

# [백문이불여일타] 데이터 분석을 위한 고급 SQL

본 슬라이드는 강의를 위한 자료입니다  
더 자세한 내용을 원하시면, [이곳](#)을 찾아주세요 :)

만든이: 데이터리안 SQL 팀 (S, 윤선미, 한예은)

# Intro

1. DML (Data Manipulation Language)
2. ERD (Entity Relationship Diagram)
3. 서브쿼리 Subquery
4. 윈도우함수 Window functions
5. 정규표현식 Regular Expressions
6. 사용자정의함수 User-defined function

DML

# DML이란?

DML : Data Manipulation Language

1. INSERT
2. UPDATE
3. DELETE

INSERT

# INSERT

## 1. 테이블 전체에 데이터 추가하는 방법

**INSERT INTO** 테이블명 **VALUES** (VALUE\_LIST);

-> 컬럼 순서대로 입력

# 예시

**Salary**

Id	Name	Salary	Date
----	------	--------	------

**INSERT INTO** Salary **VALUES** ('1', 'A', '250', '2020-03-31');

**Salary**

Id	Name	Salary	Date
1	A	250	2020-03-31

# INSERT

## 2. 값을 저장할 열 지정하기

```
INSERT INTO 테이블명 (COLUMN_LIST) VALUES (VALUE_LIST);
```

# 예시

**Salary**

Id	Name	Salary	Date
1	A	250	2020-03-31

```
INSERT INTO Salary (Id, Salary) VALUES ('2', '550');
```

**Salary**

Id	Name	Salary	Date
1	A	250	2020-03-31
2	Null	550	Null

- Null은 디폴트



UPDATE

# UPDATE

## 1. 컬럼 전체에 데이터 업데이트

**UPDATE** 테이블명 **SET** 컬럼 = 값;

# 예시

**Salary**

Id	Name	Salary
1	A	250
2	B	550

->

**Salary**

Id	Name	Salary
1	A	<b>350</b>
2	B	<b>650</b>

**UPDATE** Salary **SET** Salary = Salary + 100;

# UPDATE

## 2. 지정 행만 업데이트

**UPDATE** 테이블명 **SET** 컬럼 = 값 **WHERE** 조건식;

# 예시

**Salary**

Id	Name	Salary
1	A	250
2	B	550

->

**Salary**

Id	Name	Salary
1	A	250
2	B	<b>650</b>

**UPDATE** Salary **SET** Salary = Salary + 100 **WHERE** Id = 2;

DELETE

# DELETE

## 1. 테이블 전체에 데이터 삭제

**DELETE FROM** 테이블명 ;

# 예시

**Salary**

Id	Name	Salary
1	A	250
2	B	550

->

**Salary**

Id	Name	Salary

**DELETE FROM** Salary;

# DELETE

## 2. WHERE 조건에 일치하는 행 삭제

**DELETE FROM 테이블명 WHERE 조건식;**

# 예시

**Salary**

Id	Name	Salary
1	A	250
2	B	550

->

**Salary**

Id	Name	Salary
1	A	250

**DELETE FROM Salary WHERE Id = 2;**

# UPDATE, DELETE

문제풀이

# LeetCode 672. Swap Salary

<https://leetcode.com/problems/swap-salary/>

```
UPDATE 테이블명
SET 컬럼명 = CASE
                WHEN condition1 THEN value_if_condition1_true
                WHEN condition2 THEN value_if_condition2_true
                ELSE value_other_cases
            END
(WHERE 조건식) ;
```



# LeetCode 672. Swap Salary

# 예시

Salary

Id	Name	Salary
1	A	250
2	B	300
3	C	500
4	D	450
5	E	600
6	Z	350

->

Salary

Id	Name	Salary
1	가	250
2	나	300
3	다	500
4	라	450
5	마	600
6	확인필요	350

```
UPDATE Salary
```

```
SET Name = CASE
```

```
    WHEN Id = 1 THEN '가'
```

```
    WHEN Id = 2 THEN '나'
```

```
    WHEN Id = 3 THEN '다'
```

```
    WHEN Id = 4 THEN '라'
```

```
    WHEN Id = 5 THEN '마'
```

```
    ELSE '확인필요'
```

```
END;
```

# LeetCode 196. Delete Duplicate Emails

<https://leetcode.com/problems/delete-duplicate-emails/>

# 서브쿼리

```
DELETE FROM Person
WHERE Id NOT IN (Write a subquery which only
contains Ids should be deleted);
```

# LeetCode 196. Delete Duplicate Emails

# [심화] DELETE에 JOIN 활용하기

<https://www.mysqltutorial.org/mysql-delete-join/>

```
DELETE T1, T2  
FROM T1  
INNER JOIN T2 ON T1.key = T2.key  
WHERE condition;
```

# LeetCode 196. Delete Duplicate Emails

# [심화] DELETE에 JOIN 활용하기

<https://www.mysqltutorial.org/mysql-delete-join/>

```
DELETE t1, t2
FROM t1
INNER JOIN t2 ON t1.id = t2.ref
WHERE t1.id = 1;
```

**t1**

id
1
2
3

**t2**

id	ref
A	1
B	2
C	3

# LeetCode 196. Delete Duplicate Emails

# [심화] DELETE에 JOIN 활용하기

<https://www.mysqltutorial.org/mysql-delete-join/>

```
DELETE t1, t2
FROM t1
INNER JOIN t2 ON t1.id = t2.ref
WHERE t1.id = 1;
```

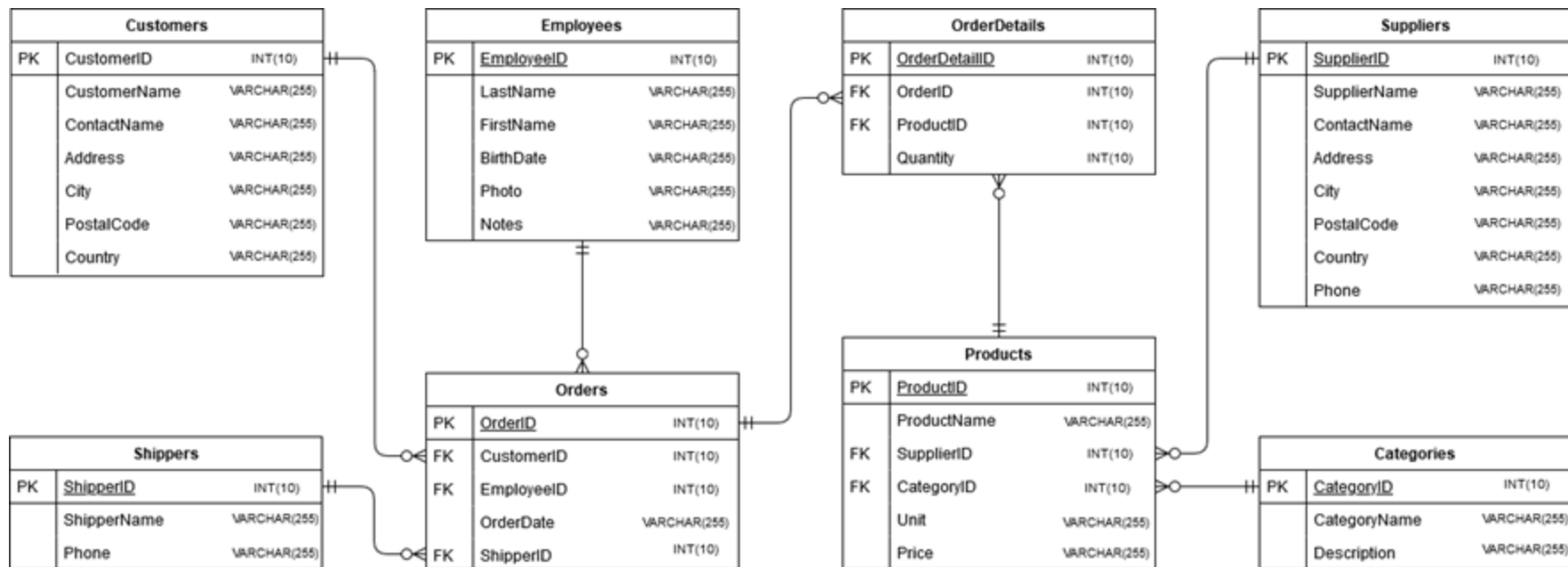
t1

id
4
2
3

t2

id	ref
A	4
B	2
C	3

ERD



# 구성 요소

- Entity
- Relationship
- Attribute
  - 컬럼 이름
  - 데이터 타입
    - 숫자
      - 정수 tinyint(), smallint(), mediumint(), int(), bigint()
      - 실수 decimal(), double(), float()
    - 문자 varchar(), various character 의 약자 / char()
    - 날짜, 시간
      - date() 1000-01-01 ~ 9999-12-31
      - datetime() 1000-01-01 00:00:00.000000 ~ 9999-12-31 23:59:59.999999
      - timestamp() = datetime() + timezone
      - Orders 테이블에서 OrderDate는 왜 varchar()?

<https://stackoverflow.com/questions/38677002/mysql-convert-string-to-datetime>



subquery

# From 절 서브쿼리

## crimes

incident_id	week	date	type	...
3001	2	2020-01-05	A	
3002	2	2020-01-05	A	
3003	2	2020-01-06	B	
3004	2	2020-01-07	B	
3005	2	2020-01-07	...	
3006	2	2020-01-07		
3007	2	2020-01-08		
3008	2	2020-01-09		
3009	2	2020-01-10		
3010	2	2020-01-11		
3011	2	2020-01-11		

```
SELECT daily_stats.week
      , AVG(daily_stats.incidents_daily)
FROM (
          SELECT week
            , date
            , COUNT(incident_id) AS incidents_daily
          FROM crimes
          GROUP BY week, date
        ) daily_stats
GROUP BY daily_stats.week
```

# WHERE 절 서브쿼리

```
SELECT *  
FROM crimes  
WHERE date = (SELECT MIN(date) FROM crimes)
```

```
SELECT *  
FROM crimes  
WHERE date IN (SELECT date FROM crimes ORDER BY date DESC LIMIT 5)
```

# Window Functions

# 모양새

- 함수 (컬럼) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)

# 집계 함수

- **MAX (컬럼) OVER (PARTITION BY 컬럼)**

예시출처: <https://leetcode.com/problems/department-highest-salary/>

## Employee

Id	Name	Salary	DepartmentId	MaxSalary
1	Joe	70000	1	90000
2	Jim	90000	1	90000
3	Henry	80000	2	80000
4	Sam	60000	2	80000
5	Max	90000	1	90000

```
SELECT Id
      , Name
      , Salary
      , DepartmentId
      , MAX(Salary) OVER (PARTITION BY DepartmentId) AS MaxSalary
FROM Employee
```

# 집계 함수

- **SUM (컬럼) OVER (ORDER BY 컬럼)**

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406

```
SELECT Id
      , Name
      , kg
      , Line
      , SUM(kg) OVER (ORDER BY Line) AS CumSum
FROM Elevator
```

# 집계 함수

- **SUM(컬럼) OVER (ORDER BY 컬럼 PARTITION BY 컬럼)**

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406
B	Julia	70	1	70
B	Saoirse	65	2	135

```
SELECT Id
      , Name
      , kg
      , Line
      , SUM(kg) OVER (ORDER BY Line PARTITION BY Id) AS CumSum
FROM Elevator
```



# 순위 정하기

- **ROW\_NUMBER()** , **RANK()** , **DENSE\_RANK()**

예시출처: <https://dev.mysql.com/doc/refman/8.0/en/window-function-descriptions.html>

val	row_number	rank	dense_rank
1	1	1	1
1	2	1	1
2	3	3	2
3	4	4	3
3	5	4	3
3	6	4	3
4	7	7	4
4	8	7	4
5	9	9	5

```
SELECT val
      , ROW_NUMBER() OVER (ORDER BY val) AS 'row_number'
      , RANK() OVER (ORDER BY val) AS 'rank'
      , DENSE_RANK() OVER (ORDER BY val) AS 'dense_rank'
FROM sample
```

# 데이터 위치 바꾸기

- **LEAD () , LAG ()**

- **LAG (컬럼) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**
- **LAG (컬럼, 간수) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**
- **LAG (컬럼, 간수, Default) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**
  
- **LEAD (컬럼) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**
- **LEAD (컬럼, 간수) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**
- **LEAD (컬럼, 간수, Default) OVER (PARTITION BY 컬럼 ORDER BY 컬럼)**

# 데이터 위치 바꾸기

예시출처: <https://leetcode.com/problems/rising-temperature/>

## Weather

Id	RecordDate	Temperature	lag	lead
1	2015-01-01	10	NULL	25
2	2015-01-02	25	10	20
3	2015-01-03	20	25	30
4	2015-01-04	30	20	28
5	2015-01-05	28	30	NULL

```
SELECT Id
       , RecordDate
       , Temperature
       , LAG(Temperature) OVER (ORDER BY RecordDate) AS 'lag'
       , LEAD(Temperature) OVER (ORDER BY RecordDate) AS 'lead'
FROM sample
```

# 데이터 위치 바꾸기

예시출처: <https://leetcode.com/problems/rising-temperature/>

## Weather

Id	RecordDate	Temperature	lag2	lead2
1	2015-01-01	10	NULL	20
2	2015-01-02	25	NULL	30
3	2015-01-03	20	10	28
4	2015-01-04	30	25	NULL
5	2015-01-05	28	20	NULL

```
SELECT Id
      , RecordDate
      , Temperature
      , LAG(Temperature, 2) OVER (ORDER BY RecordDate) AS 'lag2'
      , LEAD(Temperature, 2) OVER (ORDER BY RecordDate) AS 'lead2'
FROM sample
```

심화

# 누적합 | 윈도우 함수 사용 (1)

- **SUM (컬럼) OVER (ORDER BY 컬럼)**

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406

```
SELECT Id
      , Name
      , kg
      , Line
      , SUM(kg) OVER (ORDER BY Line) AS CumSum
FROM Elevator
```

# 누적합 | 윈도우 함수 사용 (2)

- **SUM(컬럼) OVER (ORDER BY 컬럼 PARTITION BY 컬럼)**

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406
B	Julia	70	1	70
B	Saoirse	65	2	135

```
SELECT Id
      , Name
      , kg
      , Line
      , SUM(kg) OVER (ORDER BY Line PARTITION BY Id) AS CumSum
FROM Elevator
```

# 누적합 | 윈도우 함수 이외의 방법(1)

- JOIN 활용

## Elevator

Id	Name	kg	Line	Id	Name	kg	Line
A	Joe	70	1	A	Joe	70	1
A	Jim	91	2	A	Joe	70	1
A	Jim	91	2	A	Jim	91	2
A	Henry	59	3	A	Joe	70	1
A	Henry	59	3	A	Jim	91	2
A	Henry	59	3	A	Henry	59	3

```
SELECT *  
FROM Elevator e1  
INNER JOIN Elevator e2  
    on e1.Id = e2.Id  
    and e1.Line >= e2.Line
```



# 누적합 | 윈도우 함수 이외의 방법(1)

- JOIN 활용

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406
B	Julia	70	1	70
B	Saoirse	65	2	135

```
SELECT e1.Id
      , e1.Name
      , e1.kg
      , e1.Line
      , SUM(e2.kg) AS CumSum
FROM Elevator e1
INNER JOIN Elevator e2
      ON e1.Id = e2.Id
      AND e1.Line >= e2.Line
GROUP BY 1,2,3,4
```

# 누적합 | 윈도우 함수 이외의 방법(2)

- **SELECT절 서브쿼리 활용**

## Elevator

Id	Name	kg	Line	CumSum
A	Joe	70	1	70
A	Jim	91	2	161
A	Henry	59	3	220
A	Sam	100	4	320
A	Max	86	5	406
B	Julia	70	1	70
B	Saoirse	65	2	135

```
SELECT e1.Id
      , e1.Name
      , e1.kg
      , e1.Line
      , (SELECT SUM(e2.kg)
        FROM Elevator e2
        WHERE e1.Id = e2.Id
        AND e1.Line >= e2.Line) AS CumSum
FROM Elevator e1
```

# User-Defined Function

사용자 정의 함수

# MySQL FUNCTION | 기본 구조

```
CREATE FUNCTION `function name` (`parameter name`, `datatype`)
    RETURNS `datatype` (DETERMINISTIC)
BEGIN
    DECLARE `variable name` `datatype`;
    SET ;
    RETURN (Query) / `variable name`;
END
```

- 사용 방법:

```
SELECT `function name` (parameter)
```

# MySQL FUNCTION | 예시

예시 출처: <https://www.mysqltutorial.org/mysql-stored-function/>

customers
* customerNumber customerName contactLastName contactFirstName phone addressLine1 addressLine2 city state postalCode country salesRepEmployeeNumber creditLimit

## customers

customerNumber	custsomerName	creditLimit	customerLevel
103	Atelier graphique	21000.00	GOLD
112	Signal Gift Stores	71800.00	PLATINUM
114	Australian Collectors, Co.	117300.00	PLATINUM
119	La Rochelle Gifts	118200.00	PLATINUM
121	Baane Mini Imports	81700.00	PLATINUM
124	Mini Gifts Distributors Ltd.	210500.00	PLATINUM
125	Havel & Zbyszek Co	0.00	SILVER

# MySQL FUNCTION | 예시

```
CREATE FUNCTION CustomerLevel(credit DECIMAL(10,2))
RETURNS VARCHAR(20) DETERMINISTIC
BEGIN
    DECLARE Level VARCHAR(20);
    IF credit > 50000 THEN
        SET Level = 'PLATINUM';
    ELSEIF (credit <= 50000 AND credit >= 10000) THEN
        SET Level = 'GOLD';
    ELSEIF credit < 10000 THEN
        SET Level = 'SILVER';
    END IF;
    -- return the customer level
    RETURN (Level);
END
```

- 사용 방법:

```
SELECT customerName,
        CustomerLevel(creditLimit)
FROM
    customers
ORDER BY
    customerName;
```

# LeetCode 177. Nth Highest Salary 힌트 1

<https://leetcode.com/problems/nth-highest-salary/>

MySQL CASE statement

- **CASE WHEN** condition **THEN NULL ELSE** value

# LeetCode 177. Nth Highest Salary 힌트 2

## MySQL IF function

- **IF**(condition, value\_if\_true, value\_if\_false)

## Example

```
SELECT IF(500<1000, "YES", "NO")
```

```
SELECT IF(500<1000, "YES", NULL)
```



# LeetCode 177. Nth Highest Salary 힌트 3

## LIMIT 심화

- **SELECT \* FROM table LIMIT 5, 10** # Retrieve rows 6~15
- **SELECT \* FROM table LIMIT 5, 1** # Retrieve rows 6
- **SELECT \* FROM table LIMIT N, 1** # Retrieve rows N+1  
= **SELECT \* FROM table LIMIT 1 OFFSET N**

서브쿼리 없이, **LIMIT**와 **OFFSET**을 이용해 풀어보세요!

N의 값을 바꾸고 싶다면, 변수를 선언하고 정의하는 **DECLARE, SET** statement를 활용하세요.

마무리

1. DML (Data Manipulation Language) [Leetcode 2문제](#)
2. ERD (Entity Relationship Diagram)
  - a. ERD 읽는 법
  - b. 데이터 타입
3. Subquery [Leetcode, HackerRank 3+문제](#)
4. Window functions [Leetcode 3문제](#)
5. Regular Expressions [HackerRank 4문제](#)
6. MySQL Function [Leetcode 1문제](#)