**DATABASE MANAGEMENT**

2018-2019 HOMEWORK

SOCIAL RESUME DATABASE

05140000645 ÖZGENUR ÇAPUK

05170000018 İREM TUNÇ

INDEX

Analaysis..........................................................................................................................2

What is the aim of each application?..................................................................3

What are the main entities of them?..................................................................3

What are the charactheristic of entitiy...............................................................4

Facebook.................................................................................................4

Linkedin...................................................................................................5

Social Resume Database..........................................................................6

What relationship exist among the entities?......................................................7

Facebook – Linkedin................................................................................7

Social Resume Database..........................................................................8

Key constraints.....................................................................................................9

Facebook – Linkedin.................................................................................9

Social Resume Database..........................................................................10

Referential integrity constraints..........................................................................11

Facebook..................................................................................................11

Linkedin....................................................................................................12

Social Resume Database......................................................................... 13

Design – Conceptual Design (Er Diagrams).....................................................................14

Design – Logical Design...................................................................................................16

Facebook.................................................................................................16

Linkedin...................................................................................................18

Social Resume Database..........................................................................20

Implementation – Physical Model..................................................................................21

Facebook.................................................................................................21

Linkedin...................................................................................................25

Social Resume Database.........................................................................28

Populate Database – Insert Values................................................................................34

Create Trigger................................................................................................................42

Insert, Delete, Update Commands................................................................................45

Select Statement...........................................................................................................47

ANALYSIS

1. **Write a brief explanation using your own words about these applications in terms of their scope.**

**FACEBOOK:** Facebook, **4 Şubat 2004** tarihinde **Harvard Üniversitesi**’nde okuyan **Mark Zuckerberg** isimli öğrenci tarafından “The Facebook” adı ile kuruldu. Facebook ilk zamanlar Harvard Üniversitesinde öğrenim gören öğrencilerin kendi aralarındaki iletişim sağlaması amacıyla kurgulanmış ücretsiz bir uygulamaydı. Fakat kısa süre içerisinde kullanıcılarının beğenisini toplamış ve Harvard Üniversitesi dışında da popüler hale gelmiştir. **11 Eylül 2006**’da açılmış olan Facebook, 31 Ağustos 2013 tarihinde yapılmış istatistiklere göre dünyada en çok ziyaret edilen 2.site olarak belirlenmiştir. Siteyi en çok kullanan ve ziyaret eden ülkeler ise ABD, Mısır, Panama, Norveç, Türkiye ve Avustralya olarak değişim gösteriyor. Peki Facebook nedir? **Kullanımı oldukça da basit** olan Facebook mobil cihazlar, bilgisayarlar ve tabletler tarafından desteklenmektedir. Mesajlaşma, fotoğraf ekleme, video paylaşma gibi özellikleri olan Facebook’un en güzel yönlerinden biri de etkinlik kısmıdır. Facebook sayesinde tüm arkadaş çevreniz ve ailenizle iletişim halinde olmanız mümkün.

**LINKEDIN:** Linkedin çalışanlar, iş arayanlar ve işverenlerin profesyonel anlamda kullanımı amacıyla tasarlanmış bir sosyal ağ sitesidir. Aynı zamanda meslektaş gruplar ile kolay yoldan iletişime geçebilme ve rahatlıkla arkadaş edinebilme platformudur. LınkedIn sayesinde iş ilanlarına kolayca ve en profesyonel şekilde ulaşılıp başvuru yapılabiliyor. Yapılan sunum, dosya ve birçok kitapların rahatlıkla paylaşılabilmesi, yazılan blogların kolay yoldan reklamını gerçekleştirmek de yine LinkedIn sayesinde mümkün. Aynı zamanda eklenen cv bölümü ile insanların yeteneklerini, başarılarını, ödüllerini ve benzeri pek çok şeyi paylaşarak profesyonel ağlarındaki insanlara kolayca ulasmayı vaad ediyor. Sitede ağınıza ne kadar insan eklerseniz fırsatları o kadar çabuk yakalayabiliyor ve daha fazla insana ulaşarak şansınızı artırabiliyorsunuz. LinkedIn bunca özelliği sayesinde hızlıca büyüyen bir ağ konumunda. Bu kadar hızlı büyümesinin en önemli nedeni dünya genelindeki profesyonellere gerçek ve etkili bir iletişim ağı kurma, network geliştirme imkanı vermesidir.

1. **Write an analysis report for each of the database model/application:**
2. **What is the aim of each application?**

**FACEBOOK:** Facebook insanların sosyal geçmişlerini tutan ve arkadaşlarıyla bir şeyler paylaşmasını sağlayan sosyal paylaşım sitesidir. İnsanlar facebookta hobilerini, ilgi alanlarını, favorilerini, eğitim ve iş bilgilerini, aile yakınlarını ve birçok gönderiyi paylaşabilirler. Gruplara katılıp, ilgi alanlarına göre oluşan sayfaları beğenebilirler. Etkinlikler düzenleyip, etkinliklere katılabilirler.

**LINKEDIN:** Linkedin ise kullanıcıların kariyer anlamda özgeçmişlerinin tutulduğu bir sitedir. Kullanıcılar profillerinde yeteneklerini, patentlerini, katıldıkları kurs ve projeleri, aldıkları ödülleri, sınav sonuçlarını, bildikleri dilleri ve katıldıkları organizasyonları paylaşabilirler. Ayrıntılı bir CV oluşturup iş deneyimlerini, eğitim bilgilerini, lisans ve sertifikalarını iş verenlere ve diğer kullanıcılara sunabilirler.

**SOCIAL RESUME DATABASE:** Social Resume Database ise bizim geliştirdiğimiz, kullanıcıların sosyal geçmişlerini ve cvlerini bir arada görebilmemizi sağlayan veri tabanı modelidir.

1. **What are the main entities of them?**

**Facebook:**

USERS, PROFILE, FRIEND, FEED, BOOKMARK, STATUS, ACTIVITIES, NOTIFICATION, ADDRESS, SCHOOL, JOB, PRIVACY, NICKNAME, BLOG, LANGUAGE, NOTIFICATION, COMMENTS, DEPENDENTS

**Linkedin:**

ADDRESS, MARITAL STATUS, ORGANIZATIONS, MEMBERS, GROUPS, CV, PROFILE

**Social Resume Database:**

ORGANIZATIONS, MEMBERS, PROFILE, ADDRESS, NICKNAME, FRIEND, GROUPS, PAGES, EVENTS, CV, BOOKMARK, FEED, STATUS, BLOG, NOTIFICATION, COMMENTS, DEPENDENTS

1. **What are the characteristics of each entity?**

**FACEBOOK:**

* **PRIVACY:** Kullanıcıların gizlilik bilgilerini tutan tablodur. Primary key: privacy\_id
* **USERS:** Hesap bilgilerini barındıran tablodur. Primary key: user\_id. Varchar tipinde password, e\_mail, fname, minit, lname, profile\_picture bilgisini tutar. Timestamp tipinde created\_at zamanını tutar.
* **NICKNAME:** Kullanıcıların takma adlarının tutulduğu tablodur. Primary key: nickname\_id. Foreign key: privacy
* **FRIEND:** Kullanıcıların arkadaş bilgisini ve subscriber olup olmadıkları bilgisini tutan tablodur.
* **PROFILE:** Kullanıcıların telefon numarası, e\_maili, doğumgünü, cinsiyeti, ilişki durumu ve favorilerini tutan tablodur. Primary key: profile\_id
* **BOOKMARK:** Kullanıcıların kaydettiği gönderileri tutan tablodur. Primary key: Bookmark\_id. Url, clicks, rating, favorite bilgilerini de tutar.
* **BCATEGORY:** Bookmark kategorilerini tutan tablodur.
* **FEED:** Kullanıcıların gördüğü tüm akıştır.
* **FCATEGORY:** Feed kategorilerini tutan tablodur.
* **STATUS:** Kullanıcıların paylaştığı status bilgilerini tutar. Primary key: Status\_id
* **ACTIVITIES:** Kullanıcıların katıldığı ya da oluşturduğu grup, page veya event bilgilerini tutan tablodur. Primary\_key: Act\_id ve varchar tipinde act\_name, act\_exp ve act\_type bilgisini, date ve time tipinde event zamanını tutar.
* **BLOG:** Kullanıcının oluşturduğu blog bilgisini tutar.
* **LANGUAGE:** Kullanıcının bildiği dilleri tutan tablodur. User\_id ve language birlikte primary keydir.
* **NOTIFICATION:** Kullanıcının aldığı bildirimleri tutan tablodur. Mesajı ve bildirimin ne zaman oluştuğu bilgisini tutar.
* **MESSAGE:** Userların mesajlaşma bilgilerini tutan tablodur. Primary key: send\_user\_id ve to\_user\_id
* **COMMENTS:** Statuslere yapılan yorumların tutulduğu tablodur. Primary key: comment\_id.
* **DEPENDENTS:** Kullanıcının aile bilgisini tutar. Dep\_id Primary key. Profile\_id foreign keydir.
* **JOBS:** Kullanıcının yaptığı işlerin bilgisini tutar.
* **SCHOOL:** Kullanıcının eğitim bilgilerini tutar.
* **ADDRESS:** Kullanıcının adres bilgisini tutar.
* **CITY:** Addreslerin şehir bilsinini tutar.
* **COUNTRY:** Ülke bilgisini tutar
* **LIKES:** Statuslere yapılan likeların hangi friendler tarafından yapıldığını tutan tablodur.

**LINKEDIN**

* **ADDRES:** Kullanıcıların adres bilgilerini tutan tablodur. Primary key: address\_id
* **MARITAL\_STATUS:** Kullanıcıların ilişki durumunu tutan tablodur.
* **ORGANIZATIONS:** Kullanıcıların çalıştığı organizasyonların bilgisini tutan tablodur.
* **MEMBERS:** Kullanıcıların tüm bilgileri bu tabloda tutulur. Primary\_key: member\_id. Doğum günü, e mail, şifre, isim, cinsiyet ve katılma tarihini tutar.
* **GROUPS:** Kullanıcıların oluşturduğu grupların bilgisini tutar. Primary key: Group\_id. Grubun başlama bitme ve en son aktif olma tarihini ve hangi member tarafından kurulduğunu görebiliriz.
* **CV:** Kullanıcıların cvlerini tutan tablodur. Primary key: Cv\_id. Cv’nin yaratılma ve güncellenme zamanlarının bilgisini verir.
* **CV\_SECTION:** Kullanıcıların CV başlıklarının ve içeriğinin tutulduğu tablodur. Primary key: Cv\_section\_id ve cv\_id kullanıcının cvsine referans eder. Cv başlıkları enum tipindedir. Work Experience, Education, Licance and Sertificated ve Volunteer Experience olabilir. Açıklama, yaratılma ve güncelleme zamanlarının bilgisini de görebiliriz.
* **PROFILE:** Kullanıcıların profilidir.
* **PROFILE\_SECTION:** Kullanıcıların profil başlıklarını ve içeriklerini tutar. Kullanıcılar profillerine Skills, patents, course, project, honors and awards, exam scores, language ve organizations bilgisi ekleyebilir.
* **CONNECTS:** Kullanıcıların bağlantıda olduğu memberların listesini ve bağlantı tarihlerini tutar.
* **RECOMMENDATION:** Kullanıcıların tavsiye ettiği memberların listesini tutar.
* **MEMBER GROUPS:** Kullanıcıların katıldığı grupların listesini tutar.
* **COMPANY:** Şirketlerin bilgilerini tutar.
* **JOB\_OFFER:** Şirketlerin iş ilanlarını tutar.
* **FOLLOW:** Memberların takip ettiği şirketlerin bilgisini tutar.

**SOCIAL RESUME DATABASE:**

* **COUNTRY:** Ülke bilgisinin tutan tablodur.
* **CITY:** Şehir bilgisini tutan tablodur.
* **ORGANIZATIONS:** Kullanıcıların çalıştığı organizasyonları tutar.
* **MEMBERS:** Kullanıcıların hesap bilgilerini tutan tablodur.
* **PROFILE:** Kullanıcıların hobileri, favorileri, ilgi alanları ve hakkındaki diğer bilgileri tutan tablodur.
* **ADDRESS:** Kullanıcıların adres bilgilerini tutan tablodur.
* **NICKNAME:** Kullanıcıların nickname bilgisini tutar.
* **FRIEND:** Kullanıcıların arkadaşlık ilişkisini tutan tablodur.
* **GROUPS:** Kullanıcıların oluşturduğu grupların bilgisini tutar.
* **EVENTS:** Kullanıcıların oluşturduğu eventlerin adres tarih açıklama bilgilerini tutar.
* **PAGES:** Kullanıcıların oluşturduğu sayfaların bilgisini tutar.
* **MEMBER\_GROUPS:** Memberların katıldığı grupların bilgisini tutar.
* **MEMBER\_PAGES:** Memberların beğendiği sayfaların bilgisini tutar.
* **MEMBER\_EVENTS:** Memberların katıldığı etkinliklerin bilgisini tutar.
* **CV:** Memberların cv bilgisini tutar.
* **CV\_SECTION:** Memberların CV başlıklarını ve içeriklerini tutar.
* **PROFILE\_SECTION:** Memberların profil başlıklarını ve içeriklerini tutar.
* **CONNECTS:** Birbiriyle bağlantıda olan memberların listesini ve bağlantı tarihlerini tutar.
* **RECOMMENDATION:** Birbirini tavsiye eden memberların listesini ve diğer detayları tutar.
* **BCATEGORY:** Bookmark kategorilerini tutan tablodur.
* **BOOKMARK:** Kullanıcının kaydettiği gönderileri tutan tablodur.
* **FCATEGORY:** Feed kategorilerini tutan tablodur.
* **FEED:** Kullanıcıların gördüğü akışı tutan tablodur.
* **STATUS:** Kullanıcıların paylaştığı statusleri tutan tablodur.
* **BLOG:** Kullanıcının oluşturduğu blog bilgisini tutar.
* **NOTIFICATION:** Kullanıcıya gelen bildirimlerin listesini tutar.
* **MESSAGE:** Kullanıcıların birbirleriyle mesajlaşma bilgisini tutar.
* **COMMENTS:** Statuslere gelen yorumların bilgisini tutar.
* **DEPENDENTS:** Memberların yakınlarının bilgisini tutar.
* **LIKES:** Statuslere gelen likeların tutulduğu tablodur.
* **COMPANY:** Şirketlerin bilgilerini tutar.
* **JOB\_OFFER:** Şirketlerin iş ilanlarını tutar.
* **FOLLOW:** Memberların takip ettiği şirketlerin bilgisini tutar.

1. **What relationship exist among the entities?**

**FACEBOOK:**

* **USERS** – have1 – **PROFILE:** 1-1 Relationship
* **PROFILE** – dependents\_of – **DEPENDENTS:** 1-N Relationship
* **PROFILE** – work – **JOBS:** 1-N Relationship
* **PROFILE** – edu – **SCHOOL:** 1-N Relationship
* **PROFILE** – live – **ADDRESS:** 1-N Relationship
* **CITY** – in1 – **ADDRESS:** 1-N Relationship
* **COUNTRY** – in2 – **CITY:** 1-N Relationship
* **USERS** – join – **ACTIVITIES:** 1-N Relationship
* **USERS** – create – **ACTIVITIES:** 1-N Relationship
* **ACTIVITIES** – overlapping act\_type – **EVENT, PAGE, GROUP**
* **EVENT** – in3 – **ADDRESS:** 1-1 Relationship
* **USERS** – share – **STATUS:** 1-N Relationship
* **USERS** – friendship – **FRIEND:** 1-N Relationship
* **FRIEND** – make – **COMMENT:** 1-N Relationship
* **USERS** – make – **COMMENT:** 1-N Relationship
* **STATUS** – take – **COMMENT:** 1-N Relationship
* **FRIEND** – like – **STATUS:** M-N Relationship
* **USERS** – have2 – **NICKNAME:** 1-1 Relationship
* **USERS** – write – **BLOG:** 1-1 Relationship
* **USERS** – record – **BOOKMARK:** 1-N Relationship
* **BOOKMARK** – have3 – **BCATEGORY:** 1-1 Relationship
* **USERS** – see – **FEED:** 1-N Relationship
* **FEED** – have4 – **FCATEGORY:** 1-1 Relationship
* **USERS** – take2 – **NOTIFICATION:** 1-N Relationship
* **USERS** – message – **USERS:** M-N Relationship
* **PRIVACY** – union – **FEED, BOOKMARK, NICKNAME, FRIEND, PROFILE, STATUS, ACTIVITIES**

**LINKEDIN:**

* **MEMBERS** – live – **ADDRESS:** 1-1 Relationship
* **MEMBERS** – works – **ORGANIZATIONS:** 1-1 Relationship
* **MEMBERS** – current – **MARITAL\_STATUS:** 1-1 Relationship
* **MEMBERS** – create – **GROUPS:** 1-N Relationship
* **MEMBERS** – join – **GROUPS:** M-N Relationship
* **MEMBERS** – generate – **CV:** 1-1 Relationship
* **CV** – has1 – **CV\_SECTION:** 1-N Relationship
* **MEMBERS** – have – **PROFILE:** 1-1 Relationship
* **PROFILE** – has2 – **PROFILE\_SECTION:** 1-N Relationship
* **MEMBERS** – follows – **MEMBERS:** M-N Relationship
* **MEMBERS** – connects – **MEMBERS:** M-N Relationship
* **MEMBERS** – recommendation – **MEMBERS:** M-N Relationship

**SOCIAL RESUME DATABASE**:

* **MEMBER** – live – **ADDRESS:** 1-N Relationship
* **CITY** – in1 – **ADDRESS:** 1-N Relationship
* **COUNTRY** – in2 – **CITY:** 1-N Relationship
* **MEMBERS** – works – **ORGANIZATIONS:** 1-1 Relationship
* **MEMBERS** – have – **PROFILE:** 1-1 Relationship
* **PROFILE** – has1 – **PROFILE\_SECTION:** 1-N Relationship
* **MEMBERS** – have2 – **NICKNAME:** 1-1 Relationship
* **MEMBERS** – friendship – **FRIEND:** 1-N Relationship
* **MEMBERS** – create1 – **GROUPS:** 1-N Relationship
* **MEMBERS** – join 1– **GROUPS:** M-N Relationship
* **MEMBERS** – create2 – **EVENTS:** 1-N Relationship
* **MEMBERS** – join2 – **EVENTS:** M-N Relationship
* **MEMBERS** – create3 – **PAGES:** 1-N Relationship
* **MEMBERS** – join3 – **PAGES:** M-N Relationship
* **MEMBERS** – generate – **CV:** 1-1 Relationship
* **CV** – has2 – **CV\_SECTION:** 1-N Relationship
* **MEMBERS** – follows – **MEMBERS:** M-N Relationship
* **MEMBERS** – connects – **MEMBERS:** M-N Relationship
* **MEMBERS** – recommendation – **MEMBERS:** M-N Relationship
* **MEMBERS** – record – **BOOKMARK:** 1-N Relationship
* **BOOKMARK** – have3 – **BCATEGORY:** 1-1 Relationship
* **MEMBERS** – see – **FEED:** 1-N Relationship
* **FEED** – have4 – **FCATEGORY:** 1-1 Relationship
* **MEMBERS** – share – **STATUS:** 1-N Relationship
* **MEMBERS** – write – **BLOG:** 1-1 Relationship
* **MEMBERS** – take1 – **NOTIFICATION:** 1-N Relationship
* **MEMBERS** – message – **MEMBERS:** M-N Relationship
* **FRIEND** – make – **COMMENT:** 1-N Relationship
* **MEMBERS** – make – **COMMENT:** 1-N Relationship
* **STATUS** – take2 – **COMMENT:** 1-N Relationship
* **MEMBERS** – dependents\_of – **DEPENDENTS:** 1-N Relationship
* **FRIEND** – like – **STATUS:** M-N Relationship

1. **What are the constraints related to entities, their characteristics and the relationship among them?**

**KEY CONSTRAINTS:**

**FACEBOOK KEY CONSTRAINTS:**

* PRIVACY (privacy\_id)
* USERS (user\_id)
* NICKNAME (nickname\_id)
* FRIEND (friend\_user\_id, user\_id)
* PROFILE (profile\_id, user\_id)
* BCATEGORY (bookmark\_category\_id)
* BOOKMARK (bookmark\_id)
* FCATEGORY (feed\_category\_id)
* FEED (feed\_id)
* STATUS (status\_id)
* ACTIVITIES (act\_id)
* BLOG (blog\_id, user\_id)
* LANGUAGE (language, user\_id)
* NOTIFICATION (notification\_id)
* MESSAGE (send\_user\_id, to\_user\_id)
* COMMENTS (comment\_id)
* DEPENDENTS (dep\_id)
* JOBS (job\_id)
* SCHOOL (school\_id)
* ADDRESS (address\_id)
* CITY (city\_id)
* COUNTRY (country\_id)
* LIKES (friend\_user\_id, status\_id)

**LINKEDIN KEY CONSTRAINTS**

* ADDRESS (address\_id)
* MARITAL\_STATUS (marital\_status\_id)
* ORGANIZATIONS (org\_id)
* MEMBERS (member\_id)
* GROUPS (group\_id)
* CV (cv\_id, member\_id)
* CV\_SECTION (cv\_section\_id)
* PROFILE (profile\_id, member\_id)
* PROFILE\_SECTION (profile\_section\_id)
* CONNECTS (member\_id, connected\_member\_id)
* RECOMMENDATION (member\_id, recommended\_id)
* MEMBER\_GROUPS (member\_id, group\_id)
* COMPANY (company\_id)

**SOCIAL RESUME DATABASE KEY CONSTRAINTS**

* CITY (city\_id)
* COUNTRY (country\_id)
* ADDRESS (address\_id)
* ORGANIZATIONS (org\_id)
* MEMBERS (member\_id)
* PROFILE (profile\_id, member\_id)
* PROFILE\_SECTION (profile\_section\_id)
* NICKNAME (nickname\_id, member\_id)
* FRIEND (member\_id, friend\_member\_id)
* GROUPS (group\_id)
* MEMBER\_GROUPS (member\_id, group\_id)
* PAGES (page\_id)
* MEMBER\_PAGES (member\_id, page\_id)
* EVENTS (event\_id)
* MEMBER\_EVENTS (member\_id, event\_id)
* CV (cv\_id, member\_id)
* CV\_SECTION (cv\_section\_id)
* CONNECTS (member\_id, connected\_member\_id)
* FOLLOWS (follower\_id, followed\_id)
* RECOMMENDATION (member\_id, recommended\_id)
* BCATEGORY (bookmark\_category\_id)
* BOOKMARK (bookmark\_id)
* FCATEGORY (feed\_category\_id)
* FEED (feed\_id)
* STATUS (status\_id)
* BLOG (blog\_id, member\_id)
* NOTIFICATION (notification\_id)
* MESSAGE (send\_member\_id, to\_member\_id)
* COMMENTS (comment\_id)
* DEPENDENTS (dep\_id)
* LIKES (friend\_member\_id, status\_id)

Yukarıdakiler primary key oldukları için not null ve unique’dir.

**REFERENTIAL INTEGRITY CONSTRAINTS**

**FACEBOOK REFERENTIAL INTEGRITY CONSTRAINTS**

**FOREIGN KEY--------------------- REFERENCES---------------------PRIMARY KEY**

* NICKNAME.user\_id 🡪 USERS.user\_id
* NICKNAME.privacy 🡪 PRIVACY.privacy\_id
* FRIEND.friend\_user\_id 🡪 USERS.user\_id
* FRIEND.user\_id 🡪 USERS.user\_id
* PROFILE.user\_id 🡪 USERS.user\_id
* PROFILE.privacy 🡪 PRIVACY.privacy\_id
* BOOKMARK.bookmark\_category\_id 🡪 BCATEGORY.id
* BOOKMARK.user\_id 🡪 USERS.user\_id
* BOOKMARK.privacy 🡪 PRIVACY.privacy\_id
* FEED.feed\_category\_id 🡪 FCATEGORY.id
* FEED.user\_id 🡪 USERS.user\_id
* FEED.privacy 🡪 FEED.privacy\_id
* STATUS.privacy 🡪 PRIVACY.privacy\_id
* STATUS.user\_id 🡪 USERS.user\_id
* ACTIVITIES.privacy 🡪 PRIVACY.privacy\_id
* ACTIVITIES.create\_user\_id 🡪 USERS.user\_id
* ACTIVITIES.join\_user\_id 🡪 USERS.user\_id
* ACTIVITIES.event\_address\_id 🡪 ADDRESS.addres\_id
* BLOG.user\_id 🡪 USERS.user\_id
* LANGUAGE.user\_id 🡪 USERS.user\_id
* NOTIFICATION.user\_id 🡪 USERS.user\_id
* MESSAGE.send\_user\_id 🡪 USERS.user\_id
* MESSAGE.to\_user\_id 🡪 USERS.user\_id
* COMMENTS.status\_id 🡪 STATUS.status\_id
* COMMENTS.friend\_user\_id 🡪 FRIEND.friend\_user\_id
* COMMENTS.user\_id 🡪 USERS.user\_id
* DEPENDENTS.profile\_id 🡪 PROFILE.profile\_id
* JOBS.profile\_id 🡪 PROFILE.profile\_id
* SCHOOL.profile\_id 🡪 PROFILE.profile\_id
* ADDRES.city\_id 🡪 CITY.city\_id
* ADDRESS.profile\_id 🡪 PROFILE.profile\_id
* CITY.country\_id 🡪 COUNTRY.country\_id
* LIKES. friend\_user\_id 🡪 FRIEND.friend\_user\_id
* LIKES. status\_id 🡪 STATUS.status\_id

**LINKEDIN REFERENTIAL INTEGRITY CONSTRAINTS**

**FOREIGN KEY--------------------- REFERENCES---------------------PRIMARY KEY**

* MEMBERS.address\_id 🡪 ADDRES.address\_id
* MEMBERS.current\_organization\_id 🡪 ORGANIZATIONS.org\_id
* MEMBERS.marital\_status\_id 🡪 MARITAL\_STATUS.id
* GROUPS.create\_member\_id 🡪 MEMBERS.member\_id
* CV.member\_id 🡪 MEMBERS.member\_id
* CV\_SECTION.cv\_id 🡪 CV.cv\_id
* PROFILE.member\_id 🡪 MEMBERS.member\_id
* PROFILE\_SECTION.profile\_id 🡪 PROFILE.profile\_id
* CONNECTS.member\_id 🡪 MEMBERS.member\_id
* CONNECTS.connected\_member\_id 🡪 MEMBERS.member\_id
* FOLLOWS.follower\_id 🡪 MEMBERS.member\_id
* FOLLOWS.followed\_id 🡪 MEMBERS.member\_id
* RECOMMENDATION.member\_id 🡪 MEMBERS.member\_id
* RECOMMENDATION.rec\_id 🡪 MEMBERS.member\_id
* MEMBER\_GROUPS.member\_id 🡪 MEMBERS.member\_id
* MEMBER\_GROUPS.group\_id 🡪 GROUPS.group\_id

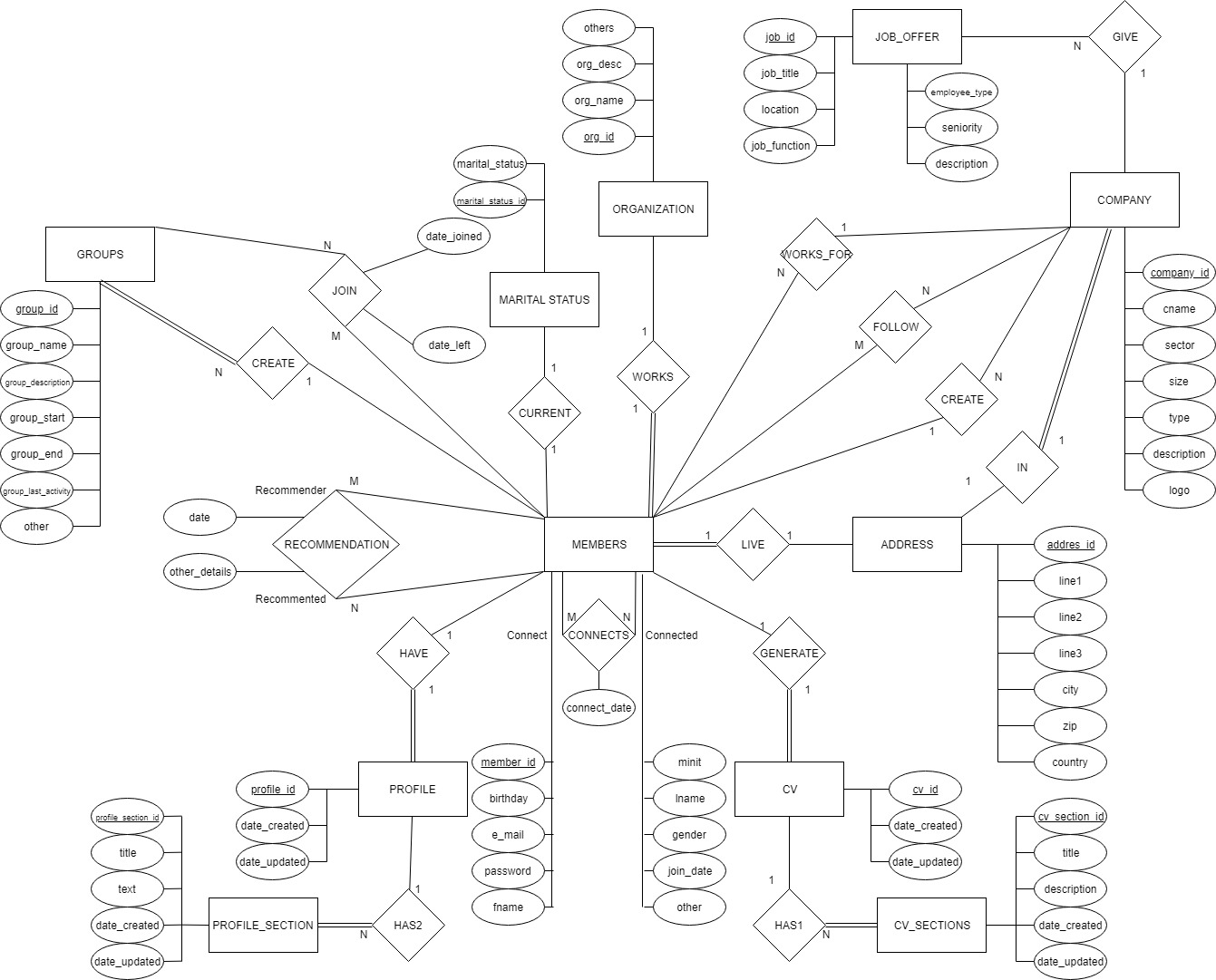
**SOCIAL RESUME DATABASE REFERENTIAL INTEGRITY CONSTRAINTS**

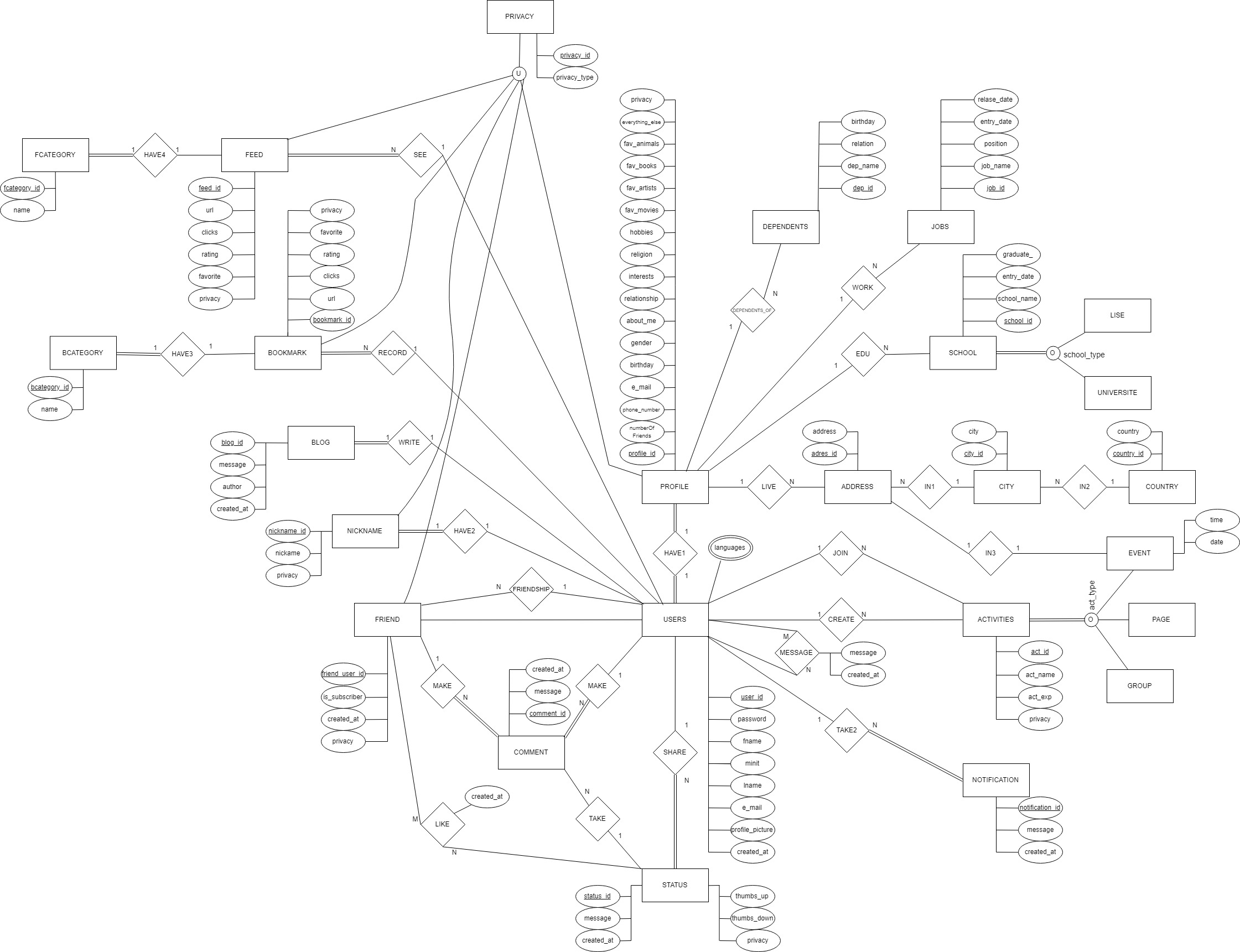
**FOREIGN KEY--------------------- REFERENCES---------------------PRIMARY KEY**

* CITY.country\_id 🡪 COUNTRY.country\_id
* MEMBERS.current\_organization\_id 🡪 ORGANIZATIONS.org\_id
* PROFILE.member\_id 🡪 MEMBERS.member\_id
* PROFILE\_SECTION.profile\_id 🡪 PROFILE.profile\_id
* ADDRES.city\_id 🡪 CITY.city\_id
* ADDRESS.member\_id 🡪 MEMBERS.member\_id
* NICKNAME.member\_id 🡪 MEMBERS.member\_id
* FRIEND.friend\_member\_id 🡪 MEMBERS.member\_id
* FRIEND.member\_id 🡪 MEMBERS.member\_id
* GROUPS.create\_member\_id 🡪 MEMBERS.member\_id
* EVENTS.event\_addres\_id 🡪 ADDRES.address\_id
* EVENTS.create\_member\_id 🡪 MEMBERS.member\_id
* PAGES.create\_member\_id 🡪 MEMBERS.member\_id
* CV.member\_id 🡪 MEMBERS.member\_id
* CV\_SECTION.cv\_id 🡪 CV.cv\_id
* CONNECTS.member\_id 🡪 MEMBERS.member\_id
* CONNECTS.connected\_member\_id 🡪 MEMBERS.member\_id
* FOLLOWS.follower\_id 🡪 MEMBERS.member\_id
* FOLLOWS.followed\_id 🡪 MEMBERS.member\_id
* RECOMMENDATION.member\_id 🡪 MEMBERS.member\_id
* RECOMMENDATION.rec\_id 🡪 MEMBERS.member\_id
* BOOKMARK.bookmark\_category\_id 🡪 BCATEGORY.id
* BOOKMARK.member\_id 🡪 MEMBERS.member\_id
* FEED.feed\_category\_id 🡪 FCATEGORY.id
* FEED.member\_id 🡪 MEMBERS.member\_id
* STATUS.member\_id 🡪 MEMBERS.member\_id
* BLOG.member\_id 🡪 MEMBERS.member\_id
* NOTIFICATION.member\_id 🡪 MEMBERS.member\_id
* MESSAGE.send\_member\_id 🡪 MEMBERS.member\_id MESSAGE.to\_member\_id 🡪 MEMBERS.member\_id
* COMMENTS.status\_id 🡪 STATUS.status\_id
* COMMENTS.friend\_member\_id 🡪 FRIEND.friend\_member\_id
* COMMENTS.member\_id 🡪 MEMBERS.member\_id
* DEPENDENTS.member\_id 🡪 MEMBERS.member\_id
* LIKES. friend\_member\_id 🡪 FRIEND.friend\_member\_id
* LIKES. status\_id 🡪 STATUS.status\_id
* MEMBER\_GROUPS.member\_id 🡪 MEMBERS.member\_id
* MEMBER\_GROUPS.group\_id 🡪 GROUPS.group\_id
* MEMBER\_PAGES.member\_id 🡪 MEMBERS.member\_id
* MEMBER\_PAGES.page\_id 🡪 PAGES.page\_id
* MEMBER\_EVENTS.member\_id 🡪 MEMBERS.member\_id
* MEMBER\_EVENTS.event\_id 🡪 EVENTS.event\_id

DESIGN – CONCEPTUAL DESIGN

1. **Create an EER diagram for each of the application. The output of this step is 2 EER diagrams, one for Facebook and the other one for LinkedIn.**





DESIGN – LOGICAL MODEL

1. **Convert each EER diagram into relational model using the methodology that will be introduced in your course.**

**FACEBOOK:**

**1st Itteration - Step 9:** MAPPING OF UNION TYPES

**PRIVACY** (privacy\_id, privacy\_type)

**PROFILE** (profile\_id, numberofFriends, phone\_number, e\_mail, birthday, gender, about\_me, relationship, interest, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, **privacy**)

**BOOKMARK** (bookmark\_id, url, clicks, rating, favorite, **privacy**)

**FEED** (feed\_id, url, clicks, rating, favorite, **privacy**)

**STATUS** (status\_id, message, created\_at, thumbs\_up, thumbs\_down, **privacy**)

**ACTIVITIES** (act\_id, act\_name, act\_exp, **privacy**)

**NICKNAME** (nickname\_id, nickname, **privacy**)

**FRIEND** (friend\_user\_id, is\_subscriber, created\_at, **privacy**)

**2nd Itteration - Step 1:** MAPPING OF REGULAR ENTITY TYPES

**USERS** (user\_id, password, fname, minit, lname, e\_mail, profile\_picture, created\_at)

**PROFILE** (profile\_id, numberofFriends, phone\_number, e\_mail, birthday, gender, about\_me, relationship, interest, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, privacy)

**BOOKMARK** (bookmark\_id, url, clicks, rating, favorite, privacy)

**FEED** (feed\_id, url, clicks, rating, favorite, privacy)

**STATUS** (status\_id, message, created\_at, thumbs\_up, thumbs\_down, privacy)

**ACTIVITIES** (act\_id, act\_name, act\_exp, privacy)

**NICKNAME** (nickname\_id, nickname, privacy)

**FRIEND** (friend\_user\_id, is\_subscriber, created\_at, privacy)

**DEPENDENTS** (dep\_id, dep\_name, relation, birthday)

**JOBS** (job\_id, job\_name, position, entry\_date, relase\_date)

**SCHOOL** (school\_id, school\_name, entry\_date, graduate)

**ADDRES** (address\_id, address)

**CITY** (city\_id, city)

**COUNTRY** (country\_id, country)

**BCATEGORY** (bookmark\_category\_id, bname)

**FCATEGORY** (feed\_category\_id, fname)

**BLOG** (blog\_id, message, author, created\_at)

**COMMENT** (comment\_id, message, created\_at)

**NOTIFICATION**(notification\_id, message, created\_at)

**Step 3:** MAPPING OF BINARY 1 – 1 RELATION

have1 relationship:

**PROFILE** (profile\_id, **user\_id**, numberofFriends, phone\_number, e\_mail, birthday, gender, about\_me, relationship, interest, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, privacy)

have2 relationship:

**NICKNAME** (nickname\_id, **user\_id**, nickname, privacy)

write relationship:

**BLOG** (blog\_id, **user\_id**, message, author, created\_at)

have3 relationship:

**BOOKMARK** (bookmark\_id, url, clicks, rating, favorite, privacy, **bookmark\_category\_id**)

have4 relationship:

**FEED** (feed\_id, url, clicks, rating, favorite, privacy, **feed\_category\_id**)

**Step 4:** MAPPING OF BINARY 1 – N RELATION

see relationship:

**FEED** (feed\_id, url, clicks, rating, favorite, privacy, feed\_category\_id, **user\_id**)

record relationship:

**BOOKMARK** (bookmark\_id, url, clicks, rating, favorite, privacy, bookmark\_category\_id, **user\_id**)

friendship relationship:

**FRIEND** (friend\_user\_id, is\_subscriber, created\_at, privacy, **user\_id**)

make relationship:

**COMMENT** (comment\_id, message, created\_at, **user\_id** )

make relationship:

**COMMENT** (comment\_id, message, created\_at, user\_id, **friend\_user\_id** )

take relationship:

**COMMENT** (comment\_id, message, created\_at, user\_id, friend\_user\_id, **status\_id** )

share relationship:

**STATUS** (status\_id, message, created\_at, thumbs\_up, thumbs\_down, privacy, **user\_id**)

take2 relationship:

**NOTIFICATION**(notification\_id, message, created\_at, **user\_id**)

create relationship:

**ACTIVITIES** (act\_id, act\_name, act\_exp, privacy, **create\_user\_id**)

join relationship:

**ACTIVITIES** (act\_id, act\_name, act\_exp, privacy, create\_user\_id, **join\_user\_id**)

live relationship:

**ADDRES** (address\_id, address, **profile\_id**)

in1 relationship:

**ADDRES** (address\_id, address, profile\_id, **city\_id**)

in2 relationship:

**CITY** (city\_id, city, **country\_id**)

edu relationship:

**SCHOOL** (school\_id, school\_name, entry\_date, graduate, **profile\_id**)

work relationship:

**JOBS** (job\_id, job\_name, position, entry\_date, relase\_date, **profile\_id**)

dependents\_of relationship:

**DEPENDENTS** (dep\_id, dep\_name, relation, birthday, **profile\_id**)

**Step 5:** MAPPING OF BINARY M – N RELATION

**MESSAGE** (send\_user\_id, to\_user\_id, message, created\_at)

**LIKES** (friend\_user\_id, status\_id, created\_at)

**Step 8:** MAPPING FOR SPECALIZATION OR GENERALIZATION

8C)

**ACTIVITIES** (act\_id, act\_name, act\_exp, privacy, create\_user\_id, join\_user\_id, **act\_type,** **event\_date, event\_time**)

**SCHOOL** (school\_id, school\_name, entry\_date, graduate, profile\_id, **school\_type**)

**3rd Itteration - Step 3:** MAPPING OF BINARY 1 – 1 RELATION

**ACTIVITIES** (act\_id, act\_name, act\_exp, privacy, create\_user\_id, join\_user\_id, act\_type, event\_date, event\_time, **event\_addres\_id**)

**LINKEDIN:**

**Step 1:** MAPPING OF REGULAR ENTITY TYPES

**MEMBERS** (member\_id, birthday, e\_mail, password, fname, minit, lname, gender, join\_date, other)

**PROFILE** (profile\_id, date\_created, date\_updated)

**PROFILE\_SEC** (profile\_section\_id, title, text, date\_created, date\_updated)

**CV** (cv\_id, date\_created, date\_updated)

**CV\_SEC** (cv\_section\_id, title, description, date\_created, date\_updated)

**ADDRESS** (addres\_id, line1, line2, line3, city, zip, country)

**ORG** (org\_id, org\_name, org\_desc, others)

**MARITAL\_ST** (marital\_status\_id, marital\_status)

**GROUPS** (group\_id, group\_name, group\_description, group\_start, group\_end, group\_last\_activity, other)

**COMPANY** (company\_id, cname, sector, size, type, description, logo)

**JOB\_OFFER** (job\_id, job\_title, location, job\_function, employee\_type, seniority, description)

**Step 3:** MAPPING OF BINARY 1 – 1 RELATION

live relationship:

**MEMBERS** (member\_id, birthday, e\_mail, password, fname, minit, lname, gender, join\_date, other, **addres\_id**)

works relationship:

**MEMBERS** (member\_id, birthday, e\_mail, password, fname, minit, lname, gender, join\_date, other, addres\_id, **current\_organization\_id**)

current relationship:

**MEMBERS** (member\_id, birthday, e\_mail, password, fname, minit, lname, gender, join\_date, other, addres\_id, current\_organization\_id, **marital\_status\_id**)

have relationship:

**PROFILE** (profile\_id, **member\_id**, date\_created, date\_updated)

generate relationship:

**CV** (cv\_id, **member\_id**, date\_created, date\_updated)

in relationship:

**COMPANY** (company\_id, cname, sector, size, type, description, logo, **addres\_id**)

**Step 4:** MAPPING OF BINARY 1 – N RELATION

create relationship:

**GROUPS** (group\_id, group\_name, group\_description, group\_start, group\_end, group\_last\_activity, other, **create\_member\_id**)

has2 relationship:

**PROFILE\_SEC** (profile\_section\_id, title, text, date\_created, date\_updated, **profile\_id**)

has1 relationship:

**CV\_SEC** (cv\_section\_id, title, description, date\_created, date\_updated, **cv\_id**)

create relationship:

**COMPANY** (company\_id, cname, sector, size, type, description, logo, addres\_id, **boss**)

works\_for relationship:

**MEMBERS** (member\_id, birthday, e\_mail, password, fname, minit, lname, gender, join\_date, other, addres\_id, current\_organization\_id, marital\_status\_id, **work\_comp\_id**)

give relationship:

**JOB\_OFFER** (job\_id, job\_title, location, job\_function, employee\_type, seniority, description, **company\_id**)

**Step 5:** MAPPING OF BINARY M – N RELATION

**FOLLOWS** (member\_id, company\_id)

**RECOMMEND** (member\_id, recommended\_id, date, others)

**CONNECTS** (member\_id, connected\_member\_id, connect\_date)

**MEMBER\_GROUPS** (member\_id, group\_id, date\_joined, date\_left)

1. **Integrate the relational models into the social resume database.**

**SOCIAL RESUME DATABASE:**

**ADDRES** (address\_id, address, member\_id, city\_id)

**BOOKMARK** (bookmark\_id, url, clicks, rating, favorite, bookmark\_category\_id, member\_id)

**BLOG** (blog\_id, message, author, created\_at, member\_id)

**BCATEGORY** (bookmark\_category\_id, bname)

**CITY** (city\_id, city\_name, country\_id)

**CONNECTS** (member\_id, connected\_member\_id, connection\_date)

**COMPANY** (company\_id, cname, sector, size, ctype, cdesc, logo, address\_id, boss)

**COMMENTS** (comment\_id, message, created\_at, status\_id, friend\_member\_id, member\_id)

**COUNTRY** (country\_id, county\_name)

**FOLLOWS** (follower\_id, followed\_id, start\_date, stop\_date)

**FEED** (feed\_id, url, clicks, rating, favorite, privacy, feed\_category\_id, member\_id)

**FCATEGORY** (feed\_category\_id, fname)

**EVENTS** (event\_id, event\_name, event\_description, event\_start, event\_end, event\_address\_id, other, create\_member\_id)

**DEPENDENTS** (dep\_id, dep\_name, relation, birthday, member\_id)

**CV\_SECTION** (cv\_section\_id, title, description\_, date\_created, date\_updated, cv\_id)

**CV** (cv\_id, date\_created, date\_updated, member\_id)

**FRIEND** (member\_id, friend\_member\_id, is\_subscriber, created\_at)

**FOLLOWS** (member\_id, company\_id)

**GROUPS** (group\_id, group\_name, group\_description, group\_start, group\_end, group\_last\_activity, other, create\_member\_id)

**JOB\_OFFER** (job\_id, job\_title, location, job\_function, emplooye\_type, seniority, offer\_desc, created\_at, company\_id)

**LIKES** (friend\_member\_id, status\_id, created\_at)

**MEMBER\_PAGES** (member\_id, page\_id, date\_joined, date\_left)

**MEMBER\_GROUPS** (member\_id, group\_id, date\_joined, date\_left)

**MEMBER\_EVENTS** (member\_id, event\_id)

**MEMBERS** (member\_id, e\_mail, password\_, fname, minit, lname, gender, marital\_status, profile\_picture, current\_organization\_id, join\_date, other)

**MESSAGE** (send\_member\_id, to\_member\_id, message, created\_at)

**NICKNAME** (nickname\_id, nickname, member\_id)

**NOTIFICATION** (notification\_id, message, created\_at, member\_id)

**ORGANIZATIONS** (org\_id, org\_name, org\_description, other)

**PAGES** (page\_id, page\_name, page\_description, other, create\_member\_id)

**PROFILE** (profile\_id, numberOfFriends, phone\_number, birthday, about\_me, interests, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, date\_created, date\_updated, member\_id)

**PROFILE\_SECTION** (profile\_section\_id, title, text\_, date\_created, date\_updated, profile\_id)

**RECOMMENDATION** (member\_id, recommended\_id, start\_date, others\_)

**STATUS** (status\_id, message, created\_at, thumbs\_up, thumbs\_down, member\_id)

IMPLEMENTATION – PHYSICAL MODEL

1. **Write down the appropriate SQL scripts (DDL statements) for creating the database and its relational model.**

**CREATE SCHEMA FACEBOOK;**

USE FACEBOOK;

CREATE TABLE IF NOT EXISTS PRIVACY (

privacy\_id TINYINT(3) NOT NULL DEFAULT '0',

privacy\_type VARCHAR(45),

PRIMARY KEY(privacy\_id)

);

CREATE TABLE IF NOT EXISTS USERS (

user\_id MEDIUMINT(8) NOT NULL DEFAULT '0',

password\_ VARCHAR(45) DEFAULT NULL,

fname VARCHAR(45) DEFAULT NULL,

minit VARCHAR(45) DEFAULT NULL,

lname VARCHAR(45) DEFAULT NULL,

e\_mail VARCHAR(100) DEFAULT NULL,

profile\_picture VARCHAR(255) DEFAULT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (user\_id)

);

CREATE TABLE IF NOT EXISTS NICKNAME (

nickname\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

nickname VARCHAR(45) DEFAULT NULL,

user\_id MEDIUMINT(8) NOT NULL,

privacy TINYINT(3) DEFAULT NULL,

PRIMARY KEY(nickname\_id, user\_id),

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (privacy) REFERENCES PRIVACY(privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FRIEND (

user\_id MEDIUMINT(8) NOT NULL,

friend\_user\_id MEDIUMINT(8) NOT NULL,

is\_subscriber BOOLEAN DEFAULT '0',

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

privacy TINYINT(3) DEFAULT NULL,

PRIMARY KEY(user\_id, friend\_user\_id),

FOREIGN KEY (friend\_user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROFILE\_(

profile\_id MEDIUMINT(8) NOT NULL,

numberOfFriends tinyint(3) NOT NULL DEFAULT '0',

phone\_number VARCHAR(45) DEFAULT NULL,

e\_mail VARCHAR(100)DEFAULT NULL,

birthday DATE DEFAULT NULL,

gender ENUM('Male','Female') DEFAULT NULL,

about\_me VARCHAR(255) DEFAULT NULL,

relationship ENUM('Single','in Relationship','Married','Separated') DEFAULT NULL,

interests VARCHAR(255) DEFAULT NULL,

religion VARCHAR(255) DEFAULT NULL,

hobbies VARCHAR(255) DEFAULT NULL,

fav\_movies VARCHAR(255) DEFAULT NULL,

fav\_artists VARCHAR(255) DEFAULT NULL,

fav\_books VARCHAR(255) DEFAULT NULL,

fav\_animals VARCHAR(255) DEFAULT NULL,

everything\_else VARCHAR(255) DEFAULT NULL,

user\_id MEDIUMINT(8) NOT NULL UNIQUE,

privacy TINYINT(3) DEFAULT NULL,

PRIMARY KEY(profile\_id, user\_id),

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (privacy) REFERENCES PRIVACY(privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS BOOKMARK (

bookmark\_id BIGINT(20) NOT NULL AUTO\_INCREMENT,

url VARCHAR(255) DEFAULT NULL,

clicks SMALLINT(5) NOT NULL,

rating SMALLINT(5) DEFAULT NULL,

favorite BOOLEAN DEFAULT '1',

privacy TINYINT(3) DEFAULT '3',

bookmark\_category\_id SMALLINT(5) DEFAULT NULL,

user\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (bookmark\_id),

KEY bookmark\_ibfk\_3 (bookmark\_category\_id),

FOREIGN KEY (privacy) REFERENCES PRIVACY (privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FCATEGORY (

feed\_category\_id SMALLINT(5) NOT NULL AUTO\_INCREMENT,

fname VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(feed\_category\_id)

);

CREATE TABLE IF NOT EXISTS FEED (

feed\_id BIGINT(20) NOT NULL AUTO\_INCREMENT,

url VARCHAR(255) DEFAULT NULL,

clicks SMALLINT(5) NOT NULL,

rating SMALLINT(5) DEFAULT NULL,

favorite BOOLEAN DEFAULT '1',

privacy TINYINT(3) DEFAULT '3',

feed\_category\_id SMALLINT(5) DEFAULT NULL,

user\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (feed\_id),

FOREIGN KEY (privacy) REFERENCES PRIVACY (privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (feed\_category\_id) REFERENCES FCATEGORY (feed\_category\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS STATUS\_ (

status\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) DEFAULT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

thumbs\_up TINYINT(3) DEFAULT NULL,

thumbs\_down TINYINT(3) DEFAULT NULL,

privacy TINYINT(3) DEFAULT NULL,

user\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (status\_id),

FOREIGN KEY (privacy) REFERENCES PRIVACY(privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS ACTIVITIES (

act\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

act\_name VARCHAR(45) DEFAULT NULL,

act\_exp VARCHAR(255) DEFAULT NULL,

act\_type ENUM('Group','Page','Event') NOT NULL,

event\_date DATE DEFAULT NULL,

event\_time TIME DEFAULT NULL,

event\_address\_id MEDIUMINT(8) DEFAULT NULL,

privacy TINYINT(3) DEFAULT NULL,

join\_user\_id MEDIUMINT(8) DEFAULT NULL,

create\_user\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (act\_id),

KEY activities\_ibfk\_4 (event\_address\_id),

FOREIGN KEY (privacy) REFERENCES PRIVACY(privacy\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (create\_user\_id) REFERENCES USERS(user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (join\_user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS BCATEGORY (

bookmark\_category\_id SMALLINT(5) NOT NULL AUTO\_INCREMENT,

bname VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(bookmark\_category\_id)

);

CREATE TABLE IF NOT EXISTS BLOG (

blog\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) DEFAULT NULL,

author VARCHAR(45) DEFAULT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

user\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(blog\_id, user\_id),

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS LANGUAGE\_ (

language\_ VARCHAR(45),

user\_id MEDIUMINT(8),

PRIMARY KEY(language\_, user\_id),

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS NOTIFICATION (

notification\_id MEDIUMINT (8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

user\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(notification\_id),

FOREIGN KEY (user\_id) REFERENCES USERS (user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MESSAGE(

send\_user\_id MEDIUMINT(8) NOT NULL,

to\_user\_id MEDIUMINT(8) NOT NULL,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (send\_user\_id, to\_user\_id),

FOREIGN KEY (send\_user\_id) REFERENCES USERS(user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (to\_user\_id) REFERENCES USERS(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS COMMENTS (

comment\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

status\_id MEDIUMINT(8) DEFAULT NULL,

friend\_user\_id MEDIUMINT(8) DEFAULT NULL,

user\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (comment\_id),

FOREIGN KEY (status\_id) REFERENCES STATUS\_(status\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (friend\_user\_id) REFERENCES FRIEND (friend\_user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (user\_id) REFERENCES USERS(user\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS DEPENDENTS (

dep\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

dep\_name VARCHAR(45) DEFAULT NULL,

relation VARCHAR(45) DEFAULT NULL,

birthday DATE DEFAULT NULL,

profile\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (dep\_id),

FOREIGN KEY (profile\_id) REFERENCES PROFILE\_(profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS JOBS(

job\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

job\_name VARCHAR(45) DEFAULT NULL,

position\_ VARCHAR(45) DEFAULT NULL,

entry\_date YEAR DEFAULT NULL,

relase\_date YEAR DEFAULT NULL,

profile\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(job\_id),

FOREIGN KEY(profile\_id) REFERENCES PROFILE\_(profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS SCHOOL(

school\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

school\_type ENUM('Lise','Üniversite') NOT NULL,

school\_name VARCHAR(100) DEFAULT NULL,

entry\_date YEAR DEFAULT NULL,

graduate YEAR DEFAULT NULL,

profile\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(school\_id),

FOREIGN KEY(profile\_id) REFERENCES PROFILE\_(profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS ADDRESS(

address\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

address VARCHAR(255) DEFAULT NULL,

profile\_id MEDIUMINT(8) DEFAULT NULL,

city\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY(address\_id),

KEY address\_ibfk\_2 (city\_id),

FOREIGN KEY(profile\_id) REFERENCES PROFILE\_(profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CITY(

city\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

city\_name VARCHAR(45) DEFAULT NULL,

country\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(city\_id),

KEY city\_ibfk\_1 (country\_id)

);

CREATE TABLE IF NOT EXISTS COUNTRY(

country\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

country\_name VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(country\_id)

);

CREATE TABLE IF NOT EXISTS LIKES(

friend\_user\_id MEDIUMINT(8) NOT NULL,

status\_id MEDIUMINT(8) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY( friend\_user\_id,status\_id),

FOREIGN KEY(friend\_user\_id) REFERENCES FRIEND(friend\_user\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(status\_id) REFERENCES STATUS\_(status\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

ALTER TABLE BOOKMARK

ADD CONSTRAINT bookmark\_ibfk\_3 FOREIGN KEY (bookmark\_category\_id) REFERENCES BCATEGORY (bookmark\_category\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE ADDRESS

ADD CONSTRAINT address\_ibfk\_2 FOREIGN KEY (city\_id) REFERENCES CITY (city\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE CITY

ADD CONSTRAINT city\_ibfk\_1 FOREIGN KEY (country\_id) REFERENCES COUNTRY (country\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE ACTIVITIES

ADD CONSTRAINT activities\_ibfk\_4 FOREIGN KEY (event\_address\_id) REFERENCES ADDRESS (address\_id) ON DELETE CASCADE ON UPDATE CASCADE;

**CREATE SCHEMA LINKEDIN;**

USE LINKEDIN;

CREATE TABLE IF NOT EXISTS ADDRESS(

address\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

line1 VARCHAR(45) DEFAULT NULL,

line2 VARCHAR(45) DEFAULT NULL,

line3 VARCHAR(45) DEFAULT NULL,

city VARCHAR(45) DEFAULT NULL,

zip VARCHAR(45) DEFAULT NULL,

country VARCHAR(45) DEFAULT NULL,

PRIMARY KEY (address\_id)

);

CREATE TABLE IF NOT EXISTS MARITAL\_STATUS(

marital\_status\_id MEDIUMINT(8) NOT NULL,

marital\_status ENUM('Single','in Relationship','Married','Separated') DEFAULT NULL,

PRIMARY KEY(marital\_status\_id)

);

CREATE TABLE IF NOT EXISTS ORGANIZATIONS(

org\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

org\_name VARCHAR(45) DEFAULT NULL,

org\_description VARCHAR(255) DEFAULT NULL,

other VARCHAR(255) DEFAULT NULL,

PRIMARY KEY(org\_id)

);

CREATE TABLE IF NOT EXISTS COMPANY (

company\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

cname VARCHAR(45) NOT NULL,

sector VARCHAR(45) DEFAULT NULL,

size VARCHAR(45) DEFAULT NULL,

ctype VARCHAR(45) DEFAULT NULL,

cdesc VARCHAR(255) DEFAULT NULL,

logo VARCHAR(45) DEFAULT NULL,

address\_id MEDIUMINT(8) DEFAULT NULL,

boss MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (company\_id),

FOREIGN KEY (address\_id) REFERENCES ADDRESS (address\_id) ON DELETE CASCADE ON UPDATE CASCADE,

KEY company\_ibfk\_2 (boss)

);

CREATE TABLE IF NOT EXISTS MEMBERS(

member\_id MEDIUMINT(8) NOT NULL DEFAULT '0',

birthday DATE DEFAULT NULL,

e\_mail VARCHAR(100) DEFAULT NULL,

password\_ VARCHAR(45) DEFAULT NULL,

fname VARCHAR(45) DEFAULT NULL,

minit VARCHAR(45) DEFAULT NULL,

lname VARCHAR(45) DEFAULT NULL,

gender ENUM('Male','Female') DEFAULT NULL,

join\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

other VARCHAR(255) DEFAULT NULL,

address\_id MEDIUMINT(8) NOT NULL,

current\_organization\_id MEDIUMINT(8) NOT NULL,

marital\_status\_id MEDIUMINT(8) NOT NULL,

work\_comp\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (member\_id),

FOREIGN KEY (work\_comp\_id) REFERENCES COMPANY (company\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (address\_id) REFERENCES ADDRESS (address\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (current\_organization\_id) REFERENCES ORGANIZATIONS (org\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (marital\_status\_id) REFERENCES MARITAL\_STATUS (marital\_status\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

ALTER TABLE COMPANY

ADD CONSTRAINT company\_ibfk\_2 FOREIGN KEY (boss) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE;

CREATE TABLE IF NOT EXISTS JOB\_OFFER (

job\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

job\_title VARCHAR(45) DEFAULT NULL,

location VARCHAR(45) DEFAULT NULL,

job\_function VARCHAR(255) DEFAULT NULL,

emplooye\_type VARCHAR(45) DEFAULT NULL,

seniority VARCHAR(45) DEFAULT NULL,

offer\_desc VARCHAR(255) DEFAULT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

company\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (job\_id),

FOREIGN KEY (company\_id) REFERENCES COMPANY(company\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FOLLOWS\_ (

member\_id MEDIUMINT(8) NOT NULL,

company\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (member\_id, company\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (company\_id) REFERENCES COMPANY (company\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS GROUPS\_(

group\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

group\_name VARCHAR(45) DEFAULT NULL,

group\_description VARCHAR(255) DEFAULT NULL,

group\_start TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

group\_end TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

group\_last\_activity TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

other VARCHAR(255) DEFAULT NULL,

create\_member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(group\_id),

FOREIGN KEY (create\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CV(

cv\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

date\_created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(cv\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CV\_SECTION(

cv\_section\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

title ENUM('Work Experience','Education','Licance and Sertificated','Volunteer Experience') DEFAULT NULL,

description\_ VARCHAR(255) DEFAULT NULL,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

cv\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(cv\_section\_id),

FOREIGN KEY (cv\_id) REFERENCES CV (cv\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROFILE\_(

profile\_id MEDIUMINT(8) NOT NULL UNIQUE,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8),

PRIMARY KEY(profile\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROFILE\_SECTION(

profile\_section\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

title ENUM('Skills','Patent','Course','Project','Honors and Awards','Exam Scores','Languages','Organizations'),

text\_ VARCHAR(255) DEFAULT NULL,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

profile\_id MEDIUMINT(8),

PRIMARY KEY(profile\_section\_id),

FOREIGN KEY (profile\_id) REFERENCES PROFILE\_ (profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CONNECTS(

member\_id MEDIUMINT(8) NOT NULL,

connected\_member\_id MEDIUMINT(8) NOT NULL,

connection\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(member\_id,connected\_member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (connected\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS RECOMMENDATION(

member\_id MEDIUMINT(8) NOT NULL,

recommended\_id MEDIUMINT(8) NOT NULL,

start\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

others\_ VARCHAR(255) DEFAULT NULL,

PRIMARY KEY(member\_id,recommended\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (recommended\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MEMBER\_GROUPS(

member\_id MEDIUMINT(8) NOT NULL,

group\_id MEDIUMINT(8) NOT NULL,

date\_joined TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_left TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(member\_id,group\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (group\_id) REFERENCES GROUPS\_ (group\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

**CREATE SCHEMA SOCIAL\_RESUME\_DATABASE;**

USE SOCIAL\_RESUME\_DATABASE;

CREATE TABLE IF NOT EXISTS COUNTRY(

country\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

country\_name VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(country\_id)

);

CREATE TABLE IF NOT EXISTS CITY(

city\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

city\_name VARCHAR(45) DEFAULT NULL,

country\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(city\_id),

FOREIGN KEY(country\_id) REFERENCES COUNTRY (country\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS ORGANIZATIONS(

org\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

org\_name VARCHAR(45) DEFAULT NULL,

org\_description VARCHAR(255) DEFAULT NULL,

other VARCHAR(255) DEFAULT NULL,

PRIMARY KEY(org\_id)

);

CREATE TABLE IF NOT EXISTS MEMBERS(

member\_id MEDIUMINT(8) NOT NULL DEFAULT '0',

e\_mail VARCHAR(100) DEFAULT NULL,

password\_ VARCHAR(45) DEFAULT NULL,

fname VARCHAR(45) DEFAULT NULL,

minit VARCHAR(45) DEFAULT NULL,

lname VARCHAR(45) DEFAULT NULL,

gender ENUM('Male','Female') DEFAULT NULL,

marital\_status ENUM('Single','in Relationship','Married','Separated') DEFAULT NULL,

profile\_picture VARCHAR(255) DEFAULT NULL,

current\_organization\_id MEDIUMINT(8) NOT NULL,

join\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

other VARCHAR(255) DEFAULT NULL,

work\_comp\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (member\_id),

FOREIGN KEY (current\_organization\_id) REFERENCES ORGANIZATIONS (org\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROFILE\_(

profile\_id MEDIUMINT(8) NOT NULL UNIQUE,

numberOfFriends tinyint(3) NOT NULL DEFAULT '0',

phone\_number VARCHAR(45) DEFAULT NULL,

birthday DATE DEFAULT NULL,

about\_me VARCHAR(255) DEFAULT NULL,

interests VARCHAR(255) DEFAULT NULL,

religion VARCHAR(255) DEFAULT NULL,

hobbies VARCHAR(255) DEFAULT NULL,

fav\_movies VARCHAR(255) DEFAULT NULL,

fav\_artists VARCHAR(255) DEFAULT NULL,

fav\_books VARCHAR(255) DEFAULT NULL,

fav\_animals VARCHAR(255) DEFAULT NULL,

everything\_else VARCHAR(255) DEFAULT NULL,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8),

PRIMARY KEY(profile\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS ADDRESS(

address\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

address VARCHAR(255) DEFAULT NULL,

member\_id MEDIUMINT(8) DEFAULT NULL,

city\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY(address\_id),

FOREIGN KEY (city\_id) REFERENCES CITY (city\_id),

FOREIGN KEY(member\_id) REFERENCES MEMBERS(member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS COMPANY (

company\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

cname VARCHAR(45) NOT NULL,

sector VARCHAR(45) DEFAULT NULL,

size VARCHAR(45) DEFAULT NULL,

ctype VARCHAR(45) DEFAULT NULL,

cdesc VARCHAR(45) DEFAULT NULL,

logo VARCHAR(45) DEFAULT NULL,

address\_id MEDIUMINT(8) DEFAULT NULL,

boss MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (company\_id),

FOREIGN KEY (address\_id) REFERENCES ADDRESS (address\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (boss) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

ALTER TABLE MEMBERS

ADD CONSTRAINT members\_ibfk\_2 FOREIGN KEY (work\_comp\_id) REFERENCES COMPANY (company\_id) ON DELETE CASCADE ON UPDATE CASCADE;

CREATE TABLE IF NOT EXISTS JOB\_OFFER (

job\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

job\_title VARCHAR(45) DEFAULT NULL,

location VARCHAR(45) DEFAULT NULL,

job\_function VARCHAR(255) DEFAULT NULL,

emplooye\_type VARCHAR(45) DEFAULT NULL,

seniority VARCHAR(45) DEFAULT NULL,

offer\_desc VARCHAR(45) DEFAULT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

company\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (job\_id),

FOREIGN KEY (company\_id) REFERENCES COMPANY(company\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FOLLOWS\_ (

member\_id MEDIUMINT(8) NOT NULL,

company\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (member\_id, company\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (company\_id) REFERENCES COMPANY (company\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS NICKNAME (

nickname\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

nickname VARCHAR(45) NOT NULL,

member\_id MEDIUMINT(8) NOT NULL UNIQUE,

PRIMARY KEY(nickname\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FRIEND (

member\_id MEDIUMINT(8) NOT NULL,

friend\_member\_id MEDIUMINT(8) NOT NULL,

is\_subscriber BOOLEAN DEFAULT '0',

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(member\_id, friend\_member\_id),

FOREIGN KEY (friend\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS GROUPS\_(

group\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

group\_name VARCHAR(45) DEFAULT NULL,

group\_description VARCHAR(255) DEFAULT NULL,

group\_start TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

group\_end TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

group\_last\_activity TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

other VARCHAR(255) DEFAULT NULL,

create\_member\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY(group\_id),

FOREIGN KEY (create\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS EVENTS\_ (

event\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

event\_name VARCHAR(45) DEFAULT NULL,

event\_description VARCHAR(255) DEFAULT NULL,

event\_start TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

event\_end TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

event\_address\_id MEDIUMINT(8) DEFAULT NULL,

other VARCHAR(255) DEFAULT NULL,

create\_member\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (event\_id),

FOREIGN KEY (event\_address\_id) REFERENCES ADDRESS (address\_id),

FOREIGN KEY (create\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PAGES (

page\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

page\_name VARCHAR(45) DEFAULT NULL,

page\_description VARCHAR(255) DEFAULT NULL,

other VARCHAR(255) DEFAULT NULL,

create\_member\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (page\_id),

FOREIGN KEY (create\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CV(

cv\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(cv\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CV\_SECTION(

cv\_section\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

title ENUM('Work Experience','Education','Licance and Sertificated','Volunteer Experience') DEFAULT NULL,

description\_ VARCHAR(255) DEFAULT NULL,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

cv\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(cv\_section\_id),

FOREIGN KEY (cv\_id) REFERENCES CV (cv\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROFILE\_SECTION(

profile\_section\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

title ENUM('Skills','Patent','Course','Project','Honors and Awards','Exam Scores','Languages','Organizations'),

text\_ VARCHAR(255) DEFAULT NULL,

date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_updated TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

profile\_id MEDIUMINT(8),

PRIMARY KEY(profile\_section\_id),

FOREIGN KEY (profile\_id) REFERENCES PROFILE\_ (profile\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS CONNECTS(

member\_id MEDIUMINT(8) NOT NULL,

connected\_member\_id MEDIUMINT(8) NOT NULL,

connection\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(member\_id,connected\_member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (connected\_member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS RECOMMENDATION(

member\_id MEDIUMINT(8) NOT NULL,

recommended\_id MEDIUMINT(8) NOT NULL,

start\_date TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

others\_ VARCHAR(255) DEFAULT NULL,

PRIMARY KEY(member\_id,recommended\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (recommended\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS BCATEGORY (

bookmark\_category\_id SMALLINT(5) NOT NULL AUTO\_INCREMENT,

bname VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(bookmark\_category\_id)

);

CREATE TABLE IF NOT EXISTS BOOKMARK (

bookmark\_id BIGINT(20) NOT NULL AUTO\_INCREMENT,

url VARCHAR(255) DEFAULT NULL,

clicks SMALLINT(5) NOT NULL,

rating SMALLINT(5) DEFAULT NULL,

favorite BOOLEAN DEFAULT '1',

bookmark\_category\_id SMALLINT(5) DEFAULT NULL,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (bookmark\_id),

FOREIGN KEY (bookmark\_category\_id) REFERENCES BCATEGORY (bookmark\_category\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS FCATEGORY (

feed\_category\_id SMALLINT(5) NOT NULL AUTO\_INCREMENT,

fname VARCHAR(45) DEFAULT NULL,

PRIMARY KEY(feed\_category\_id)

);

CREATE TABLE IF NOT EXISTS FEED (

feed\_id BIGINT(20) NOT NULL AUTO\_INCREMENT,

url VARCHAR(255) DEFAULT NULL,

clicks SMALLINT(5) NOT NULL,

rating SMALLINT(5) DEFAULT NULL,

favorite BOOLEAN DEFAULT '1',

privacy TINYINT(3) DEFAULT '3',

feed\_category\_id SMALLINT(5) DEFAULT NULL,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (feed\_id),

FOREIGN KEY (feed\_category\_id) REFERENCES FCATEGORY (feed\_category\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS STATUS\_ (

status\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) DEFAULT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

thumbs\_up TINYINT(3) DEFAULT NULL,

thumbs\_down TINYINT(3) DEFAULT NULL,

member\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (status\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS BLOG (

blog\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) DEFAULT NULL,

author VARCHAR(45) DEFAULT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8) NOT NULL UNIQUE,

PRIMARY KEY(blog\_id, member\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS NOTIFICATION (

notification\_id MEDIUMINT (8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(notification\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MESSAGE(

send\_member\_id MEDIUMINT(8) NOT NULL,

to\_member\_id MEDIUMINT(8) NOT NULL,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(send\_member\_id, to\_member\_id),

FOREIGN KEY (send\_member\_id) REFERENCES MEMBERS(member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (to\_member\_id) REFERENCES MEMBERS(member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS COMMENTS (

comment\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

message VARCHAR(255) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

status\_id MEDIUMINT(8) DEFAULT NULL,

friend\_member\_id MEDIUMINT(8) DEFAULT NULL,

member\_id MEDIUMINT(8) DEFAULT NULL,

PRIMARY KEY (comment\_id),

FOREIGN KEY (status\_id) REFERENCES STATUS\_(status\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (friend\_member\_id) REFERENCES FRIEND (friend\_member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (member\_id) REFERENCES MEMBERS(member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS DEPENDENTS (

dep\_id MEDIUMINT(8) NOT NULL AUTO\_INCREMENT,

dep\_name VARCHAR(45) DEFAULT NULL,

relation VARCHAR(45) DEFAULT NULL,

birthday DATE DEFAULT NULL,

member\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY (dep\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS(member\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS LIKES(

friend\_member\_id MEDIUMINT(8) NOT NULL,

status\_id MEDIUMINT(8) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY(friend\_member\_id,status\_id),

FOREIGN KEY(friend\_member\_id) REFERENCES FRIEND(friend\_member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY(status\_id) REFERENCES STATUS\_(status\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MEMBER\_GROUPS(

member\_id MEDIUMINT(8) NOT NULL,

group\_id MEDIUMINT(8) NOT NULL,

date\_joined TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_left TIMESTAMP DEFAULT NULL,

PRIMARY KEY(member\_id,group\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (group\_id) REFERENCES GROUPS\_ (group\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MEMBER\_EVENTS(

member\_id MEDIUMINT(8) NOT NULL,

event\_id MEDIUMINT(8) NOT NULL,

PRIMARY KEY(member\_id,event\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (event\_id) REFERENCES EVENTS\_ (event\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS MEMBER\_PAGES(

member\_id MEDIUMINT(8) NOT NULL,

page\_id MEDIUMINT(8) NOT NULL,

date\_joined TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

date\_left TIMESTAMP DEFAULT NULL,

PRIMARY KEY(member\_id,page\_id),

FOREIGN KEY (member\_id) REFERENCES MEMBERS (member\_id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (page\_id) REFERENCES PAGES (page\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

1. **Populate the database you just created again using SQL script file loaded with sample tuples.**

**FACEBOOK INSERT VALUES:**

INSERT INTO PRIVACY (privacy\_id, privacy\_type) VALUES

(0, 'Only Me'),

(1, 'Friends'),

(2, 'Friends of Friends'),

(3, 'Everyone');

INSERT INTO USERS (user\_id, password\_, fname, minit, lname, e\_mail, profile\_picture, created\_at) VALUES

(1, '123456', 'Özgenur', null , 'Çapuk', 'ozgenurcapuk@gmail.com', 'www.facebook.com/photos', '2018-12-15 16:35:00'),

(2, '654321', 'İrem', null, 'Tunç', 'iremtunc@gmail.com', 'www.facebook.com/photos2', '2018-12-15 16:36:00'),

(3, '456789', 'Yasin', null, 'Kaya', 'yasinkaya@gmail.com', null, '2018-12-15 16:36:00'),

(4, '987654', 'Salih', 'Mustafa', 'Kaya', 'salihkaya@hotmail.com', null, '2018-12-15 16:38:50'),

(5, '456789', 'Merve', 'Eda', 'Bilgili', 'merve@hotmail.com', null, '2018-12-16 15:30:25'),

(6, '789456', 'Özgür', null, 'Çapuk', 'ozgur26@gmail.com', null, '2018-12-16 15:30:20'),

(7, '789654', 'Ahmet', null, 'Altan', null,null, '2018-12-16 15:40:00'),

(8, '123456', 'Nur', 'Gizem', 'Çalışkan', 'nurgizem@gmail.com','www.facebook.com/photos3' ,'2018-12-17 15:50:00');

INSERT INTO NICKNAME (nickname\_id, nickname, user\_id, privacy) VALUES

(1, 'ozgenurc', 1, 1),

(2, 'iremtnc', 2, 1),

(3, 'ysnky', 3, 0),

(4, 'slhky', 4, 3),

(5, 'mevy', 5,3),

(6, 'özgr26',6,3),

(7, 'ahmtaltn',7,2),

(8, 'nurç',8,0);

INSERT INTO PROFILE\_ (profile\_id, numberOfFriends , phone\_number, e\_mail, birthday, gender, about\_me, relationship, interests, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, user\_id, privacy) VALUES

(1, 0, '5434605702', 'ozgenurcapuk@gmail.com', '1996-07-04', 'Female', 'Bilgisayar mühendisliği öğrencisi', 'in Relationship', null, null, 'Uyumak', 'Interstellar, Inception', 'Johnny Depp, Jim Carry', 'Harry Potter Series', 'Dogs', null, 1,1),

(2, 0, '5371234545', 'iremtunc@gmail.com', '1997-02-13', 'Female', null, 'Single', null, 'Muslim', 'Programlama', null, 'Benedict Cumberbatch', 'Uçurtma Avcısı', 'Cat', null, 2,1),

(3, 0, '5415412323', 'yasinkaya@gmail.com', '1991-08-27', 'Male', 'Elektrik elektronik okuyor', 'in Relationship',null, null, 'Bilgisayar oyunları', 'Game of Thsrones', 'Emilia Clarke', 'Yüzüklerin Efendisi', 'bütün hayvanlar',null, 3,1),

(4, 0, '5478962312', 'salihkaya@hotmail.com', '1990-09-02', 'Male', 'Yapı Kredi İş Analisti', 'Married',null,null,null,null,null,null,'Cats',null,4,3),

(5, 0, '5461237889', 'merve@hotmail.com', '1980-01-01', 'Female', null, 'Separated',null,'Inancsız', 'Youtube, Video editlemek', 'Notebook', 'Brad Pitt', null, 'Cats', 'kanalımı takip etmeyi unutmayın', 5,3),

(6, 0, null, null, null, 'Male',null, 'Married', null, 'Muslim', 'Yüzmek', 'Marvel filmleri', 'Eminem',null, 'Fish',null, 6,1),

(7, 0, null, 'ahmetaltan@hotmail.com', null, null, 'Tiyatro her şey.', 'Single', null, null, null,null,null,null,null,null, 7, 2),

(8, 0, null,null, '1992-07-15', 'Female', null, null, null, null, 'Kitap yazmak, şarkı söylemek, resim yapmak', 'Doctor Who, Memento', 'Queen, Sting', 'Kişisel gelişim', 'Tüm hayvanları severim:)', null,8,3);

INSERT INTO FRIEND (user\_id, friend\_user\_id, privacy, is\_subscriber, created\_at) VALUES

(1, 2, 1, 1, default),

(1, 3, 1, 1, default),

(1, 4, 1, 1, default),

(1, 5, 1, 1, default),

(5, 7, 1, 1, default),

(5, 8, 1, 1, default),

(2, 3, 1, 1, default),

(2, 4, 1, 1, default),

(2, 8, 1, 1, default),

(3, 6, 1, 1, default),

(4, 7, 1, 1, default);

INSERT INTO BCATEGORY (bookmark\_category\_id, bname) VALUES

(1, 'Page'),

(2, 'Group'),

(3, 'Event'),

(4, 'Application');

INSERT INTO BOOKMARK (bookmark\_id, url, clicks, rating, privacy, favorite, bookmark\_category\_id, user\_id) VALUES

(1, 'www.faceboook.com/mezunsayfasi', 1000, 5, default, default, 1, 1),

(2, 'www.facebook.com/mezunlargrubu', 500,4, default, default, 2, 1),

(3, 'www.facebook.com/mezunlartoplantısı',250, 4, 2, default, 3, 1),

(4, 'www.facebook.com/games/candycrush',10000, 5, 3, default, 4, 3);

INSERT INTO STATUS\_ (status\_id, message, created\_at, thumbs\_up, thumbs\_down, privacy, user\_id) VALUES

(1, 'We can do it!', '2018-12-15 19:30:00', 2,1,2,1),

(2, 'Havalar çok soğuk', '2018-12-15 19:30:00', 2, 2, 3, 2),

(3, 'Çok uykum var', '2018-12-16 02:00:00', 50,1,3,5);

INSERT INTO COUNTRY (country\_id, country\_name) VALUES

(1, 'Türkiye'),

(2, 'Fransa'),

(3, 'İngiltere'),

(4, 'Amerika'),

(5, 'Japonya');

INSERT INTO CITY (city\_id, city\_name, country\_id) VALUES

(1, 'İzmir', 1),

(2, 'Eskişehir', 1),

(3, 'Tekirdağ', 1),

(4, 'Paris', 2),

(5, 'Tokyo', 5),

(6, 'Londra', 3),

(7, 'Ankara', 1);

INSERT INTO ADDRESS (address\_id, address, profile\_id, city\_id) VALUES

(1, 'Ergene mah. 544 sok.' , 1, 1),

(2, 'Atatürk mah. 120 sok.', 1, 2),

(3, 'Champ de Mars, 5 Avenue Anatole', 3, 4),

(4, '17 Place du Trocadéro et du 11 Novembre', 2, 4),

(5, '221b Baker St, Marylebone', 8, 6),

(6, 'İnönü mah. 152 sokak', 7, 1),

(7, 'yasemin mah, 70 sokak', null, 2);

INSERT INTO ACTIVITIES (act\_id, act\_name, act\_exp, act\_type, event\_date, event\_time, event\_address\_id, privacy, join\_user\_id, create\_user\_id) VALUES

(1, 'Ege Üniversitesi 2014 Girişliler', 'Ders notları ve iletişim için grubumuzu kullanabilirsiniz.', 'Group', null, null, null, 3, 2, 1),

(2, '2.El Eşya Alım Satım', null, 'Group', null, null, null, 3, 1, null),

(3, 'Dr. Who Fan Page', 'Whovianlar buraya', 'Page', null, null, null, 3, 2, 1),

(5, 'Programlama Sevenler', 'Programlama hakkında her şey', 'Page', null, null, null, 3, 5, 2),

(4, 'Orhan Oğuz And Lisesi Mezunlar Yemeği', '2014 mezunlarını bekleriz.', 'Event', '2019-01-15', '20:00:00', 7, 1, 1, null);

INSERT INTO BLOG (blog\_id, message, author, created\_at, user\_id) VALUES

(1, 'İzlenecek filmler listesi', 'Ben', default, 1),

(2, 'Ölmeden önce yapmam gereken şeyler', 'İrem', default, 2);

INSERT INTO LANGUAGE\_ (language\_, user\_id) VALUES

( 'Türkçe', 1),

( 'İngilizce', 2),

( 'Fransızca', 8),

( 'İngilizce', 3);

INSERT INTO NOTIFICATION (notification\_id, message, created\_at, user\_id) VALUES

(1, '.....Fotoğrafını beğendi.', default,1),

(2, '.....Bir video paylaştı.', default,1),

(3, '.....Durumunuza yotum yaptı.', default,2),

(4, '.....Fotoğrafınızı beğendi.', default,3);

INSERT INTO COMMENTS (comment\_id, message, created\_at, status\_id, friend\_user\_id, user\_id) VALUES

(1, 'Neyi başarıyorsunuz?', default, 1, 3, null),

(2, 'Görürsün yakında.', default, 1, null, 1),

(3, 'İzmir soğuksa biz burda ölelim.', default , 2, 8, null),

(4, 'Ne zaman yok ki uykun', default, 3, 5, null);

INSERT INTO MESSAGE (send\_user\_id, to\_user\_id, message) VALUES

(1,2,'Bebeğim proje nasıl gidiyor?'),

(2,1,'İyi gidiyor sanırım.'),

(1,3,'Yarın benimle yemeğe geliyor musun?'),

(2,8, 'Nasılsın?'),

(1,5,'Bu aralar çok uykusuz kaldım.');

INSERT INTO DEPENDENTS (dep\_id, dep\_name, relation, birthday, profile\_id) VALUES

(1, 'Serpil Çapuk', 'Anne', '1974-11-28', 1),

(2, 'Eylül Çapuk', 'Kardeş', '2010-10-26',1),

(3, 'Sema Altan', 'Abla', default, 7);

INSERT INTO JOBS (job\_id, job\_name, position\_, entry\_date, relase\_date, profile\_id) VALUES

(1, 'Yapı Kredi', 'İş Analisti', '2014', default, 4),

(2, 'Garanti Bankası', 'Stajer', '2013', '2013',4),

(3, 'Can Yayınları', 'Yazar', '2016', default, 8),

(4, 'Yapı Kredi', 'Genel Müdür', '2018', default, 5),

(5, 'Riot Games', 'Software Engineer', '2018', default, 3);

INSERT INTO SCHOOL (school\_id, school\_type, school\_name, entry\_date, graduate, profile\_id) VALUES

(1, 'Lise', 'Prof. Dr. Orhan Oğuz And. Lisesi', '2010', '2014', 1),

(2, 'Üniversite', 'Ege Üniversitesi', '2014', default, 1),

(3, 'Üniversite', ' Orta Doğu Teknik Üniversitesi', '2010', '2014', 3),

(4, 'Üniversite', 'Ege Üniversitesi', '2010', '2014', 4);

INSERT INTO LIKES (friend\_user\_id, status\_id, created\_at) VALUES

(2, 1, default),

(3, 1, default),

(4, 1, default),

(5, 1, default),

(6, 3, default),

(7, 3, default);

**INSERT LINKEDIN VALUES**

USE LINKEDIN;

INSERT INTO ADDRESS (address\_id, line1, line2,line3,city,zip,country) VALUES

(1, 'Yunus Emre Mah.', '6444 Sok. No:3','Baris Apartmani', 'izmir', NULL, 'Turkey'),

(2, 'Altintop Mah.', 'Lise Cad. No:2','Zafer Apartmani', 'ankara', NULL,'Turkey'),

(3, 'Ataturk Mah.', '22. Cad. No:5', 'Database Apartmani', 'izmit', NULL, 'Turkey'),

(4, 'Bornova Mah.', 'Universite Cad.', 'Ege Apartmani', 'izmir', NULL, 'Turkey'),

(5, 'Bahçelievler Mah.', 'Anavatan Cad.', 'Beşiktaş', 'istanbul', default, 'Turkey');

INSERT INTO MARITAL\_STATUS (marital\_status\_id, marital\_status) VALUES

(1,'Single'),

(2,'in Relationship'),

(3,'Married'),

(4,'Separated');

INSERT INTO ORGANIZATIONS (org\_id, org\_name, org\_description, other) VALUES

(1,'Tanisma Partisi','Tanismak isteyenlerin katilimi rica olunur.',NULL),

(2,'Klup Toplantisi','Etkinlik planlari icin toplanilacaktir.','Okulda toplanilacak.'),

(3,'Aksam Yemegi','Tum uyelerimiz davetlidir.',NULL),

(4,'Kutlama','Eglence icin herkesi bekliyoruz.',NULL);

INSERT INTO MEMBERS (member\_id, birthday, e\_mail, password\_, fname, minit, lname, gender, join\_date, other, address\_id, current\_organization\_id, marital\_status\_id) VALUES

(1,'1997-12-22','iremtunc@gmail.com','irem123','Irem',NULL,'Tunc','Female','2014-10-22 07:32:13',NULL,1,4,2),

(2,'1995-10-28','salihatar@gmail.com','salih96','Salih','Muhammed','Atar','Male', '2017-07-27 10:32:13',NULL,1,2,4),

(3,'1993-06-21','ozge12@gmail.com','ozge1212','Ozge','Nur','Capuk','Female','2013-11-25 15:32:13',NULL,3,1,3),

(4,'1997-05-12','yasinbey@gmail.com','123yasin123','Yasin',NULL,'Ayhan','Male','2011-08-25 12:32:13',NULL,2,2,2),

(5,'1992-07-08','nurcan\_92@gmail.com','asdfgh32','Nurcan',NULL,'Ozcan','Female','2018-07-20 14:32:13',NULL,2,1,3),

(6,'1998-07-10','ahmet\_izmirli@gmail.com','izmir35ahmet','Ahmet',NULL,'Bircan','Male','2012-01-14 07:32:13',NULL,3,2,4),

(7,'1996-10-11','mehmet\_gulay@hotmail.com','23568ege','Mehmet','Birol','Ayyildiz','Male','2014-02-13 09:32:13',NULL,2,4,3),

(8,'1999-04-17','aysekaraman@gmail.com','karaman458','Ayse',NULL,'Karaman','Male','2012-03-18 08:42:13',NULL,1,3,3);

INSERT INTO COMPANY (company\_id, cname, sector, size, ctype, cdesc, logo, address\_id, boss) VALUES

(1, 'Trenkwalder Turkey', 'Satış-Pazarlama', '1000-5000 Çalışan', 'Özel şirket', default, 'jpeg.', 5, 5),

(2, 'Turktelekom', 'Telekomünikasyon', '10000+ Çalışan', 'Halka açık özel şirket', default, 'jpef', 5, null),

(3, 'Ciceksepeti.com', 'Online alışveris', '51-200 Çalışan', 'Halka açık özel şirket', 'Sevdiklerinize çiçek hediye edin', 'cicek.jpeg', 4, 3);

INSERT INTO JOB\_OFFER (job\_id, job\_title, location, job\_function, emplooye\_type, seniority, offer\_desc, created\_at, company\_id) VALUES

(1, 'Video procuder', 'istanbul', 'Bilgi teknolojisi, sanat, yaratıcılık', 'Tam zamanlı', 'Orta-üst düzey', null, default, 2),

(2, 'Mobility specialist', 'İstanbul', 'Bilgi teknolojisi', 'Tam zamanlı', 'Senior', null, default, 2),

(3, 'Satış Pazarlama Uzmanı', 'İzmir', 'Satış, Pazarlama, Müşteri hizmetleri', 'Yarı zamanlı', 'Uzman', null, default, 1);

INSERT INTO FOLLOWS\_ (member\_id, company\_id) VALUES

(1, 1),

(2, 1),

(4, 2),

(5, 2),

(6, 2),

(3, 3);

INSERT INTO GROUPS\_ (group\_id,group\_name,group\_description,group\_start,group\_end,group\_last\_activity,other,create\_member\_id) VALUES

(1,'Gida Muhendisleri', 'Gida Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2016-01-13 22:00:00', '2015-01-13 18:00:00',NULL,2),

(2,'Bilgisayar Muhendisleri', 'Bilgisayar Muhendisleri giris yapabilir :)', '2013-01-13 22:00:00', '2014-01-13 23:00:00', '2013-01-13 22:00:00',NULL,7),

(3,'Elektirik Muhendisleri', 'Elektirik Elektronik Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2016-01-13 22:00:00', '2015-01-13 22:00:00',NULL,5),

(4,'Makine Muhendisleri', 'Makine Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2017-01-13 21:00:00', '2016-09-13 21:00:00',NULL,3),

(5,'Insaat Muhendisleri ', 'Insaat Muhendisleri giris yapabilir :)', '2012-01-13 22:00:00', '2018-01-13 20:00:00', '2017-10-13 21:00:00',NULL,8);

INSERT INTO CV (cv\_id,date\_created,date\_updated,member\_id) VALUES

(1,'2016-05-13 15:00:00','2017-05-13 15:00:00',2),

(2,'2016-05-13 15:00:00','2017-01-13 22:00:00',6),

(3,'2016-11-21 15:00:00','2017-01-13 15:00:00',8),

(4,'2016-05-13 15:00:00','2017-01-13 22:00:00',4);

INSERT INTO CV\_SECTION (cv\_section\_id, title, description\_, date\_created, date\_updated, cv\_id) VALUES

(1,'Work Experience','Software Engineer','2016-05-13 15:00:00','2017-01-13 22:00:00', 3),

(2,'Education','Ege Üniversitesi Bilgisayar Mühendisliği','2016-05-13 15:00:00','2018-01-13 22:00:00', 1),

(3,'Licance And Sertificated','Girisimcilik Sertifikası','2016-05-13 15:00:00','2017-01-13 22:00:00', 2),

(4,'Volunteer Experience','Toplum Gönüllüleri Vakfı','2016-05-13 15:00:00','2018-01-13 22:00:00', 4);

INSERT INTO PROFILE\_ (profile\_id, date\_created, date\_updated, member\_id) VALUES

(1,'2016-05-13 15:00:00','2017-01-13 22:00:00',4),

(2,'2012-05-14 15:00:00','2017-01-13 22:00:00',7),

(3,'2013-05-21 15:00:00','2017-01-13 22:00:00',8),

(4,'2011-05-17 15:00:00','2017-01-13 22:00:00',3),

(5,'2016-05-26 15:00:00','2018-02-13 22:00:00',2),

(6,'2017-03-27 15:00:00','2017-05-13 15:00:00',1),

(7,'2018-01-02 15:00:00','2018-05-13 15:00:00',5),

(8,'2010-05-09 15:00:00','2016-05-13 13:00:00',6);

INSERT INTO PROFILE\_SECTION (profile\_section\_id, title, text\_, date\_created, date\_updated, profile\_id) VALUES

(1,'course','Gitar Kursu,Resim Kursu','2016-05-13 15:00:00','2017-01-13 22:00:00',1),

(2,'languages','Ingilizce,Almanca','2016-05-13 15:00:00','2017-01-13 22:00:00',3);

INSERT INTO CONNECTS(member\_id, connected\_member\_id, connection\_date) VALUES

(1,2,'2017-03-27 15:00:00'),

(2,3,'2017-03-27 15:00:00'),

(3,4,'2017-03-27 15:00:00'),

(4,2,'2017-03-27 15:00:00'),

(5,1,'2017-03-27 15:00:00'),

(6,4,'2017-03-27 15:00:00'),

(7,1,'2017-03-27 15:00:00'),

(8,2,'2017-03-27 15:00:00');

INSERT INTO RECOMMENDATION (member\_id, recommended\_id, start\_date, others\_) VALUES

(1, 2, '2014-12-23 10:05:55', NULL),

(1, 3, '2014-12-23 10:05:55', NULL);

INSERT INTO MEMBER\_GROUPS (member\_id, group\_id, date\_joined, date\_left) VALUES

(1, 1, '2012-01-13 22:00:00',default),

(2, 2, '2012-01-13 22:00:00',default);

**SOCIAL RESUME DATABASE INSERT VALUES**

INSERT INTO COUNTRY (country\_id, country\_name) VALUES

(1, 'Türkiye'),

(2, 'Fransa'),

(3, 'İngiltere'),

(4, 'Amerika'),

(5, 'Japonya');

INSERT INTO CITY (city\_id, city\_name, country\_id) VALUES

(1, 'İzmir', 1),

(2, 'Eskişehir', 1),

(3, 'Tekirdağ', 1),

(4, 'Paris', 2),

(5, 'Tokyo', 5),

(6, 'Londra', 3),

(7, 'Ankara', 1);

INSERT INTO ORGANIZATIONS (org\_id, org\_name, org\_description, other) VALUES

(1,'Tanisma Partisi','Tanismak isteyenlerin katilimi rica olunur.',NULL),

(2,'Klup Toplantisi','Etkinlik planlari icin toplanilacaktir.','Okulda toplanilacak.'),

(3,'Aksam Yemegi','Tum uyelerimiz davetlidir.',NULL),

(4,'Kutlama','Eglence icin herkesi bekliyoruz.',NULL);

INSERT INTO MEMBERS (member\_id, e\_mail, password\_, fname, minit, lname, gender, marital\_status, profile\_picture, current\_organization\_id, join\_date, other) VALUES

(1, 'ozgenurcapuk@gmail.com', '123456', 'Özgenur', null , 'Çapuk', 'Female', 'in Relationship', 'ben.jpg', 4, default, null),

(2, 'iremtunc@gmail.com', '123456', 'İrem', null, 'Tunç', 'Female', 'Single', 'me.jpeg', 1, default, null),

(3, 'yasinbey@gmail.com','123yasin123','Yasin', null ,'Kaya', 'Male', 'in Relationship', null, 3, default, null),

(4, 'salihatar@gmail.com', 'salih123', 'Salih', 'Muhammed', 'Atar', 'Male', 'Single', null, 1, default, null),

(5, 'merve@hotmail.com', '987654', 'Merve', 'Eda', 'Bilgili', 'Female', 'Married', '1.jpeg', 2, default, null),

(6, 'mehmet\_gulay@hotmail.com', '1457854', 'Mehmet', 'Birol', 'Ayyildiz', 'Male', 'Separated' , null , 3, default, null);

INSERT INTO PROFILE\_ (profile\_id, numberOfFriends , phone\_number, birthday, about\_me, interests, religion, hobbies, fav\_movies, fav\_artists, fav\_books, fav\_animals, everything\_else, date\_created, date\_updated, member\_id) VALUES

(1, 0, '5434605055', '1996-07-04', 'Bilgisayar mühendisliği öğrencisi', null, null, 'Uyumak', 'Interstellar, Inception, Doctor Who', 'Johnny Depp, Jim Carry', 'Harry Potter Series', 'Dogs', null, default, default, 1),

(2, 0, '5371234545', '1997-02-13', null, null, 'Muslim', 'Programlama', null, 'Benedict Cumberbatch', 'Uçurtma Avcısı', 'Cat', null, default, default, 2),

(3, 0, '5415412323', '1991-08-27', 'Elektrik elektronik okuyor', null, null, 'Bilgisayar oyunları', 'Game of Thsrones', 'Emilia Clarke', 'Yüzüklerin Efendisi', 'bütün hayvanlar', null, default, default, 3),

(4, 0, '5478962312', '1990-09-02', 'Yapı Kredi İş Analisti',null ,null ,null ,null ,null ,null ,'Cats' ,null , default, default, 4),

(5, 0, '5461237889', '1980-01-01', null, null,'Inancsız', 'Youtube, Video editlemek', 'Notebook', 'Brad Pitt', null, 'Cats', 'kanalımı takip etmeyi unutmayın', default, default, 5),

(6, 0, null, '1992-07-15', null, null, null, 'Kitap yazmak, şarkı söylemek, resim yapmak', 'Doctor Who, Memento', 'Queen, Sting', 'Kişisel gelişim', 'Tüm hayvanları severim:)', null, default, default, 6);

INSERT INTO ADDRESS (address\_id, address, member\_id, city\_id) VALUES

(1, 'Ergene mah. 544 sok.' , 1, 1),

(2, 'Atatürk mah. 120 sok.', 1, 2),

(3, 'Champ de Mars, 5 Avenue Anatole', 3, 4),

(4, '17 Place du Trocadéro et du 11 Novembre', 2, 4),

(5, '221b Baker St, Marylebone', null, 6),

(6, 'İnönü mah. 152 sokak',5, 1),

(7, 'yasemin mah, 70 sokak', null, 2),

(8, 'orman findanlıgı', null, 2);

INSERT INTO COMPANY (company\_id, cname, sector, size, ctype, cdesc, logo, address\_id, boss) VALUES

(1, 'Trenkwalder Turkey', 'Satış-Pazarlama', '1000-5000 Çalışan', 'Özel şirket', default, 'jpeg.', 5, 5),

(2, 'Turktelekom', 'Telekomünikasyon', '10000+ Çalışan', 'Halka açık özel şirket', default, 'jpef', null, null),

(3, 'Ciceksepeti.com', 'Online alışveris', '51-200 Çalışan', 'Halka açık özel şirket', 'Sevdiklerinize çiçek hediye edin', 'cicek.jpeg', 6, 3);

INSERT INTO JOB\_OFFER (job\_id, job\_title, location, job\_function, emplooye\_type, seniority, offer\_desc, created\_at, company\_id) VALUES

(1, 'Video procuder', 'istanbul', 'Bilgi teknolojisi, sanat, yaratıcılık', 'Tam zamanlı', 'Orta-üst düzey', null, default, 2),

(2, 'Mobility specialist', 'İstanbul', 'Bilgi teknolojisi', 'Tam zamanlı', 'Senior', null, default, 2),

(3, 'Satış Pazarlama Uzmanı', 'İzmir', 'Satış, Pazarlama, Müşteri hizmetleri', 'Yarı zamanlı', 'Uzman', null, default, 1);

INSERT INTO FOLLOWS\_ (member\_id, company\_id) VALUES

(1, 1),

(2, 1),

(4, 2),

(5, 2),

(6, 2),

(3, 3);

INSERT INTO NICKNAME (nickname\_id, nickname, member\_id) VALUES

(1, 'ozgenurc', 1),

(2, 'iremtnc', 2),

(3, 'ysnky', 3),

(4, 'slhky', 4),

(5, 'mevy', 5),

(6, 'mhmt26',6);

INSERT INTO FRIEND (member\_id, friend\_member\_id, is\_subscriber, created\_at) VALUES

(1, 2, 1, default),

(1, 3, 1, default),

(1, 4, 1, default),

(1, 5, 1, default),

(5, 6, 1, default),

(5, 4, 1, default),

(5, 3, 1, default);

INSERT INTO GROUPS\_ (group\_id, group\_name, group\_description, group\_start, group\_end, group\_last\_activity, other, create\_member\_id) VALUES

(1,'Gida Muhendisleri', 'Gida Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2016-01-13 22:00:00', '2015-01-13 18:00:00',NULL,1),

(2,'Bilgisayar Muhendisleri', 'Bilgisayar Muhendisleri giris yapabilir :)', '2013-01-13 22:00:00', '2014-01-13 23:00:00', '2013-01-13 22:00:00',NULL,2),

(3,'Elektirik Muhendisleri', 'Elektirik Elektronik Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2016-01-13 22:00:00', '2015-01-13 22:00:00',NULL,1),

(4,'Makine Muhendisleri', 'Makine Muhendisleri giris yapabilir :)', '2015-01-13 22:00:00', '2017-01-13 21:00:00', '2016-09-13 21:00:00',NULL,3),

(5,'Insaat Muhendisleri ', 'Insaat Muhendisleri giris yapabilir :)', '2012-01-13 22:00:00', '2018-01-13 20:00:00', '2017-10-13 21:00:00',NULL,5);

INSERT INTO EVENTS\_ (event\_id, event\_name, event\_description, event\_start, event\_end, event\_address\_id, other, create\_member\_id) VALUES

(1, 'Mezun Yemeği', 'Mezunlarımızı bekleriz', '2019-01-01', '2019-01-02', 7, null, 1),

(2, 'Piknik', null, '2019-05-25', null, 8, null, null),

(3, 'Bilmüh Toplantısı', null, '2018-12-16', null, 5, null,1);

INSERT INTO PAGES (page\_id, page\_name, page\_description, other, create\_member\_id) VALUES

(1, '2014 Girişliler', 'Ege Üni 2014 girişliler sayfası', null, 1),

(2, 'Oyun Dünyası', 'Oyun severler buraya', null, 3),

(3, 'Bilmüh', null, null, 4);

INSERT INTO CV (cv\_id,date\_created,date\_updated,member\_id) VALUES

(1,'2016-05-13 15:00:00','2017-05-13 15:00:00',1),

(2,'2016-05-13 15:00:00','2017-01-13 22:00:00',2),

(3,'2016-11-21 15:00:00','2017-01-13 15:00:00',3),

(4,'2016-05-13 15:00:00','2017-01-13 22:00:00',4);

INSERT INTO CV\_SECTION (cv\_section\_id, title, description\_, date\_created, date\_updated, cv\_id) VALUES

(1,'Work Experience','Software Engineer','2016-05-13 15:00:00','2017-01-13 22:00:00', 1),

(2,'Education','Ege Üniversitesi Bilgisayar Mühendisliği','2016-05-13 15:00:00','2018-01-13 22:00:00', 1),

(3,'Licance And Sertificated','Girisimcilik Sertifikası','2016-05-13 15:00:00','2017-01-13 22:00:00', 1),

(4,'Volunteer Experience','Toplum Gönüllüleri Vakfı','2016-05-13 15:00:00','2018-01-13 22:00:00', 1),

(5,'Work Experience','iş Analisti','2016-06-13 15:00:00','2017-04-13 22:00:00', 2),

(6,'Education','ODTU Bilgisayar Mühendisliği','2016-08-13 15:00:00','2018-09-13 22:00:00', 4);

INSERT INTO PROFILE\_SECTION (profile\_section\_id,title,text\_,date\_created,date\_updated,profile\_id) VALUES

(1,'Course','Gitar Kursu,Resim Kursu','2016-05-13 15:00:00','2017-01-13 22:00:00',1),

(2,'Languages','Ingilizce,Almanca','2016-05-13 15:00:00','2017-01-13 22:00:00',3);

INSERT INTO CONNECTS(member\_id,connected\_member\_id,connection\_date) VALUES

(1, 2, default),

(1, 3, default),

(2, 4, default),

(2, 5, default),

(1, 4, default),

(5, 1, default),

(5, 6, default);

INSERT INTO RECOMMENDATION (member\_id, recommended\_id, start\_date, others\_) VALUES

(1, 2, '2014-12-23 10:05:55', NULL),

(1, 3, '2014-12-23 10:05:55', NULL),

(4, 6, default, null);

INSERT INTO BCATEGORY (bookmark\_category\_id, bname) VALUES

(1, 'Page'),

(2, 'Group'),

(3, 'Event'),

(4, 'Application');

INSERT INTO BOOKMARK (bookmark\_id, url, clicks, rating, favorite, bookmark\_category\_id, member\_id) VALUES

(1, 'www.faceboook.com/mezunsayfasi', 1000, 5, default, 1, 1),

(2, 'www.facebook.com/mezunlargrubu', 500,4, default, 2, 1),

(3, 'www.facebook.com/mezunlartoplantısı',250, 4, default, 3, 1),

(4, 'www.facebook.com/games/candycrush',10000, 5, default, 4, 3);

INSERT INTO STATUS\_ (status\_id, message, created\_at, thumbs\_up, thumbs\_down, member\_id) VALUES

(1, 'We can do it!', '2018-12-15 19:30:00', 2,1,1),

(2, 'Havalar çok soğuk', '2018-12-15 19:30:00', 2, 2, 2),

(3, 'Çok uykum var', '2018-12-16 02:00:00', 50,1,5);

INSERT INTO BLOG (blog\_id, message, author, created\_at, member\_id) VALUES

(1, 'İzlenecek filmler listesi', 'Ben', default, 1),

(2, 'Ölmeden önce yapmam gereken şeyler', 'İrem', default, 2);

INSERT INTO NOTIFICATION (notification\_id, message, created\_at, member\_id) VALUES

(1, '.....Fotoğrafını beğendi.', default,1),

(2, '.....Bir video paylaştı.', default,1),

(3, '.....Durumunuza yotum yaptı.', default,2),

(4, '.....Fotoğrafınızı beğendi.', default,3);

INSERT INTO COMMENTS (comment\_id, message, created\_at, status\_id, friend\_member\_id, member\_id) VALUES

(1, 'Neyi başarıyorsunuz?', default, 1, 3, null),

(2, 'Görürsün yakında.', default, 1, null, 1),

(3, 'İzmir soğuksa biz burda ölelim.', default , 2, 6, null),

(4, 'Ne zaman yok ki uykun', default, 3, 5, null);

INSERT INTO MESSAGE (send\_member\_id, to\_member\_id, message) VALUES

(1,2,'Bebeğim proje nasıl gidiyor?'),

(2,1,'İyi gidiyor sanırım.'),

(1,3,'Yarın benimle yemeğe geliyor musun?'),

(2,6, 'Nasılsın?'),

(1,5,'Bu aralar çok uykusuz kaldım.');

INSERT INTO DEPENDENTS (dep\_id, dep\_name, relation, birthday, member\_id) VALUES

(1, 'Serpil Çapuk', 'Anne', '1974-11-28', 1),

(2, 'Eylül Çapuk', 'Kardeş', '2010-10-26',1),

(3, 'Sema Altan', 'Abla', default, 5);

INSERT INTO LIKES (friend\_member\_id, status\_id, created\_at) VALUES

(2, 1, default),

(3, 1, default),

(4, 1, default),

(5, 1, default),

(6, 3, default);

INSERT INTO MEMBER\_GROUPS (member\_id, group\_id, date\_joined, date\_left) VALUES

(1, 1, '2012-01-13 22:00:00',default),

(2, 1, '2012-01-13 22:00:00',default),

(3, 2, default, default),

(4, 2, default, default),

(5, 2, default, default),

(6, 3, default, default);

INSERT INTO MEMBER\_EVENTS (member\_id, event\_id) VALUES

(1, 3),

(2, 3),

(5, 3),

(6, 3),

(1, 2),

(1, 1),

(4, 1),

(3, 3);

INSERT INTO MEMBER\_PAGES (member\_id, page\_id, date\_joined, date\_left) VALUES

(1, 2, default, default),

(1, 3, default, default),

(2, 2, default, default),

(5, 1, default, default);

1. **Write down 3 triggers for 3 different tables. Triggers should be meaninful.**

DELIMITER//

CREATE TRIGGER selfrequest2 BEFORE UPDATE ON FRIEND

FOR EACH ROW BEGIN

IF NEW.user\_id=New.friend\_user\_id

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot be friend yourself';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER selfrequest BEFORE INSERT ON FRIEND

FOR EACH ROW BEGIN

IF NEW.user\_id=New.friend\_user\_id

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot be friend yourself';

END IF;

END

//

DELIMITER ;

Veri eklerken veya güncellerken kişinin kendi kendini eklemesine engel olan trigger

DELIMITER //

CREATE TRIGGER arkadasmi BEFORE INSERT ON FRIEND

FOR EACH ROW BEGIN

IF EXISTS(SELECT friend\_user\_id

FROM FRIEND

WHERE user\_id=new.friend\_user\_id AND friend\_user\_id=new.user\_id)

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'You are already friends';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER arkadasmi2 BEFORE UPDATE ON FRIEND

FOR EACH ROW BEGIN

IF EXISTS(SELECT friend\_user\_id

FROM FRIEND

WHERE user\_id=new.friend\_user\_id AND friend\_user\_id=new.user\_id)

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'You are already friends';

END IF;

END

//

DELIMITER ;

Veri eklerken veya güncellerken kişilerin daha önce arkadaş olup olmadığını kontrol eden trigger

Arkadaş eklediğimizde ya da sildiğimizde arkadaşları saymaya yarayan trigger

DELIMITER //

CREATE TRIGGER FRIEND\_NUMBER\_DECREMENT BEFORE DELETE ON Friend

FOR EACH ROW BEGIN

UPDATE Profile\_ SET Profile\_.numberOfFriends = Profile\_.numberOfFriends -1

WHERE Profile\_.user\_id =(select OLD.friend\_user\_id );

UPDATE Profile\_ SET Profile\_.numberOfFriends = numberOfFriends -1 WHERE Profile\_.user\_id =OLD.user\_id;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER FRIEND\_NUMBER\_INCREMENT BEFORE INSERT ON Friend

FOR EACH ROW BEGIN

UPDATE Profile\_ SET Profile\_.numberOfFriends = Profile\_.numberOfFriends +1

WHERE Profile\_.user\_id =(select NEW.friend\_user\_id );

UPDATE Profile\_ SET Profile\_.numberOfFriends = numberOfFriends +1 WHERE Profile\_.user\_id =NEW.user\_id;

END

//

DELIMITER ;

Kişinin kendi kendine mesaj göndermesini engelleyen trigger

DELIMITER //

CREATE TRIGGER update\_selfmessage BEFORE UPDATE ON Message

FOR EACH ROW BEGIN

IF NEW.send\_user\_id=New.to\_user\_id

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot send message to yourself';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER before\_selfmessage BEFORE INSERT ON Message

FOR EACH ROW BEGIN

IF NEW.send\_user\_id=New.to\_user\_id

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot send message to yourself';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER schooldatecontrol BEFORE INSERT ON SCHOOL

FOR EACH ROW BEGIN

IF New.entry\_date>New.graduate

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Can not entry date bigger than graduate date';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER jobdatecontrol BEFORE INSERT ON JOBS

FOR EACH ROW BEGIN

IF New.entry\_date>New.relase\_date

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Can not entry date bigger than relase date';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER schooldatecontrol2 BEFORE UPDATE ON SCHOOL

FOR EACH ROW BEGIN

IF New.entry\_date>New.graduate

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Can not entry date bigger than graduate date';

END IF;

END

//

DELIMITER ;

DELIMITER //

CREATE TRIGGER jobdatecontrol2 BEFORE UPDATE ON JOBS

FOR EACH ROW BEGIN

IF New.entry\_date>New.relase\_date

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Can not entry date bigger than relase date';

END IF;

END

//

DELIMITER ;

Veri eklerken veya güncellerken giriş tarihlerinin çıkış tarhilerinden büyük olmasını engelleyen triggerlar

1. **Write down the following SQL statements:**
2. **Write sample INSERT, DELETE and UPDATE statements for 3 of the tables you have chosen.**

**USE FACEBOOK;**

/\*BLOG tablosu için\*/

UPDATE BLOG SET message='Tavsiye edilen diziler' WHERE blog\_id=1;

DELETE FROM BLOG where blog\_id=2;

INSERT INTO BLOG (message, author, created\_at, user\_id) VALUES

('İzlenecek filmler listesi', 'Ben', default, 2);

/\*STATUS tablosu için\*/

UPDATE STATUS\_ SET message='Dışarıda güzel bir hava var' WHERE status\_id=3;

DELETE FROM STATUS\_ WHERE status\_id=1;

INSERT INTO STATUS\_ (message, created\_at, thumbs\_up, thumbs\_down, privacy, user\_id) VALUES

('Başarabilirsin', '2018-12-15 19:30:00', 7,1,2,1);

/\*DEPENDENTS tablosu için\*/

UPDATE DEPENDENTS SET dep\_name='Neriman Capuk' WHERE dep\_id=1;

DELETE FROM DEPENDENTS WHERE dep\_id=3;

INSERT INTO DEPENDENTS (dep\_name, relation, birthday, profile\_id) VALUES

('Ayhan Çapuk', 'kuzen', '1974-11-28', 1);

**USE LINKEDIN;**

/\* MEMBERS tablosu için \*/

UPDATE MEMBERS SET lname='Atar' WHERE member\_id=1;

DELETE FROM MEMBERS WHERE member\_id=7;

INSERT INTO MEMBERS (birthday, e\_mail, password\_, fname, minit, lname, gender, join\_date, other, address\_id, current\_organization\_id, marital\_status\_id) VALUES

('1997-12-22','iremtunc@gmail.com','irem123','Irem',NULL,'Tunc','Female','2014-10-22 07:32:13',NULL,2,4,2);

/\* CV\_SECTION tablosu için\*/

UPDATE CV\_SECTION SET description\_='Front-End Developer' WHERE cv\_section\_id=1;

DELETE FROM CV\_SECTION WHERE cv\_section\_id=2;

INSERT INTO CV\_SECTION (title, description\_, date\_created, date\_updated, cv\_id) VALUES

('Education','Teacher','2016-05-13 15:00:00','2017-01-13 22:00:00', 3);

/\* PROFILE\_SECTION tablosu için\*/

UPDATE PROFILE\_SECTION SET title='Course' WHERE profile\_section\_id=1;

DELETE FROM PROFILE\_SECTION WHERE profile\_section\_id=2;

INSERT INTO PROFILE\_SECTION (title, text\_, date\_created, date\_updated, profile\_id) VALUES

('Languages','Yabacnı dil ve dans kursu','2016-05-13 15:00:00','2017-01-13 22:00:00',1);

**USE SOCIAL\_RESUME\_DATABASE;**

/\* MESSAGE tablosu için\*/

UPDATE MESSAGE SET message='Hey!Yarınki toplantıyı hatırlatmak istedim.' WHERE send\_member\_id=1 AND to\_member\_id=2;

DELETE FROM MESSAGE WHERE send\_member\_id=1 AND to\_member\_id=3;

INSERT INTO MESSAGE (send\_member\_id, to\_member\_id, message) VALUES

(2,3,'Nasılsın');

/\*STATUS\_ tablosu için \*/

UPDATE STATUS\_ SET message='İyi günler dileriz!' WHERE status\_id=2;

DELETE FROM STATUS\_ WHERE status\_id=3;

INSERT INTO STATUS\_ (message, created\_at, thumbs\_up, thumbs\_down, member\_id) VALUES

('At least you tried', '2018-12-15 19:30:00', 6,1,1);

/\*EVENTS\_ tablosu için \*/

UPDATE EVENTS\_ SET event\_description='Toplantı okulda yapılacaktır.' WHERE event\_id=3;

DELETE FROM EVENTS\_ WHERE event\_id=2;

INSERT INTO EVENTS\_ (event\_name, event\_description, event\_start, event\_end, event\_address\_id, other, create\_member\_id) VALUES

('Bisiklet Turu', 'Tüm bisiklet tutkunlarını bekleriz', '2019-01-01', '2019-01-02', 5, null, 1);

1. **Write 10 SELECT statements for the database you have implemented.**

**USE FACEBOOK;**

/\*Bir kullanıcının tüm arkadaş bilgileri\*/

SELECT \*

FROM USERS

WHERE USERS.user\_id IN

(SELECT Friend.friend\_user\_id

FROM Friend

WHERE Friend.user\_id IN

(SELECT USERS.user\_id

FROM USERS

WHERE USERS.fname='İrem' AND USERS.lname='Tunç'));

/\* Bir Kullanicinin bildigi diller\*/

SELECT LANGUAGE\_.language\_

FROM LANGUAGE\_

WHERE LANGUAGE\_.user\_id IN (

SELECT USERS.user\_id

FROM USERS

WHERE USERS.fname='Yasin' AND USERS.lname='Kaya');

/\*İngilizce dilini bilen kullanıcılar\*/

SELECT USERS.fname,USERS.minit, USERS.lname

FROM USERS

WHERE USERS.user\_id IN ( SELECT LANGUAGE\_.user\_id

FROM LANGUAGE\_

WHERE language\_='İngilizce')

/\*Bir kullanıcının Begendigi durumlar\*/

SELECT \*

FROM Status\_

WHERE Status\_.status\_id IN

(SELECT status\_id

FROM LIKES

WHERE LIKES.friend\_user\_id IN

(SELECT friend\_user\_id

FROM Friend

WHERE Friend.user\_id IN (

SELECT user\_id

FROM USERS WHERE USERS.fname='Merve' AND USERS.lname='Bilgili')));

**USE LINKEDIN;**

/\* izmir'de yaşayanların tüm bilgileri\*/

SELECT \*

FROM ADDRESS

WHERE address.city='izmir'

/\*single olanlar\*/

SELECT \*

FROM MARITAL\_STATUS

WHERE marital\_status='Single'

/\* title'ı course olanın textini yazdır \*/

SELECT text\_

FROM PROFILE\_SECTION

WHERE title='Course'

/\*2YE BAĞLANANLAR\*/

SELECT DISTINCT \*

FROM CONNECTS

WHERE connected\_member\_id=2

/\*2yle bağlantı kuranların dogum günleri\*/

SELECT DISTINCT birthday

FROM CONNECTS,MEMBERS

WHERE current\_organization\_id=2

/\*ADI AYLA BASLAYANLARIN ORGANİZATION İDLERİ\*/

SELECT DISTINCT current\_organization\_id

FROM ORGANIZATIONS,MEMBERS

WHERE fname LIKE 'A%'

/\*şartı sağlayanların isimleri\*/

SELECT fname,minit,lname

FROM MEMBERS

WHERE current\_organization\_id=1 AND marital\_status\_id=3

/\*İlişki durumu married olanlar\*/

FROM MEMBERS

WHERE MEMBERS.member\_id IN (

SELECT MARITAL\_STATUS.marital\_status\_id

FROM MARITAL\_STATUS

WHERE MARITAL\_STATUS.marital\_status='Married');

**USE SOCIAL\_RESUME\_DATABASE;**

/\*İZMİRDE YASAYANLARIN ALFABETIK SIRALAMAYA GÖRE LİSTESİ\*/

SELECT DISTINCT MEMBERS.member\_id,MEMBERS.fname, MEMBERS.lname

FROM ADDRESS,PROFILE\_,MEMBERS,CITY

WHERE ADDRESS.city\_id = '1' AND ADDRESS.Profile\_id = PROFILE\_.Profile\_id AND PROFILE\_.Member\_id=MEMBERS.Member\_id

ORDER BY fname,lname

/\*member\_id si 1 olanların arkadaşlarını ad soyad olarak listeleme (sıralı)\*/

SELECT MEMBERS.fname,MEMBERS.minit,MEMBERS.lname

FROM MEMBERS

WHERE MEMBERS.member\_id IN (SELECT FRIEND.Friend\_member\_id

FROM MEMBERS,FRIEND

WHERE MEMBERS.Member\_id=1 AND MEMBERS.Member\_id=FRIEND.Member\_id)

ORDER BY fname,minit,lname

/\*mail uzantısı @gmail olanların isimleri\*/

SELECT fname,minit,lname

FROM MEMBERS

WHERE MEMBERS.e\_mail LIKE '%gmail.com%'

ORDER BY lname DESC