

SQL로 입문하는 데이터분석 레시피

LEE SANG YEOL

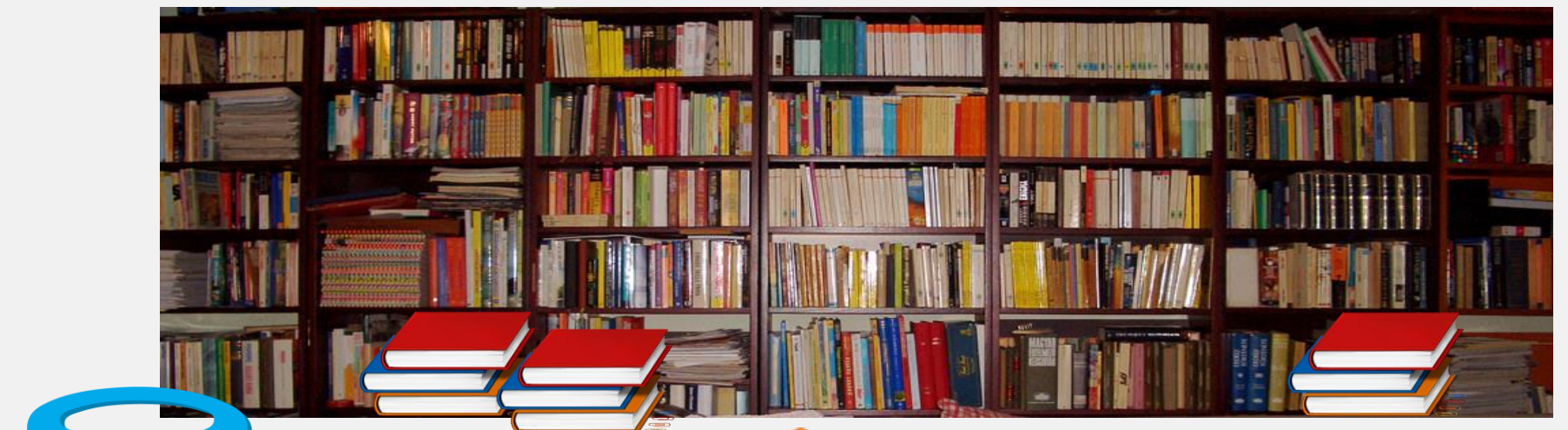
DATA ANALYST

syleeie@gmail.com

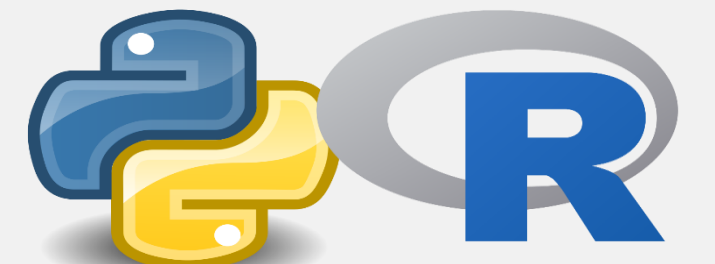
▲ 자기소개

- 데이터뽀개기 운영진 (이전 캐글뽀개기 커뮤니티 그룹)
- 석사 제외하고 5년 째 데이터 분석가로 업무 진행 중
- (현) 게임 회사 다니는 중

- 주로 진행했던 업무
 - 알고리즘 및 추천시스템 개발
 - 지표 및 로그 분석



SQL 데이터
분석 환경



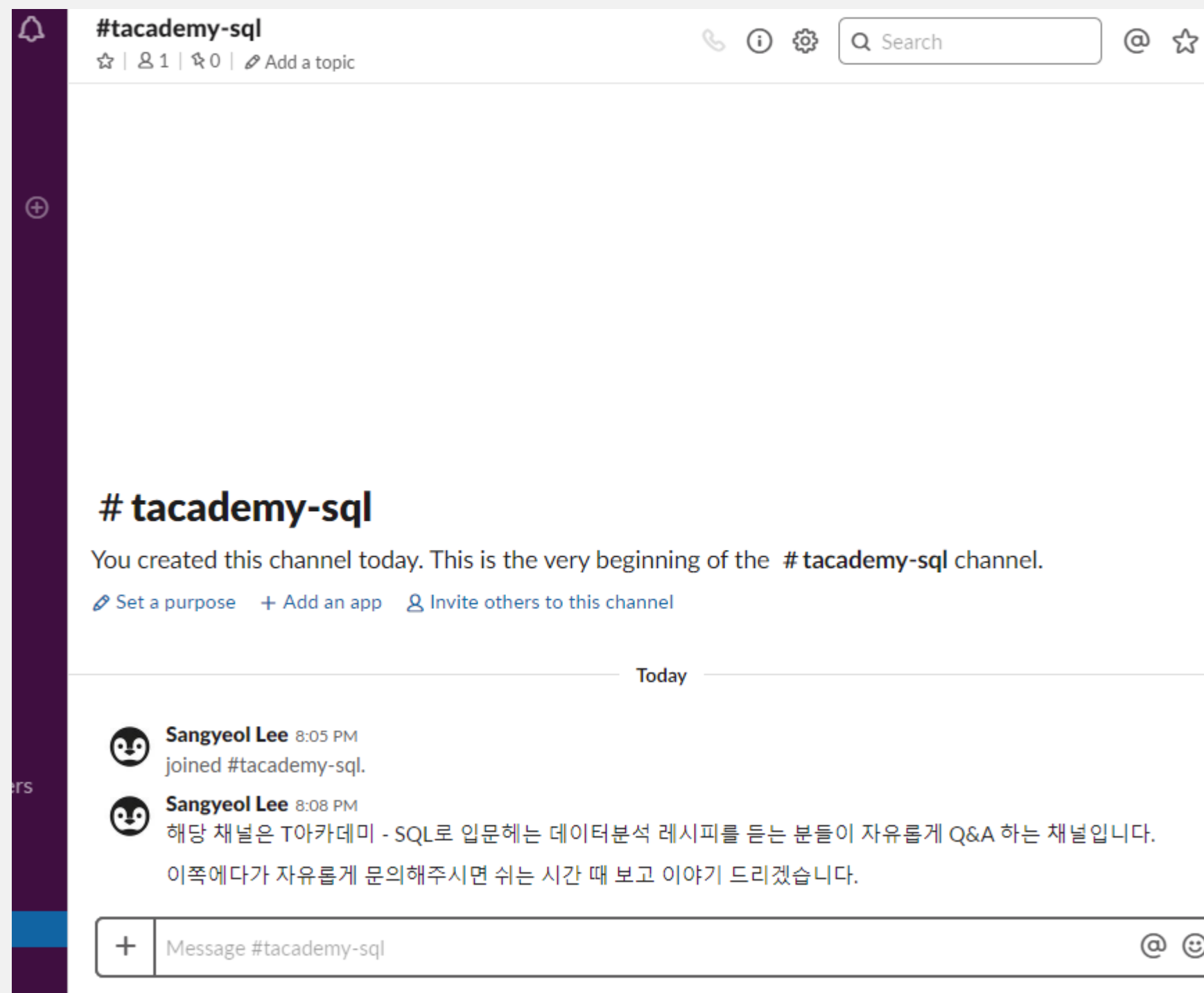
프로그래밍 언어
기반 분석 환경
(제플린, Jupyter
노트북)

조교 소개

- 김건희
 - 데이터보개기 운영자, 통계 전공



발표 질문 (Slack 채널 이용)

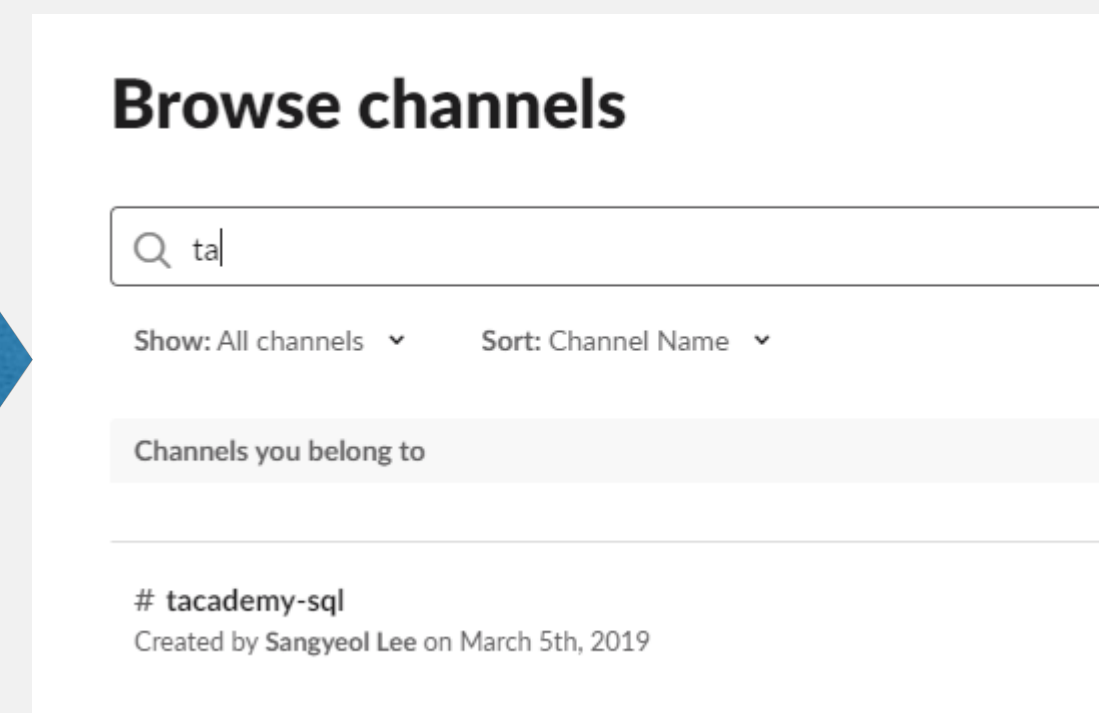
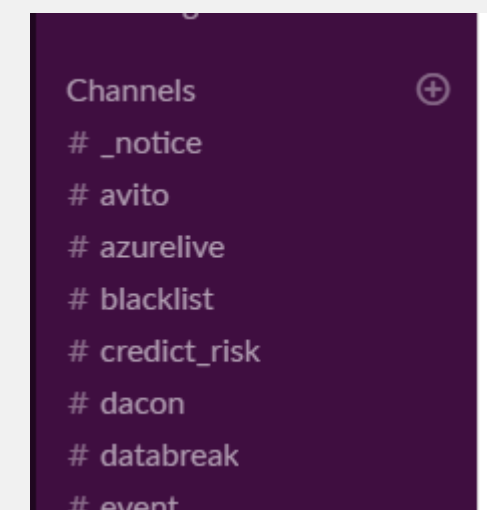


<데이터보내기 Slack 채널방>

슬랙 주소 : databreak.slack.com

가입 방법 : <https://kaggle-break.herokuapp.com/>

사이트 접속하셔서 이메일 적으시고,
토큰은 kagglebreak 입력하시면 초대 메일이 옵니다.
제한 : 누구나 가능



슬랙 사용법은 [여기](#) 참고하세요.

데이터뽀개기 커뮤니티 소개



캐글뽀개기

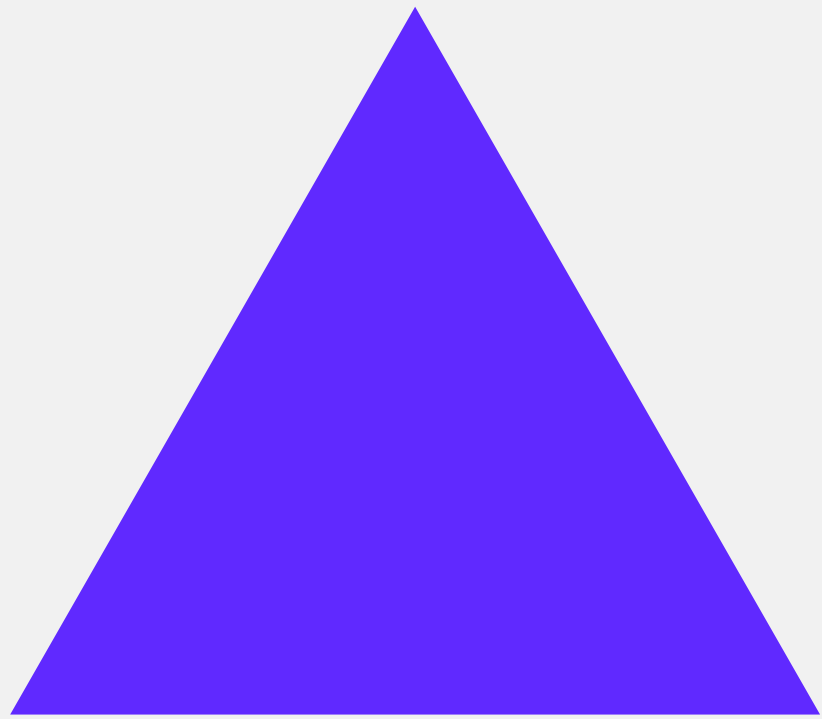
작년까지 캐글뽀개기 커뮤니티 이름이었습니다.



데이터뽀개기

<https://www.facebook.com/groups/databreak/>
비공개 페이스북 커뮤니티 그룹, 인원 약 3,700명

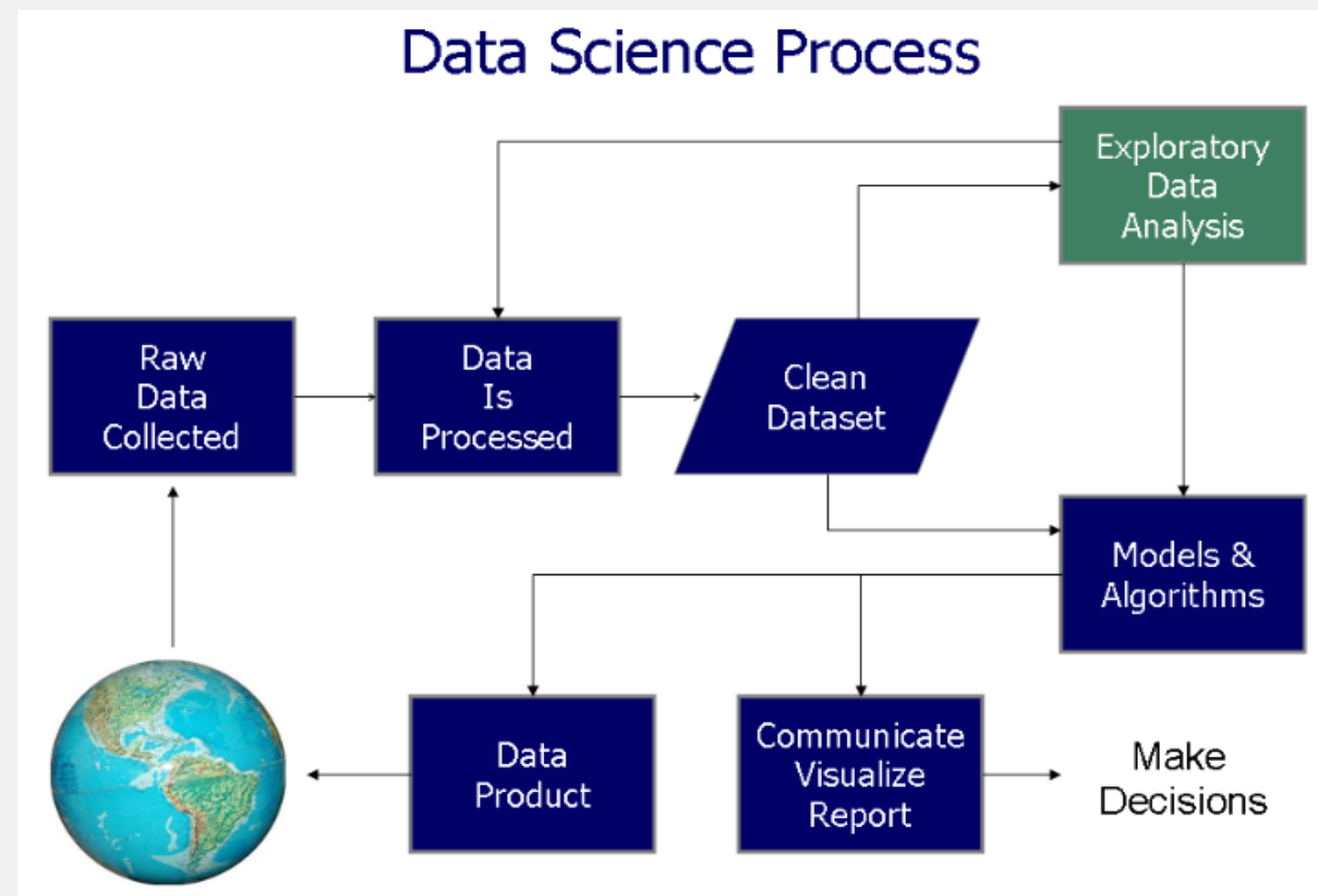
목적 : 데이터 현장에서 일하거나, 데이터 분야로
이직 / 취업 하려는 사람의 Practical Data Playground.
“오프라인, Meetup, Event, Study”
에서 데이터를 이야기하고, 연구할 수 있는 네트워킹 장소



1. 분석 입문 -

AARRRR 관점에서 데이터 분석이란

분석 입문



“The goal of a model is to provide a simple low-dimensional summary of a dataset”

Hadley Wickham

Data Driven 서비스는 시나리오 기반으로 비즈니스 모델의 개입하여 최적화하고 플랫폼 기능을 데이터(추천, 예측, 시뮬레이션 등)로 제공하는 것

인사이드는 가설기반 접근으로 고객의 경험을 숫자로 표현하는 것 (How to : 탐색, 모델링, 시각화)

▲ AARRR

Funnel 중에서 마케팅,
Growth 관련해서 가장 유명한
개념이 AARRR

실리콘 밸리의 스타트업 투자
및 엑셀러레이터인 500
Startups의
Dave McClure라는 사람이
이야기해서 유명해졌음



Funnel Analysis

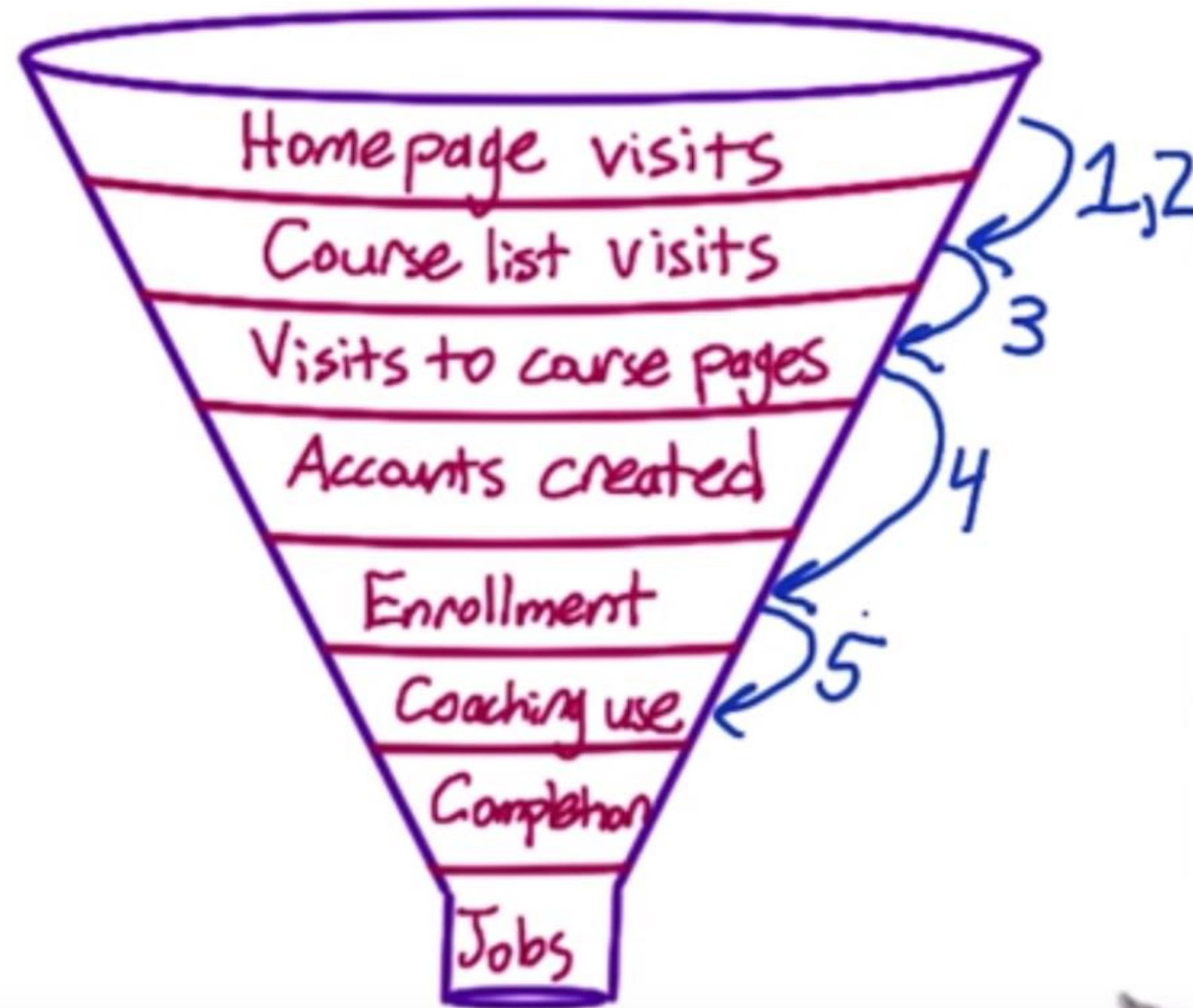
Ex) MOOC (Massive Open Online Course)

Business Objective : Helping students get jobs.
Financial sustainability

홈페이지를 접속한 **유니크** 숫자가 100,000명
강의 리스트를 접속한 **유니크** 숫자가 80,000명
계정을 만든 **유니크** 숫자가 60,000명
강의를 등록한 **유니크** 숫자가 40,000명
완료한 **유니크** 숫자가 5,000명
직업을 구한 **유니크** 숫자가 1,000명

Unit of diversion

- User Id, Anonymous ID(cookie), Event
- Device Id, IP address

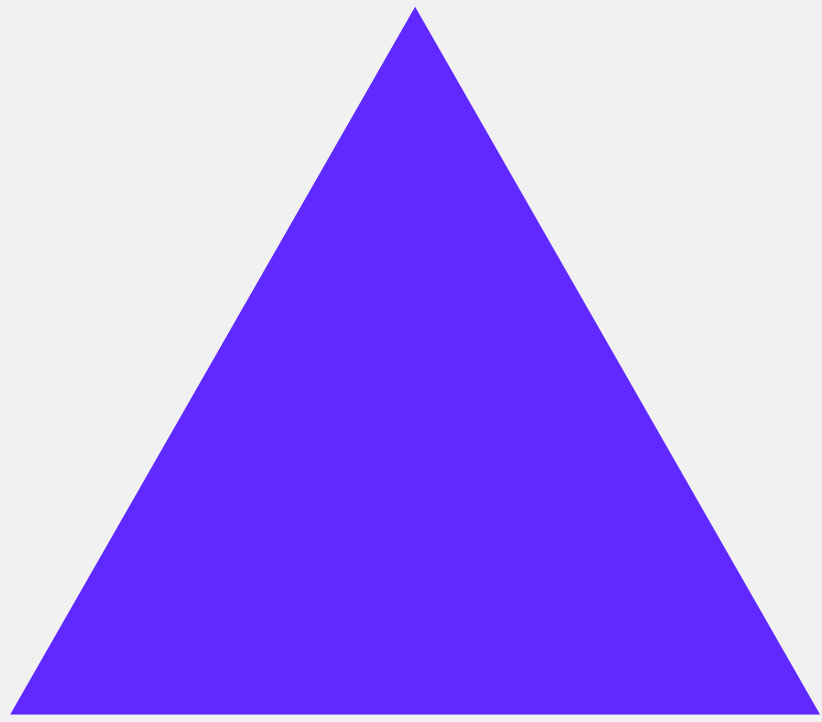




KPI 지표분석

KPI = Key Performance Indicator, 핵심 지표

분류	용어명	설명
유 저	DAU	Daily Active User. 일 방문자수(로그인 기준)
	액티브 유저	최근 n일 이내 방문 유저수
	휴면 유저	최근 n일 이내 방문 기록이 없는 유저
	잔존율	특정일의 유저를 100으로 가정했을 때, 특정일 이후 일별로 유입된 유저의 비율
매 출	매출액	유저가 결제한 금액
	BU	Buying User, 결제 유저 = Paying User (PU)라고도 부름
	BUR(구매전환율)	Buying User Ratio, 전체 유저 중 구매 유저로 전환된 비중
	ARPU	Average Revenue Per User, 전체 유저당 결제 금액
	ARPPU	Average Revenue Per Paying User, 결제 유저당 결제 금액



2. SQL 기초

- AWS분석환경 세팅
- 데이터베이스 기초
- Selecting columns & Filtering rows
- Aggregate Functions & Sorting, grouping and joins



이번 강의는 “분석가/기획자” 위주의 접근이 이루어집니다.



1. 데이터베이스에 대해서 다루지 않습니다.

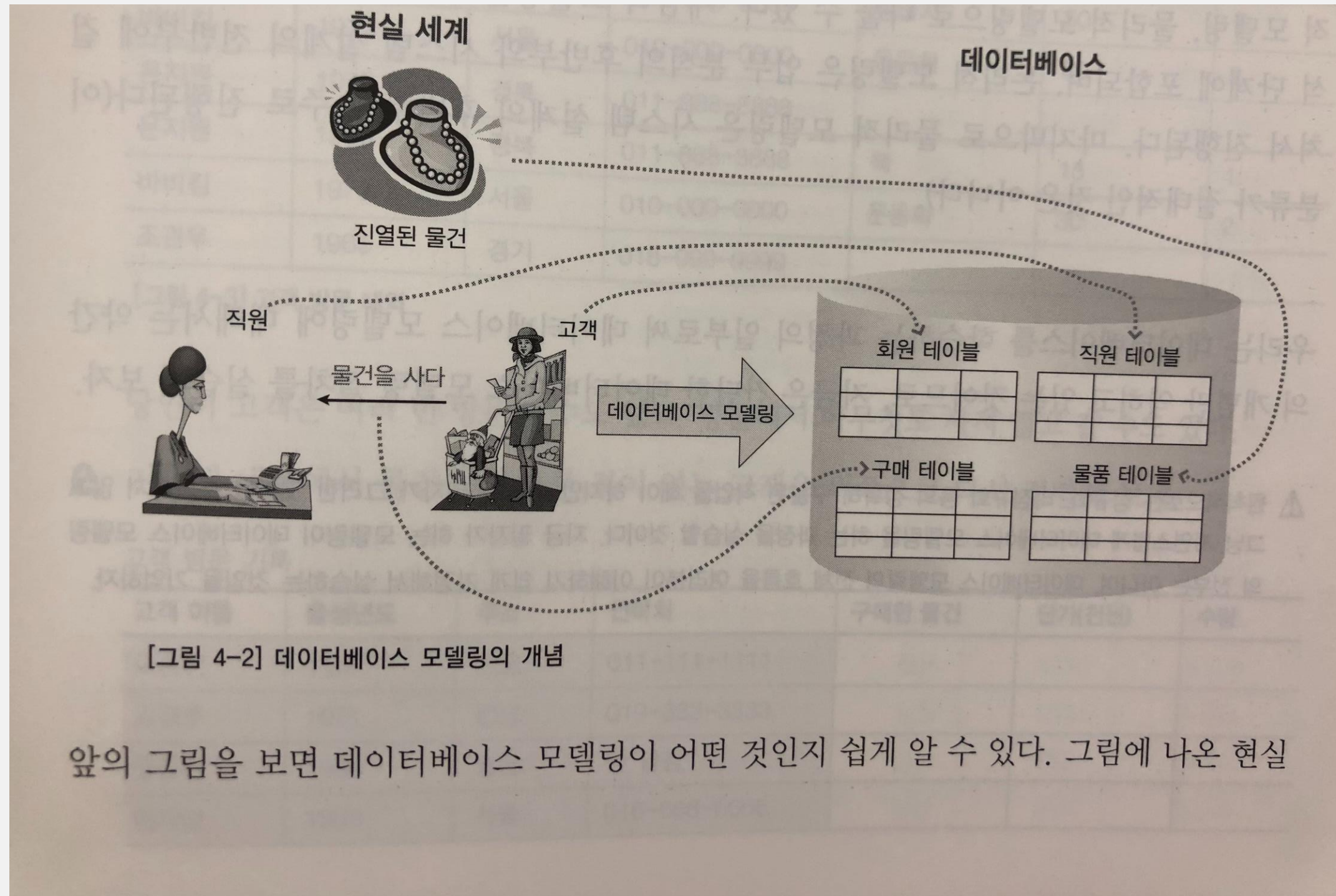
2. 정규화나 인덱스나, SQL Server를 다루는 방법은
뇌를 자극하는 SQL Server 2012 기본편 책을 보시는게
좋습니다.

저는 DBA도 아니고 개발자도 아니라서 쿼리 튜닝을 다루지
않습니다.

3. SQL Query 위주 샘플 데이터로 다루고 오늘 4시간 안에
SQL를 완벽하게 이해하긴 어려울 수 있기 때문에
간단한 SQL 사용법과 분석 레시피 관점으로 진행합니다.



데이터베이스



데이터베이스는 여러 사람에 의해 공유되어 사용할 목적으로 통합하여 관리되는 데이터의 집합, 줄여서 **DB**라고 말함

관계형 데이터베이스 (RDBMS)는 데이터를 로우(Row : 행), 컬럼(Column : 열)과 같이 일종의 표 형태로 저장하고 종속성은 관계(Relation)으로 표현함

컬럼의 구조와 데이터의 관계가 테이블 스키마(Schema)로 사전 정의되고, SQL(구조화 질의 언어)에 의해서 정교한 검색 쿼리를 제공하며 상상하는 거의 모든 방식으로 데이터를 다룰 수 있게 됨

SQL

- **2차원의 정형데이터를 다루는 가장 효율적인 언어.**
(Data를 사용하는 거의 모든 프로그램에서 SQL을 사용하지 않고 효율적인 개발 불가)[DBMS, Hive, Spark, R(sqldf), ...]
- **Data의 처리 과정을 기술하는 방법이 아닌 원하는 Data의 요건을 기술하는 방식**[SQL = Data 요건정의서]
(단순성, 높은 생산성)
- **Data의 요건이 기술된 요건 정의서 형태이므로 요건의 변경시 적은 수정으로 요건을 반영할 수 있음**
(효율적인 유지보수)





SQL

- **DDL(Data Definition Language : 데이터 정의 언어)**
 - CREATE
 - DROP
 - ALTER
- **DML(Data Manipulation Language : 데이터 조작 언어)**
 - INSERT
 - UPDATE
 - DELETE
 - SELECT
- **DCL(Data Control Language : 데이터 제어 언어)**
 - GRANT
 - REVOKE
- ...



기본 문법(SELECT)

SELECT ... 선택하고자 하는 속성목록을 기술.

FROM ... Source Data Set을 기술.

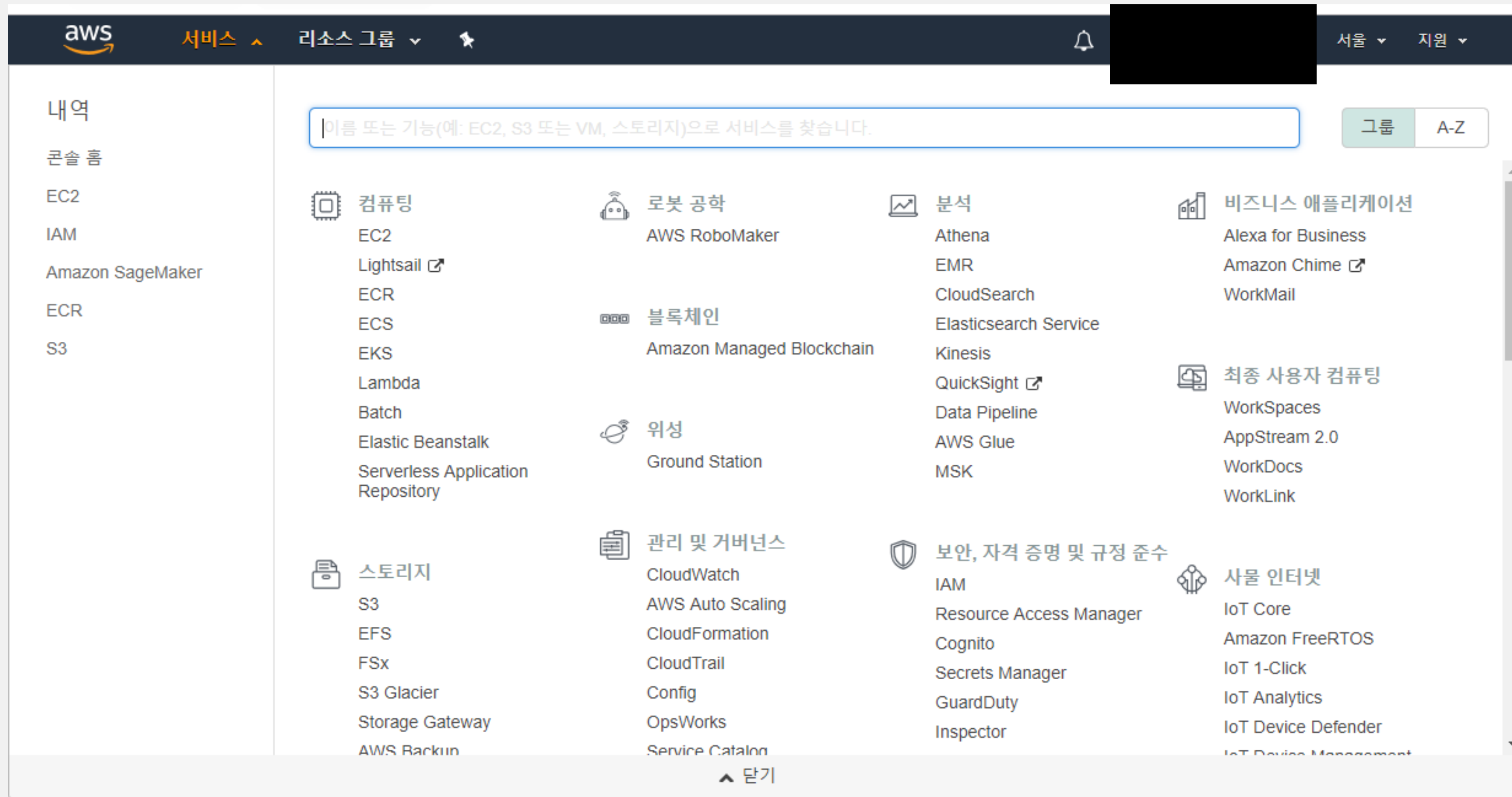
[WHERE] ... 선택하고자 하는 행의 조건의 목록을 기술.

[GROUP BY ... 집계하고자 하는 기준속성목록을 기술.

[HAVING]] ... 집계된 행을 기준으로 선택하고자 하는 행의
조건목록을 기술함.(GROUP BY가 선행조건)

[ORDER BY] ... 정렬의 기준이 되는 속성목록과 각 속성 목록의
정렬방식(올림차순, 내림차순)를 기술.

AWS 분석환경 구축



Amazon
Web
Service
(클라우드)

제공해주는
많은
서비스가 있음



AWS 분석환경 구축

단계 1:
엔진 선택

단계 2:
DB 세부 정보 지정

단계 3:
고급 설정 구성

RDS > 데이터베이스 생성

엔진 선택

엔진 옵션

☐ Amazon Aurora

Amazon
Aurora

☐ MySQL



☐ MariaDB



☐ PostgreSQL



☐ Oracle

ORACLE

☒ Microsoft SQL Server



Microsoft SQL Server

에디션

- ☒ **SQL Server Express Edition**
최대 10GiB의 데이터베이스 크기를 지원하는 경제적인 데이터베이스 관리 시스템입니다.
- ☐ **SQL Server Web Edition**
Microsoft의 라이선스 정책에 따라 Microsoft SQL Server Web Edition은 퍼블릭 및 인터넷 액세스 웹 페이지, 웹 사이트, 웹 애플리케이션 및 웹 서비스 지원 목적으로만 사용될 수 있습니다.
- ☐ **SQL Server Standard Edition**
미션 크리티컬 애플리케이션과 복합 워크로드를 위한 핵심 데이터 관리 및 비즈니스 인텔리전스 기능입니다.
- ☐ **SQL Server Enterprise Edition**
부담이 큰 데이터베이스 워크로드와 비즈니스 인텔리전스 요구 사항이 적용되는 미션 크리티컬 애플리케이션을 위한 포괄적인 고급 기능입니다.

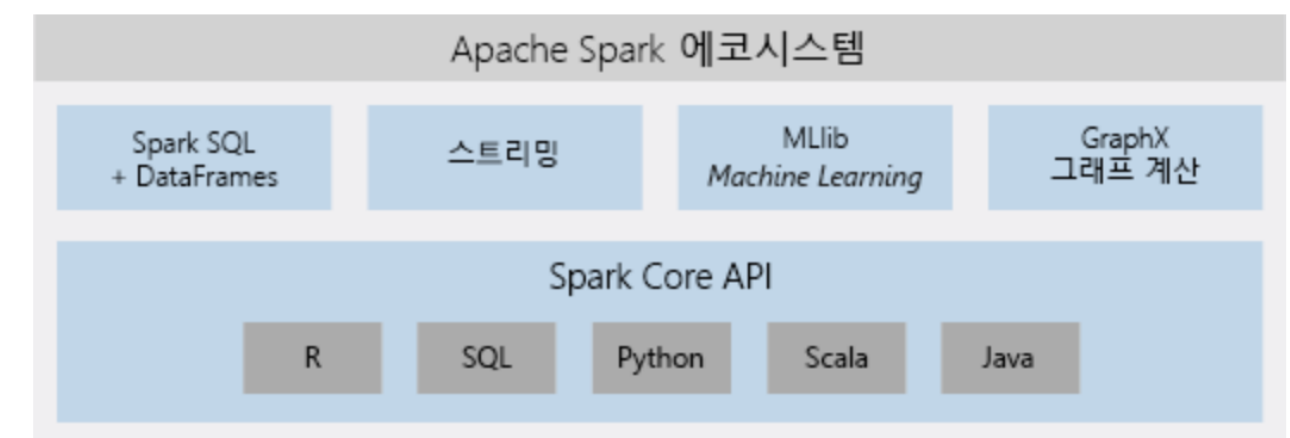
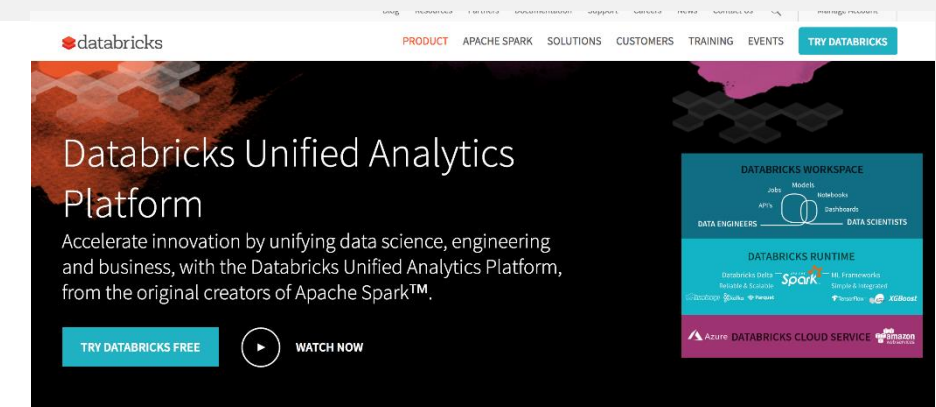
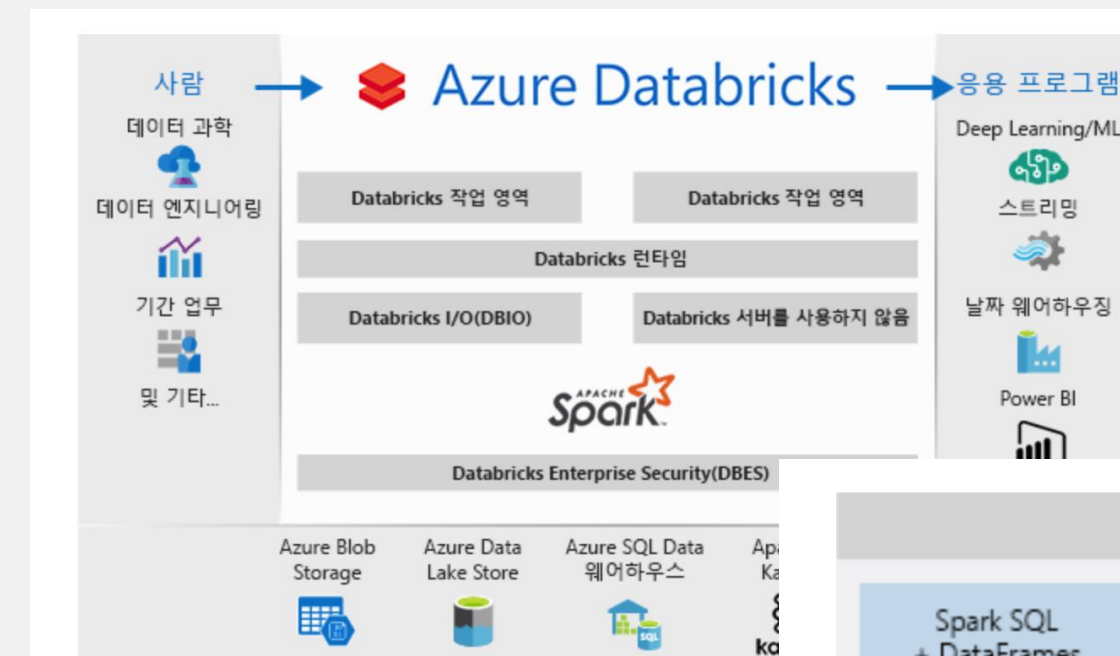
오늘 실습은
제가 구성한 SQL
Server Standard
Edition으로 진행할
예정입니다.

강의 종료하고
당일 없앨 예정
(비용이 많이 듦)

AWS 분석환경 구축

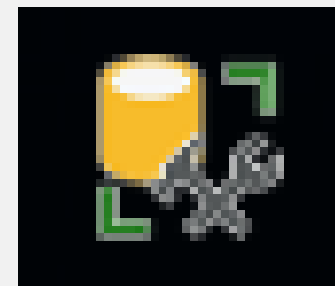
RDS(MS-SQL Server) 이외 EC2로 R, Python 등을 설치하여 분석 환경 구축을 원하시면 다음 Python Korea 2017 - [튜토리얼](#) (이상열)을 따라하시면 됩니다.

혹은 직접 구축이 귀찮으면 다음과 같은 방법도 있습니다.
AWS SageMaker Or Azure Databricks (둘다 노트북 기반)



클라이언트 설치

클라이언트를 다운로드 받아봅시다!



[SSMS\(SQL Server Management Studio\) 설치 \(ssms 2017\)](#)

맥에서는 SSMS가 없습니다
클라이언트 대체품이 많은데요.
[해당 링크](#)에 잘 나와있습니다.

맥 사용하시는 분들이 있어서
범용성 있는 클라이언트인
DBeaver 로 오늘 진행하겠습니다.

SSMS 17.9.1은 SSMS의 현재 GA(일반 공급) 버전입니다.

[SQL Server Management Studio 17.9.1 다운로드](#)

[SQL Server Management Studio 17.9.1 업그레이드 패키지 다운로드\(17.x에서 17.9.1로 업그레이드\)](#)

버전 정보

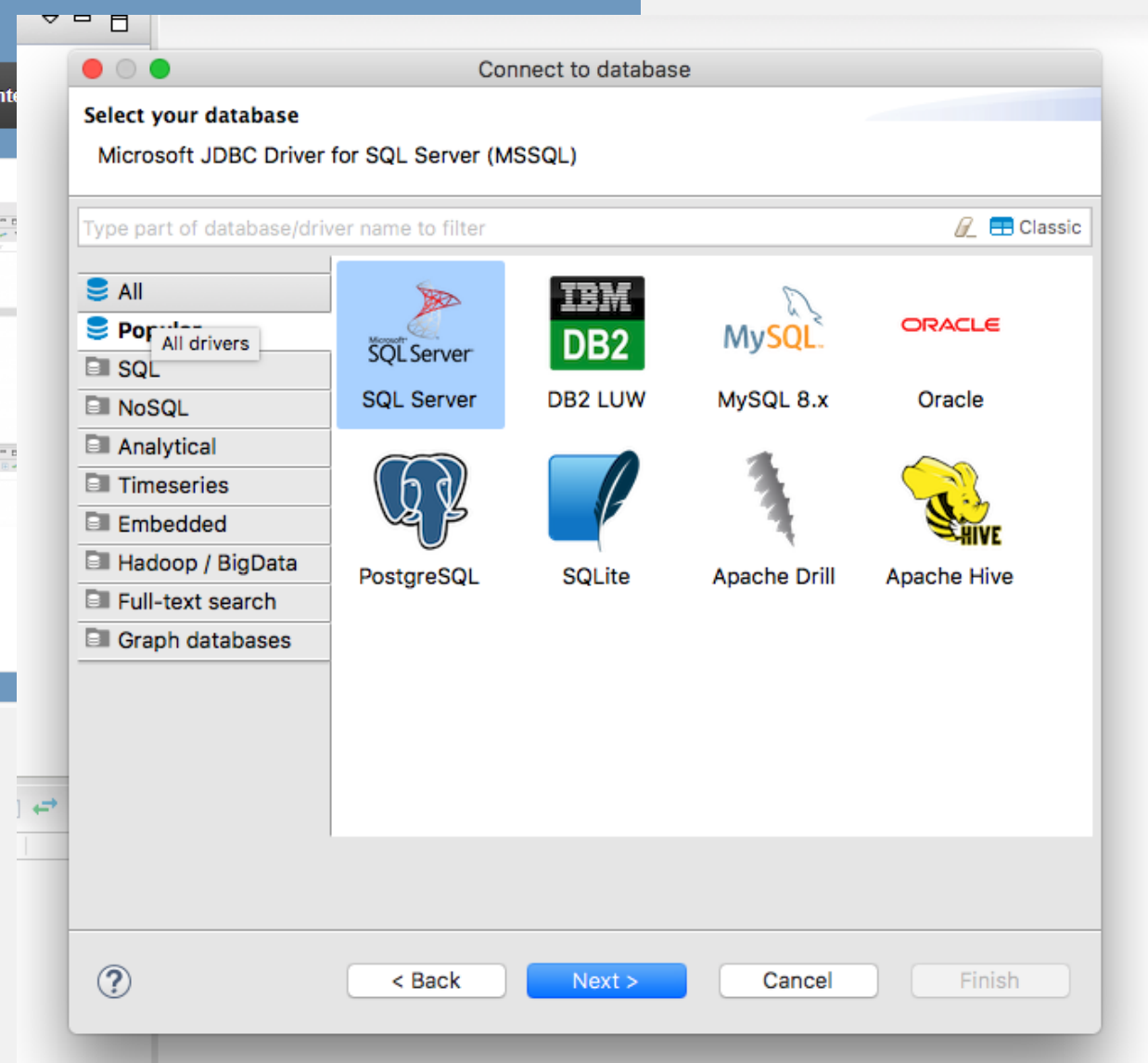
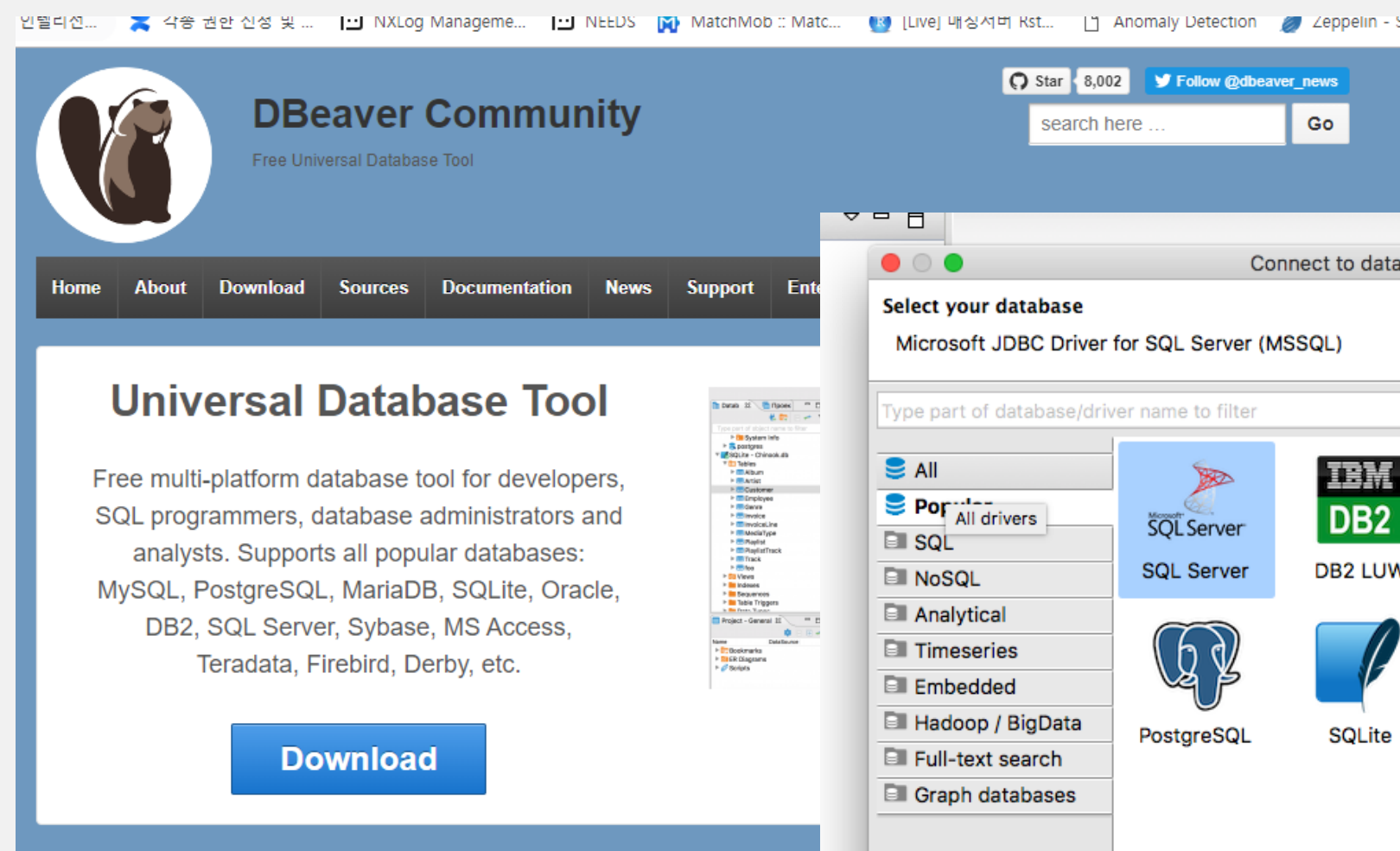
- 릴리스 번호: 17.9.1
- 빌드 번호: 14.0.17289.0
- 릴리스 날짜: 2018년 11월 21일

사용 가능한 언어(SSMS 17.9.1)



클라이언트 설치

<https://dbeaver.io/>



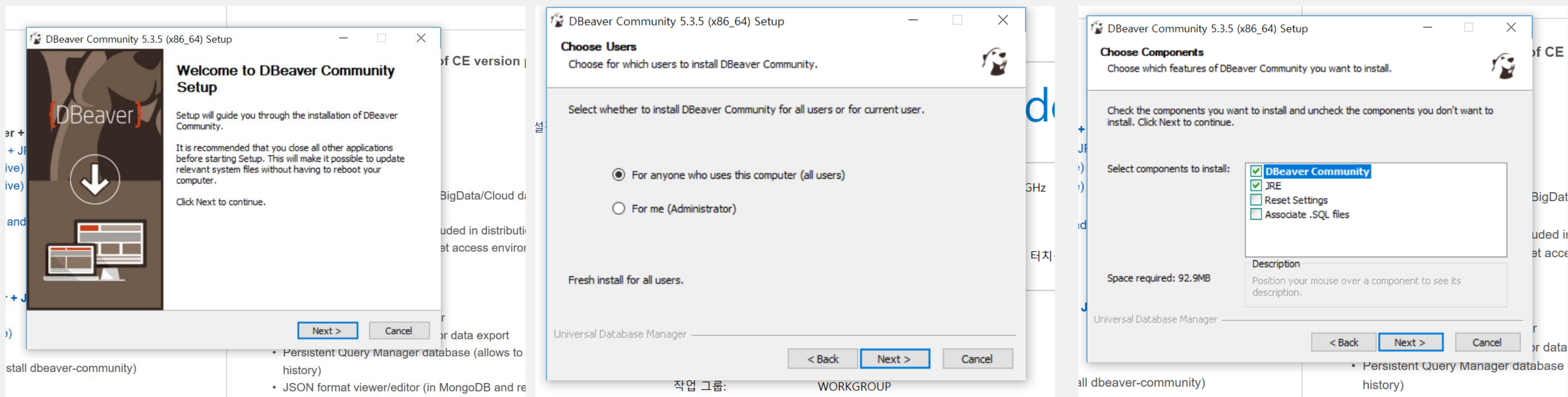
Community Edition 5.3.5 다운로드
(Free Universal Database Tool)

DBeaver 내에는 NoSQL Databases
또는
다양한 형태의 JDBC Driver 제공



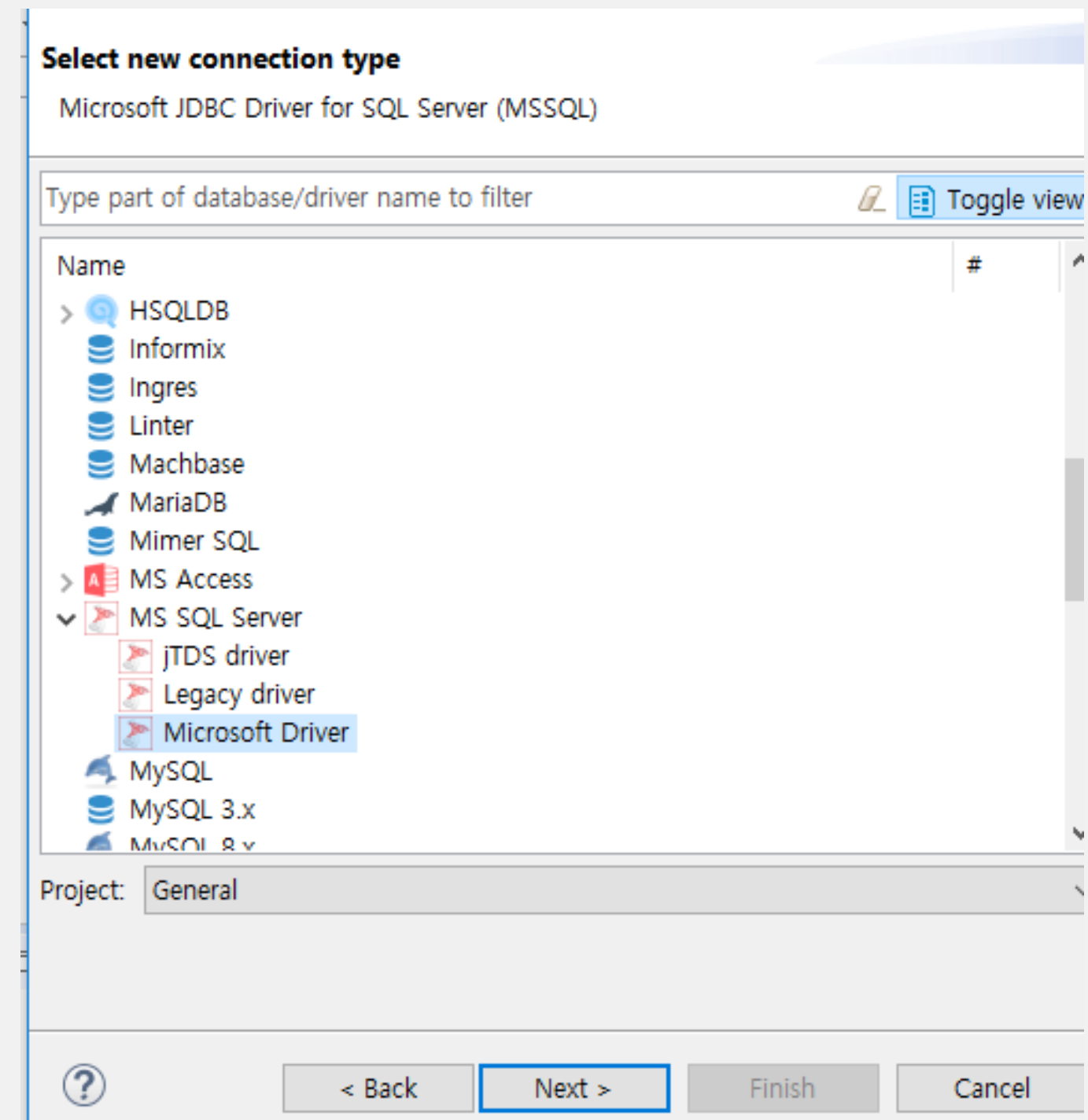
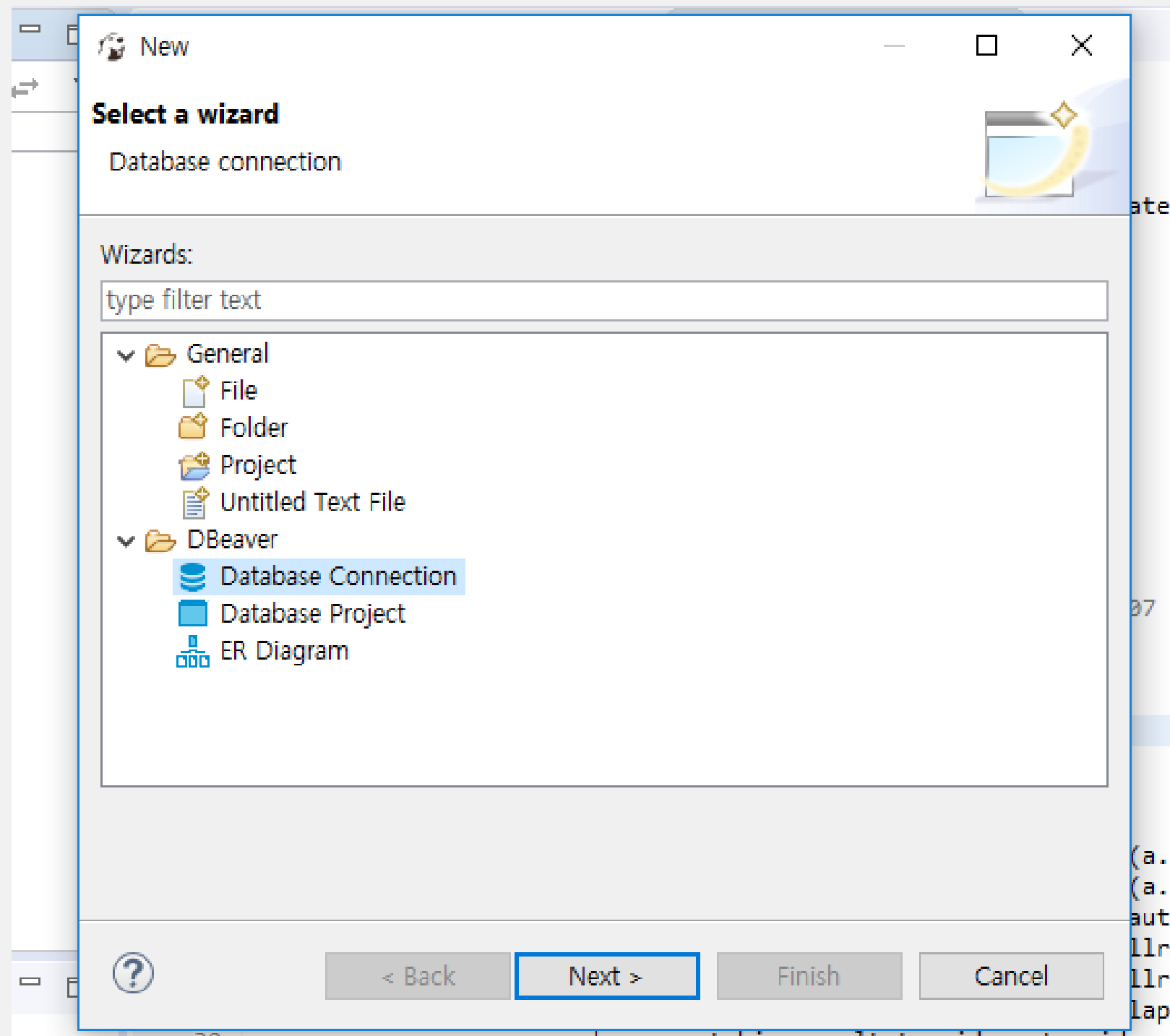
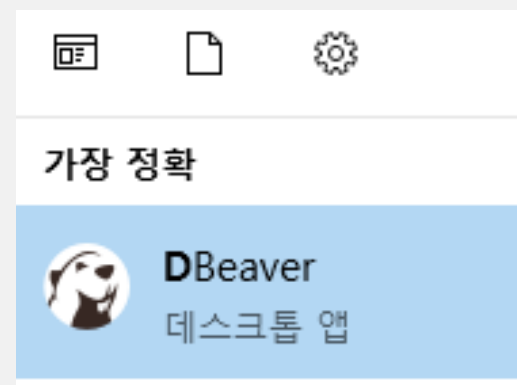
클라이언트 설치

<https://dbeaver.io/>



파일 다운로드 받아서 실행, 그대로 진행해도 됨

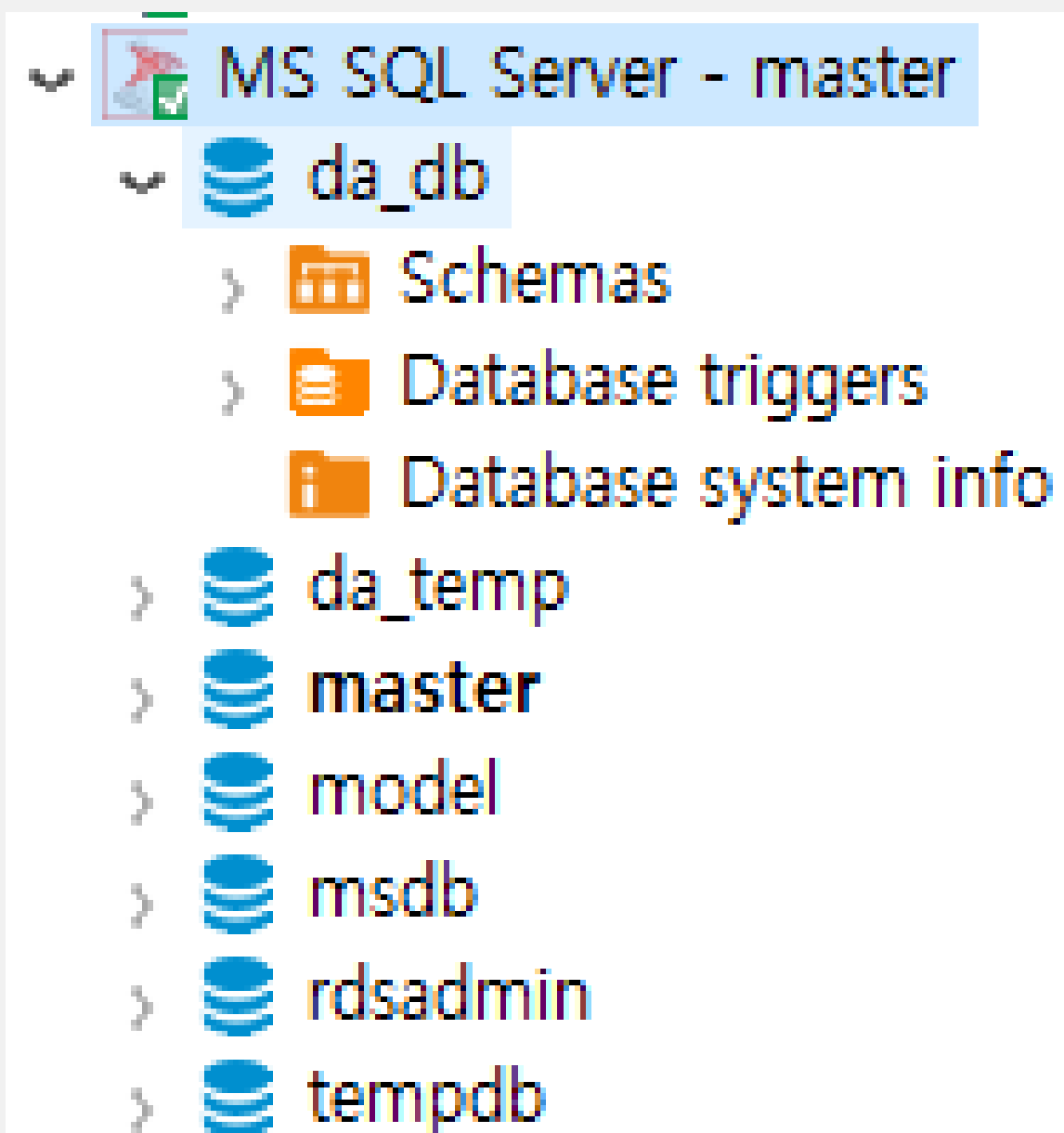
클라이언트 설치



File -> New에서 DBEaver -> Database Connection 클릭
Select new connection type에서 MS SQL Server ->Microsoft Driver 누르고 Next



클라이언트 접속



da_db : 데이터베이스 (조회용 모든 데이터)가 있습니다

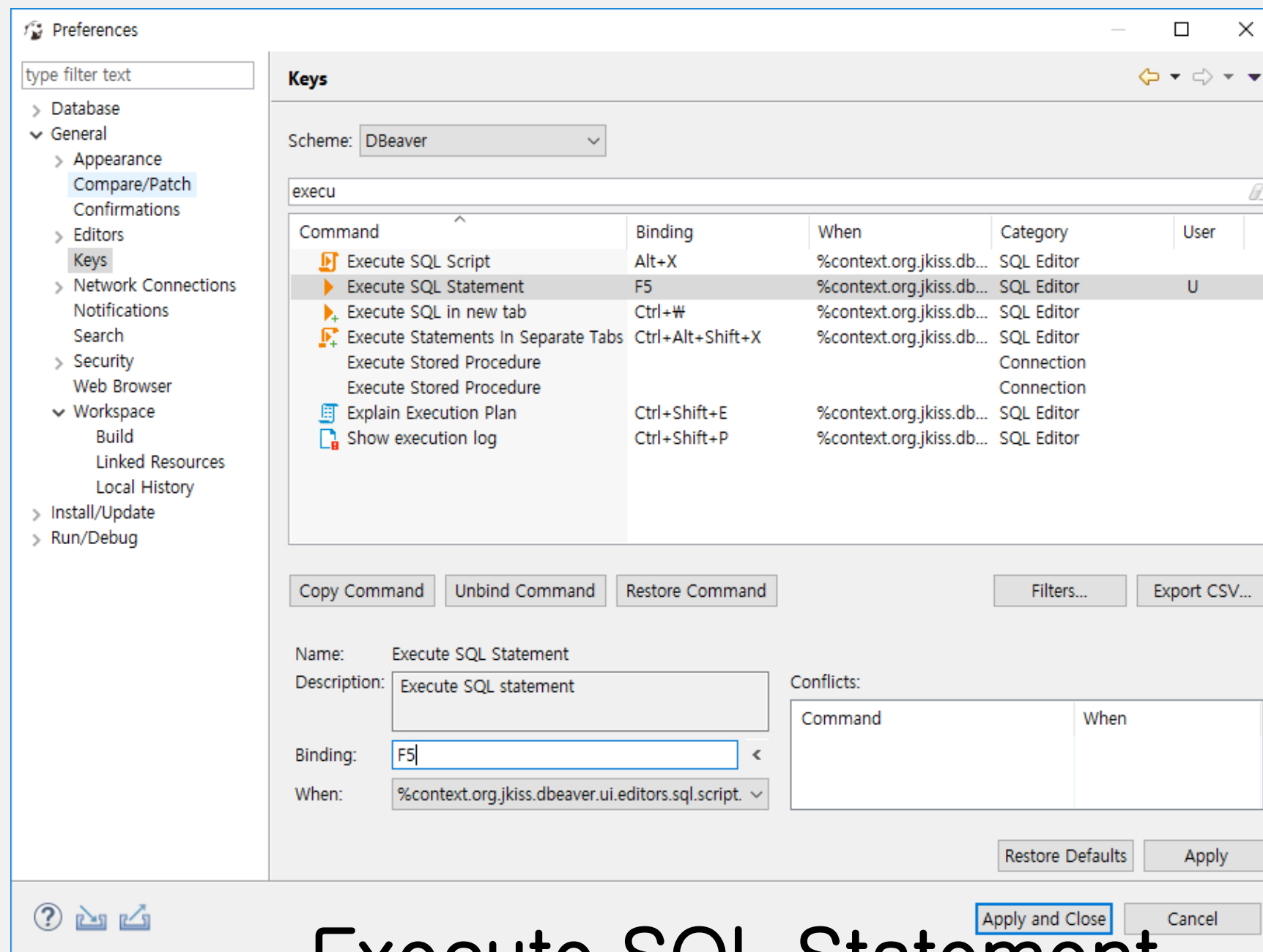
da_temp : 생성 및 삭제 등 자유롭게 만들어진 temp DB

그 외는 데이터베이스는 안보셔도 됩니다.

클라이언트 접속 (tip)

Windows -> Preferences->General 에 가면

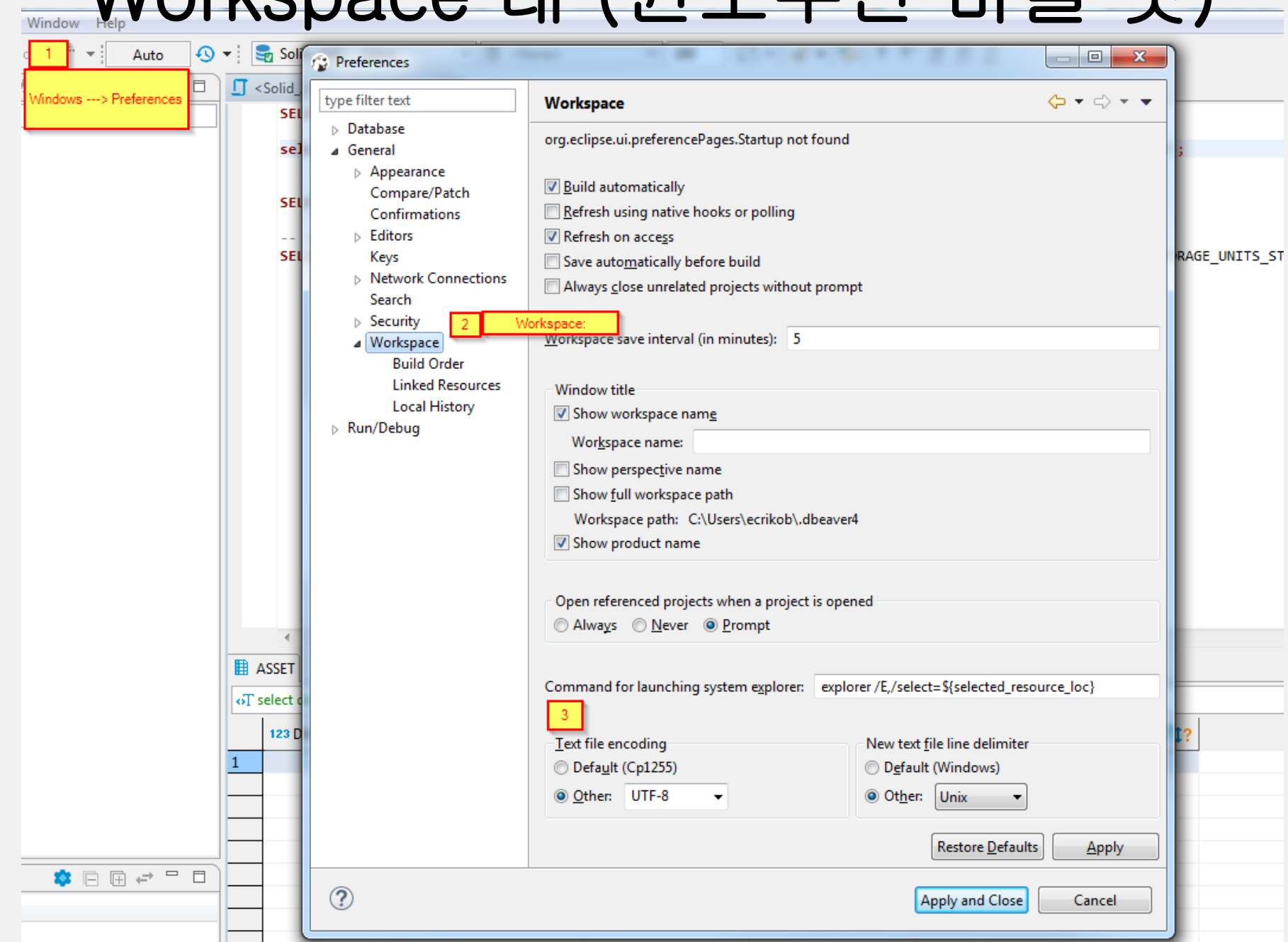
Workspace 내 (윈도우만 바꿀 것)



Execute SQL Statement

Ctrl+enter에서 F5로 변경 (Binding)

Mac은 커맨드 A로 변경



한글 인코딩 깨질 경우 UTF-8로 변경



기본 문법(SELECT)

	123 userid	ABC order_date	ABC method	123 amount	123 discount
1	1	2015-08-01	CALL	10,000	-999
2	1	2015-08-03	TOUCH	10,000	[NULL]
3	1	2015-08-10	TOUCH	10,000	-950
4	1	2015-08-14	CALL	10,000	-1,000
5	1	2015-08-25	TOUCH	10,000	[NULL]
6	2	2015-08-03	TOUCH	5,000	-500
7	2	2015-08-11	TOUCH	5,000	-300
8	2	2015-08-12	TOUCH	5,000	-700
9	2	2015-08-22	TOUCH	5,000	-1,000
10	2	2015-08-28	TOUCH	5,000	-600
11	3	2015-08-07	CALL	10,000	-1,000
12	3	2015-08-19	TOUCH	10,000	-1,000
13	3	2015-08-30	CALL	10,000	-1,000

userid : 유저 아이디 (식별자)
order_date : 주문 일자
method : 주문 방법
amount : 주문 금액
discount : 할인 금액

주문 데이터



데이터 타입

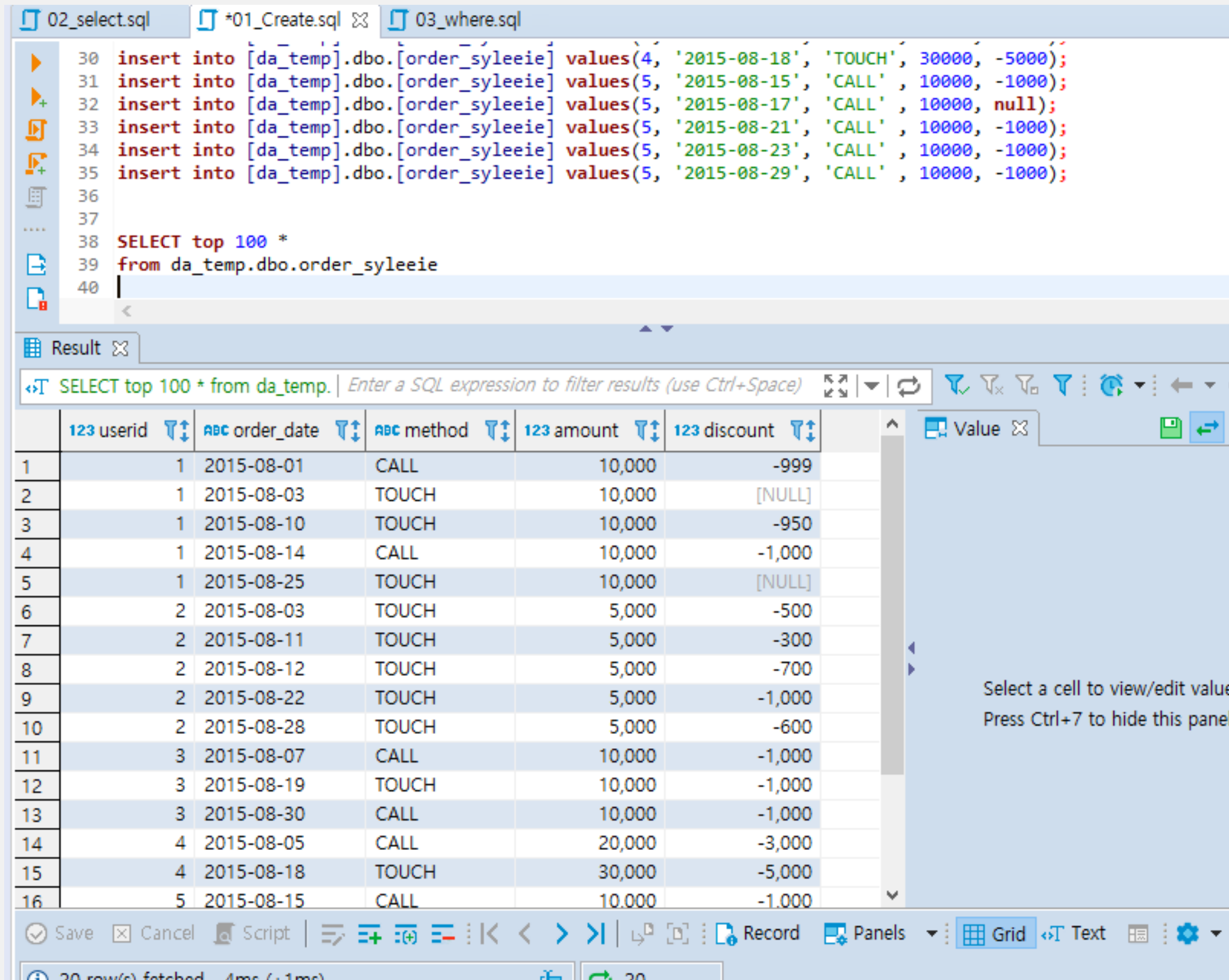
분류	데이터 타입	범위	저장소크기
정수	Bit	0 또는 1	bit
	Int	-2,147,483,648 ~ 2,147,483,647	4 바이트
	Smallint	-32,768 ~ 32,767	2 바이트
	Tinyint	0 ~ 255	1 바이트
	Bigint	-2^63 ~ 2^63-1	8 바이트
부동소수점	Float[n]	-1.79E+308 ~ 1.79E+308 n = 1~24	4 바이트
	Float[n]	-1.79E+308 ~ 1.79E+308 n = 25~53	8 바이트
	Real	-3.40E + 38 ~ 3.40E + 38	4 바이트

문자데이터	char[n]	n = 1~8000	n 바이트
	Varchar[n]	n = 1~8000 varchar [(n max)] max 는 최대 저장소 크기가 2^31-1바이트임	입력한 데이터의 길이
	Text	최대 2,147,483,647자의 가변길이	
유니코드 문자데이터	Nchar	n = 1~4000	n*2 바이트
	nvarchar	n = 1~4000 nvarchar [(n max)] max 는 최대 저장소 크기가 2^31-1바이트임	입력한 데이터의 길이*2 바이트
	Ntext	최대 1,073,741,823자의 가변길이	

이진데이터	binary	n = 1~8000	n+4 바이트
	varbinary	n = 1~8000 varbinary [(n max)] max 는 최대 저장소 크기가 2^31-1바이트임	입력한 데이터의 길이+4 바이트
	Image	최대 2,147,483,647자의 가변길이	
날짜와시간	datetime	1753/1/1~9999/12/31	8 바이트
	smalldatetime	1900/1/1~2079/6/6	4 바이트
화폐	money	-922,337,203,685,477.5808~ +922,337,203,685,477.5807	8 바이트
	smallmoney	-214,748.3648~214,748.3647	4 바이트

자세한 것은 여기 보시면 됩니다.

기본 문법(SELECT)



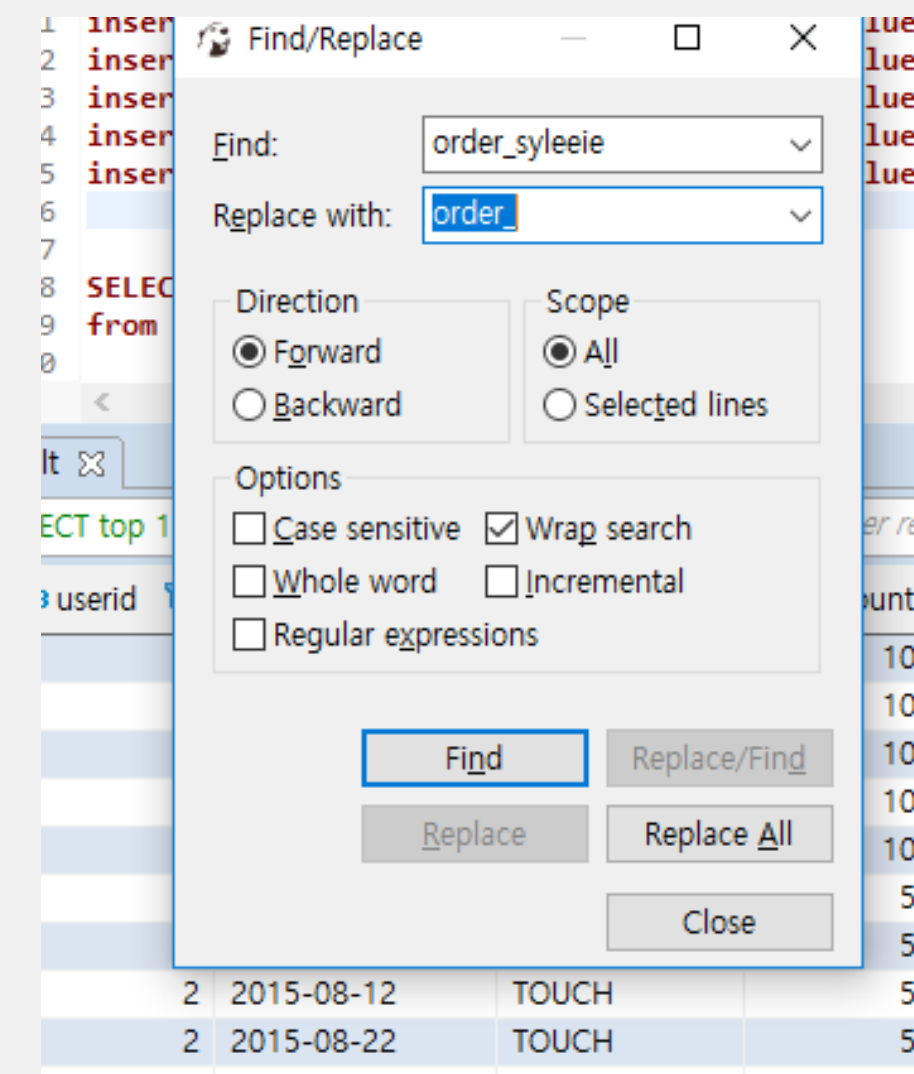
The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
30 insert into [da_temp].dbo.[order_syleeie] values(4, '2015-08-18', 'TOUCH', 30000, -5000);
31 insert into [da_temp].dbo.[order_syleeie] values(5, '2015-08-15', 'CALL', 10000, -1000);
32 insert into [da_temp].dbo.[order_syleeie] values(5, '2015-08-17', 'CALL', 10000, null);
33 insert into [da_temp].dbo.[order_syleeie] values(5, '2015-08-21', 'CALL', 10000, -1000);
34 insert into [da_temp].dbo.[order_syleeie] values(5, '2015-08-23', 'CALL', 10000, -1000);
35 insert into [da_temp].dbo.[order_syleeie] values(5, '2015-08-29', 'CALL', 10000, -1000);
36
37
38 SELECT top 100 *
39 from da_temp.dbo.order_syleeie
40
```

The result grid shows the following data:

	userid	order_date	method	amount	discount
1	1	2015-08-01	CALL	10,000	-999
2	1	2015-08-03	TOUCH	10,000	[NULL]
3	1	2015-08-10	TOUCH	10,000	-950
4	1	2015-08-14	CALL	10,000	-1,000
5	1	2015-08-25	TOUCH	10,000	[NULL]
6	2	2015-08-03	TOUCH	5,000	-500
7	2	2015-08-11	TOUCH	5,000	-300
8	2	2015-08-12	TOUCH	5,000	-700
9	2	2015-08-22	TOUCH	5,000	-1,000
10	2	2015-08-28	TOUCH	5,000	-600
11	3	2015-08-07	CALL	10,000	-1,000
12	3	2015-08-19	TOUCH	10,000	-1,000
13	3	2015-08-30	CALL	10,000	-1,000
14	4	2015-08-05	CALL	20,000	-3,000
15	4	2015-08-18	TOUCH	30,000	-5,000
16	5	2015-08-15	CALL	10,000	-1,000

테이블 생성하는 방법



Replace_with 에
자신의 영어
아이디를 넣고

Replace_all로
바꿔주세요



01_Create.sql

기본 문법(SELECT)

02_select.sql 01_Create.sql 03_where.sql

```
1 -- =====
2 -- 02. select
3 -- [da_temp].dbo.[order]
4 -- =====
5
6 select userid, order_date
7 from [da_temp].dbo.[order]
```

컨트롤 F5가 실행 (tip에서 안바꿨으면 ctrl+enter)

또는

특정 영역 드래그하여 F5

	userid	order_date
1	1	2015-08-01
2	1	2015-08-03
3	1	2015-08-10
4	1	2015-08-14
5	1	2015-08-25
6	2	2015-08-03
7	2	2015-08-11
8	2	2015-08-12
9	2	2015-08-22
10	2	2015-08-28
11	3	2015-08-07
12	3	2015-08-19
13	3	2015-08-30
14	4	2015-08-05
15	4	2015-08-18
16	5	2015-08-15
17	5	2015-08-17

Value

Select a cell to view/edit value
Press Ctrl+7 to hide this panel



02_select.sql

테이블은
Excel에서
Sheet 지정과 같음



기본 문법(SELECT)

SELECT

ALL / DISTINCT	중복 표현 / 중복 제거	select [all] order_date from “order” select distinct order_date from “order”
*	전체 열 선택	select * from “order”
ALIAS	컬럼명 재정의	select distinct order_date [as] ord_dt from “order”

Top N Query

Top	출력행 수 제한	Select top 5 * from “order”order by amount desc
-----	----------	--

기본 문법(WHERE)

02_select.sql 01_Create.sql 03_where.sql 04_group by.sql

```
1 -- =====
2 -- 03. where
3 -- [da_temp].dbo.[order]
4 -- =====
5
6 select userid, order_date
7 from [da_temp].dbo.[order]
8 where userid = 1
9
10 -- OPERATOR(WHERE 조건절)
11 -- =====
12 -- 비교 연산자
13
14 select * from [da_temp].dbo.[order] where order_date = '2015-08-05';
15 select * from [da_temp].dbo.[order] where order_date > '2015-08-05' order by order_date;
16 select * from [da_temp].dbo.[order] where order_date >= '2015-08-05' order by order_date;
17 select * from [da_temp].dbo.[order] where order_date < '2015-08-05' order by order_date;
18 select * from [da_temp].dbo.[order] where order_date <= '2015-08-05' order by order_date;
19
20 -- =====
21 -- OPERATOR(WHERE 조건절)
22
```

컨트롤 F5가 실행 (tip에서 안바꿨으면 ctrl+enter) 또는 특정 영역 드래그하여 F5

order

select amount , discount , abs(dis | Enter a SQL expression to filter results (use Ctrl+Space)

	123 amount	123 discount	123 abs	123 ceiling	123 floor	123 round	123 Value
1	10,000	-999	999	-999	-999	-999	2.7
2	10,000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	2.7
3	10,000	-950	950	-950	-950	-950	2.7
4	10,000	-1,000	1,000	-1,000	-1,000	-1,000	2.7
5	10,000	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	2.7



03_where.sql

필터 기능



기본 문법(WHERE)

- OPERATOR
 - 비교연산자
 - SQL연산자
 - 논리연산자
 - 부정 비교 연산자
 - 부정 SQL 연산자

비교 연산자

=	같다.	where order_date = '2015-08-05'
>	보다 크다.	where order_date > '2015-08-05'
>=	보다 크거나 같다.	where order_date >= '2015-08-05'
<	보다 작다.	where order_date < '2015-08-05'
<=	보다 작거나 같다.	where order_date <= '2015-08-05'

SQL 연산자

BETWEEN a AND	a와 b의 값 사이에 있으면 된다. (a와 b 값이 포함됨)	where order_date between '2015-08-01' and '2015-08-20'
IN (list)	리스트에 있는 값 중에서 어느 하 라도 일치 하면 된다.	where order_date in ('2015-08-01', '2015-08-07')
LIKE '비교문자열'	비교문자열과 형태가 일치하면 된 다. (%, _ 사용)	where order_date like '2015%2%' where order_date like '2015-08-01'
IS NULL	NULL 값인 경우 (NULL은 특별 취급)	where discount is null



기본 문법(WHERE)

- OPERATOR
 - 비교연산자
 - SQL연산자
 - 논리연산자
 - 부정 비교 연산자
 - 부정 SQL 연산자

논리 연산자

AND	앞에 있는 조건과 뒤에 오는 조건이 참(TRUE)이 되면 결과도 참(TRUE)이 된다. 즉, 앞의 조건과 뒤의 조건을 동시에 만족해야 한다.	where order_date = '2015-08-12' and userid = 2
OR	앞의 조건이 참(TRUE)이거나 뒤의 조건이 참(TRUE)이 되어야 결과도 참(TRUE)이 된다. 즉, 앞뒤의 조건 중 하나만 참(TRUE)이면 된다.	where order_date = '2015-08-12' or userid = 2
NOT	뒤에 오는 조건에 반대되는 결과를 되돌려 준다.	where not order_date = '2015-08-12' and userid = 2

부정 비교 연산자

!=	같지 않다.	where order_date != '2015-08-12' and userid = 2
^=	같지 않다.(동작X)	where order_date ^= '2015-08-12' and userid = 2
<>	같지 않다.(ISO 표준)	where order_date <> '2015-08-12' and userid = 2
NOT 컬럼명 =	~와 같지 않다.	where not order_date = '2015-08-12' and userid = 2
NOT 컬럼명 >	~보다 크지 않다.	where not order_date > '2015-08-12' and userid = 2



기본 문법(WHERE)

- OPERATOR
 - 비교연산자
 - SQL연산자
 - 논리연산자
 - 부정 비교 연산자
 - 부정 SQL 연산자

부정 SQL 연산자

NOT BETWEEN a AND b	a와 b의 값 사이에 있지 않다. (a, b 값을 포함하지 않는다.)	where order_date not between '2015-08-05' and '2015-08-20'
NOT IN (list)	list 값과 일치하지 않는다.	where order_date not in ('2015-08-15', '2015-08-23') and userid = 5
IS NOT NULL	NULL 값을 갖지 않는다.	where order_date is not null

연산자의 우선순위

연산 우선순위	설명
1	괄호 ()
2	NOT 연산자
3	비교 연산자, SQL 비교 연산자
4	AND
5	OR

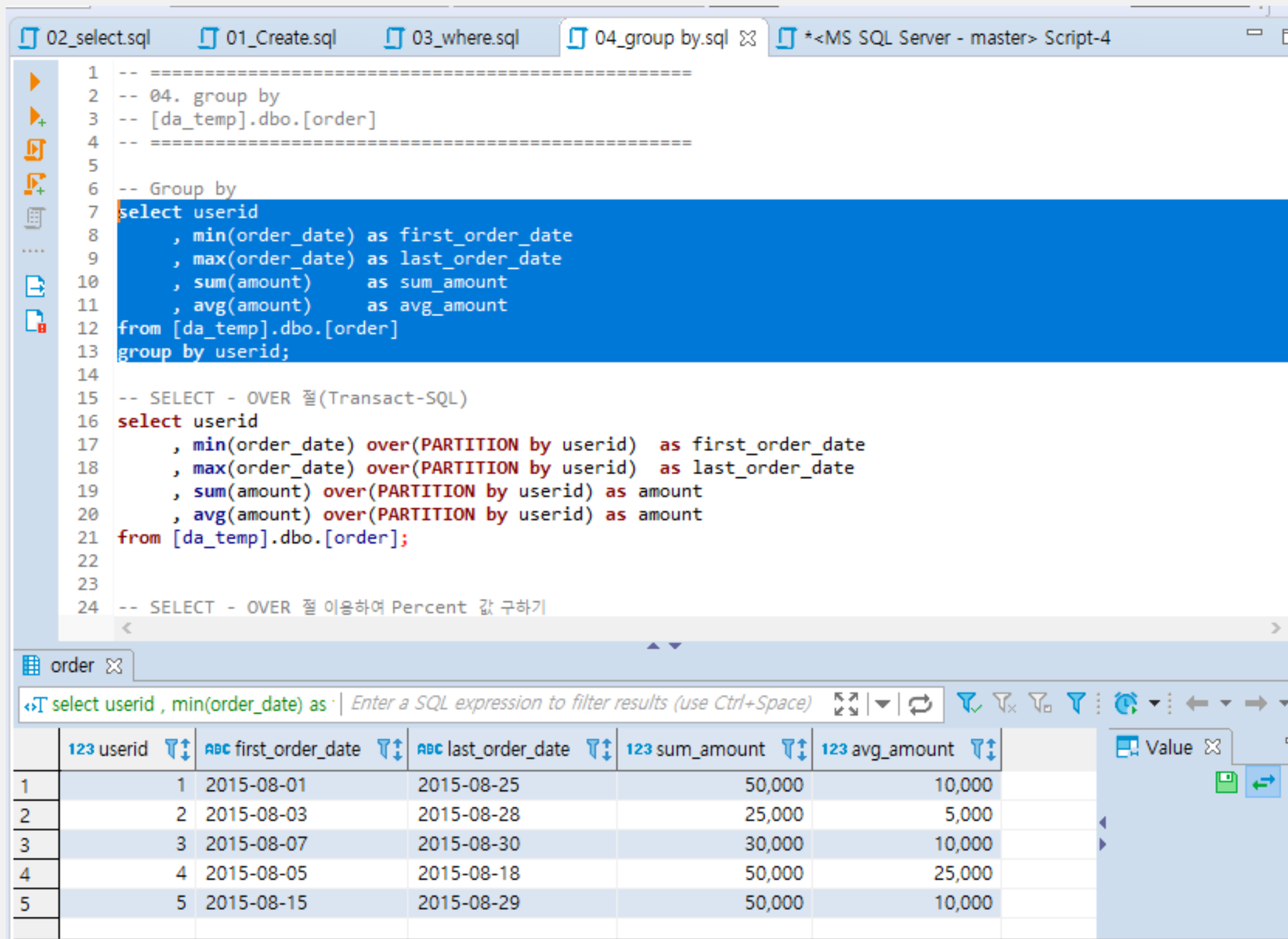


잠깐 쉬는 시간...

문제 1 : 회원별 Touch 와 Call 주문수를 알고 싶다.

문제 2. 8월 11일에서 8월 20일까지 주문이 2개 이상인 회원을 알고 싶다.

기본 문법(group by)



```
1  -- =====
2  -- 04. group by
3  -- [da_temp].dbo.[order]
4  -- =====
5
6  -- Group by
7  select userid
8      , min(order_date) as first_order_date
9      , max(order_date) as last_order_date
10     , sum(amount)      as sum_amount
11     , avg(amount)      as avg_amount
12 from [da_temp].dbo.[order]
13 group by userid;
14
15 -- SELECT - OVER 절(Transact-SQL)
16 select userid
17     , min(order_date) over(PARTITION by userid) as first_order_date
18     , max(order_date) over(PARTITION by userid) as last_order_date
19     , sum(amount) over(PARTITION by userid) as amount
20     , avg(amount) over(PARTITION by userid) as amount
21 from [da_temp].dbo.[order];
22
23
24 -- SELECT - OVER 절 이용하여 Percent 값 구하기
```

	123 userid	ABC first_order_date	ABC last_order_date	123 sum_amount	123 avg_amount
1	1	2015-08-01	2015-08-25	50,000	10,000
2	2	2015-08-03	2015-08-28	25,000	5,000
3	3	2015-08-07	2015-08-30	30,000	10,000
4	4	2015-08-05	2015-08-18	50,000	25,000
5	5	2015-08-15	2015-08-29	50,000	10,000



04_group by.sql

집계 통계량
([aggregate
Function](#))



기본 문법(WHERE)

- CASE Expression – 값변환(Value Change)

case simple_case_expression 조건 then 표현절 else 표현절 end	Simple_case_expression 조건이 맞으면 simple_case_expression 조건 내의 then 절을 수 행하고, 조건이 맞지 않으면 else 절을 수행한다.
case searched_case_expression 조건 then 표현절 else 표현절 end	Searched_case_expression 조건이 맞으면 searched_case_expression 조건내의 then 절을 수행하고, 조건이 맞지 않으면 else 절을 수행한다.



기본 문법(WHERE)

- [CAST Expression – 형 변환\(Data Type Convert\)](#)

Convert

개인적으로 가장 많이 사용하는 데이터변환 함수라고 생각합니다.

날짜 변환 하는데도 유용하게 사용가능합니다.

※ 부동 소수점 또는 숫자에서 정수로 변환할때 CONVERT()함수는 결과를 자릅니다. 다른 변환일 경우에는 반올림합니다.

사용법

```
--문법--
CONVERT(data_type[(length)], expression[style])
--예시--
SELECT CONVERT(NVARCHAR(10),칼럼) AS 칼럼명 FROM MY_TABLE --VARCHAR로 변환
SELECT CONVERT(INT,칼럼) AS 칼럼명 FROM MY_TABLE --INT로 변환
SELECT CONVERT(CHAR,칼럼) AS 칼럼명 FROM MY_TABLE --CHAR로 변환
```

expression : 유효한 식

data_type : 대상 데이터 형식 별칭 데이터 형식은 사용할 수 없습니다.

length : 대상 데이터 형식의 길이를 지정하는 선택적 정수입니다. 기본값은 30입니다.

style : Convert함수가 식을 변환하는 방법을 지정하는 정수 식입니다. style이 Null이면 Null값이 반환됩니다.

Cast

※ FLOAT,또는 NUMERIC에서 INTEGER로 변환할때 CAST()함수는 결과를 자릅니다.

사용법

```
--문법--
CAST(expression AS data_type(length))
--예시--
SELECT CAST(칼럼 AS INT) FROM MY_TABLE
```

예제

```
--테이블(MY_TALBE)에서 가격(PRICE)칼럼을 INT에서 VARCHAR로 형변환
SELECT CAST(PRICEAS AS VARCHAR)AS 가격 FROM MY_TABLE
```

기본 문법(Having)

1	-- =====
2	-- 05. Having
3	-- [da_temp].dbo.[order]
4	-- =====
5	
6	-- Group by
7	select userid
8	, min(order_date) as first_order_date
9	, max(order_date) as last_order_date
10	, sum(amount) as amount
11	from [da_temp].dbo.[order]
12	group by userid
13	having sum(amount) >= 30000
14	
15	
16	

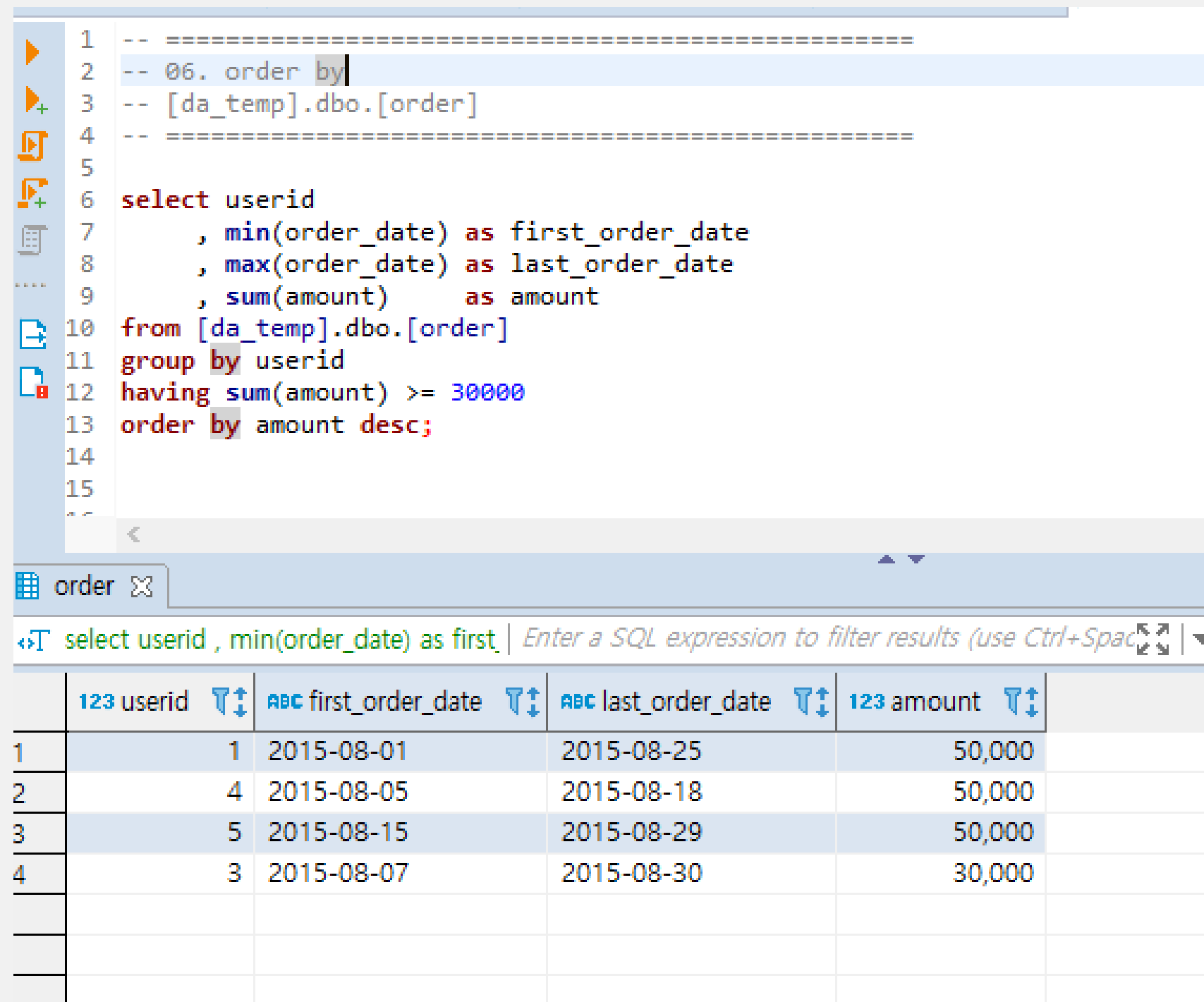
order	Σ			
select userid , min(order_date) as first_ Enter a SQL expression to filter results (use Ctrl+Space)				
	123 userid	ABC first_order_date	ABC last_order_date	123 amount
1	1	2015-08-01	2015-08-25	50,000
2	3	2015-08-07	2015-08-30	30,000
3	4	2015-08-05	2015-08-18	50,000
4	5	2015-08-15	2015-08-29	50,000



05_Having.sql

집계 후 어떤 행을 볼지?

기본 문법(order by)



The screenshot shows a SQL IDE with a query editor and a results pane. The query is as follows:

```
1 -- =====
2 -- 06. order by
3 -- [da_temp].dbo.[order]
4 -- =====
5
6 select userid
7     , min(order_date) as first_order_date
8     , max(order_date) as last_order_date
9     , sum(amount)      as amount
10 from [da_temp].dbo.[order]
11 group by userid
12 having sum(amount) >= 30000
13 order by amount desc;
```

The results pane shows the following data:

	123 userid	ABC first_order_date	ABC last_order_date	123 amount
1	1	2015-08-01	2015-08-25	50,000
2	4	2015-08-05	2015-08-18	50,000
3	5	2015-08-15	2015-08-29	50,000
4	3	2015-08-07	2015-08-30	30,000



06_order by.sql

순서대로 볼까?



상세 문법(common syntax)

SET OPERATOR	Row의 추가 및 제거.
JOIN	Column의 연결 및 선택.
Subquery	Row, Column의 연결 및 선택.

UNION	합집합으로써 모든 행의 중복 제거.
UNION ALL	중복을 제거하지 않는 합집합.
INTERSECT	교집합으로 중복 제거.
EXCEPT	차집합으로 중복 제거.



상세 문법(common syntax)

Table : Set1	
COL1	COL2
VARCHAR(5)	VARCHAR(5)
A	B
A	B
B	C

Table : Set2	
COL1	COL2
VARCHAR(5)	VARCHAR(5)
A	B
C	D



07_set.sql

union all

```
select col1, col2
  from set1
union all
select col1, col2
  from set2
```

COL1	COL2
A	B
A	B
B	C
A	B
C	D

union

```
select col1, col2
  from set1
union
select col1, col2
  from set2
```

COL1	COL2
C	D
B	C
A	B

intersect

```
select col1, col2
  from set1
intersect
select col1, col2
  from set2
```

COL1	COL2
A	B

except

```
select col1, col2
  from set1
except
select col1, col2
  from set2
```

COL1	COL2
B	C



상세 문법(Join)

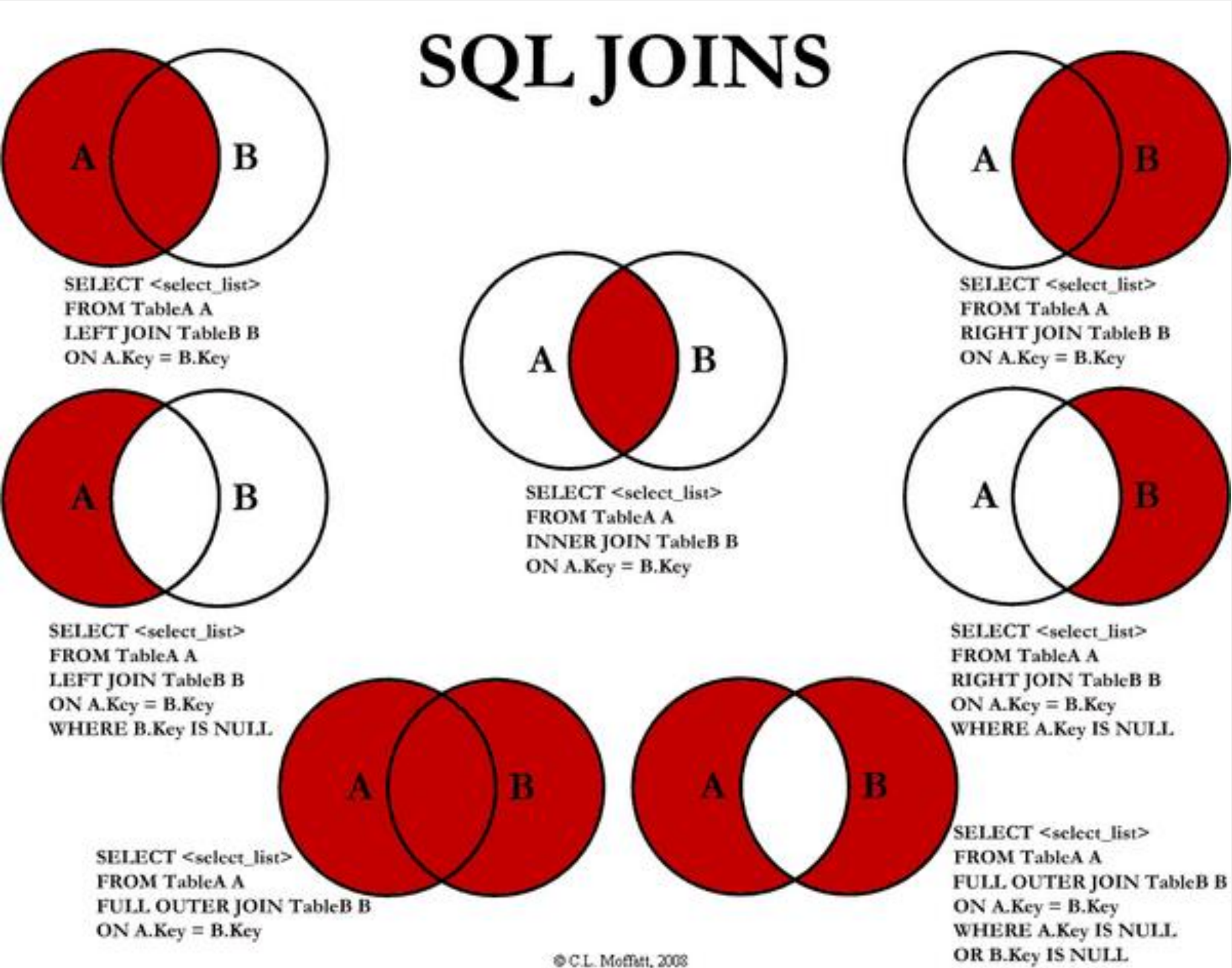
Table : Customer		
USREID	USERNAME	JOIN_DATE
INTEGER	VARCHAR(10)	VARCHAR(10)
1	A	2015-08-01
2	B	2015-08-02
3	C	2015-08-01
4	D	2015-08-03
5	E	2015-08-07
6	F	2015-08-22

Table : Order				
USREID	ORDER_DATE	METHOD	AMOUNT	DISCOUNT
INTEGER	VARCHAR(10)	VARCHAR(10)	INTEGER	NUMERIC
1	2015-08-01	CALL	10,000	-998.7
1	2015-08-03	TOUCH	10,000	
1	2015-08-10	TOUCH	10,000	-950.4
1	2015-08-14	CALL	10,000	-1,000.0
1	2015-08-25	TOUCH	10,000	
2	2015-08-03	TOUCH	5,000	-500.0
2	2015-08-11	TOUCH	5,000	-300.0
2	2015-08-12	TOUCH	5,000	-700.0
2	2015-08-22	TOUCH	5,000	-1,000.0
2	2015-08-28	TOUCH	5,000	-600.0
3	2015-08-07	CALL	10,000	-1,000.0
3	2015-08-19	TOUCH	10,000	-1,000.0
3	2015-08-30	CALL	10,000	-1,000.0
4	2015-08-05	CALL	20,000	-3,000.0
4	2015-08-18	TOUCH	30,000	-5,000.0
5	2015-08-15	CALL	10,000	-1,000.0
5	2015-08-17	CALL	10,000	
5	2015-08-21	CALL	10,000	-1,000.0
5	2015-08-23	CALL	10,000	-1,000.0
5	2015-08-29	CALL	10,000	-1,000.0



상세 문법(Join)

INNER JOIN	연결(JOIN) 되는 양쪽 데이터 집합의 연결 KEY 값들이 정확하게 일치하는 경우만 데이터를 연결.(EQUI JOIN이라고도 함)
LEFT OUTER JOIN	연결(JOIN) 되는 기준(좌측) 데이터 집합의 연결 KEY 값을 기준으로 데이터를 연결, 데이터 연결이 실패하더라도 기준(좌측) 데이터는 유지.
RIGHT OUTER JOIN	연결(JOIN) 되는 기준(우측) 데이터 집합의 연결 KEY 값을 기준으로 데이터를 연결, 데이터 연결이 실패하더라도 기준(우측) 데이터는 유지. (LEFT OUTER JOIN에 비해 사용 빈도 떨어짐)
CROSS JOIN	연결의 기준 없이 연결(Cartesian Product)를 수행. 집합 A의 row수가 n, 집합 B의 row수가 m이라면 A cross join B의 row수는 n*m이 됨.(데이터 중복이 발생하므로 주의).





상세 문법(Join)

```
57 -- inner join
58 -----
59 select cus.userid
60     , count(ord.userid) as orders
61   from #customer cus
62   inner join
63     #order ord
64   on cus.userid = ord.userid
65  group by cus.userid
66  order by cus.userid
67 ;
68
69 -----
70 -- left outer join
71 -----
```

#customer

select cus.userid , count(ord.userid) | Enter a SQL expression to filter results (use Ctrl)

userid	orders
1	5
2	5
3	3
4	2
5	5

```
69 -----
70 -- left outer join
71 -----
72 select cus.userid
73     , count(ord.userid) as orders
74   from #customer cus
75   left outer join
76     #order ord
77   on cus.userid = ord.userid
78  group by cus.userid
79  order by cus.userid
80 ;
81
82 -----
83 -- right outer join
84 -----
```

#customer

select cus.userid , count(ord.userid) | Enter a SQL expression to filter results (use Ctrl)

userid	orders
1	5
2	5
3	3
4	2
5	5
6	0



08_join.sql

Join을 해보자~!



상세 문법(subquery)

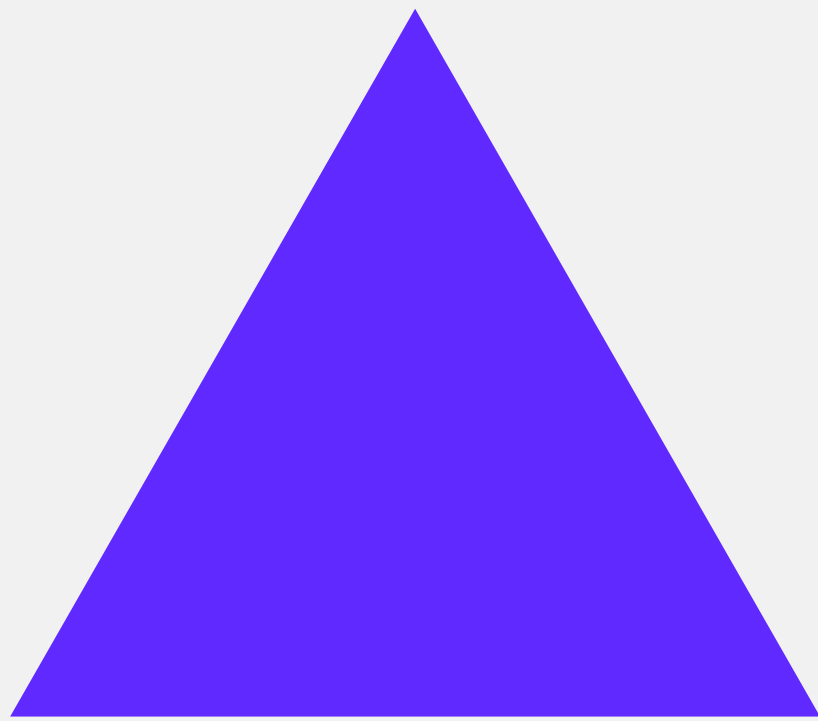
Scalar Subquery (SELECT 절)	Un-Correlated Subquery	연결 조건 없이 수행되는 상수와 같은 Scalar Subquery. (Subquery 내에서 1건만 반환되어야 함.)
	Correlated Subquery	From절 내의 컬럼을 이용하여 데이터 연결을 하는 Scalar Subquery. (Subquery 내에서 1건만 반환되어야 함.)
Inline View (FROM 절)		FROM 절 내부에 존재하는 Subquery.
Subquery (WHERE 절)	Un-Correlated Subquery	연결 조건 없이 수행되는 상수와 같은 Subquery.
	Correlated Subquery	From절 내의 컬럼을 이용하여 데이터 연결을 하는 Subquery.



08_join.sql

Subquery

Select, from, where 절에서 사용할 수 있음



3. 분석레시피 1

- 실전 문제 풀기

(DAU, ARPPU, BUR, Retention 지표 계산하기)



실전 문제 풀기

KPI = Key Performance Indicator, 핵심 지표

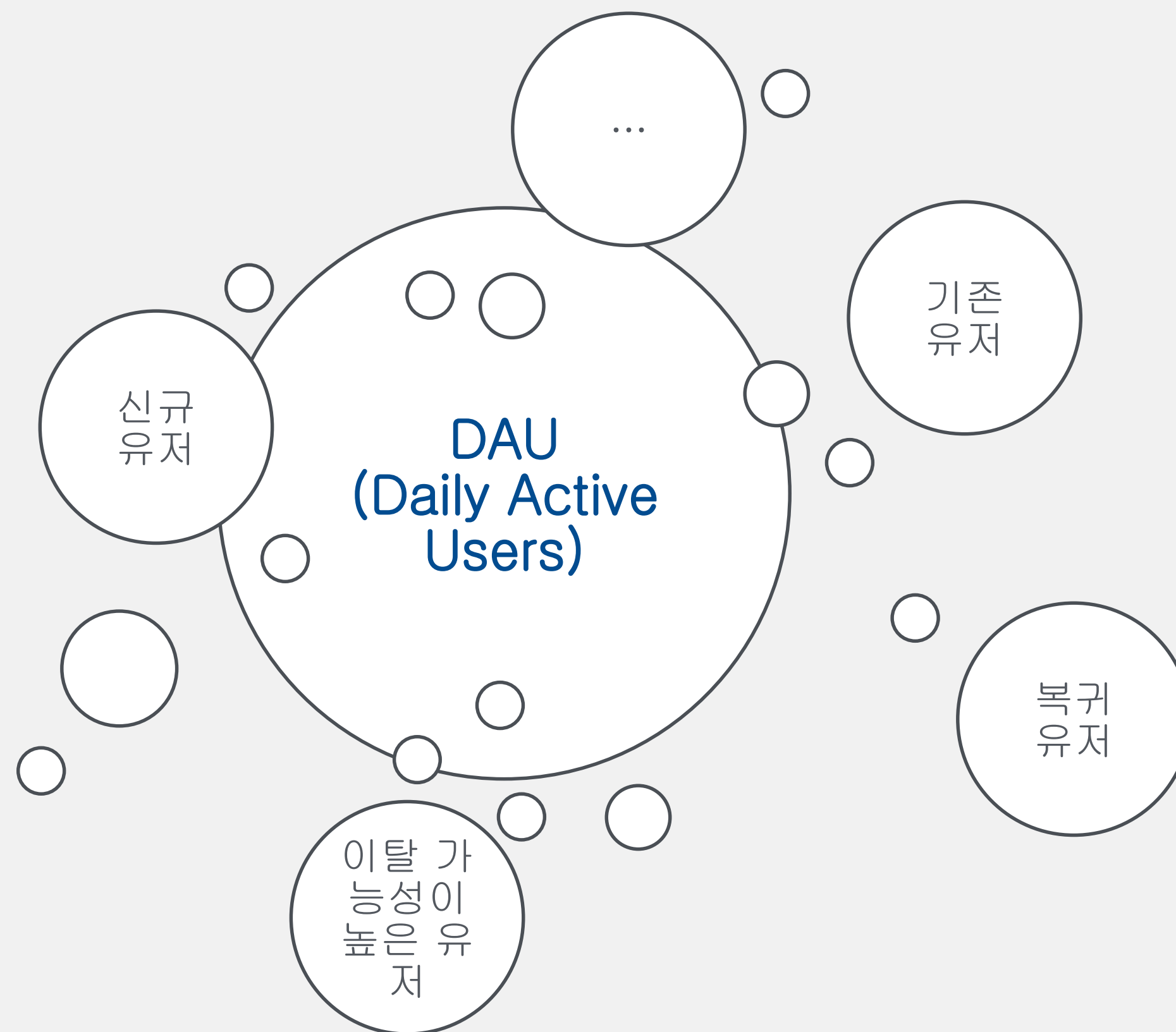
분류	용어명	설명
유 저	DAU	Daily Active User. 일 방문자수(로그인 기준)
	액티브 유저	최근 n일 이내 방문 유저수
	휴면 유저	최근 n일 이내 방문 기록이 없는 유저
	잔존율	특정일의 유저를 100으로 가정했을 때, 특정일 이후 일별로 유입된 유저의 비율
매 출	매출액	유저가 결제한 금액
	BU	Buying User, 결제 유저 = Paying User (PU)라고도 부름
	BUR(구매전환율)	Buying User Ratio, 전체 유저 중 구매 유저로 전환된 비중
	ARPU	Average Revenue Per User, 전체 유저당 결제 금액
	ARPPU	Average Revenue Per Paying User, 결제 유저당 결제 금액



실전 문제 풀기

DAU는 어떻게 구성되어 있지?

– Segment가 중요

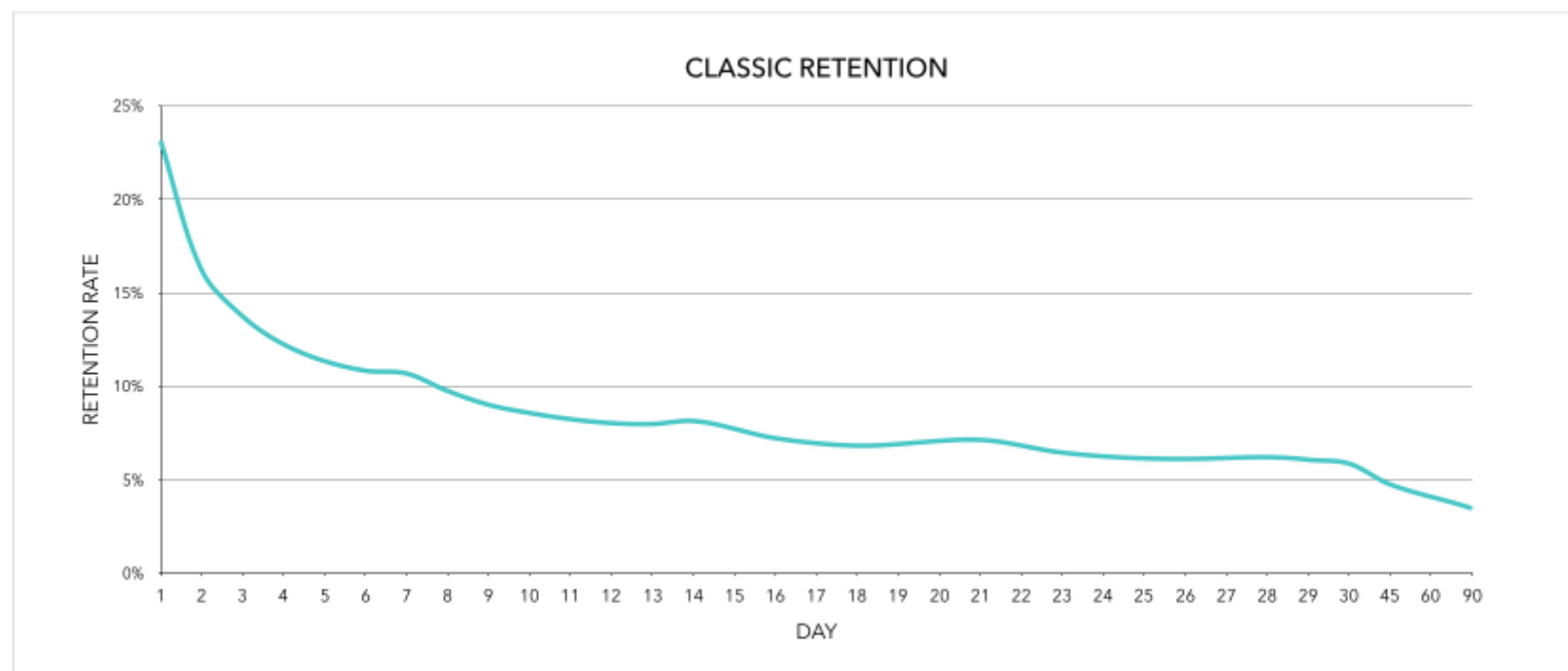




실전 문제 풀기

리텐션을 구하자!

리텐션이란? 재방문율, 특정 날짜(D+1, D+2, ..., D+14, D+31)에 다시 방문하는 유저의 비율



Retention Statistics											
COHORT	NEW USERS	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10
Aug 8 - Aug 14	150.1K	43.0%									
Aug 1 - Aug 7	146.5K	47.7%	30.2%								
Jul 25 - Jul 31	158.8K	48.0%	36.7%	26.1%							
Jul 18 - Jul 24	149.2K	47.3%	35.2%	31.6%	23.2%						
Jul 11 - Jul 17	151.7K	47.8%	36.7%	30.7%	30.2%	22.1%					
Jul 4 - Jul 10	176.9K	47.7%	37.0%	30.6%	26.5%	27.5%	20.0%				
Jun 27 - Jul 3	182.6K	47.6%	36.4%	31.3%	26.9%	23.9%	25.4%	18.5%			
Jun 20 - Jun 26	184.4K	45.7%	34.5%	29.7%	26.9%	24.9%	22.1%	23.0%	16.9%		
Jun 13 - Jun 19	152.9K	45.9%	36.5%	30.2%	27.9%	26.9%	24.5%	22.0%	22.8%	16.9%	
Jun 6 - Jun 12	152.8K	46.7%	35.4%	31.9%	27.3%	26.6%	24.9%	23.1%	20.9%	21.7%	16.1%

실전 문제 풀기

리텐션을 한 방에 구하자! (Tip)

- 테이블 압축
- 클러스터 인덱싱

```
7
3  -- 테이블 압축 및 인덱싱
9  alter table #dau_churn rebuild with (data_compression = page );
9  raiserror('테이블 압축 완료' , 0 , 1) with nowait ;
1
2  create unique clustered index ucix on #dau_churn(account_id, std_dt)
3  with (data_compression = page);
4  raiserror('테이블 인덱싱 완료' , 0 , 1) with nowait ;
5
6  |
```

Row - 고정 데이터 형식 크기를 가변적으로 바꾸어 압축
Page - 접두사 압축 후 사전 압축

압축 시 용량 감소

	name	rows	reserved	data	index_size	unused	
1	Customers	264485	26920 KB	26504 KB	176 KB	240 KB	압축 X
1	Customers_2	264485	25096 KB	25008 KB	8 KB	80 KB	ROW 압축
1	Customers_3	264485	15688 KB	15536 KB	8 KB	144 KB	PAGE 압축

클러스터 인덱스

- 테이블당 한 개만 생성 가능
- 행 데이터를 인덱스로 지정한 열에 맞춰서 자동 정렬 (중복X)
 - 영어사전처럼 책의 내용 자체가 순서대로 정렬이 되어있음
- where, order by , group by 등 자주 사용되는 컬럼



실전 문제 풀기

리텐션을 한 방에 구하자! (Tip)

- [With 문 : CTE \(Common Table Expression\)](#)
- CTE는 CREATE VIEW 문 (하나의 실행 범위 내에서 정의되는 임시결과 집합)
Inline View와 달리 자체 참조가 가능하며 동일 쿼리에서 여러 번 참조 가능

```
-- UserTotalCost라는 이름의 CTE
WITH UserTotalCost (ID, TotalCost)
AS
(
    SELECT ID, SUM(Amount * Price) AS TotalCost FROM BuyTable
    GROUP BY ID
)

-- 마치 실제 테이블처럼 사용이 가능하다
SELECT * FROM UserTotalCost ORDER BY TotalCost DESC
```



실전 문제 풀기

```
57
58 -- 잔존을 계산하기
59 if object_id('tempdb..#retention') is not null
60     drop table #retention
61
62 with x as (
63     select a.std_dt, a.account_id, a.new_yn, a.comeback_yn, case when amount = 0 then '01. '
64                                     when amount <= 30000 then '02. 3만원 이하'
65                                     when amount <= 100000 then '03. 10만원 이하'
66                                     when amount > 100000 then '04. 10만원 이상'
67                                     else '01. 무과금' end amount_group
68     from #dau_churn a
69     where 1=1
70 )
71 ,
72 y as (
73     select b.std_dt, b.new_yn, b.comeback_yn, case when b.amount = 0 then '01. 무과금'
74                                     when b.amount <= 30000 then '02. 3만원 이하'
```

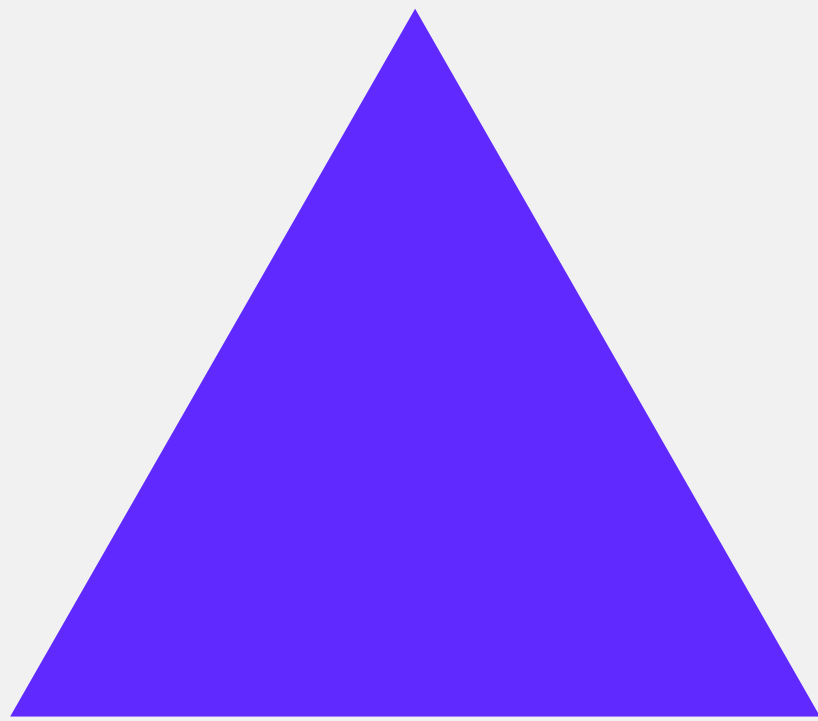
Statistics

Name	Value
Updated Rows	0
Query	with x as (select a.std_dt, a.account_id, a.new_yn, a.comeback_yn, case when amount = 0 then '01. '
Finish time	Sat Mar 02 20:50:52 KST 2019



12_Retention.sql

With 문 (x, y, z) 사용하여 retention 구하기



3. 분석레시피 2

- 실전 문제 풀기 (LTV, RFM 계산하기)



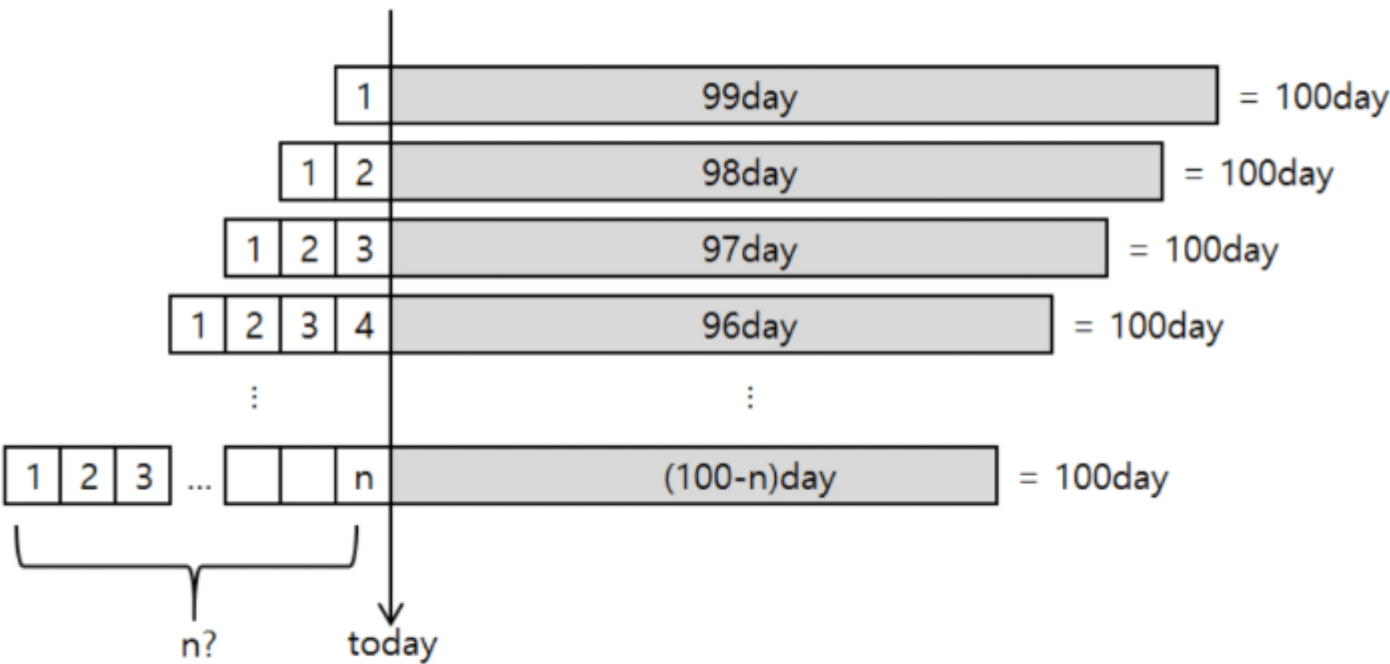
실전 문제 풀기

LTV를 구하자

- LTV를 구하는 가장 쉬운 방법은?
 - ARPU / Churn rate, 다만 이탈율(Churn rate)는 시간이 지나야 알 수 있음, 미리 알 수 없음

데이터로 확인하기

과거 100일 동안의 사용자 접속 세션 자료가 있다면, 어떤 사용자가 실제 이탈하였는지 알 수 있다. 이를 토대로 세션 자료에서 처음 며칠을 확인하면 이탈로 판단할 수 있는지 확인이 가능하다. 아래 그림에서 적절한 n 을 찾기 위해 n 을 변경하며 이탈 비율을 계산할 수 있다.



초반 n 일 동안의 접속 일수가 0일 경우에 사용자가 이탈할 확률 $p(churn = 1 | count.days_n = 0)$ 의 추이를 n 의 변화에 따라 관찰하면 적절한 n 의 값을 정할 수 있다.

이탈 기준 정하기 (Box and whisker)

실전 문제 풀기

LTV를 구하자

```
48 -- LTV를 구하기 위한 데이터 산출
49
50 SELECT std_dt, count(distinct account_id) dau
51           , sum(cast(amount as bigint)) amount
52           , count(case when amount > 0 then account_id else null end) buy_users
53           , (sum(cast(amount as bigint)) * 1.0) / count(distinct account_id) as arpu
54           , (count(case when amount > 0 then account_id else null end) * 1.0) / count(distinct account_id) as bur
55           , (sum(cast(amount as bigint)) * 1.0) / count(case when amount > 0 then account_id else null end) as ar
56           , (count(case when out_YN = 1 then account_id else null end) * 1.0) / count(distinct account_id) churn_
57 FROM #dau_churn
58 WHERE 1=1
59 and new_YN = 1
60 group by std_dt
61 order by 1;
62
```



13_LTV.sql

ARPU, ARPPU, Chrun Rate를 계산하자

실전 문제 풀기

엑셀 기능 추가 (분석 도구)

빠른 실행 도구 모음 사용자 지정(C)...
리본 메뉴 아래에 빠른 실행 도구 모음 표시(S)
리본 메뉴 사용자 지정(R)
리본 메뉴 축소(N)

추가 기능

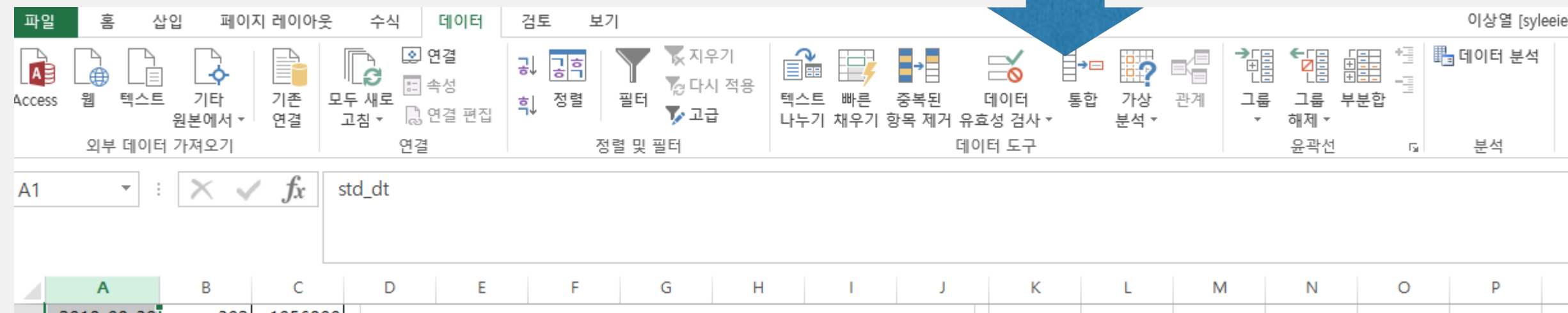
이름	위치	종류
비활성 응용 프로그램 추가 기능	C:\...Microsoft Office\Office15\DCF\NativeShim.dll	COM
Inquire	C:\...ivot Excel Add-in\PowerPivotExcelClientAddin.dll	XML
Microsoft Actions Pane 3	C:\...ower Map Excel Add-in\EXCELPLUGINSHELL.DLL	COM
Microsoft Office PowerPivot for Excel 2013	C:\...View Excel Add-in\AdHocReportingExcelClient.dll	COM
Power View	C:\...n Files\microsoft shared\Smart Tag\MOFL.DLL	동작
날짜 (XML)	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
날짜 (한국어 날짜 인식자)	C:\...ffice\Office15\Library\Analysis\ANALYS32.XLL	Excel
분석 도구	C:\...ce\Office15\Library\Analysis\ATPVBAEN.XLAM	Excel
분석 도구 - VBA	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
사람 이름 (한국어 사람 이름 인식자)	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
시간 (한국어 시간 인식자)	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
유평화 도구	C:\...rosoft Office\Office15\Library\EUROTOOL.XLAM	Excel
전화 번호 (한국어 전화 번호 인식자)	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
주소 (한국어 주소 인식자)	C:\...les\microsoft shared\Smart Tag\KSTRECOG.DLL	동작
해 찾기 추가 기능	C:\...ffice\Office15\Library\SOLVER\SOLVER.XLAM	Excel

사용 가능한 추가 기능(A):
☒ 분석 도구
☐ 분석 도구 - VBA
☐ 유평화 도구
☐ 해 찾기 추가 기능

확인
취소
찾아보기(B)...
자동화(U)...

분석 도구
통계 및 엔지니어링 분석용 데이터 분석 도구를 제공합니다.

빠른 실행 도구 모음으로 데이터 분석 도구를 추가해보자~



실전 문제 풀기

지수 평활법

> Formula

$$\begin{aligned} F_t &= F_{t-1} + a (A_{t-1} - F_{t-1}) \\ &= a * A_{t-1} + (1 - a) * F_{t-1} \end{aligned}$$

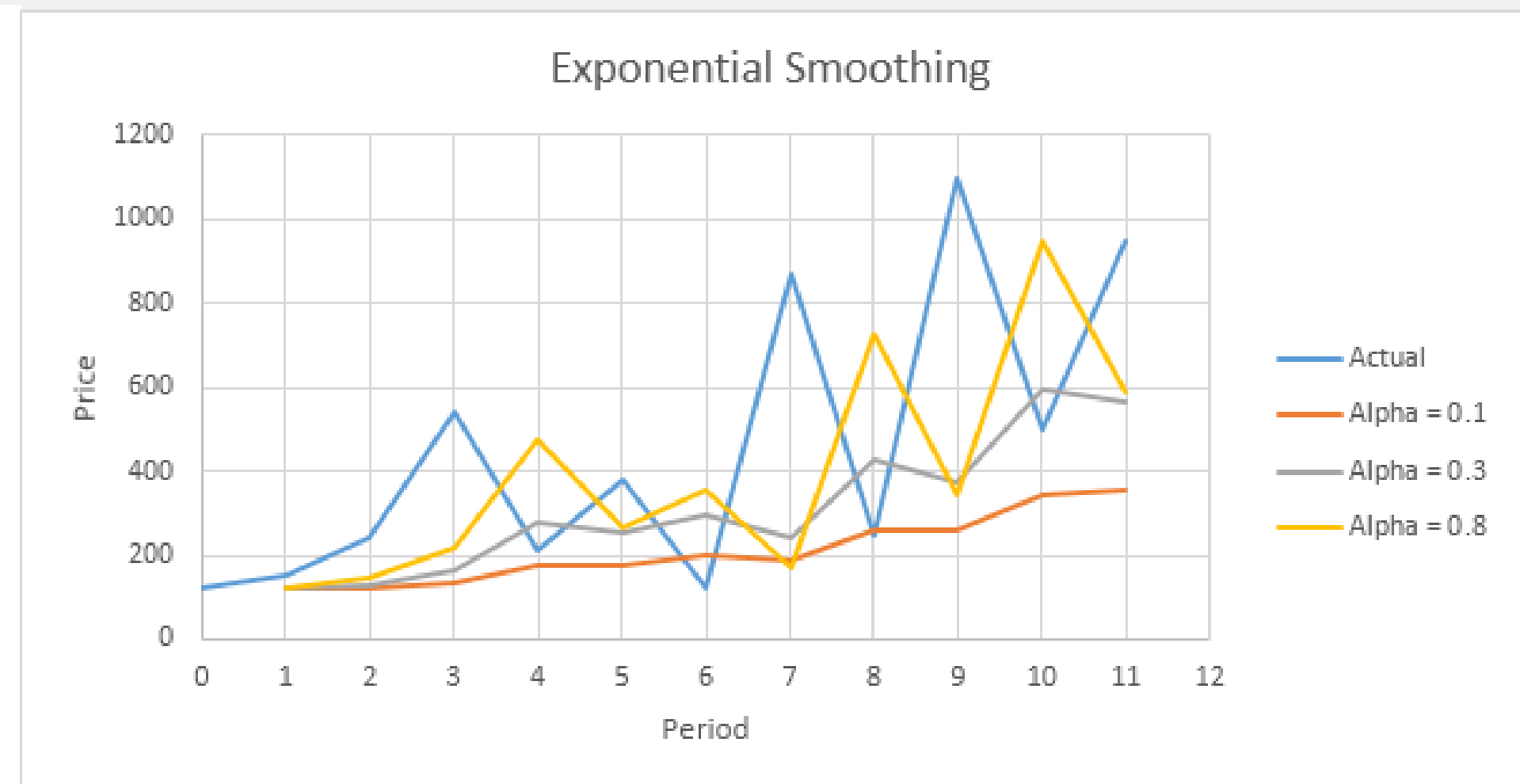
> As a gets closer to 1, the more weight put on the most recent demand number

h2. Exponential Smoothing Forecastet with $a = .3$

		Exponential Smoothing
Period	Demand	Forecast
1	12	10.00
2	15	10.60
3	11	11.92
4	9	11.64
5	10	10.85
6	8	10.60
7	14	9.82
8	12	11.07
9		11.35

$$\begin{aligned} F_2 &= .3 * 12 + .7 * 10 \\ &= 3.6 + 7 \\ &= 10.6 \end{aligned}$$

$$\begin{aligned} F_3 &= .3 * 15 + .7 * 10.6 \\ &= 11.92 \end{aligned}$$



실전 문제 풀기

엑셀 기능 추가 (분석 도구)

예측 이탈율 (지수 평활법)

지수 평활법

입력
입력 범위(I):

감쇠 인수(D):

☐ 이름표(L)

출력 옵션
출력 범위(O):

새로운 워크시트(P):

새로운 통합 문서(W):

☒ 차트 출력(C) ☐ 표준 오차(S)

확인 취소 도움말(H)

입력 범위 : \$H\$93:\$H\$198
출력 범위 : \$I\$198

예측 ARPU (지수 평활법)

지수 평활법

입력
입력 범위(I):

감쇠 인수(D):

☐ 이름표(L)

출력 옵션
출력 범위(O):

새로운 워크시트(P):

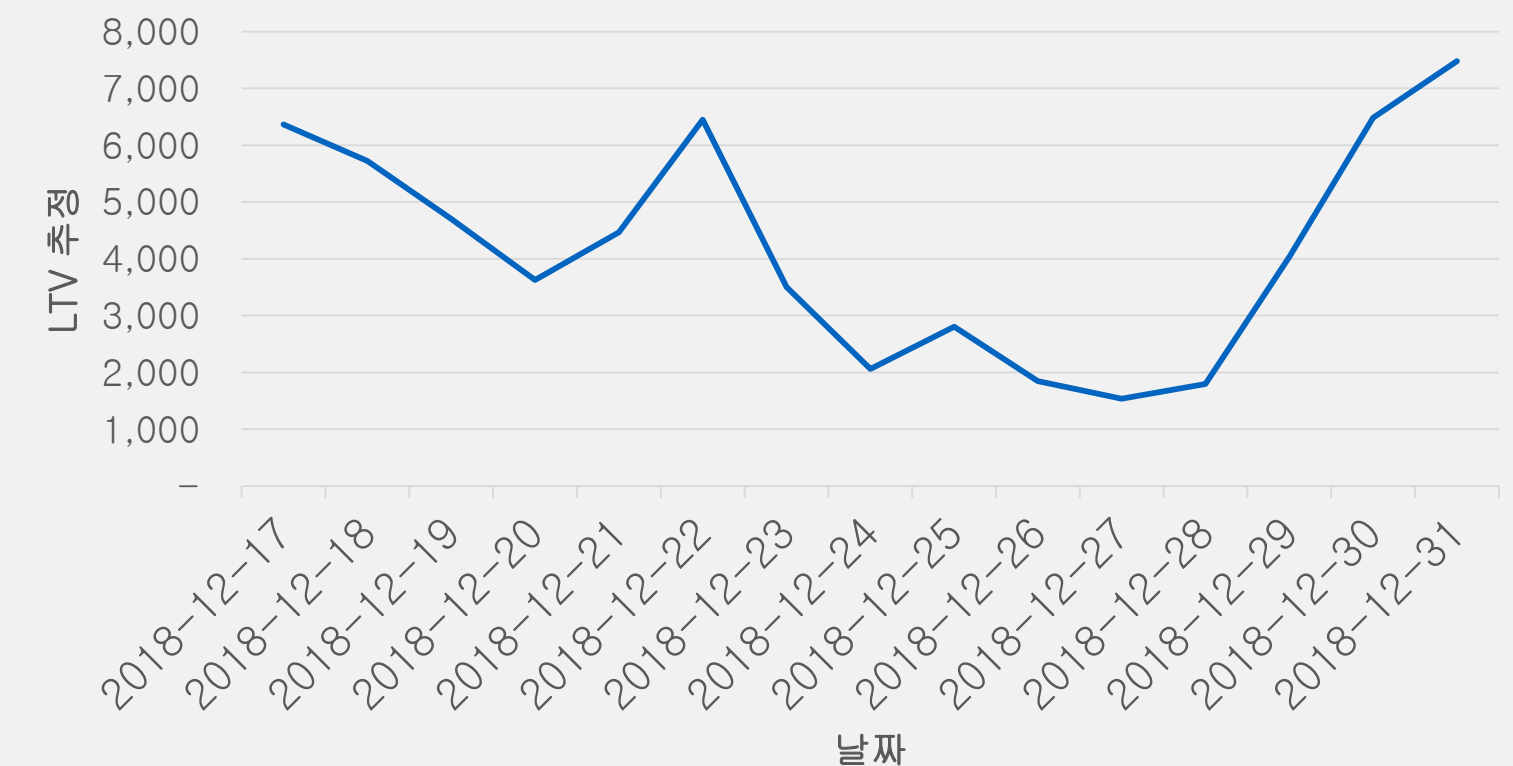
새로운 통합 문서(W):

☒ 차트 출력(C) ☐ 표준 오차(S)

확인 취소 도움말(H)

입력 범위 : \$E\$93:\$E\$198
출력 범위 : \$J\$198

일별 신규유저의 LTV 값 추정



05_LTV 지표.xlsx 참고할 것



실전 문제 풀기

RFM을 구하자 (<https://www.putler.com/rfm-analysis/>)

- RFM? 가치있는 고객을 추출해내어 이를 기준으로 고객을 분류할 수 있는 매우 간단하면서도 유용하게 사용될 수 있는 방법으로 알려져 있어서 마케팅에서 자주 사용됨

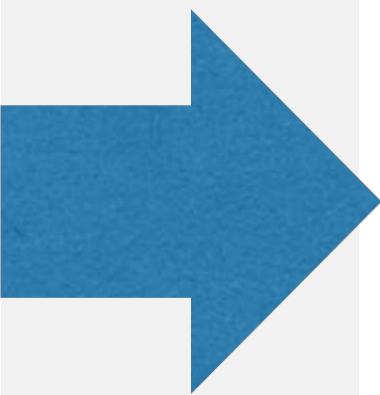
RFM은 **Recency, Frequency, Monetary**의 약자로 고객의 가치를 다음의 세 가지 기준에 의해 계산

Recency – 거래의 최근성: 고객이 얼마나 최근에 구입했는가?

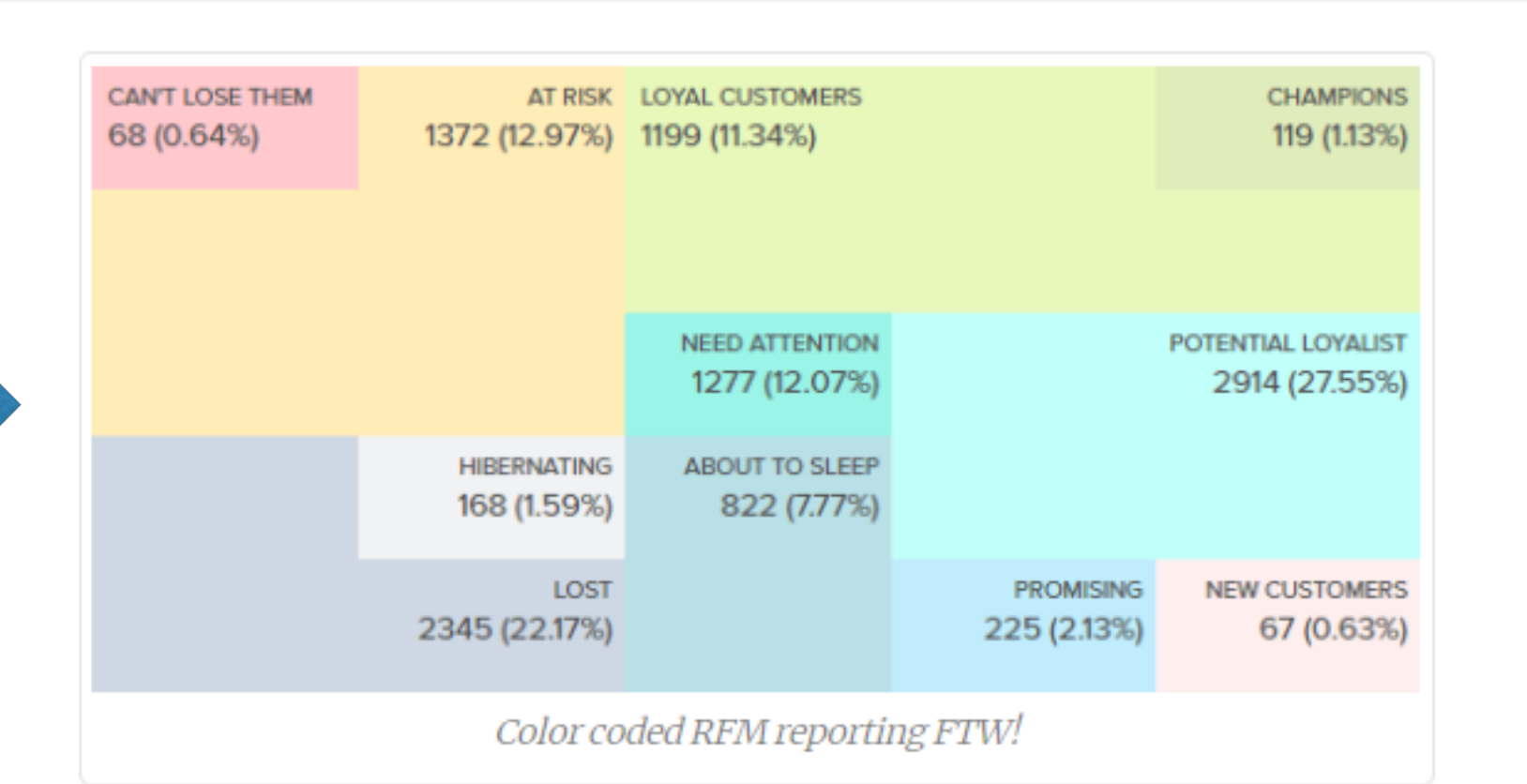
Frequency – 거래빈도: 고객이 얼마나 빈번하게 우리 상품을 구입했나?

Monetary – 거래규모: 고객이 구입했던 총 금액은 어느 정도인가?

Customer Segment	Recency Score Range	Frequency & Monetary Combined Score Range
Champions	4-5	4-5
Loyal Customers	2-5	3-5
Potential Loyalist	3-5	1-3
Recent Customers	4-5	0-1
Promising	3-4	0-1
Customers Needing Attention	2-3	2-3



11 customer segments based on RFM scores.



RFM summary report

실전 문제 풀기

RFM 구하기

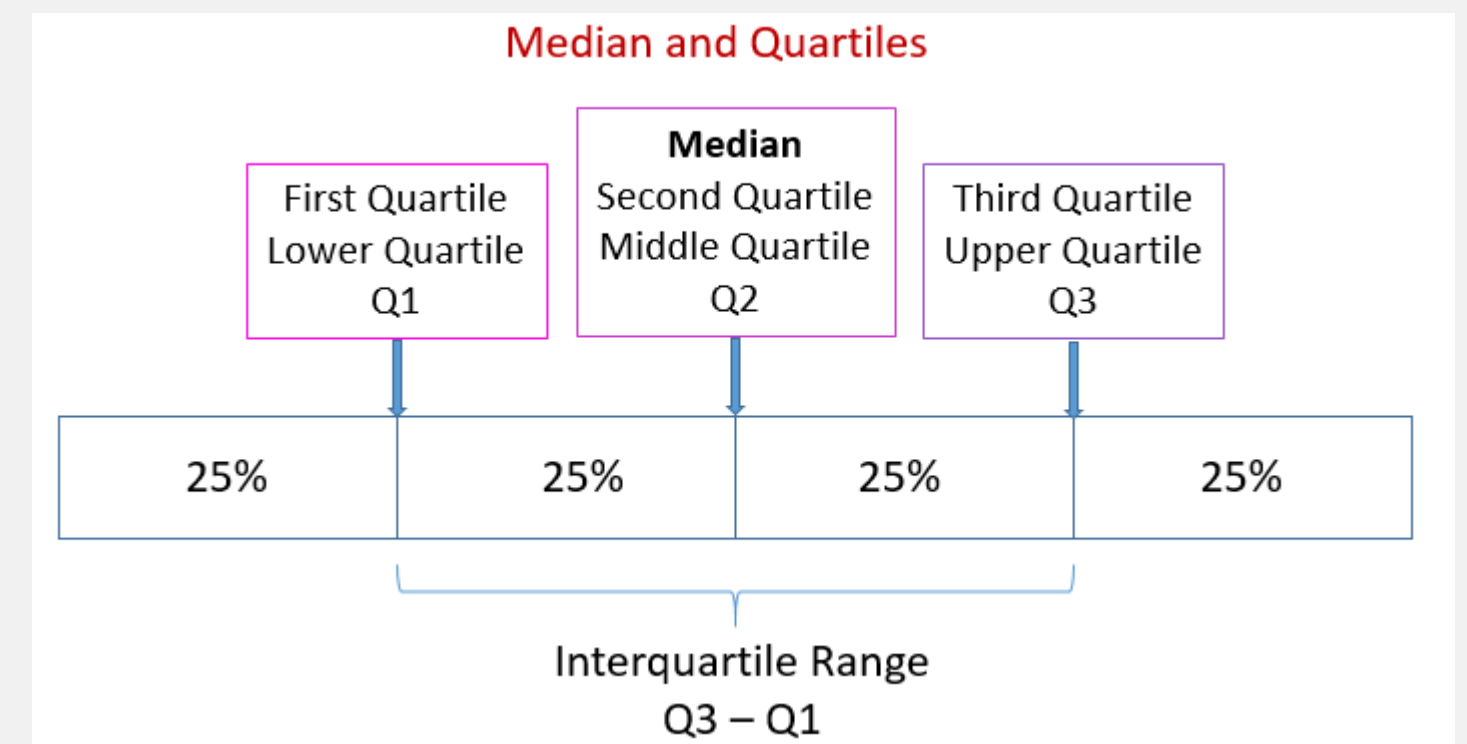
```
16 if object_id('tempdb..#PERCENTILE') is not null
17     drop table #PERCENTILE
18
19 SELECT DISTINCT
20     PERCENTILE_DISC(0.25) WITHIN GROUP (ORDER BY r) OVER (PARTITION BY NULL) r_25,
21     PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY r) OVER (PARTITION BY NULL) r_50,
22     PERCENTILE_DISC(0.75) WITHIN GROUP (ORDER BY r) OVER (PARTITION BY NULL) r_75,
23     PERCENTILE_DISC(1) WITHIN GROUP (ORDER BY r) OVER (PARTITION BY NULL) r_100,
24     PERCENTILE_DISC(0.25) WITHIN GROUP (ORDER BY f) OVER (PARTITION BY NULL) f_25,
25     PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY f) OVER (PARTITION BY NULL) f_50,
26     PERCENTILE_DISC(0.75) WITHIN GROUP (ORDER BY f) OVER (PARTITION BY NULL) f_75,
27     PERCENTILE_DISC(1) WITHIN GROUP (ORDER BY f) OVER (PARTITION BY NULL) f_100,
28     PERCENTILE_DISC(0.25) WITHIN GROUP (ORDER BY m) OVER (PARTITION BY NULL) m_25,
29     PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY m) OVER (PARTITION BY NULL) m_50,
30     PERCENTILE_DISC(0.75) WITHIN GROUP (ORDER BY m) OVER (PARTITION BY NULL) m_75,
31     PERCENTILE_DISC(1) WITHIN GROUP (ORDER BY m) OVER (PARTITION BY NULL) m_100
32 into #PERCENTILE
33 FROM [da_db].[dbo].[RFM]
34 WHERE 1=1;
35
36
37 -- r_25 r_50 r_75 r_100 f_25 f_50 f_75 f_100 m_25 m_50 m_75 m_100
38 -- 40 103 135 160 18 42 76 160 10000 20000 81000 27844000
39
```

	ABC_r_25	ABC_r_50	ABC_r_75	ABC_r_100	ABC_f_25	ABC_f_50	ABC_f_75	ABC_f_100	ABC_m_25	A
1	129	150	5	99	15	3	54	99	13000	3

Window Functions을 이용하여
분포를 구해보자!



14_RFM.sql

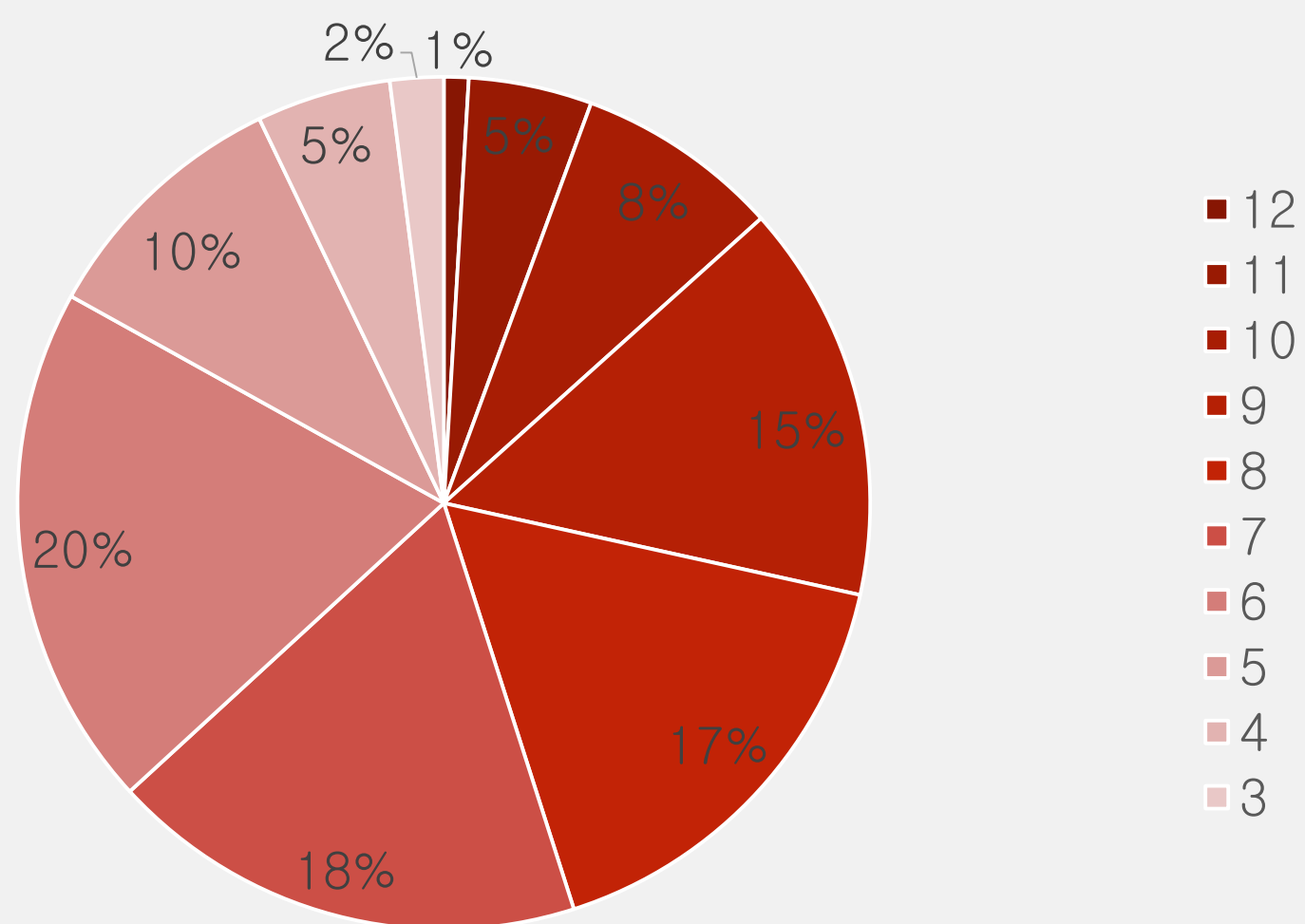




실전 문제 풀기

RFM 구하기

RFM 비중 그래프 (전체 유저)



RFM Score를 3가지로 구분해서 비교						
m_score (모두)						
RF 테이블 비율						
합계 : user 열 레이블						
행 레이블		1	2	3	4 총합계	
1		9%	4%	7%	6%	26%
2		11%	3%	5%	7%	26%
3		4%	11%	5%	4%	24%
4		3%	6%	7%	8%	24%
총합계		26%	24%	25%	25%	100%
f_score (모두)						
RM 테이블 비율						
합계 : user 열 레이블						
행 레이블		1	2	3	4 총합계	
1		7%	8%	6%	4%	26%
2		8%	6%	9%	3%	26%
3		6%	8%	6%	5%	24%
4		6%	8%	5%	5%	24%
총합계		26%	30%	27%	17%	100%



Appendix

1) 모바일 데이터 분석의 시작: AARRR (해적지표)

– <https://www.theteams.kr/teams/1043/post/64905>

2) MOOC AB-Testing

– <https://www.udacity.com/course/ab-testing--ud257>

3) 캐글보개기 - 분석툴(Part1) SQL 스터디 자료

– <https://github.com/KaggleBreak/analyticstool/tree/master/part1/sql>

4) MSSQL - 데이터 타입 알아보기 (종류, 범위, 크기)

– <https://angun2g.tistory.com/25>

5) MSSQL - 데이터 타입변환 (CONVERT,CAST) 사용법 & 예제

– <https://coding-factory.tistory.com/100>

6) 이탈기준 정하기 (Box and Whisker)

– https://boxnwhis.kr/2014/09/15/make_a_criterion_for_churn.html

7) WITH절과 CTE(비재귀적) - egloos

– <http://egloos.zum.com/sweeper/v/3000836>

8) Calculating Customer Lifetime Value: SQL Example

– <https://www.kdnuggets.com/2018/02/calculating-customer-lifetime-value-sql-example.html>

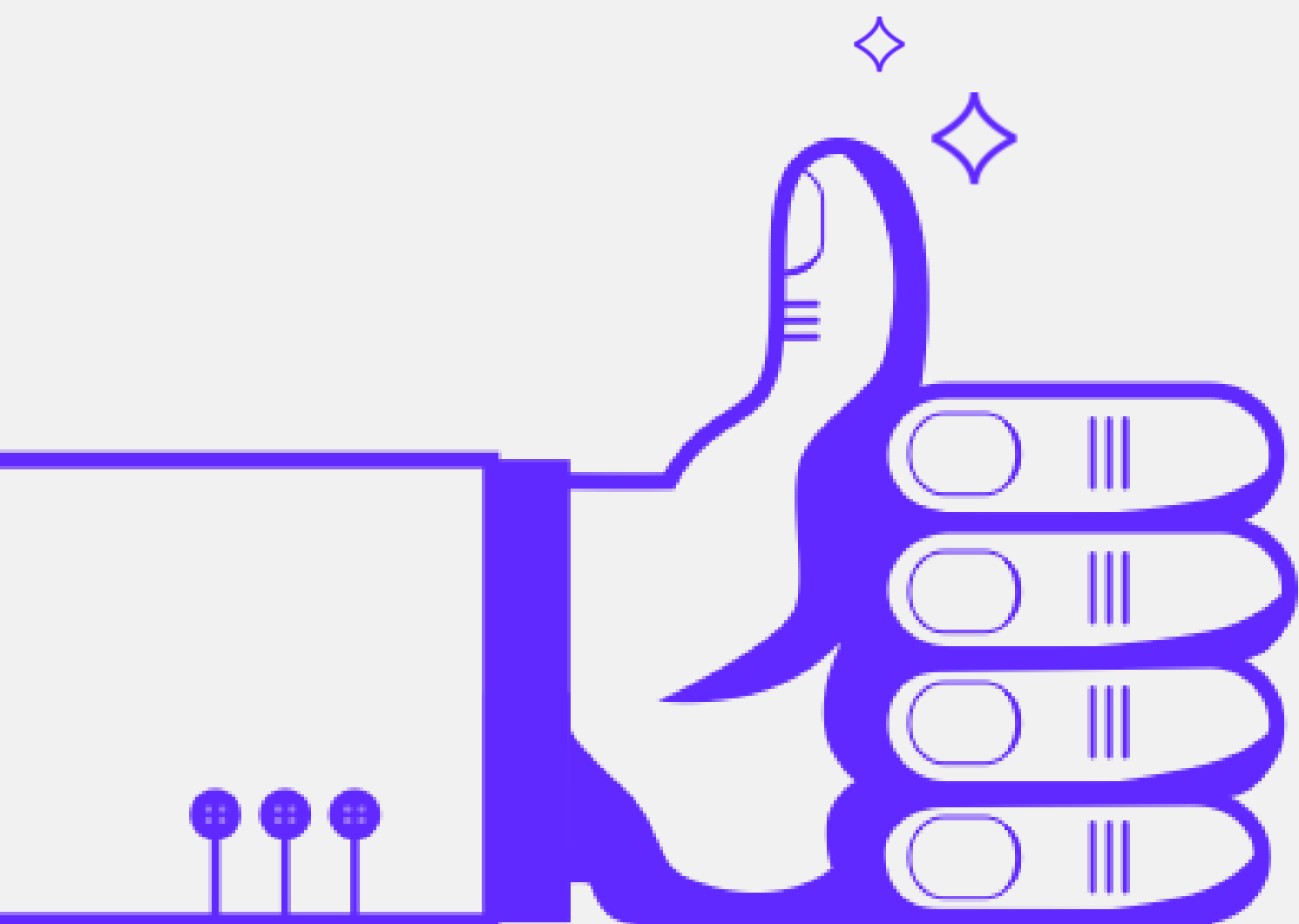
9) RFM Analysis For Successful Customer Segmentation

– <https://www.putler.com/rfm-analysis/>

발표 자료 만드는데 참고했습니다.
감사합니다.



감사합니다.



SQL 데이터 분석에서 더 궁금한 내용은...

syleeie@gmail.com 으로 이메일 문의해주세요.

데이터뽀개기 커뮤니티 그룹은
현장에서 데이터로 일하는 많은 분들의
문화가 개선할 수 있도록 노력하고 있습니다.

언제나 많은 관심 부탁드립니다 (_)