Facebook

1- (one table): Facebook kullanıcılarında yaşı 20 yaşından büyük olanların fname+mname+lname+age olarak listeleyiniz.

2- (one table): Page tablosunda company\_date i 2000 ve 2005 arası olanların isimlerini ve kuruluş tarihlerini listeleyiniz.

3- (one table): Gruplarda grup isminde “python” geçen grup isimlerini listeleyiniz.

4- (two table): user id si 4001 olan kişinin yaptığı yorumları listele

5- (two table): Tüm kullanıcıların kullandıkları uygulamalar (application) arasından en çok kullanılan uygulamanın ismini listeleyiniz. (USES\_APPLICATION, APPLICATION)

6- (two table): en az 2 albümü bulunan sayfaları listeleyiniz

7- (two table): her mesaj için mesajı, gönderileni ve alıcıyı listeleyiniz.

8- (three table): user\_id si 4001 olan kişinin gittiği etkinlikleri yöneten sayfalarnın page\_İdlerini listeleyiniz.

9- (three table): Yaşı 18 ile 30 arasında olanların en fazla katıldığı etkinliği ve katılımcı sayısını listeleyiniz. (EVENT, USER ,ATTENDS)

10- (three table): Tüm kullanıcıların yönettiği etkinlikler içinden en çok etkinliğin olduğu adresin sokağını listeleyiniz. (USER, EVENT, LOCATION)

use facebook;

#################sorgu-1

select fname,mname,lname,extract(year from CURRENT\_TIMESTAMP)-extract(year from birth\_date)

from f\_user

where extract(year from CURRENT\_TIMESTAMP)-extract(year from birth\_date)>20;

#################sorgu-2

select pname, company\_date

from f\_page

where extract(year from company\_date) between 2000 and 2005;

#################sorgu-3

select group\_name

from f\_group

where group\_name like "%python%";

#################sorgu-4

select user\_id, status\_id, comment\_id, comment\_text

from f\_comment join f\_user

on f\_comment.profile\_id=f\_user.profile\_id

where user\_id=4001;

#################sorgu-5

select a.application\_name,max(a.ct)

from

(select application\_name,count(\*) as ct

from uses\_application join application

on uses\_application.application\_id=application.application\_id

group by application.application\_id) as a;

#################sorgu-6

select a.profil,album\_sayisi

from

(select f\_page.profile\_id as profil, count(\*) as album\_sayisi

from album join f\_page

on album.profile\_id=f\_page.profile\_id

group by f\_page.profile\_id) as a

where album\_sayisi>1;

#################sorgu-7

select gonderici.fname,gonderici.lname,message.message,alici.fname,alici.lname

from message

join f\_user as gonderici

on message.sender\_profile\_id=gonderici.profile\_id

join f\_user as alici

on message.receiver\_profile\_id=alici.profile\_id;

#################sorgu-8

select a.page\_id

from

(select page\_id,manages.community\_id as com\_id

from manages join f\_page

on f\_page.profile\_id=manages.profile\_id) as a

join attends

on (select event\_id from f\_event where community\_id=a.com\_id)=attends.event\_id

where attends.user\_id=4001;

#################sorgu-9

select a.isim name,max(kisi)

from

(select f\_event.event\_name as isim,count(\*) as kisi

from f\_user,attends,f\_event

where f\_user.user\_id=attends.user\_id and attends.event\_id=f\_event.event\_id and

extract(year from CURRENT\_TIMESTAMP)-extract(year from birth\_date) between 18 and 30

group by f\_event.event\_name) as a;

#################sorgu-10

select sokak,max(event\_sayisi)

from

(select location.street as sokak,count(\*) as event\_sayisi

from attends join f\_event

on attends.event\_id=f\_event.event\_id

join location

on f\_event.location\_id=location.location\_id

group by location.street) as a

//////////////////////////////////////////////////////////////////////////////////////////////////

DROP TRIGGER IF EXISTS ins\_like;

DELIMITER |

CREATE TRIGGER ins\_like AFTER INSERT ON f\_like

FOR EACH ROW BEGIN

update f\_status set like\_count=like\_count+1 where likeable\_object\_id=NEW.likeable\_object\_id;

update f\_page set like\_count=like\_count+1 where likeable\_object\_id=NEW.likeable\_object\_id;

update f\_comment set like\_count=like\_count+1 where likeable\_object\_id=NEW.likeable\_object\_id;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS del\_like;

DELIMITER |

CREATE TRIGGER del\_like AFTER DELETE ON f\_like

FOR EACH ROW BEGIN

update f\_status set like\_count=like\_count-1 where likeable\_object\_id=OLD.likeable\_object\_id;

update f\_page set like\_count=like\_count-1 where likeable\_object\_id=OLD.likeable\_object\_id;

update f\_comment set like\_count=like\_count-1 where likeable\_object\_id=OLD.likeable\_object\_id;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS ins\_friend;

DELIMITER |

CREATE TRIGGER ins\_friend AFTER INSERT ON friends\_with

FOR EACH ROW BEGIN

update f\_user set friend\_count=friend\_count+1 where f\_user.user\_id=new.user\_id;

update f\_user set friend\_count=friend\_count+1 where f\_user.user\_id=new.friends\_user\_id;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS del\_friend;

DELIMITER |

CREATE TRIGGER del\_friend AFTER DELETE ON friends\_with

FOR EACH ROW BEGIN

update f\_user set friend\_count=friend\_count-1 where f\_user.user\_id=old.user\_id;

update f\_user set friend\_count=friend\_count-1 where f\_user.user\_id=old.friends\_user\_id;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_likeable\_status;

DELIMITER |

CREATE TRIGGER auto\_likeable\_status BEFORE INSERT ON f\_status

FOR EACH ROW BEGIN

if new.likeable\_object\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='likeable\_object' AND TABLE\_SCHEMA=DATABASE() );

insert into likeable\_object values(null);

set NEW.likeable\_object\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_likeable\_page;

DELIMITER |

CREATE TRIGGER auto\_likeable\_page BEFORE INSERT ON f\_page

FOR EACH ROW BEGIN

if new.likeable\_object\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='likeable\_object' AND TABLE\_SCHEMA=DATABASE() );

insert into likeable\_object values(null);

set NEW.likeable\_object\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_likeable\_comment;

DELIMITER |

CREATE TRIGGER auto\_likeable\_comment BEFORE INSERT ON f\_comment

FOR EACH ROW BEGIN

if new.likeable\_object\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='likeable\_object' AND TABLE\_SCHEMA=DATABASE() );

insert into likeable\_object values(null);

set NEW.likeable\_object\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_community\_page;

DELIMITER |

CREATE TRIGGER auto\_community\_page BEFORE INSERT ON f\_page

FOR EACH ROW BEGIN

if new.community\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='community' AND TABLE\_SCHEMA=DATABASE() );

insert into community values(null);

set NEW.community\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_community\_group;

DELIMITER |

CREATE TRIGGER auto\_community\_group BEFORE INSERT ON f\_group

FOR EACH ROW BEGIN

if new.community\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='community' AND TABLE\_SCHEMA=DATABASE() );

insert into community values(null);

set NEW.community\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_community\_event;

DELIMITER |

CREATE TRIGGER auto\_community\_event BEFORE INSERT ON f\_event

FOR EACH ROW BEGIN

if new.community\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='community' AND TABLE\_SCHEMA=DATABASE() );

insert into community values(null);

set NEW.community\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_profile\_page;

DELIMITER |

CREATE TRIGGER auto\_profile\_page BEFORE INSERT ON f\_page

FOR EACH ROW BEGIN

if new.profile\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='f\_profile' AND TABLE\_SCHEMA=DATABASE() );

insert into f\_profile values(null);

set NEW.profile\_id=@auto\_id;

end if;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS auto\_profile\_user;

DELIMITER |

CREATE TRIGGER auto\_profile\_user BEFORE INSERT ON f\_user

FOR EACH ROW BEGIN

if new.profile\_id is null

then

set @auto\_id := (SELECT AUTO\_INCREMENT FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME='f\_profile' AND TABLE\_SCHEMA=DATABASE() );

insert into f\_profile values(null);

set NEW.profile\_id=@auto\_id;

end if;

END|

DELIMITER ;