

A glowing light bulb with a circuit board overlay. The light bulb is on the right side of the image, with its filament glowing. A circuit board overlay is visible on the left side of the image, with lines connecting to the light bulb. The background is a solid blue color.

20200507 DB
기초 5

B반 신석환

과제



#1 -> #4



#2 -> #5



#3 -> #6



끝

#1 - #4

● 2. 실습(전체 복습하기)

[실습 #1]

· 아래와 같은 Table을 구성하고 출력하기(No는 AUTO_INCREMENT)

Employee					
No	Name	Age	Department	Title	Salary
1	"본인이름"	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	5600000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accouting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
7	Anne	31	research and development department	Research Engineer	4700000
8	Henry	32	research and development department	Research Engineer	4200000
9	William	27	research and development department	Research Engineer	5100000
10	John	41	research and development department	Principal Research Engineer	9200000
10	Thomas	39	research and development department	Senior Research Engineer	7200000

1. Salary가 5500000원 이상인 사람들 출력
2. Age가 40살 미만이면서 Salary가 5000000원 이상인 사람들 출력
3. Salary가 5000000원 이하인 사람들은 100000원씩 더한 후 출력
4. Title이 "Research Engineer"인 사람들은 모두 "Senior Engineer"로 변경 후 출력

● 2. 실습(전체 복습하기)

[실습 #4]

· 실습 #1에서 진행한 Table에 '난이도' Column 추가하기

1. 본인이 생각하는 난이도를 기준으로 각각 상, 중, 하로 레코드 입력하기
2. 레코드 전체 출력하기

#1 - 1

```
mysql> select * from employee;
```

No	Name	Age	Department	Title	Salary
1	Shin SeokHwan	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accounting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
7	Anne	31	research and development department	Research Engineer	4700000
8	Henry	32	research and development department	Research Engineer	4200000
9	William	27	research and development department	Research Engineer	5100000
10	John	41	research and development department	Principal Research Engineer	9200000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

```
11 rows in set (0.00 sec)
```

```
mysql> select * from employee where Salary >= 5500000;
```

No	Name	Age	Department	Title	Salary
1	Shin SeokHwan	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accounting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
10	John	41	research and development department	Principal Research Engineer	9200000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

```
8 rows in set (0.00 sec)
```

#1 - 2, 3

2

```
|> select * from employee where Age < 40 and Salary >= 5000000;
```

	Name	Age	Department	Title	Salary
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
5	Alice	34	accouting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
9	William	27	research and development department	Senior Engineer	5100000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

3

```
mysql> select * from employee;
```

	Name	Age	Department	Title	Salary
	Shin SeokHwan	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accouting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
7	Anne	31	research and development department	Research Engineer	4800000
8	Henry	32	research and development department	Research Engineer	4300000
9	William	27	research and development department	Research Engineer	5100000
10	John	41	research and development department	Principal Research Engineer	9200000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

#1 - 4

No	Name	Age	Department	Title	Salary
1	Shin SeokHwan	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accounting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
7	Anne	31	research and development department	Research Engineer	4800000
8	Henry	32	research and development department	Research Engineer	4300000
9	William	27	research and development department	Research Engineer	5100000
10	John	41	research and development department	Principal Research Engineer	9200000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

11 rows in set (0.06 sec)

```
mysql> update employee set Title = 'Senior Engineer' where Title =  
-> ^C
```

```
mysql> update employee set Title = 'Senior Engineer' where Title = 'Research Engineer';
```

Query OK, 3 rows affected (0.17 sec)

Rows matched: 3 Changed: 3 Warnings: 0

```
mysql> select * from employee;
```

No	Name	Age	Department	Title	Salary
1	Shin SeokHwan	45	office of president	CEO	20000000
2	Leo	36	planning department	Manager	7000000
3	Jake	30	human resources department	Assistant Manager	13000000
4	Mary	40	research and development department	Director	13000000
5	Alice	34	accounting department	Manager	7000000
6	Paul	39	research and development department	Senior Research Engineer	6800000
7	Anne	31	research and development department	Senior Engineer	4800000
8	Henry	32	research and development department	Senior Engineer	4300000
9	William	27	research and development department	Senior Engineer	5100000
10	John	41	research and development department	Principal Research Engineer	9200000
11	Thomas	39	research and development department	Senior Research Engineer	7200000

#4

No	Name	Age	Department	Title	Salary	level
1	Shin SeokHwan	45	office of president	CEO	20000000	top
2	Leo	36	planning department	Manager	7000000	mid
3	Jake	30	human resources department	Assistant Manager	13000000	top
4	Mary	40	research and development department	Director	13000000	top
5	Alice	34	accounting department	Manager	7000000	mid
6	Paul	39	research and development department	Senior Research Engineer	6800000	mid
7	Anne	31	research and development department	Senior Engineer	4800000	low
8	Henry	32	research and development department	Senior Engineer	4300000	low
9	William	27	research and development department	Senior Engineer	5100000	low
10	John	41	research and development department	Principal Research Engineer	9200000	top
11	Thomas	39	research and development department	Senior Research Engineer	7200000	mid

11 rows in set (0.00 sec)

#2 - #5 -

● 2. 실습(전체 복습하기)

[실습 #2]

· SQL 명령어 백과사전 만들기

1. Table 1개를 생성
2. Column은 No(AUTO_INCREMENT), 명령어, 설명, 예제, 비고로 구성
3. 아래의 명령어에 대한 설명과 예제 등을 넣기
 - 1) SELECT
 - 2) INSERT
 - 3) CREATE
 - 4) DELETE
 - 5) UPDATE
 - 6) TRUNCATE
 - 7) SHOW
 - 8) USE등 총 12개
4. 레코드 전체 출력하기

● 2. 실습(전체 복습하기)

[실습 #5]

· 실습 #3에서 진행한 Table에 '난이도' Column 추가하기

1. 본인이 생각하는 난이도를 기준으로 각각 상, 중, 하로 레코드 입력하기
2. 레코드 전체 출력하기

#2

-테이블을 생성
[Create table mysql]

-각각의 값 입력
[insert into mysql]

```
mysql> create table mySQL (
    -> No int not null auto_increment primary key,
    -> Orderwords char(20),
    -> Exp char(30),
    -> Example char(20),
    -> Etc char(20)
    -> );
Query OK, 0 rows affected (1.29 sec)

mysql> insert into mySQL(Orderwords, Exp, Example, Etc) values(
    -> 'SELECT', 'Searching Column', 'select * from table', ''
    -> );
Query OK, 1 row affected (0.05 sec)
```

No	Orderwords	Exp	Example	Etc
1	SELECT	Searching Column	select * from table	
2	INSERT	Add Detail	insert into table values	
3	CREATE	Create Something	create table tableName	
4	DELETE	Delete Somethin	delete from table	
5	UPDATE	Changing column detail	update tableName set	
6	TRUNCATE	Delete Something	truncate table tableName	
7	SHOW	Show databases or table	show databases	
8	USE	Use a database or table	use databaseName	
9	ALTER	Add or Delete Column	alter table tableName add colm	
10	ORDER BY	Order a column	select from tableName order by	
11	GROUP BY	Grouping Details	select * from tableName group by	
12	AUTO_INCREMENT	Making No Mechanicly	No int auto_increment	

#5

-컬럼 추가
(alter table TN add)

-추가된 칼럼에 값 입력
(update TN set ?? where)

```
mysql> alter table mysql add column level char(10);  
Query OK, 0 rows affected (0.39 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> update mysql set level = '상' where Wording = 'UPDATE';  
Query OK, 1 row affected (0.07 sec)  
Rows matched: 1 Changed: 1 Warnings: 0  
  
mysql> alter table mysql change Orderwords Wording char(20);  
Query OK, 0 rows affected (0.10 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

No	Wording	Exp	Example	Etc	level
1	SELECT	Searching Column	select * from table		중
2	INSERT	Add Detail	insert into table values		하
3	CREATE	Create Something	create table tableName		하
4	DELETE	Delete Somethin	delete from table		하
5	UPDATE	Changing column detail	update tableName set		상
6	TRUNCATE	Delete Something	truncate table tableName		하
7	SHOW	Show databases or table	show databases		하
8	USE	Use a database or table	use databaseName		하
9	ALTER	Add or Delete Column	alter table tableName add colm		상
10	ORDER BY	Order a column	select from tableName order by		상
11	GROUP BY	Grouping Details	select * from tableName group by		상
12	AUTO_INCREMENT	Making No Machaniclly	No int auto_increment		하

#7

2. 실습(전체 복습하기)

[실습 #7]

· 실습 #5에서 진행한 Table에 'ifremembered(bool)' Column 추가하기

1. 각 항목이 잘 기억나는지를 입력(true, false)

2. 레코드 전체 출력하기

Hint) bool은 true(1), false(0)의 값을 가지는 논리형 자료형임

```
mysql> desc mysql;
```

Field	Type	Null	Key	Default	Extra
No	int	NO	PRI	NULL	auto_increment
Wording	char(20)	YES		NULL	
Exp	char(30)	YES		NULL	
Example	char(40)	YES		NULL	
Etc	char(20)	YES		NULL	
level	char(10)	YES		NULL	
ifremembered	tinyint(1)	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> select * from mysql;
```

No	Wording	Exp	Example	Etc	level	ifremembered
1	SELECT	Searching Column	select * from table		중	1
2	INSERT	Add Detail	insert into table values		하	1
3	CREATE	Create Something	create table tableName		하	1
4	DELETE	Delete Somethin	delete from table		하	1
5	UPDATE	Changing column detail	update tableName set		상	1
6	TRUNCATE	Delete Something	truncate table tableName		하	1
7	SHOW	Show databases or table	show databases		하	1
8	USE	Use a database or table	use databaseName		하	1
9	ALTER	Add or Delete Column	alter table tableName add colm		상	1
10	ORDER BY	Order a column	select from tableName order by		상	1
11	GROUP BY	Grouping Details	select * from tableName group by		상	1
12	AUTO_INCREMENT	Making No Mechanicly	No int auto_increment		하	1

12 rows in set (0.02 sec)

#3 - #6

● 2. 실습(전체 복습하기)

[실습 #3]

· Database 기본 개념 백과사전 만들기

1. Table 1개를 생성
2. Column은 No(AUTO_INCREMENT), 용어, 설명, 비고로 구성
3. 아래의 명령어에 대한 설명과 예제 등을 넣기
 - 1) 데이터 모델링
 - 2) DBMS
 - 3) DML
 - 4) DDL등 총 10개
4. 레코드 전체 출력하기

● 2. 실습(전체 복습하기)

[실습 #6]

- 실습 #3에서 진행한 Table에 'DATE' Column 추가하기
1. 각 레코드마다 언제 학습(수업일)했는지 날짜를 넣기
 2. 레코드 전체 출력하기

#3

-테이블을 생성 [Create table mysql]

-각각의 값 입력 [insert into mysql]

```
mysql> create table DirectoryDB (  
-> No int not null auto_increment primary key,  
-> wording char(20),  
-> Exp char(100),  
-> Etc char(30)  
-> );  
  
mysql> insert into DirectoryDB(wording, Exp, Etc) values (  
-> 'Data Modeling',  
-> '모델링이란 현실 세계를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기술',  
-> '',  
-> );  
Query OK, 1 row affected (0.10 sec)
```

```
mysql> select * from directorydb;
```

No	wording	Exp	Etc
1	Data Modeling	모델링이란 현실 세계를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기술	
2	DBMS	사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고, 데이터베이스를 관리해주는 소프트웨어	관계형, 객체지향형, 객체관계형
3	DML	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어	select, insert, update, delete
4	DDL	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어	create, alter, drop, rename, truncate
5	DCL	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어들	grant, revoke
6	TCL	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위 트랜잭션 별로 제어하는 명령어	commit, rollback, savepoint
7	Database	여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합	식별자, 튜플, 어트리뷰트
8	Field	같은 종류의 의미를 갖는 데이터들을 저장하기 위한 항목	번호, 제목, 글쓴이 등
9	Record	Field에 연관성 있게 저장되어 있는 데이터들의 집합체	
10	Table	각 데이터들을 저장하는 항목인 Field들로 구성된 데이터의 집합체	

#6

-컬럼 추가 [alter table TN add]

```
mysql> alter table directorydb add column Date date;  
Query OK, 0 rows affected (1.42 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from directorydb;
```

No	wording	Exp	Etc	Date
1	Data Modeling	모델링이란 현실 세계를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기술		NULL
2	DBMS	사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고, 데이터베이스를 관리해주는 소프트웨어	관계형, 객체지향형, 객체관계형	NULL
3	DML	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어	select, insert, update, delete	NULL
4	DDL	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어	create, alter, drop, rename, truncate	NULL
5	DCL	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어들	grant, revoke	NULL
6	TCL	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위_트랜잭션 별로 제어하는 명령어	commit, rollback, savepoint	NULL
7	Database	여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합	식별자, 튜플, 어트리뷰트	NULL
8	Field	같은 종류의 의미를 갖는 데이터들을 저장하기 위한 항목	번호, 제목, 글쓴이 등	NULL
9	Record	Field에 연관성 있게 저장되어 있는 데이터들의 집합체		NULL
10	Table	각 데이터들을 저장하는 항목인 Field들로 구성된 데이터의 집합체		NULL

```
10 rows in set (0.00 sec)
```


#6

-추가된 칼럼에 값 입력 [update TN set ?? where]

```
mysql> update directorydb set Date = '2020-03-31' where wording = 'Date Modeling' or wording = 'DBMS';
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from directorydb;
```

No	wording	Exp	Etc	Date
1	Data Modeling	모델링이란 현실 세계를 추상화, 단순화, 명확화하기 위해 일정한 표기법에 의해 표현하는 기술		2020-03-31
2	DBMS	사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고, 데이터베이스를 관리해주는 소프트웨어	관계형, 객체지향형, 객체관계형	2020-03-31
3	DML	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어	select, insert, update, delete	2020-03-31
4	DDL	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어	create, alter, drop, rename, truncate	2020-03-31
5	DCL	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어들	grant, revoke	Not Yet
6	TCL	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위_트랜잭션 별로 제어하는 명령어	commit, rollback, savepoint	Not Yet
7	Database	여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합	식별자, 튜플, 어트리뷰트	2020-03-30
8	Field	같은 종류의 의미를 갖는 데이터들을 저장하기 위한 항목	번호, 제목, 글쓴이 등	2020-03-30
9	Record	Field에 연관성 있게 저장되어 있는 데이터들의 집합체		2020-03-30
10	Table	각 데이터들을 저장하는 항목인 Field들로 구성된 데이터의 집합체		2020-03-30

#예습과제 - 2

• 아래 내용을 포함하는 백과사전 Table 만들기

1. Column은 No(AUTO_INCREMENT), 용어, 설명, 예제, 비고로 구성
2. 데이터는 아래 내용들을 조사해서 추가하기

- 1) PRIMARY KEY
- 2) FOREIGN KEY
- 3) SubQuery
- 4) View
- 5) Stored Procedure
- 6) JDBC(Java와 MySQL 연동)
- 7) Trigger

```
mysql> insert into directory(Wording, Contents, EXP, ETC) values
-> ('PRIMARY KEY', '다른 항목과 절대로 중복되어 나타날 수 없는 단일 값을 가지는 것', '주민등록번호', ''),
-> ('FOREIGN KEY', '테이블 내의 열 중 다른 테이블의 PRIMARY KEY를 참조하는 열을 FOREIGN KEY라 한다', '회원 정보 테이블을 참조하는 판매정보 테이블', ''),
-> ('SubQuery', '하나의 SQL문 안에 포함되어 있는 또 다른 SQL문', 'Single Row Subquery, Multiple Row Subquery', ''),
-> ('View', '데이터베이스에 존재하는 일종의 가상 테이블', 'create view 뷰이름 as select 필드이름1, 필드이름2 ... from 테이블이름 where 조건;', ''),
-> ('Stored Procedure', '여러 SQL 문을 하나의 SQL 문처럼 정리하여 call xx 라는 명령으로 실행할 수 있게 만든 것', 'create procedure 저장_프로시저_이름 begin SQL문 end;', ''),
-> ('JDBC_Connect_Java', '자바에서 데이터베이스를 사용하기 위한 절차에 대한 규약', '추상화된 인터페이스를 제공', ''),
-> ('Trigger', '테이블에 대한 이벤트가 발생했을 때 자동으로 실행되는 작업', 'CREATE TRIGGER [트리거이름] [BEFORE|AFTER] [INSERT|UPDATE|DELETE] ON [테이블이름] FOR EACH ROW [실행문];');
```

Query OK, 7 rows affected (0.07 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> select * from directory;

No	Wording	Contents	EXP	ETC
8	PRIMARY KEY	다른 항목과 절대로 중복되어 나타날 수 없는 단일 값을 가지는 것	주민등록번호	
9	FOREIGN KEY	테이블 내의 열 중 다른 테이블의 PRIMARY KEY를 참조하는 열을 FOREIGN KEY라 한다	회원 정보 테이블을 참조하는 판매정보 테이블	
10	SubQuery	하나의 SQL문 안에 포함되어 있는 또 다른 SQL문	Single Row Subquery, Multiple Row Subquery	
11	View	데이터베이스에 존재하는 일종의 가상 테이블	create view 뷰이름 as select 필드이름1, 필드이름2 ... from 테이블이름 where 조건;	
12	Stored Procedure	여러 SQL 문을 하나의 SQL 문처럼 정리하여 call xx 라는 명령으로 실행할 수 있게 만든 것	create procedure 저장_프로시저_이름 begin SQL문 end	
13	JDBC_Connect_Java	자바에서 데이터베이스를 사용하기 위한 절차에 대한 규약	추상화된 인터페이스를 제공	
14	Trigger	테이블에 대한 이벤트가 발생했을 때 자동으로 실행되는 작업	CREATE TRIGGER [트리거이름] [BEFORE AFTER] [INSERT UPDATE DELETE] ON [테이블이름] FOR EACH ROW [실행문]	

7 rows in set (0.00 sec)