INFORMATION RETRIEVAL OF GOOGLE QUERIES OF POSITIVE AND NEGATIVE FEEDBACK

ABSTRACT

Numerous queries are asked and answered every single day. Inorder to keep a record of them we need to access the details of the users who post queries and the ones who rate the solutions or give a suggestion about them.The rating of the query depends on the exactness of the answer and queries with positive feedback comparatively have much higher rating than the queries with negative feedback.The rating and feedback of a particular query help the user to access the solution much quickly and effectively.

REQUIREMENTS

Table name	Attributes
Interrogators	i_id varchar2(10) i_name char(20)
Analyzers	a_id varchar2(10) a_name char(20)
Queries	q_id number(10) q_name varchar2(100)
Info_Retrieval	i_id varchar2(10) q_id number(10)
Rating	a_id varchar2(10) q_id number(10) feedback char(30)

INTEGRITY CONSTRAINTS

Attribute	Constraint
i_id	Primary key in Interrogators table Foreign key in Info_Retrieval table
a_id	Primary key in Analyzers table Foreign key in Info_Rating table
q_id	Primary key in Queries table Foreign key in Info_Retrieval table Foreign key in Rating table

MAPPING CARDINALITIES AND PARTICIPATION CONSTRAINTS

Many interrogators can post queries and many queries are accepted as well, hence the interrogators and queries will have a many to many relationship.

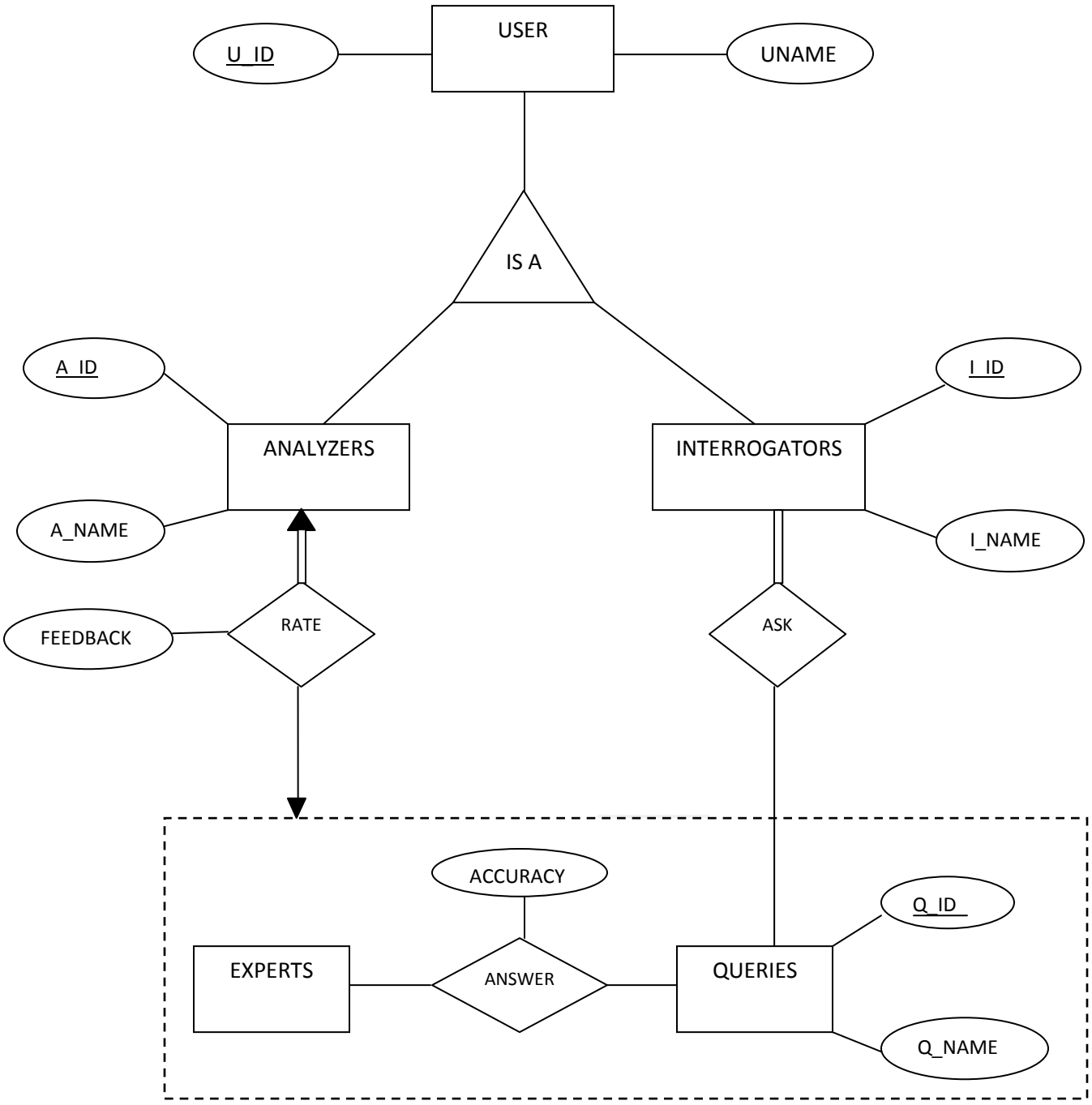
An analyzer can give only a single rating with either positive or negative feedback and the same applies to a query's solution- it can have only one rating from one user, hence they are involved in a one to one relationship.

Many experts can answer many queries, hence the experts and queries will have many to many relationship.

A user acts as an interrogator only when he/she wants to post queries, this implies total participation. The same applies for analyzers too.

All the queries may not be answered or given a rating, hence this implies partial participation.

ENTITY RELATIONSHIP DIAGRAM



DDL COMMANDS:

```
SQL> create table interrogators(
  2  i_id varchar2(10) primary key,
  3  i_name char(20));

Table created.

SQL> desc interrogators;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 I_ID                                      NOT NULL VARCHAR2(10)
 I_NAME                                             CHAR(20)

SQL> create table analyzers(
  2  a_id varchar2(10) primary key,
  3  a_name char(20));

Table created.

SQL> desc analyzers;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 A_ID                                      NOT NULL VARCHAR2(10)
 A_NAME                                             CHAR(20)

SQL> create table queries(
  2  q_id number(10) primary key,
  3  q_name varchar2(100));

Table created.

SQL> desc queries;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 Q_ID                                      NOT NULL NUMBER(10)
 Q_NAME                                             VARCHAR2(100)
```

```
SQL> create table info_retrieval(
  2  i_id varchar2(10),
  3  q_id number(10),
  4  foreign key(i_id) references interrogators(i_id),
  5  foreign key(q_id) references queries(q_id),
  6  primary key(i_id,q_id));

Table created.

SQL> desc info_retrieval;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 I_ID                                      NOT NULL VARCHAR2(10)
 Q_ID                                      NOT NULL NUMBER(10)

SQL> create table rating(
  2  a_id varchar2(10),
  3  feedback char(30),
  4  q_id number(10),
  5  foreign key(q_id) references queries(q_id),
  6  foreign key(a_id) references analyzers(a_id),
  7  primary key(a_id,q_id));

Table created.

SQL> desc rating;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 A_ID                                      NOT NULL VARCHAR2(10)
 FEEDBACK                                           CHAR(30)
 Q_ID                                      NOT NULL NUMBER(10)
```

DML COMMANDS:

```
SQL> insert into interrogators values(&i_id,'&i_name');
Enter value for i_id: 43
Enter value for i_name: piyush
old   1: insert into interrogators values(&i_id,'&i_name')
new   1: insert into interrogators values(43,'piyush')

1 row created.

SQL> /
Enter value for i_id: 31
Enter value for i_name: mohak
old   1: insert into interrogators values(&i_id,'&i_name')
new   1: insert into interrogators values(31,'mohak')

1 row created.

SQL> /
Enter value for i_id: 86
Enter value for i_name: khushi
old   1: insert into interrogators values(&i_id,'&i_name')
new   1: insert into interrogators values(86,'khushi')

1 row created.

SQL> /
Enter value for i_id: 60
Enter value for i_name: farhan
old   1: insert into interrogators values(&i_id,'&i_name')
new   1: insert into interrogators values(60,'farhan')

1 row created.

SQL> /
Enter value for i_id: 15
Enter value for i_name: devika
old   1: insert into interrogators values(&i_id,'&i_name')
new   1: insert into interrogators values(15,'devika')

1 row created.
```

```
SQL> select * from interrogators;

I_ID       I_NAME
---------- --------------------
43         piyush
31         mohak
86         khushi
60         farhan
15         devika
```

```
SQL> insert into analyzers values(10,'rohan');

1 row created.

SQL> insert into analyzers values(20,'eshan');

1 row created.

SQL> insert into analyzers values(30,'manvi');

1 row created.

SQL> insert into analyzers values(89,'kahani');

1 row created.

SQL> insert into analyzers values(45,'krish');

1 row created.

SQL> select * from analyzers;

A_ID       A_NAME
---------- --------------------
10         rohan
20         eshan
30         manvi
89         kahani
45         krish
```

```
SQL> select * from queries;

      Q_ID
----------
Q_NAME
--------------------------------------------------
       105
er_diagram

        47
is_a_hierachy

        35
integrity_constraints


      Q_ID
----------
Q_NAME
--------------------------------------------------
        29
relational algebra

        70
sql queries
```

```
SQL> insert into info_retrieval values(43,105);

1 row created.

SQL> insert into info_retrieval values(31,47);

1 row created.

SQL> insert into info_retrieval values(86,35);

1 row created.

SQL> insert into info_retrieval values(60,29);

1 row created.

SQL> insert into info_retrieval values(15,70);

1 row created.

SQL> select * from info_retrieval;

I_ID             Q_ID
---------- ----------
43                105
31                 47
86                 35
60                 29
15                 70
```

```
SQL> insert into rating values(10,'positive',105);

1 row created.

SQL> insert into rating values(20,'negative',47);

1 row created.

SQL> insert into rating values(30,'positive',35);

1 row created.

SQL> insert into rating values(89,'positive',29);

1 row created.

SQL> insert into rating values(45,'positive',70);

1 row created.

SQL> select * from rating;

A_ID       FEEDBACK                          Q_ID
---------- ------------------------------ ----------
10         positive                          105
20         negative                           47
30         positive                           35
89         positive                           29
45         positive                           70
```