

# 11강

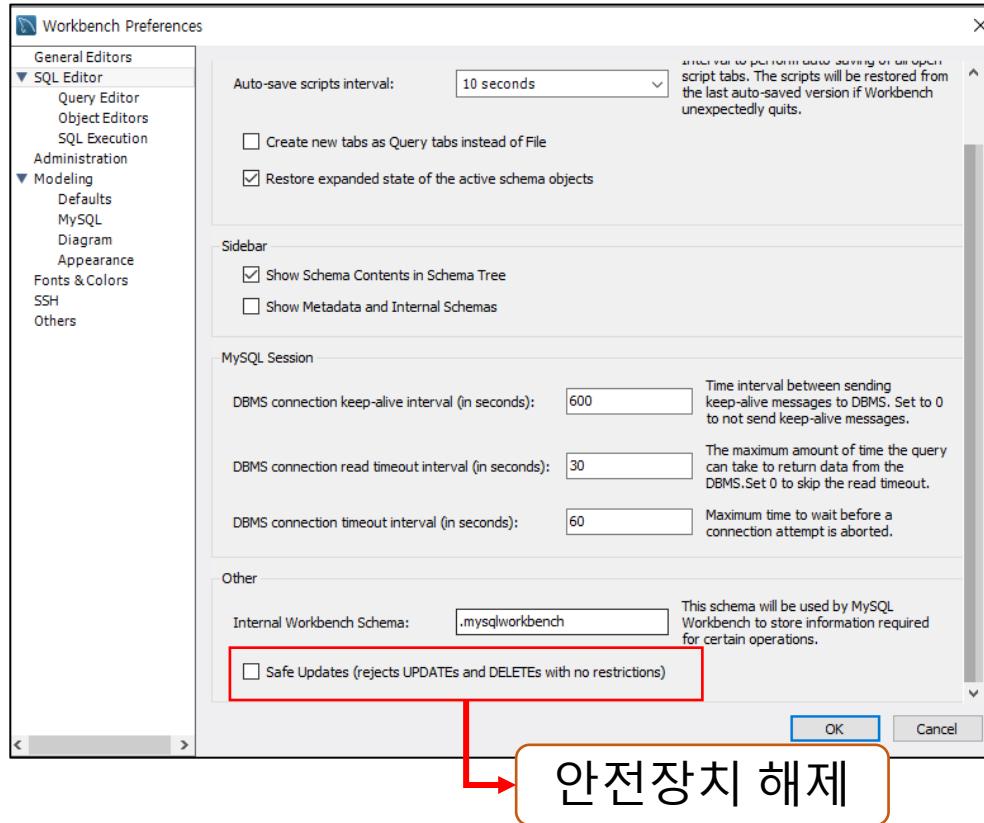
## 데이터변경 & 삭제





## 1. DELETE - 주어진 조건의 행 삭제하기

※ Edit > Preferences > SQL Editor > **Safe Updates** 항목 체크오프하고 다시 접속



### ➤ DELETE 형식

**DELETE FROM 테이블명 [WHERE 조건];**



DELETE FROM businesses WHERE status ='CLS';



	business_id	fk_section_id	business_name	status	can_takeout
▶	1	3	화룡각	OPN	1
	2	2	철구분식	OPN	1
	3	5	알코렐라	RMD	1
	4	2	바른떡볶이	OPN	1
	5	1	북극냉면	OPN	0
	6	1	보쌈마니아	OPN	1
	7	5	에그사라다	VCT	1
	8	6	달다방	OPN	1
	9	7	마카오마카롱	OPN	1
	10	2	김밥마라	OPN	1
	11	7	소소스윗	OPN	1
	12	4	사사서셔소소스시	VCT	1
	14	7	파시조아	OPN	1
	16	5	노선이탈리아	OPN	1
	17	6	커피앤크드	OPN	1
	18	2	신림동백순대	VCT	1

#### ➤ DELETE 문으로 행 전체 삭제 – 주의해야 할 부분

DELETE FROM businesses;



## DELETE FROM businesses;

Delete로 테이블의 레코드를 모두 삭제 후 insert into로 데이터 삽입 시 Auto\_increment의 마지막 번호를 인식하여 다음 번호부터 삽입됨을 주의하자 !!



	business_id	fk_section_id	business_name	status	can_takeout
▶	19	3	화伦각	OPN	1
	20	2	철구분식	OPN	1
	21	5	알코렐라	RMD	1
*	NULL	NULL	NULL	NULL	NULL

## ※ TRUNCATE 문으로 테이블 초기화

### TRUNCATE businesses;

truncate로 테이블을 모두 초기화 한 후 insert into로 데이터 삽입 시 Auto\_increment는 1번 부터 삽입됨을 주의 하자!!



	business_id	fk_section_id	business_name	status	can_takeout
▶	1	3	화伦각	OPN	1
	2	2	철구분식	OPN	1
	3	5	알코렐라	RMD	1
*	NULL	NULL	NULL	NULL	NULL



1. 속도를 비교하자면 **DELETE FROM** 이 가장 느리다.
2. 왜냐하면 **DELETE FROM** 은 DML(data manipulation language)로, **트랜잭션 로그**를 기록하는 작업 때문에 삭제가 오래 걸린다. 수백 만 건 또는 수천 만 건의 데이터를 삭제할 경우에 한참동안 삭제를 할 수도 있다.
3. DDL(data definition language)인 **DROP TABLE** 은 테이블을 통째로 지우며 트랜잭션을 발생시키지 않는다. **TRUNCATE** 또한 DDL문으로 효과는 **DELETE FROM** 과 동일하지만 트랜잭션 **로그를 기록하지 않아서 속도가 빠르다.**
4. 대용량의 테이블 전체 내용을 삭제할 때, 테이블 자체가 필요 없는 경우에는 **DROP**을, 테이블의 구조를 남기면서 트랜잭션 로그를 남기고 싶지 않다면 **TRUNCATE**를, 로그가 필요하다면 **DELETE**를 사용하면 된다.



## ☞ 여기서 잠깐 !!

green 스키마안에 <business>테이블의 status가 'VCT'인 데이터를 모두 삭제하는 SQL문을 작성하시오.(단, Delete문 사용)

출력 결과

	business_id	fk_section_id	business_name	status	can_takeout
▶	1	3	화롱각	OPN	1
	2	2	철구분식	OPN	1
	3	5	얄코렐라	RMD	1
	4	2	바른떡볶이	OPN	1
	5	1	북극냉면	OPN	0
	6	1	보쌈마니아	OPN	1
	8	6	달다방	OPN	1
	9	7	마카오마카롱	OPN	1
	10	2	김밥마라	OPN	1
	11	7	소소스윗	OPN	1
	13	3	린민짬뽕	CLS	1
	14	7	파시조아	OPN	1
	15	1	할매장국	CLS	0
	16	5	노선이탈리아	OPN	1
	17	6	커피앤클로	OPN	1
*	HULL	NULL	NULL	NULL	NULL



## 2. UPDATE - 주어진 조건의 행 수정하기

### ➤ UPDATE 형식

UPDATE 테이블명 SET 컬럼1 = 수정값1 [, 컬럼2 = 수정값2 ...] [WHERE 조건];

UPDATE menus SET menu\_name = '삼선짜장' WHERE menu\_id = 12;



	menu_id	fk_business_id	menu_name	kilocalories	price	likes
▶	1	5	물냉면	480.23	8000	3
	2	8	아메리카노	16.44	4500	6
	3	17	고르곤졸라피자	1046.27	12000	12
	4	6	보쌈	1288.24	14000	2
	5	15	장국	387.36	8500	-1
	6	17	까르보나라	619.11	9000	10
	7	9	바닐라마카롱	160.62	1500	4
	8	16	백순대	681.95	11000	24
	9	6	마늘보쌈	1320.49	16000	7
	10	16	양념순대볶음	729.17	12000	0
	11	14	단팥빵	225.88	1500	13
	12	1	삼선짜장	682.48	7000	3
	13	9	뚱카롱	247.62	2000	8
	14	5	비빔냉면	563.45	8000	4



## ※ 여러 컬럼 수정하기

```
UPDATE menus SET menu_name = '열정떡볶이', kilocalories = 492.78, price = 5000  
WHERE fk_business_id = 4 AND menu_name = '국물떡볶이';
```



	menu_id	fk_business_id	menu_name	kilocalories	price	likes
	13	9	뚱카롱	247.62	2000	8
	14	5	비빔냉면	563.45	8000	4
	15	10	참치김밥	532.39	3000	0
	16	2	치즈떡볶이	638.42	5000	15
	17	11	플레인와플	299.31	6500	2
	18	2	찹쌀순대	312.76	3000	-4
	19	15	육개장	423.18	8500	2
	20	4	열정떡볶이	492.78	5000	1
	21	10	돈가스김밥	562.72	4000	0
	22	1	삼선짬뽕	787.58	8000	32
	23	11	수플레팬케익	452.37	9500	5
▶	24	4	라볶이	423.16	5500	0



## ※ 컬럼 데이터 활용하여 수정하기

```
UPDATE menus SET price = price + 1000  
WHERE fk_business_id = 8;
```



	menu_id	fk_business_id	menu_name	kilocalories	price	likes
▶	1	5	물냉면	480.23	8000	3
	2	8	아메리카노	16.44	5500	6
	3	17	고르곤졸라피자	1046.27	12000	12
	4	6	보쌈	1288.24	14000	2
	5	15	장국	387.36	8500	-1
	6	17	까르보나라	619.11	9000	10
	7	9	바닐라마카롱	160.62	1500	4
	8	16	백순대	681.95	11000	24
	9	6	마늘보쌈	1320.49	16000	7
	10	16	양념순대볶음	729.17	12000	0
	11	14	단팥빵	225.88	1500	13
	12	1	삼선짜장	682.48	7000	3

## ※ 조건문 없이는 모든 행 변경

```
UPDATE menus SET menu_name = '훠 일화';
```



## ☞ 여기서 잠깐 !!

green 스키마안에 <menus> 테이블의 menu\_name이 떡볶이로 끝나는 모든 데이터의 price을 10%인상하여 수정하는 SQL문을 작성하시오.  
(단, Update문, Like함수 사용)

출력 결과

menu_id	fk_business_id	menu_name	kilocalories	price	likes
9	6	마늘보쌈	1320.49	16000	7
10	16	양념순대볶음	729.17	12000	0
11	14	단팥빵	225.88	1500	13
12	1	간짜장	682.48	7000	3
13	9	뚱카를	247.62	2000	8
14	5	비빔냉면	563.45	8000	4
15	10	참치김밥	532.39	3000	0
16	2	치즈떡볶이	638.42	5500	15
17	11	플레인와플	299.31	6500	2
18	2	찹쌀순대	312.76	3000	-4
19	15	육개장	423.18	8500	2
20	4	국물떡볶이	483.29	4950	1
21	10	돈가스김밥	562.72	4000	0
22	1	삼선짬뽕	787.58	8000	32
23	11	수풀레팬케익	452.37	9500	5
24	4	라볶이	423.16	5500	0
25	8	모카프라푸치노	216.39	6000	8
26	14	옛날팥빙수	382.35	8000	2

## 데이터 수정 - UPDATE

※ JOIN을 활용하여 Update(수정)하기 – 실무에서 많이 사용 함

update menus m

inner join businesses b on m.fk\_business\_id = b.business\_id

inner join sections s on s.section\_id = b.fk\_section\_id

set menu\_name = concat('전통',menu\_name)

where s.section\_name = '한식';

```
select b.business_id from menus m
inner join businesses b
on m.fk_business_id = b.business_id
inner join sections s
on s.section_id = b.fk_section_id
where s.section_name= '한식' ;
```

menu_id	fk_business_id	menu_name	kilocalories	price	likes
1	5	전통 물냉면	480.23	8000	3
2	8	아메리카노	16.44	4500	6
3	17	고르곤졸라피자	1046.27	12000	12
4	6	전통 보쌈	1288.24	14000	2
5	15	전통 장국	387.36	8500	-1
6	17	까르보나라	619.11	9000	10
7	9	바닐라마카롱	160.62	1500	4
8	16	백순대	681.95	11000	24
9	6	전통 마늘보쌈	1320.49	16000	7
10	16	양념순대볶음	729.17	12000	0
11	14	단팥빵	225.88	1500	13
12	1	간짜장	682.48	7000	3
13	9	동카를	247.62	2000	8
14	5	전통 비빔냉면	563.45	8000	4
15	10	참치김밥	532.39	3000	0
16	2	치즈떡볶이	638.42	5000	15
17	11	플레인와플	299.31	6500	2
18	2	찹쌀순대	312.76	3000	-4
19	15	전통 육개장	423.18	8500	2

<menus>테이블과 <businesses>테이블의 business\_id=fk\_business\_id가 같은 데이터를 먼저 추출하고, 추출한 결과 <sections> 테이블의 section\_id=fk\_section\_id가 같은 데이터 중 section\_name이 '한식'인 데이터의 business\_id 값을 출력한 결과이다.

business_id
15
6
5

## 코드로 확인하는 Update



### <inquiry> 수정 전

i_code	i_pw	p_code	i_title	i_content	m_name	i_date	i_readcount	ref	re_step	m_id
1	1234	56	파스텔 리빙박스 블루 입고문의	파스텔 리빙박스 블루 입고 언제 되...	오승현	2024-02-21	14	1	1	kjb1045
2	1234	35	상품 하나만 왔습니다	3단, 4단 각1개씩 시켰는데 3단 하...	오승현	2024-02-22	24	2	1	kjb1045
6	1234	79	휴지통 색이 두 종류만 있나요	사각 휴지통 색이 다양하지 않은 것...	진달래	2024-02-22	8	3	1	tom9867
9	1234	30	마인컬러 서랍장 4단 제품문의 드립니다	서랍장 4단말고 3단으로 된 것도 있...	오승현	2024-02-23	17	4	1	kjb1045
12	NULL	30	답변드립니다.	죄송합니다. 문의하신 상품은 재고...	리빙관리자	2024-03-05	4	4	2	admin
13	NULL	79	답변드립니다.	네~ 딱 2가지 색의 상품만 판매 할...	리빙관리자	2024-03-05	1	3	2	admin
14	NULL	35	답변드립니다.	죄송합니다. 빠른 시일내에 배송해...	리빙관리자	2024-03-05	4	2	2	admin
15	NULL	56	답변드립니다.	문의하신 상품은 일주일 뒤 입고 ...	리빙관리자	2024-03-05	2	1	2	admin

### <inquiry> 수정 후

i_code	i_pw	p_code	i_title	i_content	m_name	i_date	i_readcount	ref	re_step	m_id
1	1234	56	파스텔 리빙박스 블루 입고문의	파스텔 리빙박스 블루 입고 언제 되...	오승현	2024-02-21	14	1	1	kjb1045
2	1234	35	상품 하나만 왔습니다	3단, 4단 각1개씩 시켰는데 3단 하...	오승현	2024-02-22	24	2	1	kjb1045
6	1234	79	휴지통 색이 두 종류만 있나요	사각 휴지통 색이 다양하지 않은 것...	진달래	2024-02-22	8	3	1	tom9867
9	1234	30	마인컬러 서랍장 4단 제품문의~~~	서랍장 4단말고 3단으로 된 것도 있...	오승현	2024-02-23	17	4	1	kjb1045
12	NULL	30	답변드립니다.	죄송합니다. 문의하신 상품은 재고...	리빙관리자	2024-03-05	4	4	2	admin
13	NULL	79	답변드립니다.	네~ 딱 2가지 색의 상품만 판매 할...	리빙관리자	2024-03-05	1	3	2	admin
14	NULL	35	답변드립니다.	죄송합니다. 빠른 시일내에 배송해...	리빙관리자	2024-03-05	4	2	2	admin
15	NULL	56	답변드립니다.	문의하신 상품은 일주일 뒤 입고 ...	리빙관리자	2024-03-05	2	1	2	admin

#### 코드 예제

```
public void updateInquiryBoard(int code, String subject, String content) {
    getConnect();
    try {
        String sql = "update inquiry set i_title=?,i_content=? where i_code=?";
        // SQL 실행 코드
    }
}
```



UI 적용된 화면

## 상품 문의

상품 Q&amp;A입니다.



[모음전] 마인 컬러 서랍장 ( 4종 )

마인컬러 서랍장 4단 제품문의 드립니다.

오승현 2024-02-23 조회 17

서랍장 4단말고 3단으로 된 것도 있나요?

삭제

수정

## 상품 문의

제목  
마인컬러 서랍장 4단 제품문의~~~본문  
서랍장 4단말고 3단으로 된 것도 있나요?

번호

상품정보

## 상품 문의

상품 Q&amp;A입니다.

8

[모음전] 마인 컬러 서랍장 ( 4종 )  
마인컬러 서랍장 4단 제품문의~~~

작성자

작성일

조회

오승현

2024-02-23

17

7



[RE] 답변드립니다.

작성자

작성일

조회

리빙관리자

2024-03-05

4

취소

등록

# 11강

## 연습문제





## 1. greendb 스키마를 이용해 아래 <조건>에 만족하도록 SQL문을 작성하시오.

### 조건

- ① <menus> 테이블에서 likes 값이 0보다 작은 메뉴는 데이터 오류로 판단한다.
- ② 해당 메뉴의 likes 값을 모두 0으로 수정하시오.  
(단, Update문 사용)

### 출력 결과

menu_id	fk_business_id	menu_name	kilocalories	price	likes
3	17	고르곤졸라피자	1046.27	12000	12
4	6	보쌈	1288.24	14000	2
5	15	장국	387.36	8500	0
6	17	까르보나라	619.11	9000	10
7	9	바닐라마카롱	160.62	1500	4
8	16	백순대	681.95	11000	24
9	6	마늘보쌈	1320.49	16000	7
10	16	양념순대볶음	729.17	12000	0
11	14	단팥빵	225.88	1500	13
12	1	간짜장	682.48	7000	3
13	9	똥카롱	247.62	2000	8
14	5	비빔냉면	563.45	9000	4
15	10	참치김밥	532.39	3000	0
16	2	치즈떡볶이	638.42	5500	15
17	11	플레인와플	299.31	6500	2
18	2	찹쌀순대	312.76	3000	0
19	15	육개장	423.18	8500	2
20	4	국물떡볶이	483.29	4950	1
21	10	돈가스김밥	562.72	4000	0
22	1	삼선짬뽕	787.58	8000	32
23	11	수플레팬케익	452.37	9500	5
24	4	라면	423.16	5500	0
25	8	모카프라푸치노	216.39	6000	8
26	14	옛날팥빙수	382.35	8000	2



## 2. greendb 스키마를 이용해 아래 <조건>에 만족하도록 SQL문을 작성하시오.

### 조건

- ① <businesses> 테이블에서 상태(status)가 'CLS'(폐업) 인 가게들은 더 이상 포장 주문을 받을 수 없다.
- ② 해당 가게들의 can\_takeout 값을 0으로 일괄 수정하시오.  
(단, Update문 사용)

### 출력 결과

	business_id	fk_section_id	business_name	status	can_takeout
▶	1	3	화릉각	OPN	1
	2	2	철구분식	OPN	1
	3	5	얄코렐라	RMD	1
	4	2	바른떡볶이	OPN	1
	5	1	북극냉면	OPN	0
	6	1	보쌈마니아	OPN	1
	7	5	에그사라다	VCT	1
	8	6	달다방	OPN	1
	9	7	마카오마카롱	OPN	1
	10	2	김밥마라	OPN	1
	11	7	소소스윗	OPN	1
	12	4	사사서셔소쇼스시	VCT	1
	13	3	린민짬뽕	CLS	0
	14	7	파시조아	OPN	1
	15	1	할매장국	CLS	0
	16	5	노선이탈리아	OPN	1
	17	6	커피앤크드	OPN	1
	18	2	신림동백순대	VCT	1



### 3. greendb 스키마를 이용해 아래 <조건>에 만족하도록 SQL문을 작성하시오.

#### 조건

- ① <sections> 테이블의 section\_name = '카페' 이고, <menus> 테이블에서 가격(price)이 5000원 미만인 메뉴들의 가격을 5000원으로 수정하시오.
- ② (힌트: businesses → sections JOIN 필요, Update문 사용)

#### 출력 결과

	menu_id	fk_business_id	menu_name	kilocalories	price	likes
▶	1	5	쫄냉면	480.23	9000	3
	2	8	아메리카노	16.44	5000	6
	3	17	고르곤졸라피자	1046.27	12000	12
	4	6	보쌈	1288.24	14000	2
	5	15	장국	387.36	8500	0
	6	17	까르보나라	619.11	9000	10
	7	9	바닐라마카롱	160.62	1500	4
	8	16	백순대	681.95	11000	24
	9	6	마늘보쌈	1320.49	16000	7
	10	16	양념순대볶음	729.17	12000	0
	11	14	단팥빵	225.88	1500	13
	12	1	간짜장	682.48	7000	3
	13	9	뚱카롱	247.62	2000	8
	14	5	비빔냉면	563.45	9000	4
	15	10	참치김밥	532.39	3000	0
	16	2	치즈떡볶이	638.42	5500	15
	17	11	플레인와플	299.31	6500	2
	18	2	찹쌀순대	312.76	3000	0
	19	15	육개장	423.18	8500	2
	20	4	국물떡볶이	483.29	4950	1
	21	10	돈가스김밥	562.72	4000	0
	22	1	삼선짬뽕	787.58	8000	32
	23	11	수플레팬케익	452.37	9500	5
	24	4	라볶이	423.16	5500	0
	25	8	모카프라푸치노	216.39	6000	8
	26	14	옛날팥빙수	382.35	8000	2



## 4. greendb 스키마를 이용해 아래 &lt;조건&gt;에 만족하도록 SQL문을 작성하시오.

## 조건

- ① ratings 테이블에서 별점(stars)이 2점 이하인 리뷰가 달린 가게는 서비스 개선 필요로 표시한다.
- ② 해당 가게(businesses)의 status 값을 'RMD'로 수정하시오.
- ③ (힌트: ratings → businesses JOIN, Update문 사용)

출력 결과

	business_id	fk_section_id	business_name	status	can_takeout
▶	1	3	화룡각	OPN	1
	2	2	철구분식	OPN	1
	3	5	알코렐라	RMD	1
	4	2	바른떡볶이	OPN	1
	5	1	북극냉면	OPN	0
	6	1	보쌈마니아	OPN	1
	7	5	에그사라다	VCT	1
	8	6	달다방	OPN	1
	9	7	마카오마카롱	RMD	1
	10	2	김밥마라	OPN	1
	11	7	소소스윗	OPN	1
	12	4	사사서셔소쇼스시	VCT	1
	13	3	린민짬뽕	CLS	0
	14	7	파시조아	OPN	1
	15	1	할매장국	CLS	0
	16	5	노선이탈리아	OPN	1
	17	6	커피앤클로	OPN	1
	18	2	신림동백순대	VCT	1