

7강

자료형





1. 숫자 자료형

정수

자료형	바이트	SIGNED(양수/음수)	UNSIGNED(양수)
TINYINT	1	-128 ~ 127	0 ~ 255
SMALLINT	2	-32,768 ~ 32,767	0 ~ 65,535
MEDIUMINT	3	-8,388,608 ~ 8,388,607	0 ~ 16,777,215
INT	4	약 -21억 ~ +21억	0 ~ 4,294,967,295
BIGINT	8	약 -900경 ~ +900경	0 ~ $2^{64}-1$

고정 소수점(Fixed Point) 수

- 좁은 범위의 수 표현 가능, 정확한 값

자료형	설명	범위
DECIMAL(s, d)	실수 부분 총 자릿수(s) & 소수 부분 자릿수 (d)	가변 최대 65자리

예) decimal(5,2) => -999.99 ~ 999.99까지의 실수를 나타낼 수 있다.



부동 소수점(Floating Point) 수

- 넓은 범위의 수 표현 가능, 정확하지 않은 값 (일반적으로 충분히 정확)

자료형	표현 범위	설명
FLOAT(4byte)	-3.40E+38 ~ -1.17E-38	소수점 아래 7자리까지 표현
DOUBLE(8byte)	1.22E-308 ~ 1.79E+308	소수점 아래 15자리까지 표현



2. 문자 자료형

문자열

자료형	설명	차지하는 바이트	최대 바이트
CHAR(S)	고정 사이즈(남는 글자 스페이스로 채움)	S (고정값)	255
VARCHAR(S)	가변 사이즈	실제 글자 수	65,535

- 검색 시 CHAR 가 더 빠름
- VARCHAR 컬럼 길이 값이 4글자 보다 적을 경우 CHAR로 자동 변환
- **BLOB(Binary Large Object)**는 사진, 동영상, 문서 파일 등의 **대용량** 이진 데이터를 저장하는데 사용한다.

텍스트

자료형	최대 바이트 크기
TINYTEXT	255
TEXT	65,535
MEDIUMTEXT	16,777,215
LONGTEXT	4,294,967,295



자료형 CHAR와 VARCHAR은 문자열을 포함할 때 사용하는 자료형으로 사용할 때 길이를 명시해 주어야 한다. CHAR 타입은 고정 사이즈이기 때문에 CHAR(20)으로 했을 경우 20byte만큼만 공간이 할당되기 때문에 20byte를 넘어가면 데이터가 잘리게 된다. 이러한 특성은 VARCHAR도 동일하다.

그렇다면 차이점은 무엇일까 문자열 자료형 CHAR은 고정형 문자열이기 때문에 CHAR(20)일때 2byte의 문자만 넣어도 20byte만큼의 데이터를 잡게 된다. 반면 VARCHAR(Variable + char) 는 가변형 문자열이기때문에 데이터의 길이에 따라서 가변으로 길이가 정해진다.

VARCHAR(20)일 경우 입력한 크기만큼의 공간만 잡히는데, 2byte의 문자를 넣으면 2byte만큼의 데이터만 잡게 된다.

고정 사이즈인 CHAR 타입은 추후 연산할 필요가 없기 때문에 검색속도 및 읽히는 속도가 다른 타입보다 월등히 빠르다, 사이즈가 고정되어 있을때 CHAR 타입을 사용하면 더 효율적으로 데이터를 관리할 수 있다.

예) (아이디, 주민등록번호...)

예를 들어 아이디를 만드는데 데이터 베이스에 자료형이 char(100)으로 잡혀 있다면, sogood(6byte)이라는 아이디를 만들 때 6byte를 잡는데 94byte의 데이터가 낭비된다. 따라서 상황에 따라 맞는 타입을 설정해야 한다.



3. 시간 자료형

자료형	설명	비고
DATE	YYYY-MM-DD	
TIME	HHH:MI:SS	HHH : -838 ~ 838까지의 시간
DATETIME	YYYY-MM-DD HH:MI:SS	입력된 시간을 그 값 자체로 저장
TIMESTAMP	YYYY-MM-DD HH:MI:SS	<u>MySQL이 설치된 컴퓨터의 시간대를 기준으로 저장</u>

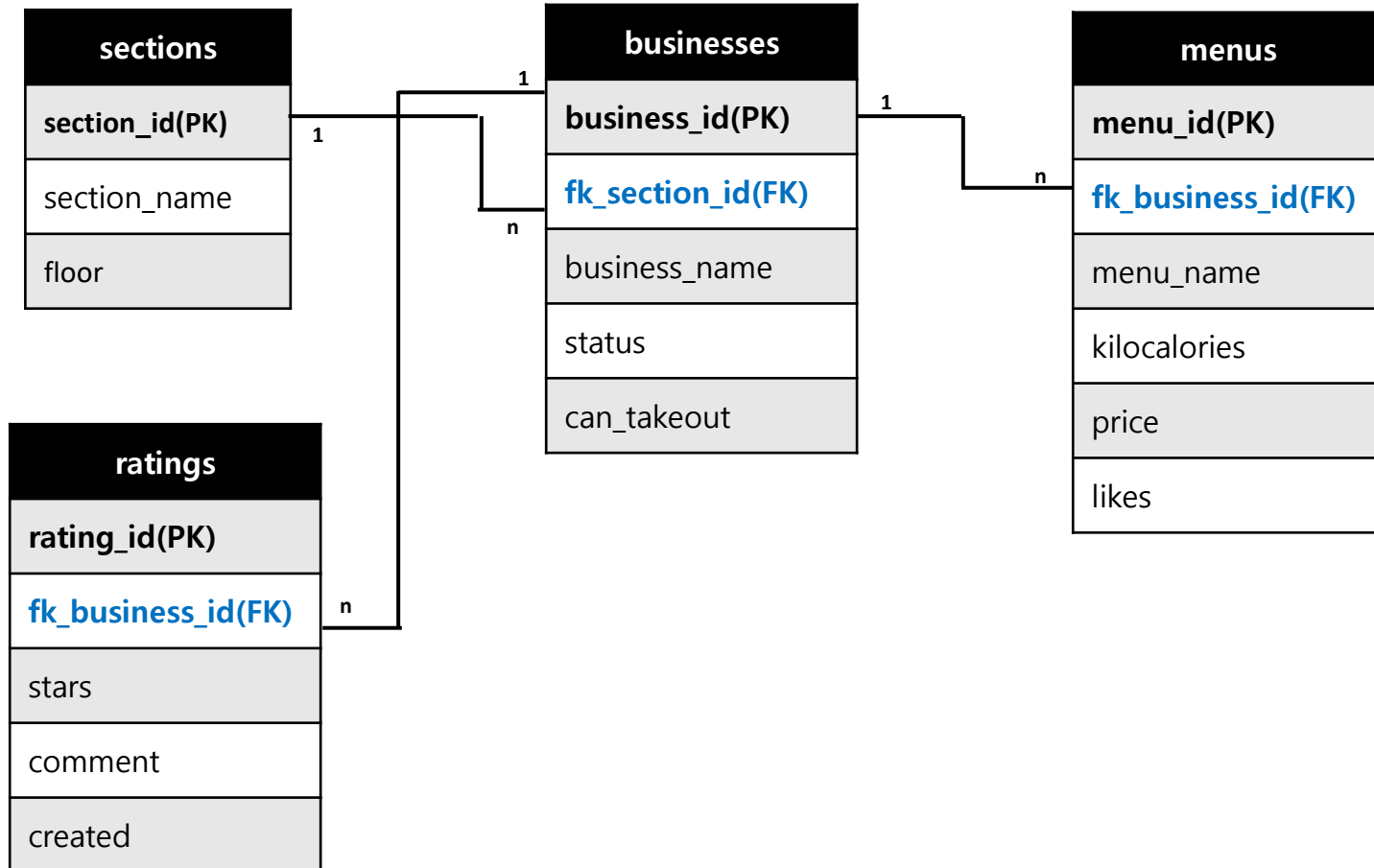
- 시간 데이터를 가감없이 기록할 때 DATETIME
- 시간 자동기록, 국제적인 서비스를 할 경우 TIMESTAMP 사용



함수	설명
DATETIME	<ul style="list-style-type: none"> 날짜와 시간을 함께 저장할 수 있는 타입으로, 기본 포맷은 'YYYY-MM-DD HH:mm:ss'이며, '1000-01-01 00:00:00' 부터 '9999-12-31 23:59:59'까지 저장할 수 있다. DATETIME은 데이터 값을 입력해주어야만 날짜가 입력이 된다. 시스템의 TIME_ZONE이 변경되어도 값이 변하지 않는다. 문자형으로 저장된다. <u>8바이트의 저장 공간을 사용한다.</u>
TIMESTAMP	<ul style="list-style-type: none"> 날짜와 시간을 타임스태프 형식으로 저장하는 타입으로, 날짜는 1001-01-01 00:00:00 ~ 9999-12-31 23:59:59까지 저장되며, 입력한 데이터가 유효한 날짜와 시간이 아니면 세 타입 모두 0으로 저장된다 TIMESTAMP는 사용자의 별도 입력이 없으면 데이터가 마지막으로 입력되거나 변경된 시간이 저장되며 기본적으로 NOT NULL이다. TIMESTAMP는 시스템의 TIME_ZONE에 의존하며, <u>UTC 시간대로 변환하여 저장 한다.</u> 숫자형으로 저장된다 <u>4바이트의 저장 공간을 사용한다.</u> UTC 란 ? 협정 세계시(UTC, Coordinated Universal Time)로서 1972년부터 시행된 <u>국제 표준시</u>이다



<menus>,<businesses>,<sections>,<ratings> 테이블 간의 관계설정 물리모델





여기서 잠깐 !!

green 스키마안에 <sections> 테이블을 생성하고, 아래 테이블 구조를 보고 데이터 타입과 제약조건을 지정하고 insert into로 데이터 삽입하시오.

테이블 구조

테이블이름	sections	테이블 한글이름				
한글컬럼명	Field	Type	Null	Key	Default	Extra
아이디	Section_id	int		PK		AUTO_INCREMENT
이름	Section_name	char(3)	Not null			
floor	floor	Tinyint	Not null			



	section_id	section_name	floor
▶	1	한식	2
	2	분식	2
	3	중식	3
	4	일식	3
	5	양식	3
	6	카페	1
	7	디저트	1
*	NULL	NULL	NULL

```

INSERT INTO sections (section_name, floor)
VALUES
('한식', 2),
('분식', 2),
('중식', 3),
('일식', 3),
('양식', 3),
('카페', 1),
('디저트', 1);

```




여기서 잠깐 !!

green 스키마안에 <ratings> 테이블을 생성하고, 아래 테이블 구조를 보고 데이터 타입과 제약조건을 지정하고 insert into로 데이터 삽입하시오.

테이블 구조

테이블이름	ratings	테이블 한글이름	평가			
한글컬럼명	Field	Type	Null	Key	Default	Extra
아이디	rating_id	int		PK		AUTO_INCREMENT
fk아이디	fk_business_id	int	Not null			
별점	stars	tinyint	Not null			
평가	comment	varchar(200)				
작성일자	created	timestamp	Not null		current_timestamp()	

rating_id	fk_business_id	stars	comment	created
1	2	4	치떡이 진리. 순대는 별로	2021-07-01 12:30:04
2	16	3	그냥저냥 먹을만해요	2021-07-01 17:16:07
3	14	5	인생팔빵. 말이 필요없음	2021-07-03 11:28:12
4	5	3	육수는 괜찮은데 면은 그냥 ...	2021-07-04 19:03:50
5	11	4	나오는데 넘 오래걸림. 맛은...	2021-07-04 13:37:42
6	9	2	빵집에서 파는 마카롱이랑 ...	2021-07-06 15:19:23
7	16	5	신림에서 먹던 맛 완벽재현	2021-07-06 20:01:39
8	7	3	그냥 그렇다	2023-05-01 19:23:08



👉 여기서 잠깐 !!

```
INSERT INTO ratings (fk_business_id, stars, comment, created)
```

```
VALUES
```

```
(2, 4, '치떡이 진리. 순대는 별로', '2021-07-01 12:30:04'),
```

```
(16, 3, '그냥저냥 먹을만해요', '2021-07-01 17:16:07'),
```

```
(14, 5, '인생팔빵. 말이 필요없음', '2021-07-03 11:28:12'),
```

```
(5, 3, '육수는 괜찮은데 면은 그냥 시판면 쓴 것 같네요.', '2021-07-04 19:03:50'),
```

```
(11, 4, '나오는데 넘 오래걸림. 맛은 있어요', '2021-07-04 13:37:42'),
```

```
(9, 2, '빵집에서 파는 마카롱이랑 비슷하거나 못합니다.', '2021-07-06 15:19:23'),
```

```
(16, 5, '신림에서 먹던 맛 완벽재현', '2021-07-06 20:01:39');
```

7강

연습문제





1. green 스키마안에 <menus> 테이블을 생성하고, 아래 화면처럼 출력되도록 데이터타입과 제약조건을 작성하고 insert into로 데이터를 삽입하시오.

INSERT INTO menus (fk_business_id, menu_name, kilocalories, price, likes)
VALUES

```
(5, '물냉면', 480.23, 8000, 3),
(8, '아메리카노', 16.44, 4500, 6),
(17, '고르곤졸라피자', 1046.27, 12000, 12),
(6, '보쌈', 1288.24, 14000, 2),
(15, '장국', 387.36, 8500, -1),
(17, '까르보나라', 619.11, 9000, 10),
(9, '바닐라마카롱', 160.62, 1500, 4),
(16, '백순대', 681.95, 11000, 24),
(6, '마늘보쌈', 1320.49, 16000, 7),
(16, '양념순대볶음', 729.17, 12000, 0),
(14, '단팔빵', 225.88, 1500, 13),
(1, '간짜장', 682.48, 7000, 3),
(9, '똥카롱', 247.62, 2000, 8),
(5, '비빔냉면', 563.45, 8000, 4),
(10, '참치김밥', 532.39, 3000, 0),
(2, '치즈떡볶이', 638.42, 5000, 15),
(11, '플레인와플', 299.31, 6500, 2),
(2, '참쌀순대', 312.76, 3000, -4),
(15, '육개장', 423.18, 8500, 2),
(4, '국물떡볶이', 483.29, 4500, 1),
(10, '돈가스김밥', 562.72, 4000, 0),
(1, '삼선짬뽕', 787.58, 8000, 32),
(11, '수플레팬케익', 452.37, 9500, 5),
(4, '라볶이', 423.16, 5500, 0),
(8, '모카프라푸치노', 216.39, 6000, 8),
(14, '옛날팔빙수', 382.35, 8000, 2);
```



<menus>

	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	단팔빵	225.88	1500	13
	12	1	간짜장	682.48	7000	3
	13	9	똥카롱	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	참쌀순대	312.76	3000	-4
	19	15	육개장	423.18	8500	2
	20	4	국물떡볶이	483.29	4500	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	옛날팔빙수	382.35	8000	2
	NULL	NULL	NULL	NULL	NULL	NULL



2. green 스키마안에 <businesses> 테이블을 생성하고, 아래 화면처럼 출력되도록 데이터 타입과 제약조건을 작성하고 insert into로 데이터를 삽입하시오.

```
INSERT INTO businesses (fk_section_id, business_name, status, can_takeout)
```

VALUES

```
(3, '화룡각', 'OPN', 1),
(2, '철구분식', 'OPN', 1),
(5, '알코렐라', 'RMD', 1),
(2, '바른떡볶이', 'OPN', 1),
(1, '북극냉면', 'OPN', 0),
(1, '보쌈마니아', 'OPN', 1),
(5, '에그사라다', 'VCT', 1),
(6, '달다방', 'OPN', 1),
(7, '마카오마카롱', 'OPN', 1),
(2, '김밥마라', 'OPN', 1),
(7, '소소스윗', 'OPN', 1),
(4, '사사서쇼소쇼스시', 'VCT', 1),
(3, '린민짬뽕', 'CLS', 1),
(7, '파시조아', 'OPN', 1),
(1, '할매장국', 'CLS', 0),
(5, '노선이탈리아', 'OPN', 1),
(6, '커피앤크드', 'OPN', 1),
(2, '신림동백순대', 'VCT', 1);
```



<businesses>

	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	사사서쇼소쇼스시	VCT	1
	13	3	린민짬뽕	CLS	1
	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
*	NULL	NULL	NULL	NULL	NULL



3. green 스키마 안에 <menus>,<businesses> 테이블을 이용하여 아래 조건에 만족하도록 SQL 문을 작성하시오.

<조건>

1. <menus>테이블과 <businesses>테이블을 inner Join 한다.
2. Status가 'OPN'인 데이터만 추출한다.
3. 오른쪽 옆의 <출력 결과>과 같이 출력되도록 작성한다.

<출력 결과>

business_name	status	menu_name	kilocalories	price
커피앤코드	OPN	까르보나라	619.11	9,000
마카오마카롱	OPN	바닐라마카롱	160.62	1,500
노선이탈리아	OPN	백순대	681.95	11,000
보쌈마니아	OPN	마늘보쌈	1320.49	16,000
노선이탈리아	OPN	양념순대볶음	729.17	12,000
파시조아	OPN	단팥빵	225.88	1,500
화룡각	OPN	간짜장	682.48	7,000
마카오마카롱	OPN	통카롱	247.62	2,000
북극냉면	OPN	비빔냉면	563.45	8,000
김밥마라	OPN	참치김밥	532.39	3,000
철구분식	OPN	치즈떡볶이	638.42	5,000
소소스윗	OPN	플레인와플	299.31	6,500
철구분식	OPN	참쌀순대	312.76	3,000
바른떡볶이	OPN	국물떡볶이	483.29	4,500
김밥마라	OPN	돈가스김밥	562.72	4,000
화룡각	OPN	삼선짬뽕	787.58	8,000
소소스윗	OPN	수플레팬케익	452.37	9,500
바른떡볶이	OPN	라볶이	423.16	5,500
달다방	OPN	모카프라푸치노	216.39	6,000
파시조아	OPN	옛날팥빙수	382.35	8,000



4. green 스키마 안에 <sections>,<businesses>,<ratings> 테이블을 이용하여 아래 조건에 만족하도록 SQL문을 작성하시오.

<조건>

1. <sections>테이블과 <businesses>테이블, <ratings> 테이블을 inner Join 한다.
2. Starts는 Repeat('반복할 문자', 반복할 숫자)함수를 이용하여 '★' 을 starts 의 입력된 숫자만큼 반복하여 출력한다.
3. 아래의 <출력 결과>과 같이 출력되도록 작성한다.

<출력 결과>

section_name	business_name	starts	comment	created
분식	철구분식	★★★★	치떡이 진리. 순대는 별로	2021-07-01 12:30:04
양식	노선이탈리아	★★★	그냥저냥 먹을만해요	2021-07-01 17:16:07
디저트	파시조아	★★★★★	인생팔빵. 말이 필요없음	2021-07-03 11:28:12
한식	북극냉면	★★★	육수는 괜찮은데 면은 그냥 시판면 쓴 것 같네요.	2021-07-04 19:03:50
디저트	소소스윗	★★★★	나오는데 넘 오래걸림. 맛은 있어요	2021-07-04 13:37:42
디저트	마카오마카롱	★★	빵집에서 파는 마카롱이랑 비슷하거나 못합니다.	2021-07-06 15:19:23
양식	노선이탈리아	★★★★★	신림에서 먹던 맛 완벽재현	2021-07-06 20:01:39