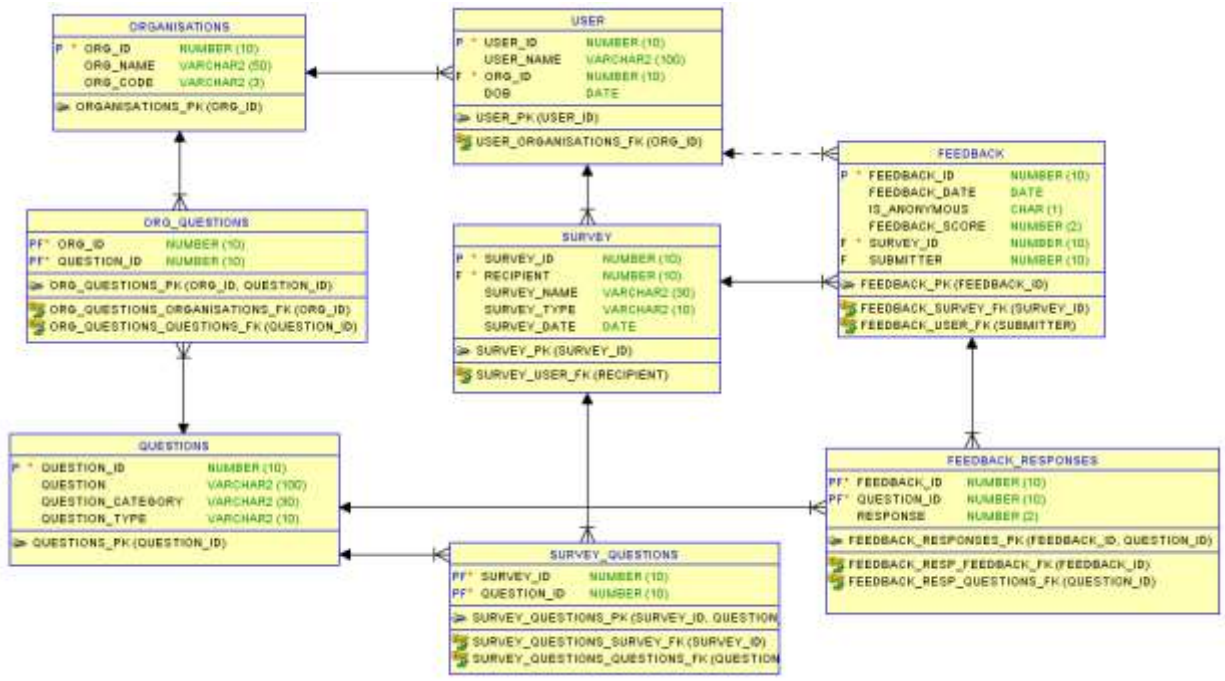
Data Model Design and Schema for 360-Degree Feedback Application

1. Data Model Schema and Design

a. ERD diagram showing all entities and relationships



Please click on the below link, if the above image is not proper/resizable

360_Degree_Feedback_DataModel_IMG

b. Description of Data Model

Entity: ORGANISATIONS

Organizations table contains different departments of a company, with ORG_ID as primary key

Entity: QUESTIONS

Questions contain survey questions across all departments with QUESTION_ID as primary key

Entity: ORG_QUESTIONS

ORG_QUESTIONS is an association table between **ORGANISATIONS** and **QUESTIONS,** to map many-many relationships between orgs and questions since each org can have multiple questions and each question can be associated with one or more orgs. (ORG_ID, QUESTION_ID) acts as primary key and both are foreign keys with respect to their associative tables.

Entity: USER

User is employee of any org and belongs to only one org. USER_ID is primary key and ORG_ID is foreign key.

Entity: SURVEY

Survey has all surveys identified by SURVEY_ID as primary key and has attributes like Survey name, date, type and recipient, conducting the survey.

Entity: SURVEY_QUESTIONS

Each survey has many questions and each question can be associated with one or more surveys. To map this association, SURVEY_QUESTIONS association table is used.

Entity: FEEDBACK

Feedback stores the feedback information for a particular survey including feedback score, submitter, whether the response is anonymous, feedback date etc. FEEDBACK_ID is the primary key and SURVEY_ID is the foreign key. Each feedback has many responses and hence there is one-many mapping between feedback and feedback responses

Entity: FEEDBACK_RESPONSES

Feedback responses is the association table between feedback questions and their responses where the primary key is (FEEDBACK_ID, QUESTION_ID). Questions table has one-may relationship with this table

c. Physical Schema.

```
create table feedback
  (
    feedback_id    number (10) not null ,
    feedback_date  date ,
    is_anonymous   char (1) ,
    feedback_score number (2) ,
    survey_id      number (10) not null ,
```

```
    submitter      number (10)
  ) ;
alter table feedback add constraint feedback_pk primary key ( feedback_id ) ;


create table feedback_responses
  (
    feedback_id number (10) not null ,
    question_id number (10) not null ,
    response    number (2)
  ) ;
alter table feedback_responses add constraint feedback_responses_pk primary key ( feedback_id, question_id ) ;


create table organisations
  (
    org_id   number (10) not null ,
    org_name varchar2 (50) ,
    org_code varchar2 (3)
  ) ;
alter table organisations add constraint organisations_pk primary key ( org_id ) ;


create table org_questions
  (
    org_id      number (10) not null ,
    question_id number (10) not null
  ) ;
alter table org_questions add constraint org_questions_pk primary key ( org_id, question_id ) ;


create table questions
  (
    question_id       number (10) not null ,
    question          varchar2 (100) ,
    question_category varchar2 (30) ,
    question_type     varchar2 (10)
  ) ;
alter table questions add constraint questions_pk primary key ( question_id ) ;


create table survey
  (
    survey_id   number (10) not null ,
    recipient   number (10) not null ,
    survey_name varchar2 (30) ,
    survey_type varchar2 (10) ,
    survey_date date
  ) ;
alter table survey add constraint survey_pk primary key ( survey_id ) ;


create table survey_questions
  (
    survey_id   number (10) not null ,
    question_id number (10) not null
  ) ;
alter table survey_questions add constraint survey_questions_pk primary key ( survey_id, question_id ) ;


create table "user"
  (
    user_id   number (10) not null ,
    user_name varchar2 (100) ,
    org_id    number (10) not null ,
    dob       date
  ) ;
alter table "user" add constraint user_pk primary key ( user_id ) ;
```

```
alter table feedback_responses add constraint feedback_resp_feedback_fk foreign key ( feedback_id ) references feedback ( feedback_id ) ;

alter table feedback_responses add constraint feedback_resp_questions_fk foreign key ( question_id ) references questions ( question_id ) ;

alter table feedback add constraint feedback_survey_fk foreign key ( survey_id ) references survey ( survey_id ) ;

alter table feedback add constraint feedback_user_fk foreign key ( submitter ) references "user" ( user_id ) ;

alter table org_questions add constraint org_questions_organisations_fk foreign key ( org_id ) references organisations ( org_id ) ;

alter table org_questions add constraint org_questions_questions_fk foreign key ( question_id ) references questions ( question_id ) ;

alter table survey_questions add constraint survey_questions_questions_fk foreign key ( question_id ) references questions ( question_id ) ;

alter table survey_questions add constraint survey_questions_survey_fk foreign key ( survey_id ) references survey ( survey_id ) ;

alter table survey add constraint survey_user_fk foreign key ( recipient ) references "user" ( user_id ) ;

alter table "user" add constraint user_organisations_fk foreign key ( org_id ) references organisations ( org_id ) ;
```

d. Insert scripts for model data

/*insert into ORG*/

```
delete from ORGANISATIONS ;
insert into ORGANISATIONS (ORG_ID, ORG_CODE, ORG_NAME) values (1, 'HR', 'Human Resources');
insert into ORGANISATIONS (ORG_ID, ORG_CODE, ORG_NAME) values (2, 'OR', 'Operations');
insert into ORGANISATIONS (ORG_ID, ORG_CODE, ORG_NAME) values (3, 'IT', 'Information Technology');
insert into ORGANISATIONS (ORG_ID, ORG_CODE, ORG_NAME) values (4, 'FI', 'Finance');
insert into ORGANISATIONS (ORG_ID, ORG_CODE, ORG_NAME) values (5, 'MR', 'Marketing');

/*insert into questions*/
delete from QUESTIONS ;
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (1, 'Rate HR Dept',
'FEEDBACK', 'Dept');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (2, 'Rate OR Dept',
'FEEDBACK', 'Dept');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (3, 'Rate FI Dept',
'FEEDBACK', 'Dept');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (4, 'Rate IT Dept',
'FEEDBACK', 'Dept');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (5, 'Rate MR Dept',
'FEEDBACK', 'Dept');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (6, 'Fun activities',
'SUGGESTIONS', 'GENERAL');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (7, 'Employee
engagement', 'SUGGESTIONS', 'GENERAL');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (8, 'Annual outings',
'SUGGESTIONS', 'GENERAL');
insert into QUESTIONS (QUESTION_ID, QUESTION,QUESTION_CATEGORY, QUESTION_TYPE) values (9, 'Workplace
improvement', 'SUGGESTIONS', 'GENERAL');

/*insert into ORG_QUESTIONS */
delete from ORG_QUESTIONS;
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (1, 1);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (1, 8);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (2, 2);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (3, 3);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (3, 7);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (3, 8);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (5, 6);
insert into ORG_QUESTIONS (ORG_ID, QUESTION_ID) values (5, 9);
```

```
/*insert into users*/
delete from "USER";
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (1, 'James', 1, '');
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (2, 'Wilson', 2, '');
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (3, 'Ryan', 3, to_date('01-19-2017', 'MM-DD-YYYY'));
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (4, 'Michael', 3, '');
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (5, 'Jared', 3, '');
insert into "USER" (USER_ID, USER_NAME, ORG_ID, DOB) values (6, 'Ramesh', 3, '');
/*

/* insert into survey_questions*/

insert into SURVEY (SURVEY_ID,SURVEY_NAME,SURVEY_TYPE,RECIPIENT, SURVEY_DATE) values (1,'Early Year
Survey','GENERAL', 3, to_date('01-05-2017', 'MM-DD-YYYY'));
insert into SURVEY (SURVEY_ID,SURVEY_NAME,SURVEY_TYPE,RECIPIENT, SURVEY_DATE) values (2,'Quarterly Year
Survey','QUARTERLY', 1, to_date('03-01-2017', 'MM-DD-YYYY'));
insert into SURVEY (SURVEY_ID,SURVEY_NAME,SURVEY_TYPE,RECIPIENT, SURVEY_DATE) values (3,'Mid Year
Survey','GENERAL', 3, to_date('06-01-2017', 'MM-DD-YYYY'));
insert into SURVEY (SURVEY_ID,SURVEY_NAME,SURVEY_TYPE,RECIPIENT, SURVEY_DATE) values (4,'End Year
Survey','GENERAL', 2, to_date('09-15-2017', 'MM-DD-YYYY'));

/*insert into survey_questions*/
delete from SURVEY_QUESTIONS;
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (1, 6);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (1, 7);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (1, 8);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (1, 9);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (1, 1);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (2, 1);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (2, 2);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (2, 3);
insert into SURVEY_QUESTIONS (SURVEY_ID, QUESTION_ID) values (2, 5);

/*insert into user_feedback*/

delete from FEEDBACK;
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (1, to_date('01-15-2017', 'MM-DD-YYYY'), 0, 8,1,1);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (2, to_date('01-17-2017', 'MM-DD-YYYY'), 0, 7, 1, 2);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (3, to_date('01-19-2017', 'MM-DD-YYYY'), 1, 10,1,null);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (4, to_date('01-23-2017', 'MM-DD-YYYY'), 0, 4,1,4);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (5, to_date('01-25-2017', 'MM-DD-YYYY'), 1, 9,1,null);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (6, to_date('03-01-2017', 'MM-DD-YYYY'), 0, 8,2,5);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (7, to_date('03-02-2017', 'MM-DD-YYYY'), 1, 5,2,null);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (8, to_date('03-03-2017', 'MM-DD-YYYY'), 0, 7,2,2);
insert into FEEDBACK (FEEDBACK_ID, FEEDBACK_DATE,IS_ANONYMOUS, FEEDBACK_SCORE, SURVEY_ID, SUBMITTER)
values (9, to_date('03-04-2017', 'MM-DD-YYYY'), 1, 9,2,null);



/*insert into feedback_responses*/

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (1, 6, 6);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (1, 7, 10);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (1, 8, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (1, 9, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (1, 1, 5);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (2, 6, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (2, 7, 7);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (2, 8, 4);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (2, 9, 10);
```

```
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (2, 1, 8);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (3, 6, 5);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (3, 7, 5);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (3, 8, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (3, 9, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (3, 1, 7);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (4, 6, 6);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (4, 7, 6);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (4, 8, 7);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (4, 1, 9);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (5, 6, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (5, 7, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (5, 8, 10);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (5, 9, 3);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (5, 1, 10);


insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (6, 1, 3);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (6, 2, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (6, 3, 7);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (6, 4, 6);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (7, 1, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (7, 2, 6);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (7, 3, 8);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (7, 4, 8);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (8, 2, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (8, 1, 10);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (8, 3, 6);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (8, 4, 8);

insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (9, 1, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (9, 2, 9);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (9, 3, 10);
insert into FEEDBACK_RESPONSES (FEEDBACK_ID, QUESTION_ID, RESPONSE) values (9, 4, 7);
```

2. QUERIES

a. Query to return feedback responses from a submitter

```
select s.survey_name, u1.user_name as recipient, u2.user_name as submitter,
q.question, fr.response, f.is_anonymous from survey s
join feedback f on s.survey_id = f.survey_id
join "user" u1 on u1.user_id = s.recipient
join feedback_responses fr on f.feedback_id = fr.feedback_id
join questions q on q.question_id = fr.question_id
left outer join "user" u2 on u2.user_id = f.submitter
where s.survey_id = 1 order by submitter;
```

Query Result

	SURVEY_NAME	RECIPIENT	SUBMITTER	QUESTION	RESPONSE	FEEDBACK_DATE	IS_ANONYMOUS
1	Early Year Survey	Ryan	Michael	Fun activities	6	23-JAN-17	0
2	Early Year Survey	Ryan	Michael	Employee engagement	6	23-JAN-17	0
3	Early Year Survey	Ryan	Michael	Annual outings	7	23-JAN-17	0
4	Early Year Survey	Ryan	Michael	Workplace improvement	9	23-JAN-17	0
5	Early Year Survey	Ryan	Michael	Rate HR Dept	9	23-JAN-17	0
6	Early Year Survey	Ryan	Wilson	Workplace improvement	10	17-JAN-17	0
7	Early Year Survey	Ryan	Wilson	Annual outings	4	17-JAN-17	0
8	Early Year Survey	Ryan	Wilson	Employee engagement	7	17-JAN-17	0
9	Early Year Survey	Ryan	Wilson	Fun activities	8	17-JAN-17	0
10	Early Year Survey	Ryan	Wilson	Rate HR Dept	8	17-JAN-17	0
11	Early Year Survey	Ryan	(null)	Annual outings	8	19-JAN-17	1
12	Early Year Survey	Ryan	(null)	Employee engagement	5	19-JAN-17	1
13	Early Year Survey	Ryan	(null)	Fun activities	5	19-JAN-17	1
14	Early Year Survey	Ryan	(null)	Workplace improvement	9	15-JAN-17	1
15	Early Year Survey	Ryan	(null)	Annual outings	8	15-JAN-17	1
16	Early Year Survey	Ryan	(null)	Employee engagement	10	15-JAN-17	1
17	Early Year Survey	Ryan	(null)	Fun activities	8	25-JAN-17	1
18	Early Year Survey	Ryan	(null)	Employee engagement	9	25-JAN-17	1
19	Early Year Survey	Ryan	(null)	Annual outings	10	25-JAN-17	1
20	Early Year Survey	Ryan	(null)	Workplace improvement	3	25-JAN-17	1
21	Early Year Survey	Ryan	(null)	Rate HR Dept	10	25-JAN-17	1
22	Early Year Survey	Ryan	(null)	Fun activities	6	15-JAN-17	1
23	Early Year Survey	Ryan	(null)	Rate HR Dept	7	19-JAN-17	1
24	Early Year Survey	Ryan	(null)	Rate HR Dept	5	15-JAN-17	1
25	Early Year Survey	Ryan	(null)	Workplace improvement	8	19-JAN-17	1

b. Query to return avg feedback score across a time period

```
select  s.SURVEY_NAME, AVG(f.FEEDBACK_SCORE) as AVG_FEEDBACK_SCORE
from SURVEY s
  join FEEDBACK f on s.SURVEY_ID = f.SURVEY_ID
  join "USER" u1 on u1.USER_ID = s.RECIPIENT
  join FEEDBACK_RESPONSES fr on f.FEEDBACK_ID = fr.FEEDBACK_ID
  join QUESTIONS q on q.QUESTION_ID = fr.QUESTION_ID
  LEFT OUTER join "USER" u2 on u2.USER_ID = f.SUBMITTER
  where s.SURVEY_DATE <= to_date('01-31-2017', 'MM-DD-YYYY') group by
s.SURVEY_NAME;
```

Query result

	SURVEY_NAME	AVG_FEEDBACK_SCORE
1	Early Year Survey	7.6

Query result across all surveys

	SURVEY_NAME	AVG_FEEDBACK_SCORE
1	Early Year Survey	7.6
2	Quarterly Year Survey	7.25

Assumptions:

1. Feedback table contains avg score across all responses for a particular submitter
2. It is normalized to be between 1 and 10

3. Algorithm to find average scores across all submitters and all feedback per survey request

```
double findAverageScore (List <FeedBack> feedBackList) {
   double totalScore = 0, avgScore = 0;
   for (FeedBack feedback : feedBackList) {
      double submitterScore = 0;
      for (Response response : feedback.getAllResponses()) {
         submitterScore += response.getScore();
      }
      totalScore += submitterScore;
   }
   int surveySize = feedBackList.getSize();
   avgScore = totalScore / surveySize;
   return avgScore;
}
```

Assumptions:

1. The datastructure is constructed as above where a list of all feedbacks for a particular survey are stored in the list
2. Each entry in the list has feedback responses for a single submitter.
3. Feedback is a wrapper where it contains list of submitter responses along with the score for each response
4. The score for every response is normalized to be between 1 and 10, since that is the range provided for feedback.

The above algorithm can be iterated for all surveys where each survey score can be normalized to be between 1 and 10, if each survey allows a different feedback scoring scale and avg normalized score can be calculated across all surveys for a particular year or years together.

Pseudocode

```
avgAcrossSurveys = 0;
totalNormalizedScore = 0;
For(survey in all Surveys):
        surveyScore = 0;
        Calculate score for each survey using previous survey score calculation algo;
        surveyScore = findAverageScore(Survey);
        Normalize the survey score to be between 1 and 10;
        surveyScore = normalized score;
        totalNormalizedScore = totalNormalizedScore+ surveyScore

avgAcrossAllSurveys = totalNormalizedScore/no of surveys
```