**Tp clone Insta :**

* Users
* Photos
* Commentaires
* Likes

**users** :

* Id : int auto increment not null primary key
* Nom : varchar(15)
* Prenom : varchar(15)
* Username : varchar(15) not null unique
* Email : varchar(25) not null unique
* Avatar : varchar(200)

Creation : CREATE TABLE users(id int auto\_increment not null primary key,

nom varchar(15), prenom varchar(15),

username varchar(15) not null unique,

email varchar(25) not null unique,

avatar varchar(200) );

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | nom | prenom | username | email | avatar |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

INSERT INTO users(nom, prenom, username, email, avatar) values(“will”, “smith”, “willy”, “[belhair@gmail.com](mailto:belhair@gmail.com)”, “https://www”) ,(“albert”, “anastasia”, “theBest”, “albert@gmail.fr” , “https://ww.com” );

**Photos**:

* Id: int auto\_increment not null primary key
* url: varchar(200) not null;
* idUser: int not null;
* FK: foreign key(idUser) references users(id) on delete cascade on update cascade,

Creation: CREATE TABLE photos(id int auto\_increment not null primary key,

url varchar(200) not null, idUser int not null,

FOREIGN KEY(idUser) REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE);

|  |  |  |
| --- | --- | --- |
| Id | url | idUser |
|  |  |  |
|  |  |  |

INSERT INTO photos(url, idUser) values(“http://image” , 1), (“http://image” , 1), (“http://image” , 2), (“http://image” , 1), (“http://image” , 2);

**Commentaires**

* Id : int auto\_increment not null primary key
* Text: varchar(300) not null
* idPhoto: int not null;
* FK: foreign key(idPhoto) references photos(id) on delete cascade on update cascade

Creation: CREATE TABLE commentaires(id int auto\_increment not null primary key,

text varchar(300) not null, idPhoto int not null,

idUser int not null,

FOREIGN KEY(idPhoto) REFERENCES photos(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(idUser) REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE);

|  |  |  |  |
| --- | --- | --- | --- |
| id | text | idPhoto | idUser |
|  |  |  |  |
|  |  |  |  |

INSERT INTO commentaires(text, idPhoto, idUser) values(‘super cette photo’, 1, 1), (‘go off’, 1, 2), (‘nice car’, 2, 1) ;

Likes:

Id: int auto increment not null primary key

Type: varchar(1) not null;

idPhoto: int

idCommentaire: int

Creation: CREATE TABLE likes(id int auto\_increment not null primary key, like\_type boolean not null, idPhoto int, idCommentaire int, FOREIGN KEY(idPhoto) REFERENCES photos(id) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(idCommentaire) REFERENCES commentaires (id) ON DELETE CASCADE ON UPDATE CASCADE);

|  |  |  |  |
| --- | --- | --- | --- |
| id | type | idPhoto | idCommentaire |
|  |  |  |  |
|  |  |  |  |

Exo LMD:

1. INSERT INTO users(nom, prenom, username, email, avatar) values(“joanne”, “kirk”, “joa\_kirk”, “[entreprise@gmail.com](mailto:entreprise@gmail.com)”, “https://www”);
2. INSERT INTO photos(url) values(“http://image/facebook” , 100);
3. SELECT commentaires.text, users.username FROM users INNER JOIN commentaires ON commentaires.idUser = users.id;
4. SELECT commentaires.text, photos.url FROM photos INNER JOIN commentaires ON commentaires.idPhoto = photos.id;
5. SELECT users.username, photos.url FROM photos INNER JOIN users ON users.id = photos.idUser;
6. SELECT commentaires.text, users.username FROM users INNER JOIN commentaires ON commentaires.idUser = users.id WHERE commentaires.idPhoto = 3;
7. SELECT commentaires.text, photo\_user.username, photo\_user.url FROM commentaires INNER JOIN

(SELECT photos.id as ID, users.username AS username, photos.url AS url FROM photos INNER JOIN users ON users.id = photos.idUser ) AS photo\_user ON photo\_user.id = commentaires.idPhoto;