

2024년 상반기 K-디지털 트레이닝

DB DML2

[KB] IT's Your Life

Q1 - DML(설치파일)

- shop7 db를 생성하고, 다음 EMP, DEPT 테이블을 이용하여 데이터를 조작하시오.

1. csv파일을 이용하여 import한 데이터로 테이블(EMP_DATA, DEPT_DATA) 생성

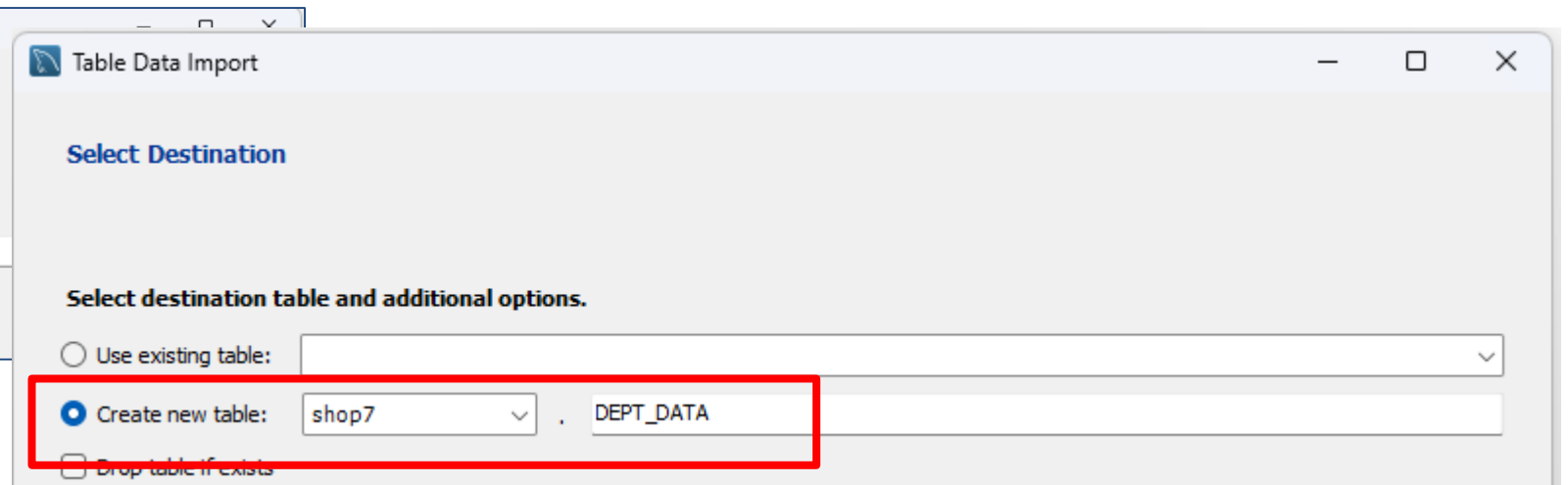
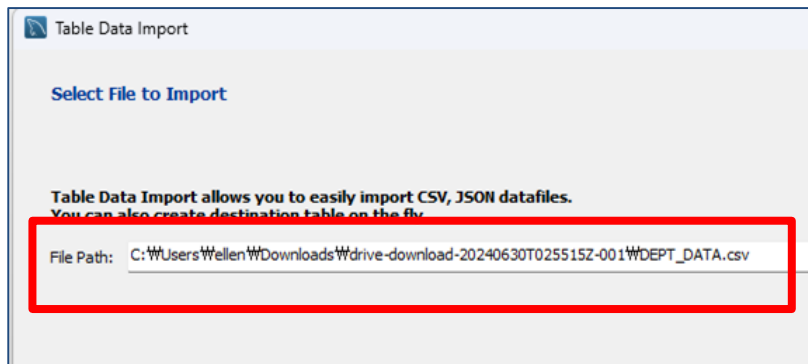
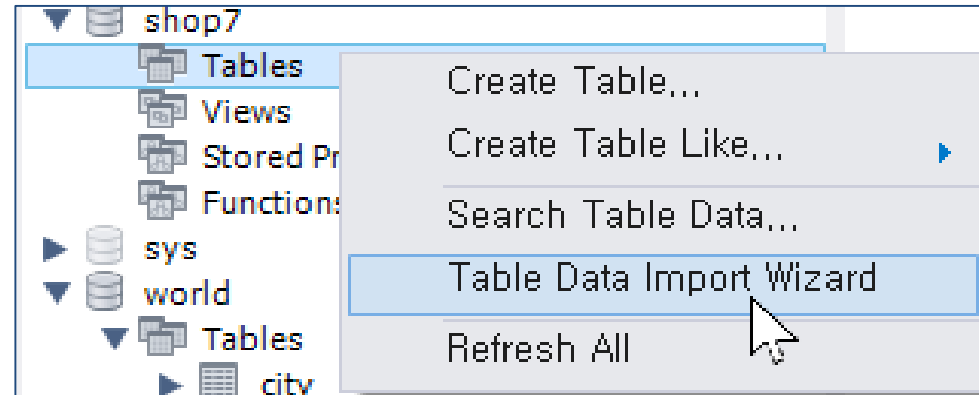
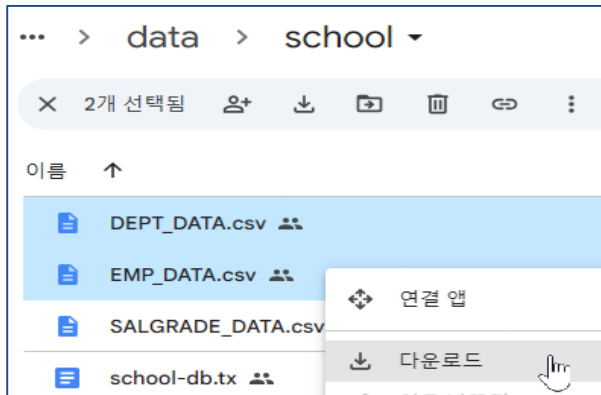


Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Field Type
DEPTNO	int
DNAME	text
LOC	text

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

ERD를 보고 INT로
설정되어있는지
확인해보고 INT로
설정되지 않은 경우 특정
컬럼을 INT로 설정

Table Data Import

Import Data

The following tasks will now be performed. Please monitor the execution.

☐ Prepare Import
☐ Import data file

Click [Next >] to execute.

Logs

< Back Next > Cancel

Table Data Import

Import Data

The following tasks will now be performed. Please monitor the execution.

☒ Prepare Import
☒ Import data file

Finished performing tasks. Click [Next >] to continue.

Show Logs

< Back Next > Cancel

shop7

Tables

Views

Stored

Function

sys

world

Tables

Create Table...

Create Table Like...

Search Table Data...

Table Data Import Wizard

Refresh All

emp_data - Table dept_data - Table

Table Name: dept_data

Charset/Collation: utf8mb4

Comments:

Column Name	Datatype	PK	NN	UQ
DEPTNO	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DNAME	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOC	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

pk설정

Table Name: emp_data

Charset/Collation: utf8mb4

Comments:

Column Name	Datatype	PK	NN	UQ
EMPNO	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ENAME	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
JOB	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MGR	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HIREDATE	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SAL	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
COMM	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DEPTNO	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2. 외래키 설정(emp_data table의 deptno와 dept_data table의 deptno)

Navigator: SCHEMAS

Filter objects

- emp
- salgrade
- Views
- Stored Procedures
- Functions
- shop
- shop2
- shop3
- shop5
- shop7
 - Tables
 - dept_data
 - emp_data
 - Views
 - Stored Procedures
 - Functions
- sys
- world

Administration Schemas

Information

emp_data - Table

Table Name: emp_data Schema: shop7

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: InnoDB

Comments:

Foreign Key Name	Referenced Table	Column	Referenced Column
dept_no_fk	`shop7`.`dept_data`	<input type="checkbox"/> EMPNO <input type="checkbox"/> ENAME <input type="checkbox"/> JOB <input type="checkbox"/> MGR <input type="checkbox"/> HIREDATE <input type="checkbox"/> SAL <input type="checkbox"/> COMM <input checked="" type="checkbox"/> DEPTNO	DEPTNO

Foreign Key Options

On Update: NO ACTION

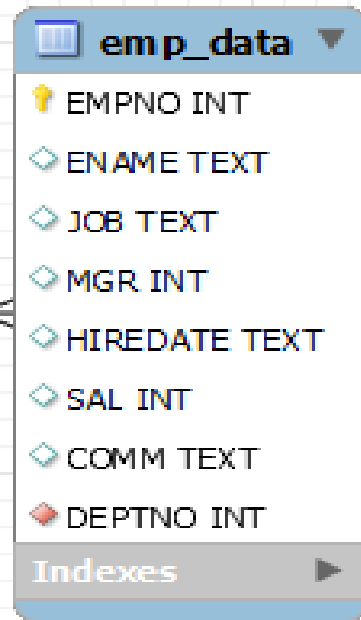
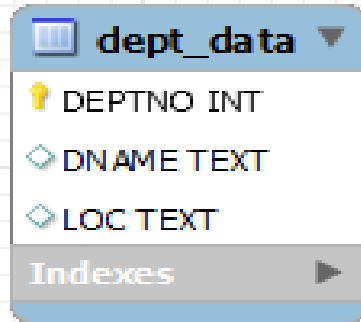
On Delete: NO ACTION

Foreign Key Comment generation

Apply Revert

3. 모든 데이터를 넣은 테이블이 생성됨. ERD로 확인 가능.

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17 00:0...	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20 00:0...	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22 00:0...	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02 00:0...	2975		20
7654	MARTIN	SALESMAN	7698	1981-09-28 00:0...	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01 00:0...	2850		30
7782	CLARK	MANAGER	7839	1981-06-09 00:0...	2450		10
7788	SCOTT	ANALYST	7566	1982-12-09 00:0...	3000		20
7844	TURNER	SALESMAN	7698	1981-09-08 00:0...	1500	0	30
7876	ADAMS	CLERK	7788	1983-01-12 00:0...	1100		20
7900	JAMES	CLERK	7698	1981-12-03 00:0...	950		30

- 문자함수, 숫자함수, 날짜함수를 이용하여 다음을 해결하시오.
 - emp_data 테이블 이용
-
1. ENAME의 글자수 검색
 2. JOB이 MANAGER인 사람의 ENAME 글자수 검색
 3. COMM이 null인 사람의 ENAME, JOB 검색
 4. COMM이 null이 아닌 사람의 HIREDATE 검색
 5. ENAME의 2번째글자부터 끝까지 검색
 6. JOB의 1번째 글자부터 3번째 글자까지 검색
 7. ENAME의 글자 중 K를 P으로 변경하여 검색
 8. ENAME과 EMPNO를 이용하여 '7369번은 SMITH입니다' 형식으로 모든 row 출력
 9. HIREDATE 이용하여 입사한 월과 연도를 출력
 10. JOB이 MANAGER이면 level1, SALESMAN이면 level2, 기타이면 level3라고 LEVEL_RESULT컬럼에 출력

1 - 5.

```
1. use shop7;
2
3 -- 1.
4. select length(ename)
5 from emp_data;
```

length(ename)
5
5
4
5
6
5
5
5
6
5
5
4
6

```
7 -- 2.
8. select length(ename)
9 from emp_data
10 where job = 'MANAGER';
```

length(ename)	job
5	MANAGER
5	MANAGER
5	MANAGER

```
12 -- 3
13. select ename, job
14 from emp_data
15 where comm is null;
```

ename	job

```
17 -- 4
18. select hiredate
19 from emp_data
20 where comm is not null;
```

hiredate
1980-12-17 00:00:00
1981-02-20 00:00:00
1981-02-22 00:00:00
1981-04-02 00:00:00

```
22 -- 5
23. select substr(ename, 2)
24 from emp_data;
```

substr(ename, 2)
MITH
LLEN

6 - 10.

```
26 -- 6
27 • select substr(job, 1, 3)
28 from emp_data;
29
```

substr(job, 1, 3)
CLE
SAL
SAI

```
30 -- 7
31 • select replace(ename, 'K', 'P')
32 from emp_data;
```

replace(ename, 'K', 'P')
JONES
MARTIN
BLAPE

```
42 -- 10
43 • select job,
44 case job
45 when 'manager' then 'level1'
46 when 'salesman' then 'level2'
47 else 'level3'
48 end as level_result
49 from emp_data;
```

job	level_result
CLERK	level3
SALESMAN	level2
SALESMAN	level2
MANAGER	level1

```
34 -- 8
35 • select concat(empno, '번은 ', ename, '입니다.')
36 from emp_data;
```

concat(empno, '번은 ', ename, '입니다.')
7369번은 SMITH입니다.
7499번은 ALLEN입니다.
7521번은 WARD입니다.

```
38 -- 9
39 • select month(hiredate), year(hiredate)
40 from emp_data;
```

month(hiredate)	year(hiredate)
12	1980
2	1981

- 집계, 그룹함수를 이용하여 다음을 해결하시오.

1. 전체 직원수 출력
2. DEPTNO가 20이거나 MGR이 7700이하인 직원수 출력
3. DEPTNO가 10 또는 20인 직원의 SAL최저값 출력
4. 직원의 SAL 최저값, 최고값, 합계 출력
5. DEPTNO별 SAL 평균 출력
6. DEPTNO별 SAL 합계를 DEPTNO 오름차순으로 출력
7. DEPTNO별 SAL 평균이 2000이상인 SAL평균을 DEPTNO역순으로 출력
8. JOB이 MANAGER인 직원의 전체 수, 월급의 평균 검색
9. ENAME에 s를 포함하는 직원의 전체 수를 검색
10. SAL이 3000이상이고, COMM이 null이 아닌 직원의 전체 수 검색, SAL의 역순으로 출력

1 - 4.

52 -- 1

53 • **select count(*)**

54 **from emp_data;**

55

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)

13

56 -- 2

57 • **select count(*)**

58 **from emp_data**

59 **where deptno = 20 or mgr <= 7700;**

60

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)

10

61 -- 3

62 • **select min(sal)**

63 **from emp_data**

64 **where deptno = 10 or deptno = 20;**

65

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

min(sal)

800

66 -- 4

67 • **select min(sal), max(sal), sum(sal)**

68 **from emp_data;**

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

min(sal)

max(sal)

sum(sal)

800

3000

24025

5 - 6.

```

71  -- 5
72  -- DEPTNO별 SAL 평균 출력
73 • select avg(sal)
74 from emp_data
75 group by deptno;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	avg(sal)			
▶	1875.0000			
	2175.0000			
	1566.6667			

```

77  -- 6
78  -- DEPTNO별 SAL 합계를 DEPTNO 오름차순으로 출력
79 • select sum(sal)
80 from emp_data
81 group by deptno
82 order by deptno;
83

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	sum(sal)			
▶	3750			
	10875			
	9400			

7- 10.

```

84  -- 7
85  -- DEPTNO별 SAL 평균이 2000이상인 SAL 평균을 DEPTNO역순으로 출력
86  select avg(sal)
87  from emp_data
88  GROUP BY DEPTNO
89  HAVING AVG(sal) >= 2000
90  ORDER BY DEPTNO DESC;
91

```

avg(sal)
2175.0000

```

99  -- 9
100 -- ENAME에 S를 포함하는 직원의 전체 수를 검색
101 select count(*)
102 from emp_data
103 where ename like '%S%';

```

count(*)
5

```

93  -- 8
94  -- JOB이 MANAGER인 직원의 전체 수, 월급의 평균 검색
95  select avg(SAL), count(*)
96  from emp_data
97  where job = 'MANAGER';
98

```

avg(SAL)	count(*)
2758.3333	3

```

105 -- 10
106 -- SAL이 3000이상이고,
107 -- COMM이 null이 아닌 직원의 전체 수 검색, SAL의 역순으로 출력
108 select count(*)
109 from emp_data
110 where sal >= 3000 and comm is not null
111 order by sal desc;

```

count(*)
2

- **EMP_DATA, DEPT_DATA 테이블을 이용하여 다음의 문제를 해결하시오.**
- 1. DEPTNO컬럼을 기준으로 두 개의 테이블을 조인 후, EMPNO, JOB, SAL, COMM, DNAME, LOC를 검색하여 출력하시오.(INNER JOIN)
- 2. EMP테이블 내에는 일반 직원과 매니저가 함께 저장되어있다. 일반 직원의 EMPNO, JOB, SAL과 일반직원을 관리하는 MGR의 EMPNO, JOB을 함께 검색하여 출력하시오.(SELF JOIN)
- 3. 모든 DEPT_DATA 테이블의 항목이 출력되고, 두 테이블 간 DEPTNO가 일치하는 것을 찾아 EMP_DATA테이블의 DEPTNO, EMPNO, ENAME 항목을 추가 출력되도록 조인하시오.(OUTER JOIN)

Q1 - DML

```

114 -- 1
115 -- DEPTNO컬럼을 기준으로 두 개의 테이블을 조인 후,
116 -- EMPNO, JOB, SAL, COMM, DNAME, LOC를 검색하여 출력하시오.
117 -- (INNER JOIN)
118 • select e.empno, e.job, e.sal, e.comm, d.dname, d.loc
119   from emp_data e, dept_data d
120  where e.DEPTNO = d.DEPTNO;

```

empno	job	sal	comm	dname	loc
7782	MANAGER	2450		ACCOUNTING	NEW YORK
7934	CLERK	1300		ACCOUNTING	NEW YORK
7369	CLERK	800		RESEARCH	DALLAS
7566	MANAGER	2975		RESEARCH	DALLAS
7788	ANALYST	3000		RESEARCH	DALLAS
7876	CLERK	1100		RESEARCH	DALLAS

```

122 -- 2
123 -- EMP테이블 내에는 일반 직원과 매니저가 함께 저장되어있다.
124 -- 일반 직원의 EMPNO, JOB, SAL과
125 -- 일반직원을 관리하는 MGR의 EMPNO, JOB을 함께 검색하여 출력하시오.
126 -- (SELF JOIN)
127 • select e.empno, e.job, e.sal, m.empno as mgrno, m.job
128   from emp_data e, emp_data m
129  where e.MGR = m.empno;

```

empno	job	sal	mgrno	job
7369	CLERK	800	7902	ANALYST
7499	SALESMAN	1600	7698	MANAGER
7521	SALESMAN	1250	7698	MANAGER
7654	SALESMAN	1250	7698	MANAGER

```
132  -- 3
133  -- 모든 DEPT_DATA 테이블이 출력되고,
134  -- EMP_DATA테이블의 항목 중 두 테이블 간 DEPTNO가 일치하는 것을 찾아
135  -- DEPTNO, EMPNO, ENAME을 검색하여 출력하시오.
136  -- (OUTER JOIN)
137  • select d.*, e.deptno, e.empno, e.ename
138  from dept_data d
139  left outer join emp_data e
140  on (d.deptno = e.deptno);
```

Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	DEPTNO	DNAME	LOC	deptno	empno	ename
▶	10	ACCOUNTING	NEW YORK	10	7782	CLARK
	10	ACCOUNTING	NEW YORK	10	7934	MILLER
	20	RESEARCH	DALLAS	20	7369	SMITH
	20	RESEARCH	DALLAS	20	7566	JONES
	20	RESEARCH	DALLAS	20	7788	SCOTT

수고하셨습니다!

