDER BY  column];
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

GROUP BY 절 사용

- 그룹 함수에 속하지 않는 SELECT list의 모든 열은 GROUP BY 절에 있어야 합니다.

```
SELECT branch, SUM(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL
GROUP BY branch;
```

SQL | 인출된 모든 행: 39(0.011초)

	BRANCH	SUM(LN_AMT)
1	46	4110000000
2	28	1280000000
3	11	2430000000
4	20	5230000000
5	26	3330000000
6	24	3500000000
7	29	1500000000
8	44	3070000000
9	39	2040000000
10	38	3700000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

GROUP BY 절 사용

•GROUP BY 열은 SELECT list에 없어도 됩니다.

```
SELECT SUM(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL
GROUP BY branch;
```

SQL | 인출된 모든 행: 39(0.003초)

	SUM(LN_AMT)
1	4110000000
2	1280000000
3	2430000000
4	5230000000
5	3330000000
6	3500000000
7	1500000000
8	3070000000
9	2040000000
10	3700000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

다중 열에서 GROUP BY 절 사용

```
SELECT branch, prod_cd, SUM(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL
GROUP BY branch, prod_cd ;
```

SQL | 인출된 모든 행: 122(0.016초)

	BRANCH	PROD_CD	SUM(LN_AMT)
1	46	201	170000000
2	28	202	60000000
3	11	209	550000000
4	20	209	1690000000
5	28	204	390000000
6	26	200	1400000000
7	24	205	460000000
8	29	102	20000000
9	28	201	350000000
10	20	208	1390000000
11	44	210	720000000
12	46	206	1480000000
13	46	210	2060000000
14	39	203	520000000
15	38	201	750000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

그룹 함수를 사용한 잘못된 query

•집계 함수가 아닌 SELECT list의 열이나 표현식은 GROUP BY 절에 있어야 합니다.

```
SELECT branch, prod_cd, SUM(ln_amt)
FROM tacct ;
```

```
ORA-00937: not a single-group group function
00937. 00000 - "not a single-group group function"
```

```
SELECT branch, prod_cd, SUM(ln_amt)
FROM tacct
GROUP BY branch ;
```

```
ORA-00979: not a GROUP BY expression
00979. 00000 - "not a GROUP BY expression"
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

그룹 함수를 사용한 잘못된 query

- WHERE 절은 그룹을 제한하는 데 사용할 수 없습니다.
- 그룹을 제한하려면 HAVING 절을 사용합니다.
- WHERE 절에서 그룹 함수를 사용할 수 없습니다.

```
SELECT branch, prod_cd, SUM(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL
  AND SUM(ln_amt) > 2000000000
GROUP BY branch, prod_cd ;
```

```
ORA-00934: group function is not allowed here
00934. 00000 - "group function is not allowed here"
*Cause:
*Action:
Error at Line: 3 Column: 9
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

HAVING 절을 사용하여 그룹 결과 제한

- HAVING 절을 사용할 경우 Oracle 서버는 다음과 같이 그룹을 제한합니다.
  1. 행이 그룹화됩니다.
  2. 그룹 함수가 적용됩니다.
  3. HAVING 절과 일치하는 그룹이 표시됩니다.

```
SELECT     column, group_function
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

HAVING 절 사용

```
SELECT branch, prod_cd, SUM(ln_amt)
FROM tacct
WHERE lmt_typ IS NULL
GROUP BY branch, prod_cd
HAVING SUM(ln_amt) > 2000000000 ;
```

SQL | 인출된 모든 행: 3(0.01초)

	BRANCH	PROD_CD	SUM(LN_AMT)
1	46	210	2060000000
2	24	201	2740000000
3	14	202	2270000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

7 조인 문장 작성

다중 테이블에서 데이터 가져오기

TID

SQL | 인출된 모든 행: 100(0.014초)

	LNID	ID_TYP	BTHDAY	GENDER	SCORE	GRADE
1	10000	2	2012/08/25	(null)	(null)	AA
2	10001	2	2015/06/14	(null)	(null)	A
3	10002	1	1997/08/09	1	970	(null)
4	10003	2	2009/10/12	(null)	(null)	AA-
5	10004	1	1962/09/30	1	920	(null)
6	10005	2	1991/07/16	(null)	(null)	A+
7	10006	2	2002/02/27	(null)	(null)	BBB-
8	10007	1	1997/01/19	2	930	(null)
9	10008	2	1989/09/19	(null)	(null)	BBB+
10	10009	2	1983/02/22	(null)	(null)	BBB-

TACCT

SQL | 200개의 행이 인출됨(0.044초)

	LNACT	LNACT_SEQ	ACCT_TYP	LMT_TYP	LNID	BRANCH
1	200000	00	2	1	10000	46
2	200000	01	2	(null)	10000	46
3	200001	00	2	1	10001	28
4	200001	01	2	(null)	10001	28
5	200002	00	2	1	10002	11
6	200002	01	2	(null)	10002	11
7	200003	00	2	1	10003	20
8	200003	01	2	(null)	10003	20
9	200004	00	2	1	10004	28
10	200004	01	2	(null)	10004	28

SQL | 인출된 모든 행: 137(0.016초)

	LNID	ID_TYP	BTHDAY	LNACT	LNACT_SEQ	LN_DT	LN_AMT
1	10000	2	2012/08/25	200000	01	2016/02/13	170000000
2	10001	2	2015/06/14	200001	01	2020/09/17	60000000
3	10002	1	1997/08/09	200002	01	2021/06/23	550000000
4	10003	2	2009/10/12	200003	01	2018/10/25	650000000
5	10004	1	1962/09/30	200004	01	2020/09/20	390000000
6	10004	1	1962/09/30	200004	02	2021/05/25	310000000
7	10005	2	1991/07/16	200005	01	2016/11/06	680000000
8	10006	2	2002/02/27	200006	01	2021/06/28	460000000
9	10007	1	1997/01/19	200007	00	2021/03/13	20000000
10	10008	2	1989/09/19	200008	01	2018/01/01	350000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

ON 절을 사용하여 레코드 검색

- ON 절을 사용하여 조인할 열을 지정합니다.
- ON 절을 사용하면 코드를 이해하기 쉽습니다.

```
SELECT a.lnid, a.id_typ, a.bthday,
       b.lnact, b.lnact_seq, b.ln_dt, b.ln_amt
FROM tid   a
JOIN tacct b
  ON a.lnid = b.lnid
WHERE b.lmt_typ IS NULL
ORDER BY b.lnact, b.lnact_seq ;
```

SQL | 인출된 모든 행: 137(0.016초)

	LNID	ID_TYP	BTHDAY	LNACT	LNACT_SEQ	LN_DT	LN_AMT
1	10000	2	2012/08/25	200000	01	2016/02/13	170000000
2	10001	2	2015/06/14	200001	01	2020/09/17	60000000
3	10002	1	1997/08/09	200002	01	2021/06/23	550000000
4	10003	2	2009/10/12	200003	01	2018/10/25	650000000
5	10004	1	1962/09/30	200004	01	2020/09/20	390000000
6	10004	1	1962/09/30	200004	02	2021/05/25	310000000
7	10005	2	1991/07/16	200005	01	2016/11/06	680000000
8	10006	2	2002/02/27	200006	01	2021/06/28	460000000
9	10007	1	1997/01/19	200007	00	2021/03/13	20000000
10	10008	2	1989/09/19	200008	01	2018/01/01	350000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

3-Way 조인 생성

```
SELECT i.lnid, i.bthday, a.lnact, a.lnact_seq, a.ln_amt,
       a.ln_dt, r.pay_dt, r.sum_mon_amt, r.ln_bb_amt
FROM tid   i
JOIN tacct a
  ON i.lnid = a.lnid
JOIN trepay r
  ON a.lnact     = r.lnact
 AND a.lnact_seq = r.lnact_seq
WHERE i.lnid = '10006'
ORDER BY r.pay_dt ;
```

SQL | 인출된 모든 행: 9(0.01초)

	LNID	BTHDAY	LNACT	LNACT_SEQ	LN_AMT	LN_DT	PAY_DT	SUM_MON_AMT	LN_BB_AMT
1	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/07/25	1801009	458808491
2	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/08/25	1801009	457615403
3	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/09/25	1801009	456420734
4	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/10/25	1801009	455224482
5	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/11/25	1801009	454026645
6	10006	2002/02/27	200006	01	460000000	2021/06/28	2021/12/25	1801009	452827221
7	10006	2002/02/27	200006	01	460000000	2021/06/28	2022/01/25	1801009	451626208
8	10006	2002/02/27	200006	01	460000000	2021/06/28	2022/02/25	1801009	450423604
9	10006	2002/02/27	200006	01	460000000	2021/06/28	2022/03/25	1801009	449219406

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

OUTER Join으로 직접 일치되지 않는 레코드 반환

DEPARTMENTS

	DEPARTMENT_NAME	DEPARTMENT_ID
1	Administration	10
2	Marketing	20
3	Shipping	50
4	IT	60
5	Sales	80
6	Executive	90
7	Accounting	110
8	Contracting	190

EMPLOYEES와 Equijoin

	DEPARTMENT_ID	LAST_NAME
1	10	Whalen
2	20	Hartstein
3	20	Fay
4	110	Higgins
5	110	Gietz
6	90	King
7	90	Kochhar
8	90	De Haan
9	60	Hunold
10	60	Ernst
...		
18	80	Abel
19	80	Taylor

부서 190에는 사원이 없습니다.

사원 "Grant"에게는 부서 ID가 할당되지 않았습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

INNER Join과 OUTER Join 비교

- SQL:1999에서 일치하는 행만 반환하는 두 테이블의 조인을 INNER join이라고 합니다.
- INNER join의 결과는 물론, 왼쪽(또는 오른쪽) 테이블의 일치하지 않는 행도 반환하는 두 테이블 간의 조인을 Left(또는 Right) OUTER join이라고 합니다.
- INNER join의 결과는 물론, Left 및 Right OUTER join의 결과를 반환하는 두 테이블 간의 조인을 Full outer join이라고 합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

LEFT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e LEFT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Vargas	50	Shipping
5	Matos	50	Shipping

...

16	Kochhar	90	Executive
17	King	90	Executive
18	Gietz	110	Accounting
19	Higgins	110	Accounting
20	Grant	(null)	(null)

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

RIGHT OUTER JOIN

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e RIGHT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Hartstein	20	Marketing
3	Fay	20	Marketing
4	Davies	50	Shipping
5	Vargas	50	Shipping
6	Rajs	50	Shipping
7	Mourgos	50	Shipping
8	Matos	50	Shipping

...

18	Higgins	110	Accounting
19	Gietz	110	Accounting
20	(null)	190	Contracting

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

FULL OUTER JOIN

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	King	90	Executive
2	Kochhar	90	Executive
3	De Haan	90	Executive
4	Hunold	60	IT

...

15	Grant	(null)	(null)
16	Whalen	10	Administration
17	Hartstein	20	Marketing
18	Fay	20	Marketing
19	Higgins	110	Accounting
20	Gietz	110	Accounting
21	(null)	190	Contracting

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Cartesian Product

- Cartesian Product는 한 테이블의 모든 행을 다른 테이블의 모든 행과 조인합니다.
- Cartesian Product는 다수의 행을 생성하므로 결과는 그다지 유용하지 않습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Cartesian Product 생성

EMPLOYEES(20행)

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	200	Whalen	10
2	201	Hartstein	20
3	202	Fay	20
4	205	Higgins	110
...			
19	176	Taylor	80
20	178	Grant	(null)

DEPARTMENTS(8행)

	DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
1	10	Administration	1700
2	20	Marketing	1800
3	50	Shipping	1500
4	60	IT	1400
5	80	Sales	2500
6	90	Executive	1700
7	110	Accounting	1700
8	190	Contracting	1700

Cartesian product:
20 x 8 = 160행

	EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
1	200	10	1700
2	201	20	1700
...			
21	200	10	1800
22	201	20	1800
...			
159	176	80	1700
160	178	(null)	1700

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Cross Join 생성

- CROSS JOIN은 두 테이블의 Cartesian Product를 생성하는 JOIN 작업입니다.
- Cartesian Product를 생성하려면 SELECT 문에 CROSS JOIN을 지정합니다.

```
SELECT last_name, department_name
FROM   employees
CROSS JOIN departments ;
```

	LAST_NAME	DEPARTMENT_NAME
1	Abel	Administration
2	Davies	Administration
3	De Haan	Administration
4	Ernst	Administration
5	Fay	Administration
...		
158	Vargas	Contracting
159	Whalen	Contracting
160	Zlotkey	Contracting

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

8 Subquery 활용

Subquery를 사용하여 문제 해결

•한국기업평가에서 평가한 등급 정보를 검색하세요.

Main query:

한국기업평가의 코드를 갖는 평가 내역을 검색

Subquery:

한국기업평가의 코드는 무엇인가요?

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Subquery 구문

- subquery는 main query*전에* 실행될 수 있습니다.
- Subquery 결과는 main query에서 사용됩니다.

```
SELECT *
  FROM tcredit
 WHERE acode = (SELECT acode
                  FROM tagency
                 WHERE aname = '한국기업평가') ;
```

SQL | 인출된 모든 행: 58(0.007초)

	LNID	ACODE	START_DT	END_DT	CODE
1	10000	01	2019/05/08	2020/05/07	04
2	10001	01	2021/10/07	2023/10/06	12
3	10003	01	2020/01/13	2022/01/12	05
4	10005	01	2022/03/05	2024/03/04	05
5	10006	01	2022/01/11	2023/01/10	12
6	10008	01	2020/04/08	2022/04/07	03
7	10011	01	2021/12/04	2023/12/03	12
8	10014	01	2021/02/20	2022/02/19	05
9	10019	01	2020/04/28	2022/04/27	13
10	10021	01	2021/05/06	2023/05/05	09

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Subquery 사용 규칙과 지침

- subquery는 괄호로 묶습니다.
- 가독성을 위해 비교 조건의 오른쪽에 subquery를 배치합니다. 그러나 subquery는 비교 연산자의 양쪽 어디에나 사용할 수 있습니다.
- single-row subquery에는 단일 행 연산자를 사용하고 multiple-row subquery에는 다중 행 연산자를 사용합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Single-Row Subquery

- 한 행만 반환합니다.
- 단일 행 비교 연산자를 사용합니다.

연산자	의미
=	같음
>	보다 큼
>=	보다 크거나 같음
<	보다 작음
<=	보다 작거나 같음
<>	같지 않음

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Single-Row Subquery 실행

```
SELECT *
  FROM tcredit
 WHERE end_dt < SYSDATE
   AND acode = (SELECT acode
                  FROM tagency
                 WHERE aname = '한국기업평가') ;
```

SQL | 인출된 모든 행: 24(0.016초)

	LNID	ACODE	START_DT	END_DT	CODE
1	10000	01	2019/05/08	2020/05/07	04
2	10003	01	2020/01/13	2022/01/12	05
3	10008	01	2020/04/08	2022/04/07	03
4	10014	01	2021/02/20	2022/02/19	05
5	10023	01	2020/08/28	2021/08/27	07
6	10024	01	2020/04/14	2021/04/13	04
7	10033	01	2021/02/02	2022/02/01	03
8	10038	01	2020/01/29	2022/01/28	08
9	10044	01	2019/11/22	2021/11/21	07
10	10045	01	2019/11/13	2020/11/12	13

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Subquery에서 그룹 함수 사용

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_term, ln_amt
  FROM tacct
 WHERE lmt_typ IS NULL
   AND ln_amt > (SELECT AVG(ln_amt)
                   FROM tacct
                  WHERE lmt_typ IS NULL) ;
```

SQL | 인출된 모든 행: 55(0.009초)

	LNACT	LNACT_SEQ	LNID	LN_DT	LN_TERM	LN_AMT
1	200002	01	10002	2021/06/23	300	550000000
2	200003	01	10003	2018/10/25	192	650000000
3	200005	01	10005	2016/11/06	300	680000000
4	200009	01	10009	2016/04/14	192	1390000000
5	200010	01	10010	2020/03/25	324	630000000
6	200011	01	10011	2021/06/09	192	710000000
7	200012	01	10012	2019/10/29	228	670000000
8	200014	01	10014	2013/06/22	324	750000000
9	200016	01	10016	2019/05/13	360	1040000000
10	200017	01	10017	2018/08/26	264	1660000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Subquery가 있는 HAVING 절

- Oracle 서버는 subquery를 먼저 실행합니다.
- Oracle 서버는 main query의 HAVING 절로 결과를 반환합니다.

```
SELECT branch, TRUNC(AVG(ln_amt))
  FROM tacct
 WHERE lmt_typ IS NULL
GROUP BY branch
HAVING AVG(ln_amt) > (SELECT AVG(ln_amt)
                        FROM tacct
                       WHERE lmt_typ IS NULL) ;
```

SQL | 인출된 모든 행: 16(0.008초)

	BRANCH	TRUNC(AVG(LN_AMT))
1	20	871666666
2	23	633000000
3	14	624444444
4	33	712500000
5	30	774285714
6	24	875000000
7	36	1005000000
8	34	567500000
9	32	935000000
10	37	573333333

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

이 명령문에서 잘못된 점은?

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_term, ln_amt
  FROM tacct
 WHERE lmt_typ IS NULL
   AND lnid = (SELECT lnid
                 FROM tid
                WHERE grade = 'A') ;
```

```
ORA-01427: single-row subquery returns more than one row
01427. 00000 - "single-row subquery returns more than one row"
*Cause:
*Action:
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Multiple-Row Subquery

- 두 개 이상의 행을 반환합니다.
- 다중 행 비교 연산자를 사용합니다.

```
SELECT lnact, lnact_seq, lnid, ln_dt, ln_term, ln_amt
  FROM tacct
 WHERE lmt_typ IS NULL
   AND lnid IN (SELECT lnid
                 FROM tid
                WHERE grade = 'A') ;
```

SQL | 인출된 모든 행: 7(0.008초)

	LNACT	LNACT_SEQ	LNID	LN_DT	LN_TERM	LN_AMT
1	200001	01	10001	2020/09/17	96	60000000
2	200045	00	10045	2018/02/10	48	50000000
3	200046	01	10046	2019/09/13	348	610000000
4	200059	01	10059	2014/07/02	84	210000000
5	200079	01	10079	2022/02/19	192	1300000000
6	200085	01	10085	2021/12/16	252	880000000
7	200086	00	10086	2020/11/20	132	170000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Multiple-Column Subquery

- multiple-column subquery는 outer query에 두 개 이상의 열을 반환합니다.
- multiple-column subquery는 SELECT 문의 FROM 절에 사용될 수도 있습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Multiple-Column Subquery: 예제

```
SELECT lnact, lnact_seq, lnid, prod_cd,
       ln_dt, ln_term, ln_amt
  FROM tacct
 WHERE lmt_typ IS NULL
   AND (prod_cd, ln_amt) IN (SELECT prod_cd, MAX(ln_amt)
                               FROM tacct
                              WHERE lmt_typ IS NULL
                             GROUP BY prod_cd) ;
```

SQL | 인출된 모든 행: 21(0.007초)

	LNACT	LNACT_SEQ	LNID	PROD_CD	LN_DT	LN_TERM	LN_AMT
1	200009	01	10009	208	2016/04/14	192	1390000000
2	200097	00	10097	104	2021/12/13	132	170000000
3	200084	00	10084	107	2021/11/27	168	280000000
4	200051	00	10051	101	2022/03/16	228	280000000
5	200082	00	10082	105	2020/04/05	60	19000000
6	200095	01	10095	202	2019/06/15	180	2270000000
7	200064	01	10064	209	2021/10/22	192	1930000000
8	200017	01	10017	210	2018/08/26	264	1660000000
9	200086	00	10086	108	2020/11/20	132	170000000
10	200091	01	10091	201	2018/12/01	180	2740000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

FROM절의 Subquery

```
SELECT a.lnact, a.lnact_seq, a.lnid, a.prod_cd,
       a.ln_dt, a.ln_amt, b.avg_ln_amt
  FROM tacct a
  JOIN (SELECT prod_cd, TRUNC(AVG(ln_amt)) AS avg_ln_amt
          FROM tacct
         WHERE lmt_typ IS NULL
        GROUP BY prod_cd) b
    ON a.prod_cd = b.prod_cd
   AND a.ln_amt  > b.avg_ln_amt
 WHERE lmt_typ IS NULL ;
```

SQL | 인출된 모든 행: 60(0.009초)

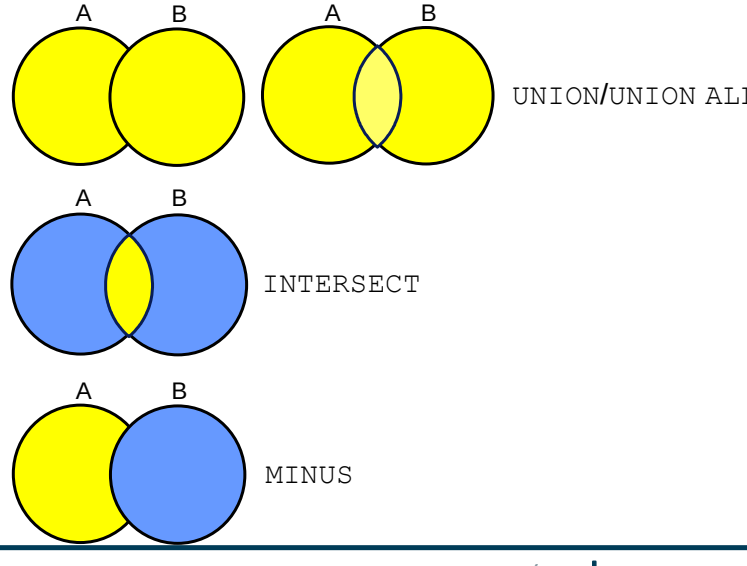
	LNACT	LNACT_SEQ	LNID	PROD_CD	LN_DT	LN_AMT	AVG_LN_AMT
1	200003	01	10003	209	2018/10/25	650000000	630000000
2	200005	01	10005	200	2016/11/06	680000000	527142857
3	200009	01	10009	208	2016/04/14	1390000000	735000000
4	200011	01	10011	206	2021/06/09	710000000	412857142
5	200016	01	10016	203	2019/05/13	1040000000	600000000
6	200017	01	10017	210	2018/08/26	1660000000	707500000
7	200019	01	10019	205	2013/11/26	1020000000	615833333
8	200021	00	10021	104	2017/08/08	130000000	110000000
9	200022	00	10022	100	2018/04/29	140000000	89500000
10	200023	00	10023	103	2019/08/04	260000000	165000000

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

9 집합 연산자

집합 연산자



ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

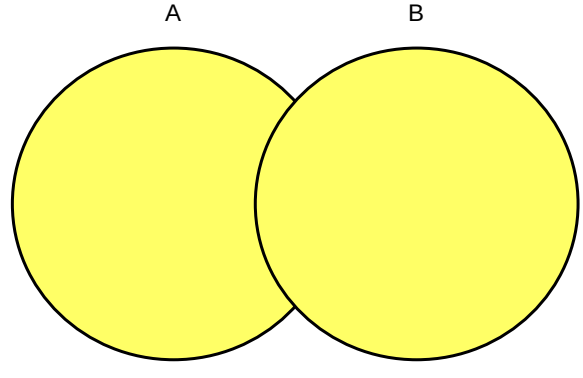
Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

집합 연산자 규칙

- SELECT 리스트의 표현식은 개수가 일치해야 합니다.
- 후속 query에 있는 각 열의 데이터 유형은 첫번째 query에 있는 상응하는 열의 데이터 유형과 일치해야 합니다.
- ORDER BY 절은 명령문의 맨 끝에만 올 수 있습니다.
- 중복 행은 UNION ALL 외에는 자동으로 제거됩니다.
- 첫번째 query의 열 이름이 결과에 나타납니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UNION 연산자



UNION 연산자는 중복 행을 제거한 후 양쪽 query에서 행을 반환합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UNION 연산자 사용

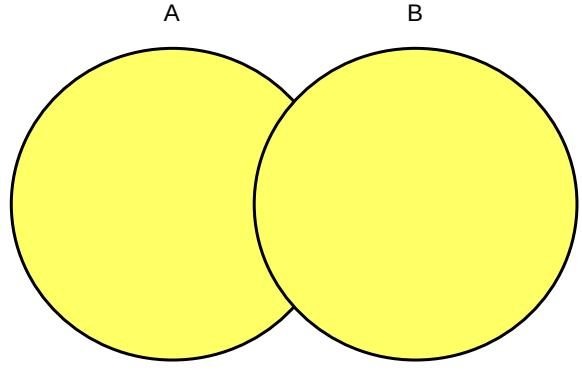
```
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2020/01/01' AND '2020/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
UNION
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2021/01/01' AND '2021/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
ORDER BY 1 ;
```

SQL | 인출된 모든 행: 32(0.015초)

	BRANCH	COUNT(LN_AMT)
1	10	1
2	11	1
3	11	3
4	12	2
5	13	1
6	14	1
7	14	3
8	15	1
9	16	2
10	17	1

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UNION ALL 연산자



UNION ALL 연산자는 모든 중복 행을 포함하여 양쪽 query의 결과를 반환합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UNION ALL 연산자 사용

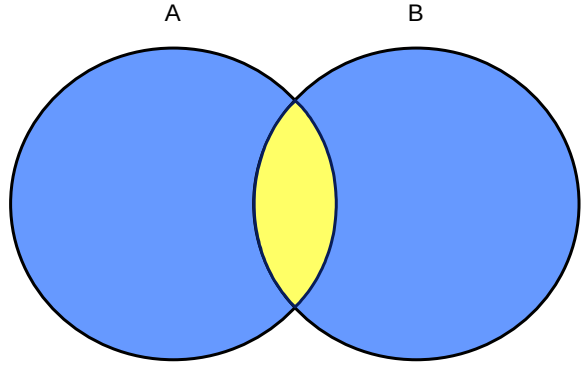
```
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2020/01/01' AND '2020/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
UNION ALL
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2021/01/01' AND '2021/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
ORDER BY 1 ;
```

SQL | 인출된 모든 행: 36(0.011초)

	BRANCH	COUNT(LN_AMT)
1	10	1
2	11	3
3	11	1
4	12	2
5	13	1
6	14	3
7	14	1
8	15	1
9	16	2
10	17	1

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

INTERSECT 연산자



INTERSECT 연산자는 양쪽 query에 공통되는 행을 반환합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

INTERSECT 연산자 사용

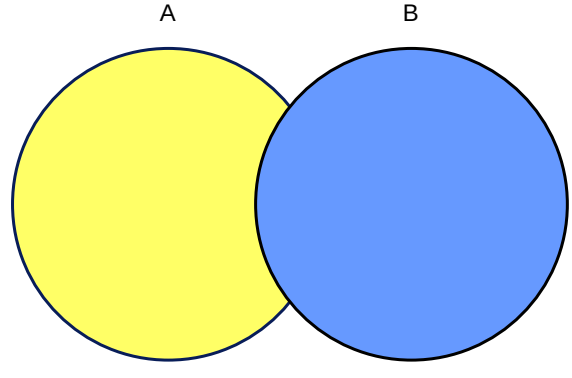
```
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2020/01/01' AND '2020/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
INTERSECT
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2021/01/01' AND '2021/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
ORDER BY 1 ;
```

SQL | 인출된 모든 행: 4(0,012초)

	BRANCH	COUNT(LN_AMT)
1	26	1
2	29	1
3	37	1
4	38	2

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

MINUS 연산자



MINUS 연산자는 첫번째 query에 의해 선택되지만 두번째 query 결과 집합에는 없는 모든 구분 행을 반환합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

MINUS 연산자 사용

```
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2020/01/01' AND '2020/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
MINUS
SELECT branch, COUNT(ln_amt)
  FROM tacct
 WHERE ln_dt BETWEEN '2021/01/01' AND '2021/12/31'
   AND lmt_typ IS NULL
GROUP BY branch
ORDER BY 1 ;
```

SQL | 인출된 모든 행: 12(0.015초)

	BRANCH	COUNT(LN_AMT)
1	10	1
2	11	1
3	14	3
4	15	1
5	17	1
6	28	2
7	30	1
8	33	1
9	39	1
10	40	1

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

10 테이블 생성 및 관리

데이터베이스 객체

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
색인	일부 query의 성능을 향상시킵니다.
동의어	객체에 다른 이름을 부여합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

이름 지정 규칙

- 테이블 이름 및 열 이름은 다음 규칙을 따라야 합니다.
  - 문자로 시작해야 합니다.
  - 길이는 1 - 30자 사이여야 합니다.
  - A - Z, a - z, 0 - 9, _, $, #만 포함할 수 있습니다.
  - 동일한 유저가 소유한 다른 객체의 이름과 중복되면 안 됩니다.
  - Oracle 서버 예약어는 사용할 수 없습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

CREATE TABLE 문

- 다음이 필요합니다.
  - CREATE TABLE 권한
  - 저장 영역

```
CREATE TABLE table
(column datatype [, ...]);
```

- 다음을 지정합니다.
  - 테이블 이름
  - 열 이름, 열 데이터 유형 및 열 크기

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

테이블 생성

- 테이블 생성:

```
CREATE TABLE borrower
(lnid   VARCHAR2(4),
 bthday DATE,
 gender CHAR(1),
 score  NUMBER(4)) ;
```

- 테이블 생성 확인:

```
DESCRIBE borrower
```

```
이름     널? 유형
------ -- -----------
LNID       VARCHAR2(4)
BTHDAY     DATE
GENDER     CHAR(1)
SCORE      NUMBER(4)
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

데이터 유형

데이터 유형	설명
VARCHAR2(*size*)	가변 길이 문자 데이터 (최대 4,000 byte)
CHAR(*size*)	고정 길이 문자 데이터 (최대 2,000 byte)
CLOB	가변 길이 문자 데이터
NUMBER(p, s)	가변 길이 숫자 데이터
DATE	날짜 및 시간 값
BLOB	이미지, 동영상, 사운드 파일 등의 바이너리 데이터 저장

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Subquery를 사용하여 테이블 생성

- CREATE TABLE 문과 AS *subquery* 옵션을 결합하여 테이블을 생성하고 행을 삽입합니다.

```
CREATE TABLE tacct_lmt
AS SELECT *
     FROM tacct
    WHERE lmt_typ = '1' ;
```

```
DESCRIBE tact_lmt
```

작업이 완료되었습니다.(0.197초)

```
이름         널?       유형
---------- -------- -----------
LNACT                CHAR(6)
LNACT_SEQ            CHAR(2)
ACCT_TYP   NOT NULL  CHAR(1)
LMT_TYP              CHAR(1)
LNID       NOT NULL  CHAR(5)
BRANCH     NOT NULL  CHAR(2)
PROD_CD              CHAR(3)
LN_DT      NOT NULL  DATE
LN_TERM    NOT NULL  NUMBER(3)
EXP_DT     NOT NULL  DATE
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

TEMPORARY TABLE 생성

- 트랜잭션 동안 데이터 유지

```
CREATE GLOBAL TEMPORARY TABLE taccount
ON COMMIT DELETE ROWS
AS SELECT *
     FROM tacct
    WHERE lmt_typ IS NULL ;
```

- 세션안에서 데이터 유지

```
CREATE GLOBAL TEMPORARY TABLE taccount
ON COMMIT PRESERVE ROWS
AS SELECT *
     FROM tacct
    WHERE lmt_typ IS NULL ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

ALTER TABLE 문

- ALTER TABLE 문을 사용하여 다음을 수행합니다.
  - 새 열 추가

```
ALTER TABLE borrower
ADD (grade  VARCHAR2(3)) ;
```

  - 기존 열 정의 수정

```
ALTER TABLE borrower
MODIFY (grade  VARCHAR2(4)) ;
```

  - 열 삭제

```
ALTER TABLE borrower
DROP (grade) ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

테이블 삭제

- 테이블을 Recycle bin으로 이동
- PURGE 절이 지정되면 테이블 및 해당 데이터를 완전히 제거

```
DROP TABLE borrower ;
```

```
DROP TABLE borrower PURGE ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

11 DML 활용

DML(데이터 조작어)

- DML 문은 다음과 같은 경우에 실행합니다.
  - 테이블에 새 행 추가
  - 테이블의 기존 행 수정
  - 테이블에서 기존 행 제거

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

INSERT 문 구문

- INSERT 문을 사용하여 테이블에 새 행을 추가합니다.

```
INSERT INTO   table [(column [, column...])]
VALUES     (value [, value...]);
```

- 이 구문을 사용할 경우 한 번에 한 행만 삽입됩니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

새 행 삽입

- 각 열에 대한 값을 포함하는 새 행을 삽입합니다.
- 테이블에 있는 열의 기본 순서로 값을 나열합니다.
- 선택적으로 INSERT 절에 열을 나열합니다.
- 문자와 날짜 값은 작은따옴표로 묶습니다.

```
INSERT INTO tid (lnid, id_typ, bthday, gender, score, grade)
VALUES ('20000', '1', '1990/01/15', '1', 800, NULL) ;
```

```
INSERT INTO tid (lnid, id_typ, bthday, grade)
VALUES ('20001', '2', '2012/01/20', 'A-') ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

다른 테이블에서 행 복사

- INSERT 문을 subquery로 작성합니다.

```
TRUNCATE TABLE taccount ;

INSERT INTO taccount
SELECT *
  FROM tacct
 WHERE lmt_typ IS NULL ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UPDATE 문 구문

- UPDATE 문을 사용하여 테이블의 기존 값을 수정합니다.

```
UPDATE     table
SET        column = value [, column = value, ...]
[WHERE     condition];
```

- 필요한 경우 한 번에 두 개 이상의 행을 갱신합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

테이블의 행 갱신

- WHERE 절을 지정하면 특정 행에서 값이 수정됩니다.

```
UPDATE tid
   SET score = 850
 WHERE lnid  = '20000' ;
```

- WHERE 절을 생략하면 테이블의 모든 행에서 값이 수정됩니다.

```
UPDATE taccount
   SET repay = 'Y'
      ,repay_dt = '2022/04/20' ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

DELETE 문

- DELETE 문을 사용하여 테이블에서 기존 행을 제거할 수 있습니다.
  - WHERE 절을 지정하면 특정 행이 삭제됩니다.

```
DELETE tid
WHERE lnid IN ('20000', '20001') ;
```

  - WHERE 절을 생략하면 테이블의 모든 행이 삭제됩니다.

```
DELETE taccount ;
```

  - TRUNCATE TABLE 명령문은 테이블의 모든 데이터를 삭제하고, 변경 사항을 자동으로 저장합니다. (ROLLBACK 불가능)

```
TRUNCATE TABLE taccount ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

12 트랜잭션 이해 및 관리

데이터베이스 트랜잭션

- 데이터베이스 트랜잭션은 다음 중 하나로 구성됩니다.
  - 데이터를 일관성 있게 변경하는 하나 이상의 DML 문
  - 하나의 DDL, DCL 문

- 트랜잭션 시작과 종료
  - 첫번째 SQL 문이 실행될 때 시작됩니다.
  - 다음 상황 중 하나가 발생하면 종료됩니다.
    - ✓COMMIT 또는 ROLLBACK 문 실행
    - ✓DDL 또는 DCL 문 실행(자동 커밋)

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

COMMIT 및 ROLLBACK 문의 이점

- COMMIT 및 ROLLBACK 문을 사용할 경우 다음과 같은 이점이 있습니다.
  - 데이터 일관성 보장
  - 변경 사항을 영구 적용하기 전에 데이터 변경 사항 검토
  - 논리적으로 관련된 작업 그룹화

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

COMMIT 또는 ROLLBACK 이전의 데이터 상태

- 이전의 데이터 상태를 복구할 수 있습니다.
- 현재 세션은 SELECT 문을 사용하여 DML 작업의 결과를 확인할 수 있습니다.
- 다른 세션은 현재 세션이 실행한 DML 문의 결과를 볼 수 *없습니다*.
- 영향을 받는 행이 *잠기므로* 다른 세션이 영향을 받는 행의 데이터를 변경할 수 없습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

COMMIT 후의 데이터 상태

- 데이터 변경 사항이 데이터베이스에 저장됩니다.
- 모든 세션이 결과를 확인할 수 있습니다.
- 영향을 받는 행의 잠금이 해제되어 이러한 행을 다른 세션에서 조작할 수 있습니다.

```
INSERT INTO tid (lnid, id_typ, bthday, grade)
VALUES ('20001', '2', '2012/01/20', 'A-') ;

COMMIT;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

ROLLBACK 후의 데이터 상태

- 데이터 변경 사항이 실행 취소됩니다.
- 영향받는 행의 잠금이 해제됩니다.

```
DELETE tid
WHERE lnid = '20001' ;

ROLLBACK;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

13 제약 조건 이해

제약 조건 포함

- 제약 조건은 테이블 레벨에서 규칙을 강제 적용합니다.
- 제약 조건은 데이터베이스의 일관성 및 무결성을 보장합니다.
- 유효한 제약 조건 유형은 다음과 같습니다.
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

NOT NULL 제약 조건

•열에 null 값이 허용되지 않도록 보장합니다.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIETZ	515.123.8181	07-JUN-94

NOT NULL 제약 조건
(Primary Key는 NOT NULL 제약 조건을 적용합니다.)

NOT NULL 제약 조건

NOT NULL 제약 조건 없음(모든 행에서 이 열에 대해 null 값을 포함할 수 있습니다.)

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

UNIQUE 제약 조건

EMPLOYEES

UNIQUE 제약 조건

	EMPLOYEE_ID	LAST_NAME	EMAIL
1	100	King	SKING
2	101	Kochhar	NKOCHHAR
3	102	De Haan	LDEHAAN
4	103	Hunold	AHUNOLD
5	104	Ernst	BERNST
6	107	Lorentz	DLORENTZ

...

INSERT INTO

208	SMITH	JSMITH
209	SMITH	JSMITH

← 허용됨

← 허용되지 않음: 이미 있음

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

PRIMARY KEY 제약 조건

DEPARTMENTS

PRIMARY KEY

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

허용되지 않음
(null 값)

INSERT INTO

(null)	Public Accounting	124	2500
50	Finance	124	1500

허용되지 않음
(이미 50이 있음)

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

FOREIGN KEY 제약 조건

DEPARTMENTS

PRIMARY KEY

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500

...

EMPLOYEES

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	200	Whalen	10
2	201	Hartstein	20
3	202	Fay	20
4	205	Higgins	110
5	206	Gietz	110

...

FOREIGN KEY

INSERT INTO

200 Ford	9
200 Ford	60

허용되지 않음
(9가 없음)

허용됨

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

CHECK 제약 조건

- 각 행이 충족해야 하는 조건을 정의합니다.
- 다른 테이블의 열을 참조할 수 없습니다.

```
...,  salary   NUMBER(2)
      CONSTRAINT emp_salary_min
             CHECK (salary > 0),...
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

제약 조건 위반

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

```
Error starting at line 1 in command:
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110
Error report:
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:    A foreign key value has no matching primary key value.
*Action:   Delete the foreign key or add a matching primary key.
```

- 부서 55가 존재하지 않습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

14 기타 객체 관리

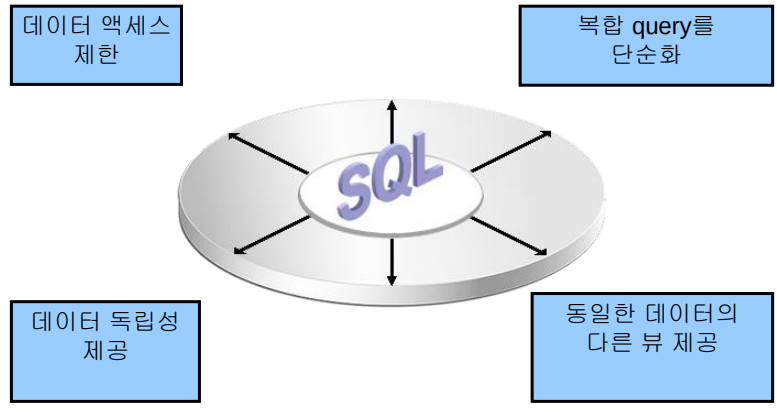
데이터베이스 객체

객체	설명
테이블	기본 저장 단위이며 행으로 구성되어 있습니다.
뷰	하나 이상의 테이블에 있는 데이터의 부분 집합을 논리적으로 나타냅니다.
시퀀스	숫자 값을 생성합니다.
색인	데이터 검색 query의 성능을 향상시킵니다.
동의어	객체에 대체 이름을 부여합니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

뷰의 이점



ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

뷰 생성

- CREATE VIEW 문에 subquery를 포함시킵니다.

```
CREATE [OR REPLACE] VIEW view
  [(alias[, alias]...)]
 AS subquery ;
```

- 이 subquery에 복합 SELECT 구문을 포함할 수 있습니다.

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

뷰 생성

```
CREATE VIEW vid
AS SELECT lnid, bthday, gender, score
     FROM tid
    WHERE id_typ = '1' ;
```

```
SELECT *
FROM   vid ;
```

SQL | 인출된 모든 행: 31(0.01초)

	LNID	BTHDAY	GENDER	SCORE
1	10002	1997/08/09	1	970
2	10004	1962/09/30	1	920
3	10007	1997/01/19	2	930
4	10010	1974/02/14	1	840
5	10012	1952/06/13	2	990
6	10013	1959/02/13	2	690
7	10016	1983/06/24	1	990
8	10017	1971/11/20	2	660
9	10020	1995/12/24	2	730
10	10031	1975/08/18	2	920

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

뷰 수정

- CREATE OR REPLACE VIEW 절을 사용하여 뷰를 수정합니다.

```
CREATE OR REPLACE VIEW vid
AS SELECT lnid, id_typ, bthday, gender, score
     FROM tid
    WHERE id_typ = '1' ;
```

```
SELECT *
FROM   vid ;
```

SQL | 인출된 모든 행: 31(0.011초)

	LNID	ID_TYP	BTHDAY	GENDER	SCORE
1	10002	1	1997/08/09	1	970
2	10004	1	1962/09/30	1	920
3	10007	1	1997/01/19	2	930
4	10010	1	1974/02/14	1	840
5	10012	1	1952/06/13	2	990
6	10013	1	1959/02/13	2	690
7	10016	1	1983/06/24	1	990
8	10017	1	1971/11/20	2	660
9	10020	1	1995/12/24	2	730
10	10031	1	1975/08/18	2	920

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

뷰 제거

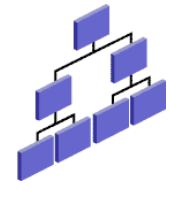
- 뷰는 데이터베이스의 기본 테이블을 기반으로 하기 때문에 뷰를 제거해도 데이터는 손실되지 않습니다.

```
DROP VIEW vid ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

인덱스

- 인덱스:
  - 오라클 서버에서 포인터를 사용하여 행 검색 속도를 높이는 데 사용할 수 있습니다.
  - 신속한 경로 액세스 방식을 사용하여 데이터를 빠르게 찾아 디스크 I/O(입/출력)를 줄일 수 있습니다.
  - 인덱스의 대상인 테이블에 종속적입니다.
  - 오라클 서버에서 자동으로 사용되고 유지 관리됩니다.



ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

인덱스 생성

- 하나 이상의 열에 인덱스를 생성합니다.

```
CREATE INDEX index
ON table (column[, column]...);
```

```
CREATE INDEX acctlmt_ix01 ON tacct_lmt(lnact) ;

SELECT *
  FROM tacct_lmt
 WHERE lnact = '200000';
```

OPERATION	OBJECT_NAME	OPTIONS
SELECT STATEMENT		
TABLE ACCESS	TACCT_LMT	BY INDEX ROWID BATCHED
INDEX	ACCTLMT_IX01	RANGE SCAN
Access Predicates		
LNACT='200000'		

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

인덱스 제거

- DROP INDEX 명령을 사용하여 인덱스를 제거합니다.

```
DROP INDEX acctlmt_ix01 ;
```

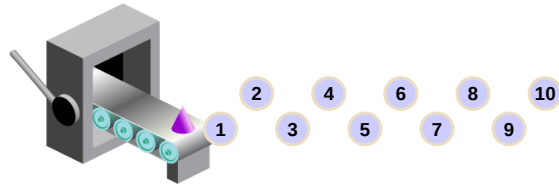
ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

시퀀스

- 시퀀스:
  - 고유 번호를 자동으로 생성할 수 있습니다.
  - 공유할 수 있는 객체입니다.
  - Primary key 값을 생성하는 데 사용할 수 있습니다.

```
CREATE SEQUENCE lnid_sq
START WITH 30000
MAXVALUE   99999 ;
```



ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

NEXTVAL 및 CURRVAL Pseudocolumn

- NEXTVAL은 사용 가능한 다음 시퀀스 값을 반환합니다. 다른 유저인 경우도 포함하여 참조될 때마다 고유 값을 반환합니다.
- CURRVAL은 가장 최근에 발생된 숫자값을 보여줍니다.
- CURRVAL을 통해 확인된 숫자 이후에 숫자가 NEXTVAL을 통해 발생됩니다.

```
INSERT INTO tid (lnid, id_typ, bthday, grade)
VALUES (lnid_sq.nextval, '2', '2012/01/20', 'A-') ;
```

```
SELECT   lnid_sq.currval
FROM     dual;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved.
All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

시퀀스 제거

- 시퀀스를 제거하려면 DROP 문을 사용합니다.

```
DROP SEQUENCE lnid_sq ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

동의어

- 동의어:
  - 데이터베이스 객체입니다.
  - 테이블 또는 기타 데이터베이스 객체에 대체 이름을 제공하기 위해 생성할 수 있습니다.
  - 저장 영역이 필요하지 않습니다.
  - 다른 유저가 소유한 테이블을 쉽게 참조할 수 있습니다.
  - 긴 객체 이름을 짧게 만듭니다.

```
CREATE SYNONYM lmt FOR tacct_lmt ;

SELECT * FROM lmt ;
```

```
DROP SYNONYM lmt ;
```

ITNVALUE Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.

Copyright 2017. ITNVALUE all rights reserved. All pictures can not be copied without permission.