

---

# **CITY(중앙 커뮤니티)**

## **ERD PRESENTATION**

# 김경모 # 박다영 # 박은경 # 박유신 # 윤태식

# INDEX

1	2	3	4	5
ERD	ENTITY	ATTRIBUTE RELATIONSHIP PARTICIPATION CONSTRAINTS		



# 1. ERD

본 프로젝트 ERD (데이터베이스 개체 설계도) 첨부

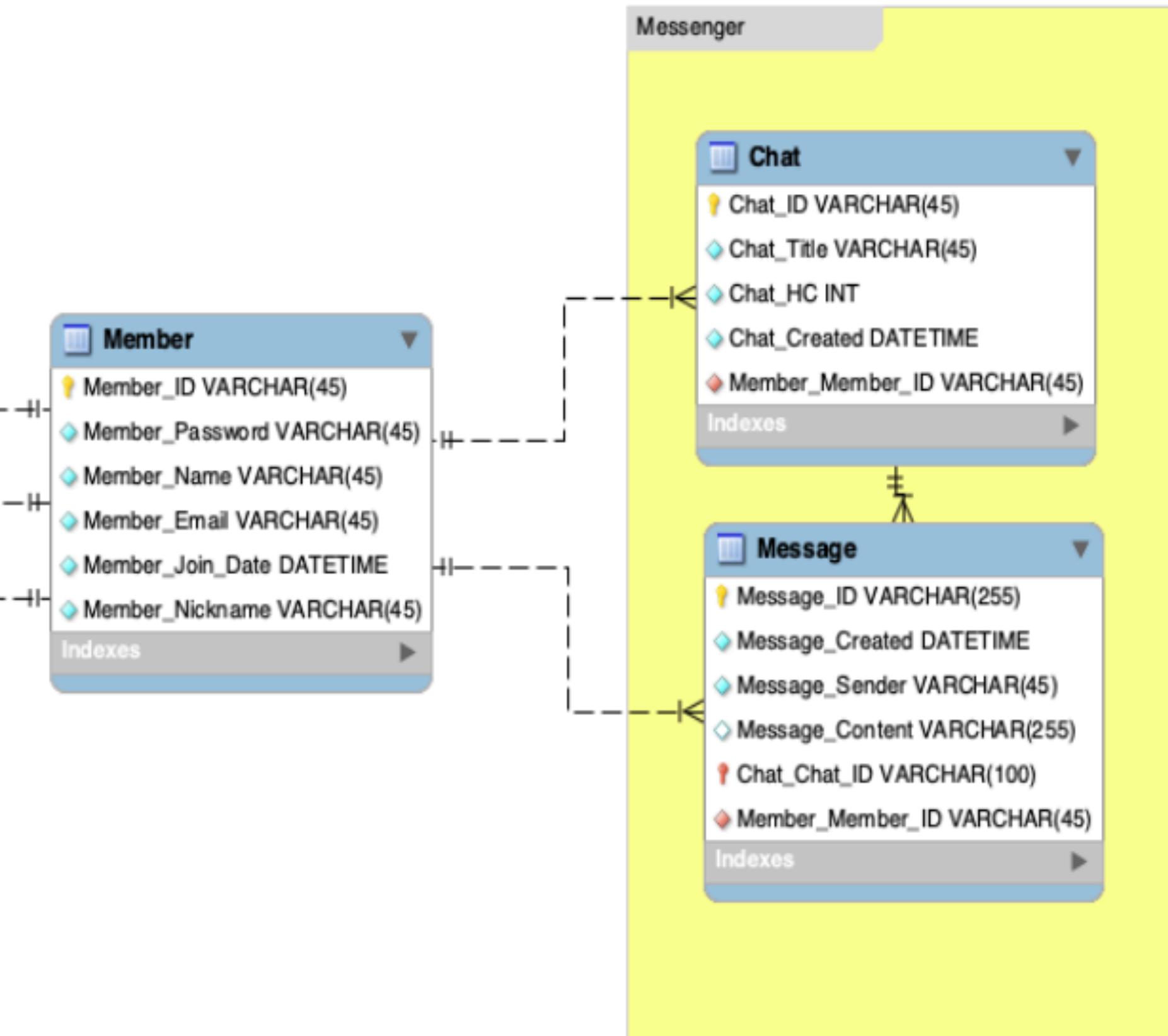
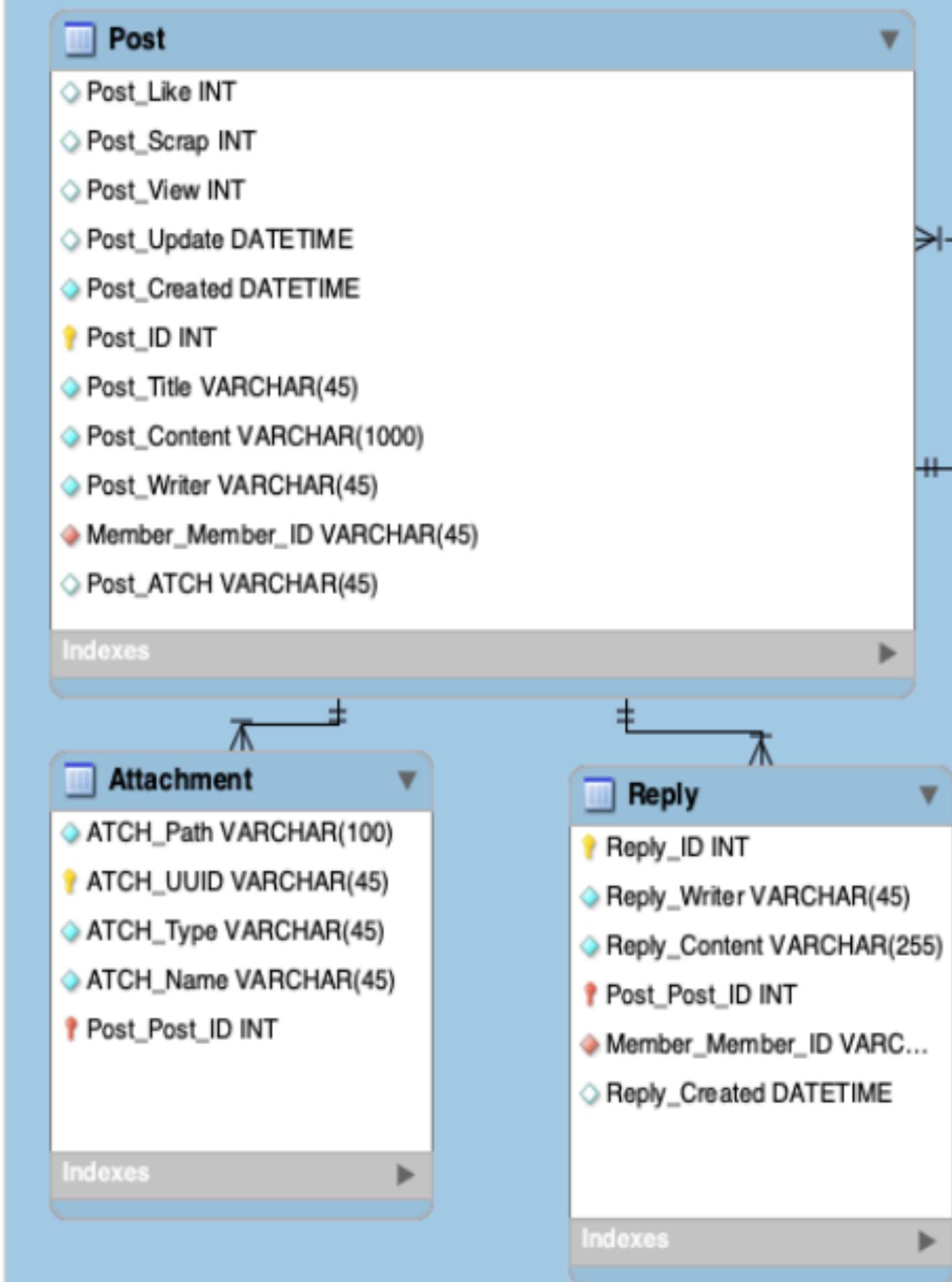


# MYSQL

# DATABASE

# WORKBENCH

## Board

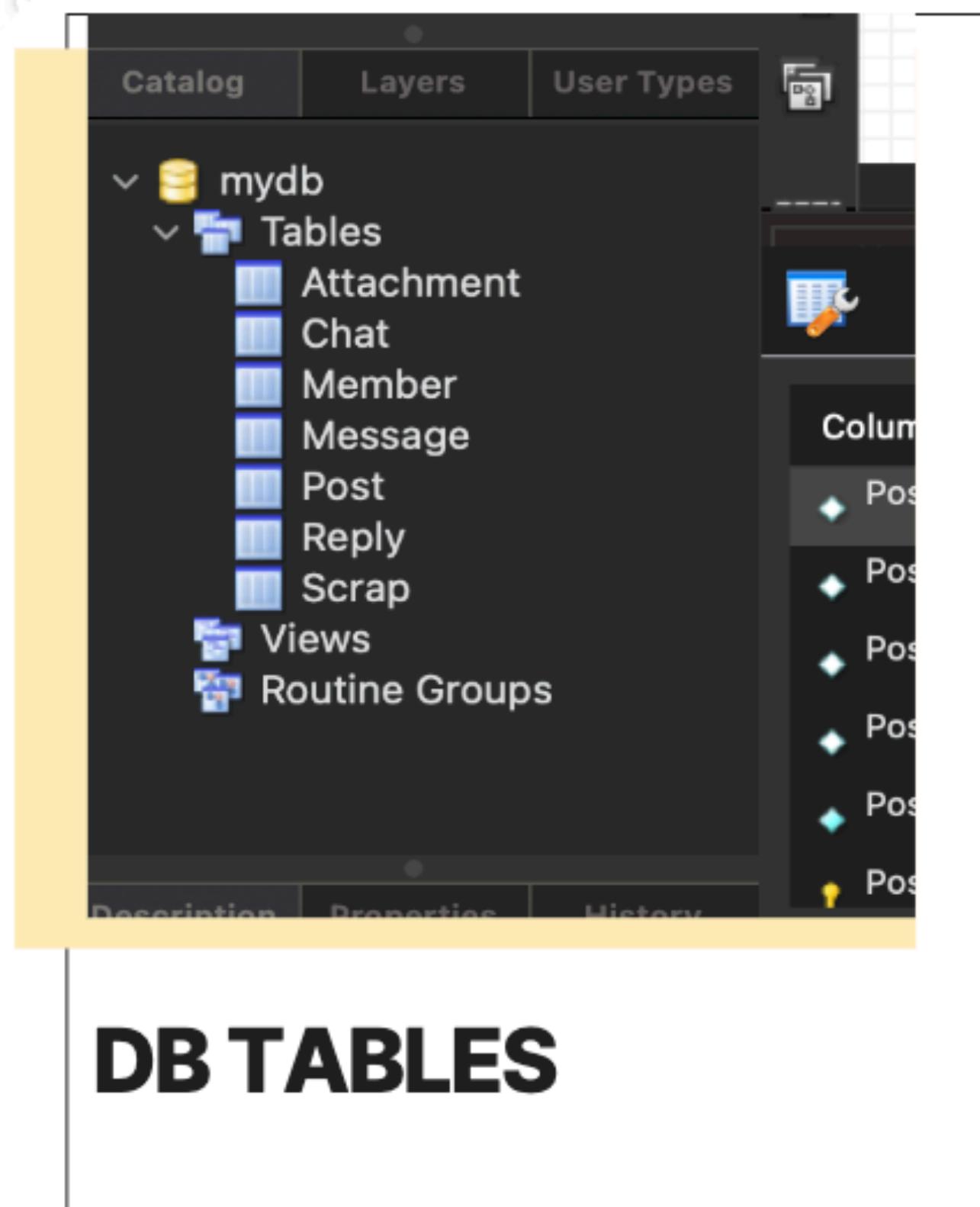


## 2. ENTITY

DB 테이블 구성

# CITY'S

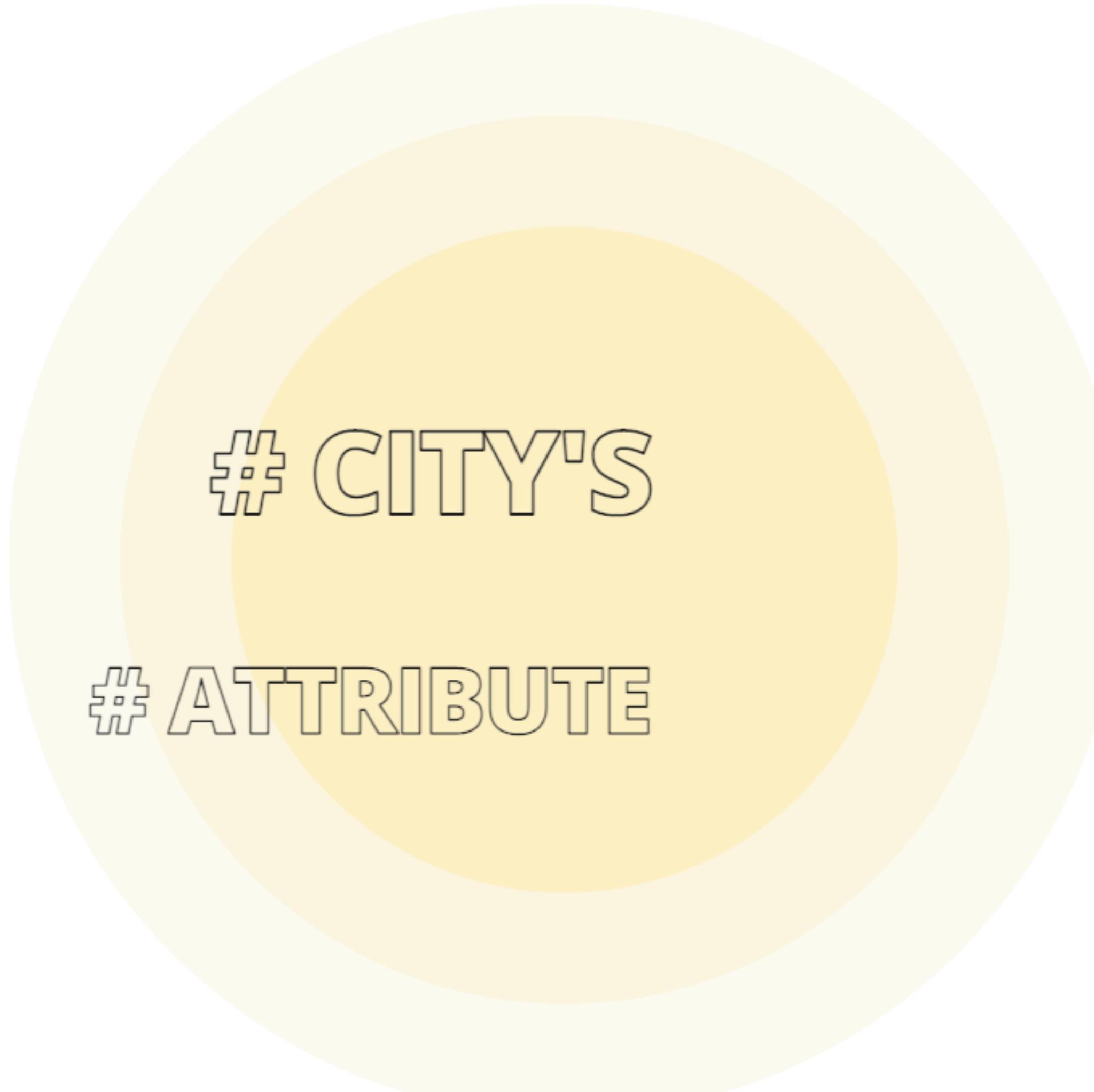
# ENTITIES



- 1. MEMBER** 유저 정보
- 2. POST** 게시글 정보
- 3. ATCH** 첨부파일 정보
- 4. REPLY** 댓글 정보
- 5. SCRAP** 스크랩 정보
- 6. CHAT** 채팅 정보
- 7. MESSAGE** 메시지 정보

# 3. ATTRIBUTE

각 ENTITY 별 속성 소개



# CITY'S  
# ATTRIBUTE

# 1. Member

(PK) Member\_ID (VARCHAR(45)) : 유저 아이디

Member\_Password (VARCHAR(45)) : 유저 비밀번호

Member\_Name (VARCHAR(45)) : 유저 이름

Member\_Email (VARCHAR(45)) : 유저 이메일

Member\_Nickname (VARCHAR(45)) : 유저 닉네임

## 2. Post

(PK) Post\_ID (INT) : 게시글 등록 순서에 따른 PK 값

Post\_Title (VARCHAR(45)) : 게시글 제목

Post\_Writer (VARCHAR(45)) : 작성자

Post\_Content (VARCHAR(45)) : 게시글 내용

Post\_ATCH (VARCHAR(45)) : 게시글 첨부파일

Post\_Like (INT) : 게시글 공감수

Post\_Scrap (INT) : 게시글 스크랩수

Post\_View (INT) : 게시글 조회수

Post\_Created (DATETIME) : 게시일, 게시글 정렬에 사용

Post\_Update (DATETIME) : 게시글 수정일

### 3. Attachment

(PK) ATCH\_UUID (VARCHAR(45)) : UUID를 생성하여 첨부파일별 아이디를 중복없이 지정, 서버에 저장될 파일 이름

ATCH\_Name (VARCHAR(45)) : 첨부파일의 원래 이름 (사용자 지정)

ATCH\_Type (VARCHAR(45)) : 첨부파일의 타입

ATCH\_Path (VARCHAR(100)) : 첨부파일 저장 경로

## 4. Reply

(PK) Reply\_ID (INT) : Post\_ID와 댓글 uid로 구성 (일반적이라고 함)

Reply\_Writer (VARCHAR(45)) : 댓글 작성자

Reply\_Content (VARCHAR(255)) : 댓글 내용

Reply\_Created (DATETIME) : 댓글 생성일, 댓글 정렬에 이용

## 5. Scrap

(PK) Scrap\_ID (VARCHAR(45)) : Post\_ID + Member\_ID 조합으로 PK 값 구성

Scrap\_Date (DATETIME) : 스크랩 날짜, 개인별 스크랩 한 게시글 정렬 시 사용

## 6. Chat

(PK) Chat\_ID (VARCHAR(45)) : UUID 생성하여 PK 값 구성 (유저 아이디 조합으로 설정할 경우 단체 채팅에서 효율성이 매우 떨어짐, Chat\_Title 과 Chat\_Created 는 중복 가능성이 있음)

Chat\_Title (VARCHAR(45)) : 채팅방 이름

Chat\_HC (INT) : 채팅 참여자 수

Chat\_Created (DATETIME) : 채팅방 생성일, 개인별 채팅목록 정렬 시 사용

## 7. Message

(PK) Message\_ID (VARCHAR(255)) : Chat\_ID + Message\_Sender + Message\_Created 조합으로 구성

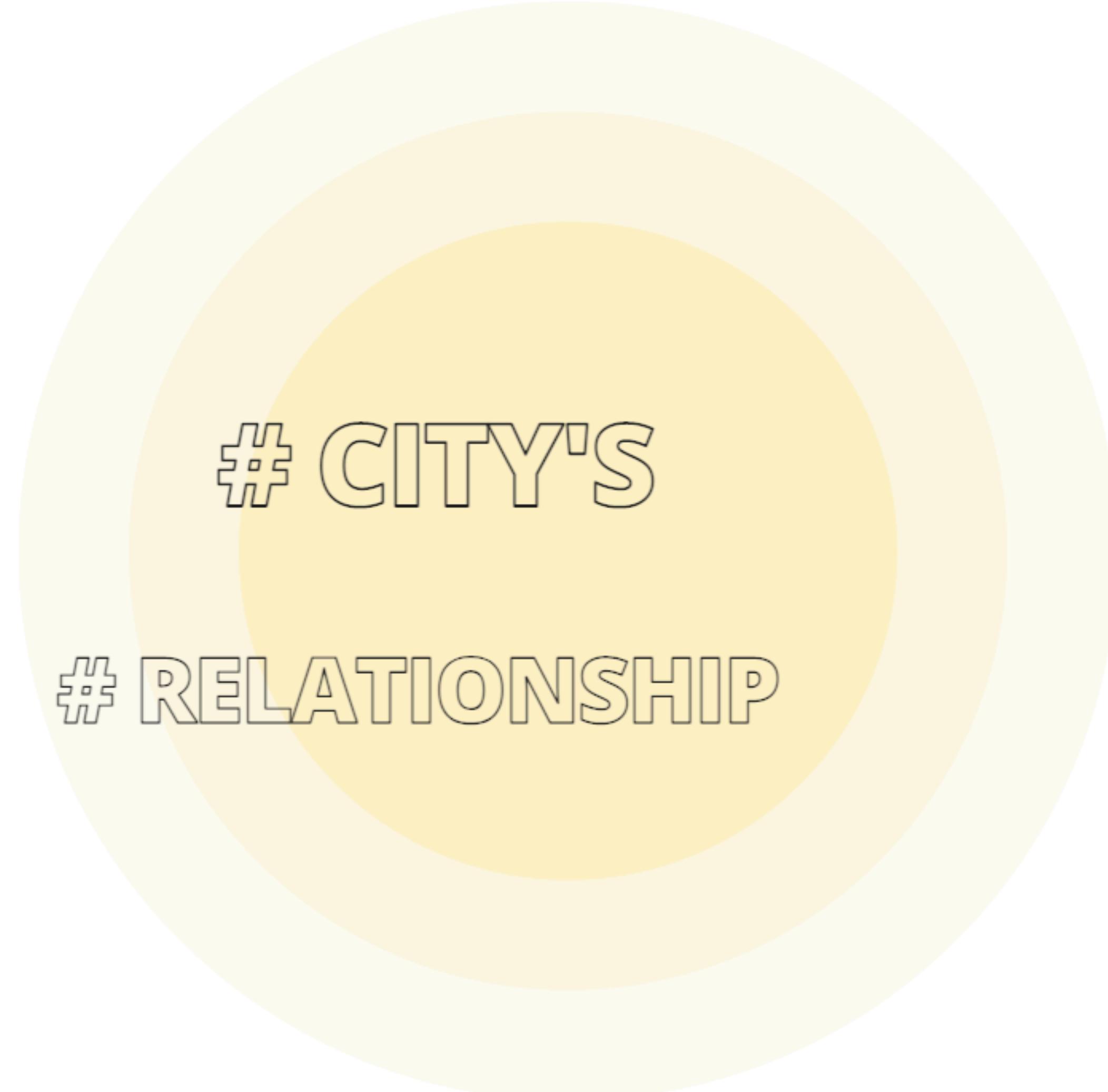
Message\_Content (VARCHAR(255)) : 메시지 내용

Message\_Created (DATETIME) : 메시지 생성일, 메시지 정렬 시 사용

Message\_Sender (VARCHAR(45)) : 메시지를 보낸 사람

# 4. RELATIONSHIP

ENTITY 간의 관계 설명



# CITY'S  
# RELATIONSHIP

# Member Post

은 erd에서 관계를 정의할때 유저와 유저가 쓴 게시글은 무슨 관계를 가질까?

 유저와 유저가 쓴 게시글은 일대다(One-to-Many) 관계를 가집니다. 즉, 한 명의 유저는 여러 개의 게시글을 작성할 수 있지만, 하나의 게시글은 오직 한 명의 유저에 의해 작성됩니다. 이러한 관계는 ERD에서 다음과 같이 표현할 수 있습니다.

sql

```
+-----+ +-----+
| User | | Post |
+-----+ +-----+
| UserId | <----> | PostId |
| Name  | | Title   |
| Email | | Content |
+-----+ | CreatedAt |
                  +-----+
```

Copy code

여기서 `UserId`는 `User` 엔티티의 기본키(PK)이고, `PostId`는 `Post` 엔티티의 PK입니다. `UserId`는 `Post` 엔티티에서 외래키(FK)로 사용되어, 두 엔티티 간의 관계를 나타냅니다. 이 관계를 통해 하나의 게시글이 오직 한 명의 유저에 의해 작성되며, 한 명의 유저는 여러 개의 게시글을 작성할 수 있음을 나타낼 수 있습니다.

# Member && Chat

온 erd에서 관계를 정의할 때 유저가 생성한 채팅방은 유저와 무슨 관계를 가질까?



유저가 생성한 채팅방은 유저와 일대다(One-to-Many) 관계를 가집니다. 즉, 한 명의 유저는 여러 개의 채팅방을 생성할 수 있지만, 하나의 채팅방은 오직 한 명의 유저에 의해 생성됩니다. 이러한 관계는 ERD에서 다음과 같이 표현할 수 있습니다.

sql

Copy code

User	ChatRoom
UserId	<----> RoomId
Name	Name
Email	CreatedAt
	CreatorId

여기서 `UserId`는 `User` 엔티티의 기본키(PK)이고, `RoomId`는 `ChatRoom` 엔티티의 PK입니다. `CreatorId`는 `ChatRoom` 엔티티에서 `User` 엔티티의 외래키(FK)로 사용되어, 두 엔티티 간의 관계를 나타냅니다. 이 관계를 통해 하나의 채팅방이 오직 한 명의 유저에 의해 생성되며, 한 명의 유저는 여러 개의 채팅방을 생성할 수 있음을 나타낼 수 있습니다.

Post

&&

Attachment  
Reply  
Like  
Scrap

온 erd에서 관계를 정의할 때 게시글의 첨부파일은 게시글과 무슨관계를 가질까?

 게시글과 게시글의 첨부파일은 일대다(One-to-Many) 관계를 가집니다. 즉, 하나의 게시글에 여러 개의 첨부파일이 있을 수 있지만, 하나의 첨부파일은 오직 한 개의 게시글에만 속할 수 있습니다. 이러한 관계는 ERD에서 다음과 같이 표현할 수 있습니다.

lua

Post	Attachment
PostId	AttachmentId
Title	FileName
Content	ContentType
Created	Size

Copy code

여기서 `PostId`는 `Post` 엔티티의 기본키(PK)이고, `AttachmentId`는 `Attachment` 엔티티의 PK입니다. `PostId`는 `Attachment` 엔티티에서 외래키(FK)로 사용되어, 두 엔티티 간의 관계를 나타냅니다. 이 관계를 통해 하나의 게시글에 여러 개의 첨부파일이 있을 수 있으며, 하나의 첨부파일은 오직 한 개의 게시글에만 속할 수 있음을 나타낼 수 있습니다.

# Member && Attachment Reply Scrap Message



유저가 쓴 게시글의 공감수, 스크랩수, 댓글, 그리고 유저가 만든 채팅방의 메시지는 모두 해당 게시글이나 채팅방에 대한 추가 정보를 제공하는 것으로 보입니다. 이러한 정보는 ERD에서 일반적으로 엔티티 속성(Attribute)으로 정의됩니다. 예를 들어, 게시글에 대한 공감수, 스크랩수, 댓글 등은 모두 `Post` 엔티티의 속성으로 정의될 수 있습니다. 마찬가지로, 채팅방에 대한 메시지는 `ChatRoom` 엔티티의 속성으로 정의될 수 있습니다. 이러한 속성은 해당 엔티티에 대한 정보를 보완하거나 확장하는 데 사용됩니다.

따라서 이러한 관계를 ERD에서는 일반적으로 다음과 같이 표현합니다.

sql

Copy code

User
UserId
Name
Email
...

writes

Post
PostId
Title
Content
Created
Updated
...

여기서 `User`와 `Post` 사이의 관계는 일대다(One-to-Many) 관계입니다. 즉, 하나의 유저는 여러 개의 게시글을 작성할 수 있지만, 하나의 게시글은 오직 한 명의 유저에 의해서만 작성될 수 있습니다.

# 5. PARTICIPATION CONSTRAINTS

ENTITIY 참여 제약 조건

# CITY'S

# PARTICIPATION  
CONSTRAINTS

- 유저(User) 엔터티는 다른 엔터티와의 관계에서 반드시 참여해야 함
- 게시글(Post) 엔터티는 반드시 한 명 이상의 유저(User) 엔터티와의 관계에서 참여해야 함
- 게시글(Post) 엔터티는 첨부파일(Attachment) 엔터티와 관계를 가질 수 있지만, 첨부파일(Attachment) 엔터티는 반드시 존재하지 않을 수도 있음
- 게시글(Post) 엔터티는 댓글(Comment) 엔터티와 관계를 가질 수 있지만, 댓글(Comment) 엔터티는 반드시 존재하지 않을 수도 있음
- 유저(User) 엔터티는 채팅방(Chat Room) 엔터티와 관계를 가질 수 있지만, 채팅방(Chat Room) 엔터티는 반드시 존재하지 않을 수도 있음
- 채팅방(Chat Room) 엔터티는 반드시 한 명 이상의 유저(User) 엔터티와의 관계에서 참여해야 함
- 채팅방 메시지(Chat Message) 엔터티는 반드시 하나의 채팅방(Chat Room) 엔터티와의 관계에서 참여해야 함

User	Comment	Post	Attachment	Chat Room	Chat Message	
	FK post_id  FK user_id  FK post_id			FK room_id		
PK	+-----+-----+-----+			+-----+-----+		
user_id  FK user_id  content	file_name	PK		PK		
content   created_at  file_type		room_id		msg_id		
+-----+ created_at+ likes	+-----+ name					
+-----+-----+-----+-----+-----+-----+-----+						

# 감사합니다

---