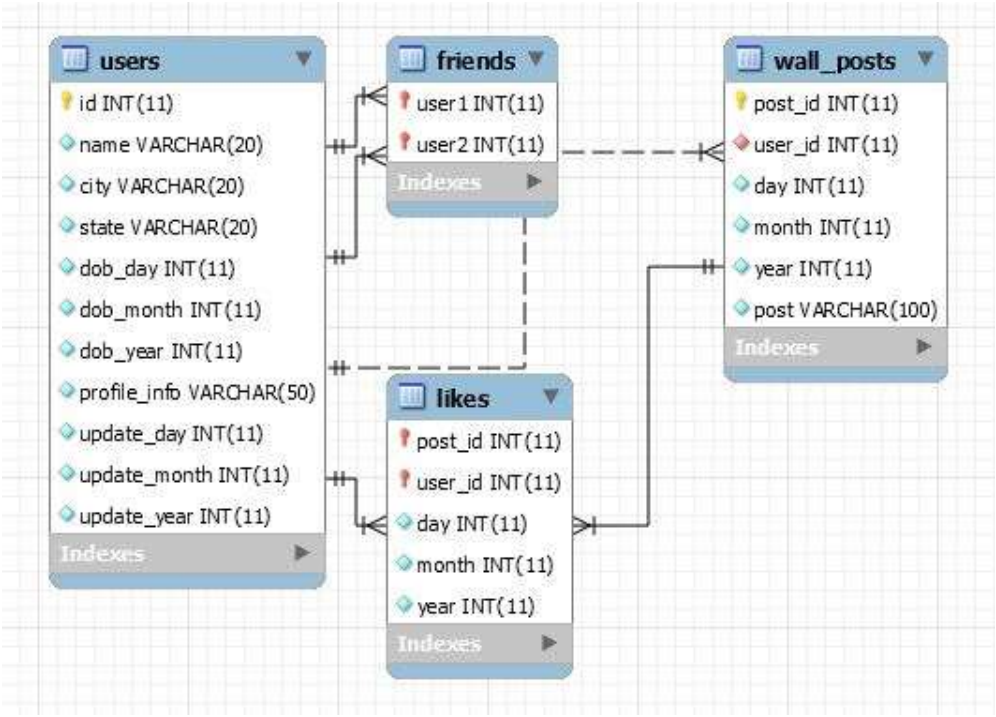
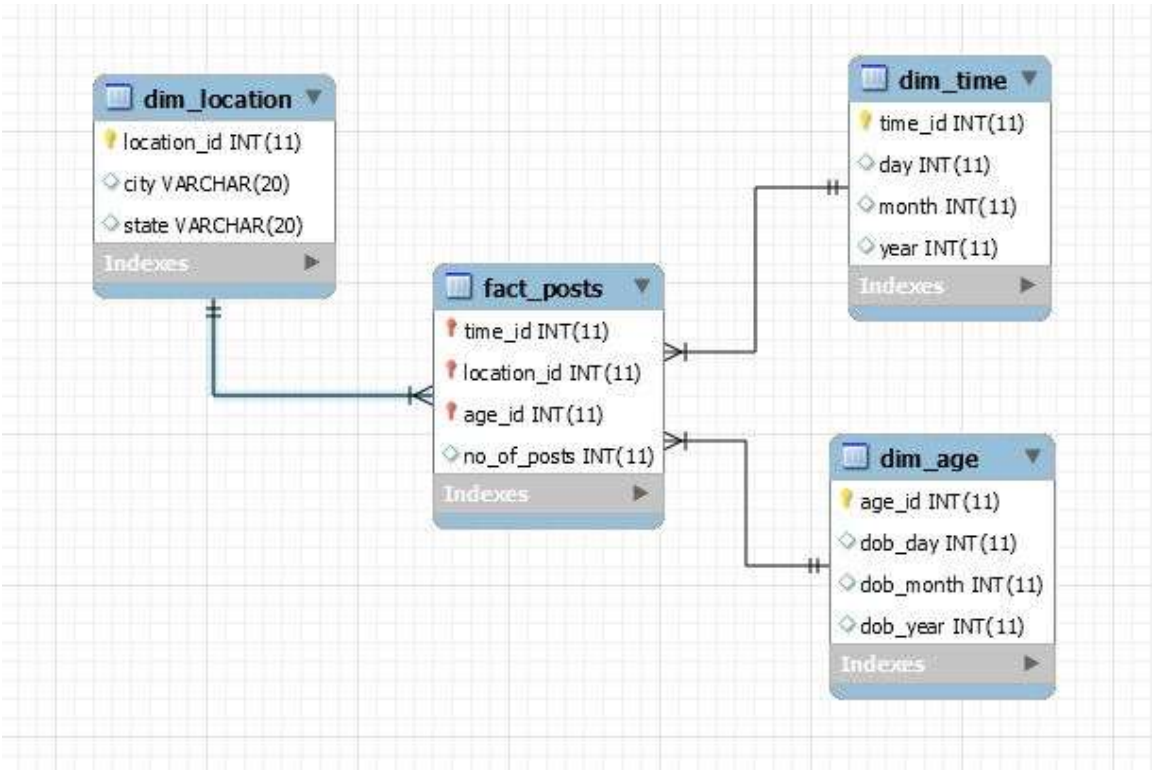
Q1. A.

The operational database is as follows:



Since the analysis is based on number of posts depending on time, location and user_age, we have designed the data warehouse schema as follows:



The dimension tables dim_time, dim_location, dim_age hold the unique values for time, location and age from the operational database. The fact table

fact_posts holds the references to these unique identifiers and the summary information (or measure) as the number of posts. Thus, analysis on the number of posts depending on the criteria such as time, location and user_age can be performed.

B. Following are the SQL queries to create and load the data warehouse from the operational database:

1. Create and Insert into DIM_TIME table queries:

```
create table dim_time
(
   time_id int primary key auto_increment,
   day int,
   month int,
   year int
);

insert into dim_time (day, month, year)
select day, month, year from wall_posts group by day, month, year;
```

2. Create and Insert into DIM_LOCATION table queries:

```
create table dim_location
(
   location_id int primary key auto_increment,
   city varchar(20),
   state varchar(20)
);

insert into dim_location (city, state)
select city, state from users group by city, state;
```

3. Create and Insert into DIM_AGE table queries:

```
create table dim_age
```

```
(
   age_id int primary key auto_increment,
   dob_day int,
   dob_month int,
   dob_year int
);

insert into dim_age (dob_day, dob_month, dob_year)
select dob_day, dob_month, dob_year from users group by dob_day, dob_month,
dob_year;
```

4. Create and Insert into FACT_POSTS table queries:

```
create table fact_posts
(
   time_id int,
   location_id int,
   age_id int,
   no_of_posts int,
   primary key (time_id, location_id, age_id),
   foreign key (time_id) references dim_time (time_id),
   foreign key (location_id) references dim_location (location_id),
   foreign key (age_id) references dim_age(age_id)
);

insert into fact_posts (time_id, location_id, age_id, no_of_posts)
select dt.time_id, dl.location_id, da.age_id, count(w.post_id)
        from wall_posts w, users u, dim_time dt, dim_location dl, dim_age da
where w.user_id = u.id and
   w.day = dt.day and
   w.month = dt.month and
   w.year = dt.year and
```

```
u.city = dl.city and
u.state = dl.state and
u.dob_day = da.dob_day and
u.dob_month = da.dob_month and
u.dob_year = da.dob_year
group by
dt.time_id, dl.location_id, da.age_id;
```

Q2) Consider Q1. Relation users is replicated between 4 servers (S1, S2, S3, S7) for dob_year >= 1970, and between 3 servers (S3, S6, S7) for dob_year < 1970

Consider the following three options for lock management: (1) majority protocol, (2) biased protocol and (3) quorum consensus protocol (choose sites' weights and Qr and Qw as you wish)

For each of the lock management options explain what locks are set and at which site for the following queries (assume that all queries are submitted at S5):

To answer this question, we know that 1. For majority protocol, if Q is replicated at n site, then a lock request message must be sent to more than half of the n sites. For biased protocol, when we do the select, each condition should be only considered once, when we do the update, each sites should be considered once. For quorum consensus protocol, we assume the total weight (S) of dob_year>=1970 is 13(1+2+3+7), and the weight of dob_year <1970 is 16(3+6+7)

(a) select dob_year, count(*) from users group by dob_year

Majority protocol: 3 shared locks from (S1, S2, S3, S7) and 2 shared locks from (S3, S6, S7)

Biased protocol: 1 shared lock from (S1, S2, S3, S7) and 1 shared lock from (S3, S6, S7)

Quorum-consensus protocol:

For the situation dob_year<1970:

2*Qw > 16

Qw>8

Because Qw=9 (suppose)

Qr+Qw>16, Qr> 7 => Therefore total site weight from which shared locks are needed should be >7

Possible combinations of shared locks from: S3+S7, S3+S6, S6+S7, or S3+S6+S7

For the situation dob_year>=1970:

2*Qw > 13

Qw>6.5;

Because Qw=7 (suppose)

Qr+Qw>13, Qr> 6 => Therefore total site weight from which shared locks are needed should be >6.

Possible combinations of shared locks from: S3+S7, S2+S7, S1+S7, S7, S1+S2+S7, S1+S3+S7, S2+S3+S7, or S1+S2+S3+S7

(b) select dob_year, count(*) from users where dob_year >= 1980 group by dob_year

Majority protocol: 3 shared locks from (S1, S2, S3, S7)

Biased protocol: 1 shared lock from (S1, S2, S3, S7)

Quorum-consensus protocol:

For the situation dob_year>=1970:

2*Qw > 13

Qw>6.5;

Because Qw=7 (suppose)

Qr+Qw>13, Qr> 6 => Therefore total site weight from which shared locks are needed should be >6

Possible combinations of shared locks from: S3+S7, S2+S7, S1+S7, S7, S1+S2+S7, S1+S3+S7, S2+S3+S7, or S1+S2+S3+S7

(c) update users set profile_info = “new information” where id = 1000

Majority protocol: 3 exclusive locks from (S1, S2, S3, S7) and 2 exclusive locks from (S3, S6, S7)

Biased protocol: 4 exclusive locks from (S1, S2, S3, S7) and 3 exclusive locks from (S3, S6, S7)

Quorum-consensus protocol:

For the condition dob_year<1970:

2*Qw > 16

Qw>8 => Therefore total site weight from which exclusive locks are needed should be > 8

Possible combinations of exclusive locks from: S3+S7, S6+S7, S6+S7, or S3+S6+S7

For the situation dob_year>=1970:

2*Qw > 13

Qw>6.5 => Therefore total site weight from which exclusive locks are needed should be > 6.5

Possible combinations of exclusive locks from: S3+S7, S2+S7, S1+S7, S7, S1+S2+S7, S1+S3+S7, S2+S3+S7, or S1+S2+S3+S7

(d) update users set update_year="2012" where dob_year >= 1969

Majority protocol: 3 exclusive locks from (S1, S2, S3, S7) and 2 exclusive locks from (S3, S6, S7)

Biased protocol: 4 exclusive locks from (S1, S2, S3, S7) and 3 exclusive locks from (S3, S6, S7)

Quorum-consensus protocol:

For the situation dob_year<1970:

2*Qw > 16

Qw>8 => Therefore total site weight from which exclusive locks are needed should be >8

Possible combinations of exclusive locks from: S3+S7, S6+S7, S6+S7, or S3+S6+S7

For the situation dob_year>=1970:

2*Qw > 13

Qw>6.5 => Therefore total site weight from which exclusive locks are needed should be >6.5

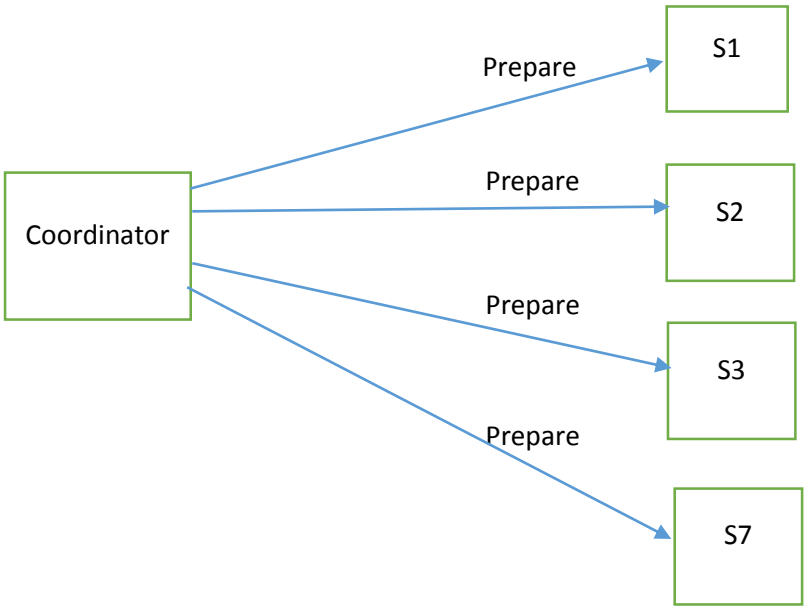
Possible combinations of exclusive locks from: S3+S7, S2+S7, S1+S7, S7, S1+S2+S7, S1+S3+S7, S2+S3+S7, or S1+S2+S3+S7

Q3) Consider the following query

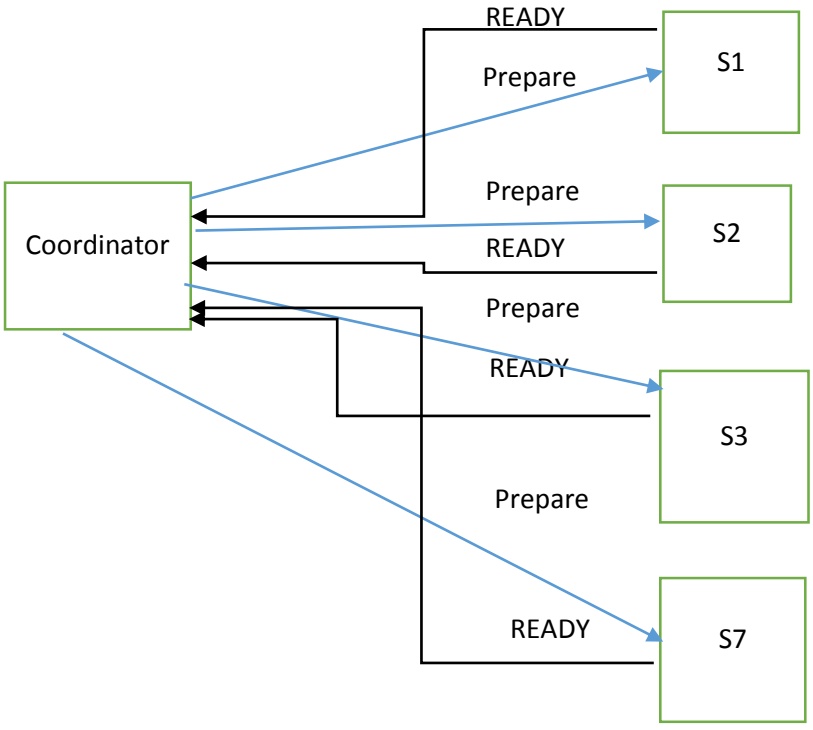
select dob_year, count(*) from users where dob_year >= 1980 group by dob_year

In this case, the data will be needed from the all the four servers S1, S2, S3 and S7. The four servers are the participants here.

The coordinator asks all the participants to prepare for the transaction.



The participants now prepare the data for the transaction. They enter the status as ‘ready’ in the log and sends the message to the coordinator.



Now after receiving the ready command from the participants, if the coordinator fails then the participants keep waiting for the coordinator to respond which will result in blocking condition. That is, the participants will continue to lock their resources until a “commit” or “abort” command is received from the co-ordinator.

Submitted by:

Vivekchander Sekaran (vic23@pitt.edu)

Chen Su (chs184@pitt.edu)

Shruti Sabusuresh (shs174@pitt.edu)