Blog Application

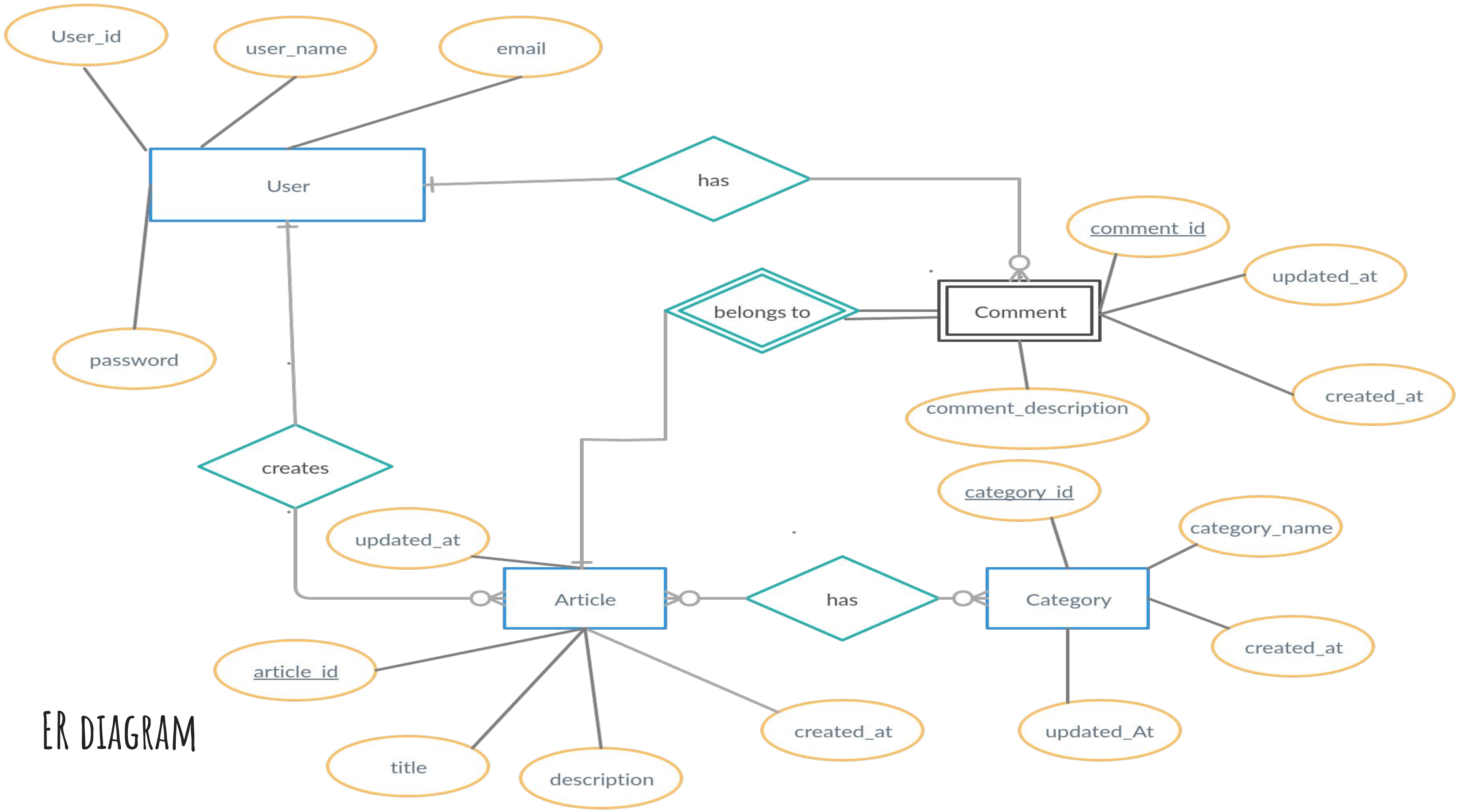
Made by: Hardik Upadhyay
Roll No: U18CO072

Project description/Aim:

To implement a blog application using DBMS concepts in which users can perform CRUD(create-read-update-delete) operations on articles. Also, they can classify articles based on their category as well as add comments to articles in the blog.

Constraints/Points to be noted

1. An article is created by one user but one user can have many articles.
2. An article can have many categories. Each category can have many articles.
3. Comments belong to an article and are written by a user. They are weak entity with complete dependency on article.


User_id
user_name
email
User
has
password
comment_id
updated_at
belongs to
Comment
created_at
comment_description
creates
category_id
category_name
updated_at
Article
has
Category
created_at
article_id
created_at
updated_At
title
description

ER DIAGRAM

Conversion of ER-diagram to relational model :

- Comment is a weak entity. Article_id which is foreign key from article table together with comment_id will form a candidate key. There is one-to-many relation between comment and user so we will keep commentor_id as a foreign key in comment table.

  Comment(comment_id, article_id, commentor_id, comment_description, created_at, updated_at)

Conversion of er to relational model(continued)

- “Each article is created by one user and one user can have many articles”. This is one-to-many relationship. We will keep creator_id as a foreign key in articles table.

Article(article_id, title, description, creator_id, created_at, updated_at)

User(user_id, username, email, password)

Conversion of ER to relational model(continued)

- Articles and categories have many to many relation between them. Hence we the table article_categories will indicate relation between article and categories.

Category(<u>category_id</u>, category_name, created_at, updated_at)

Article_Category(article_id, category_id)

Listing all the tables again :

- Article(article_id, title,description, created_at, updated_at, creator_id)
- User(user_id, username, email, password)
- Comment(comment_id, article_id, commentor_id, comment_description, created_at, updated_at)
- Category(category_id, category_name, created_at, updated_at)
- Article_Category(article_id, category_id)

NORMALISATION

Listing all the non-trivial dependencies.

For Article table:

- article_id -> title, description, user_id, created_at, updated_at

For User table:

- User_id -> user_name, email, password
- Email -> user_name, user_id, password

Normalisation

For Comment table:

- comment_id, article_id -> comment_description
- comment_id, article_id -> commentor_id
- Comment_id, article_id -> created_at, updated_at

For Category table:

- category_id -> category_name, created_at, updated_at

Check for first normal form

- As all the attributes are single-valued, the table is in first normal form.

Check for second normal form

- There are no partial dependencies in any table.

  So the table is already in second normal form.

Check for Third normal form

2 conditions for third normal form are:

- LHS should be superkey or
- RHS should be prime attribute

As all the keys on the LHS of all the FDS are super keys, there are no transitive dependencies.

So the table is already in third normal form.

BCNF FORM(3.5 NORMAL FORM)

- The table is already in 3NF. As LHS of all the dependencies are candidate keys of that table so the tables are already in 3.5NF or BCNF form.

Fourth normal form

- The table has no multivalued dependency. While handling the many-to-many case of article and category for conversion of er-diagram to relational-model we have already solved this problem
- Hence the table is already in 4NF.

Fifth normal form

- Any table cannot be decomposed further through lossless decomposition.
- Hence the table is already in fifth normal form.

Create Tables

Users:

```
create table users (
          user_id                          number generated by default on null as identity
                                          constraint users_id_pk primary key,
          username                         varchar2(20),
          email                            varchar2(255),
          password                         varchar2(20) invisible
);
```

Create tables

Articles:

```
create table articles (
    article_id                     number generated by default on null as identity
                                   constraint articles_id_pk primary key,
    title                          varchar2(20),
    description                    varchar2(200),
    created_at                     date default on null SYSDATE,
    updated_at                     date default on null SYSDATE,
    creator_id                     number
                                   constraint articles_creator_id_fk
                                   references users on delete cascade
)
;
```

Create tables

Categories

```
create table categories (
    category_id                    number generated by default on null as identity
                                   constraint categories_id_pk primary key,
    category_name                  varchar2(20),
    created_at                     date default on null SYSDATE,
    updated_at                     date default on null SYSDATE
)
;
```

Article_Categories

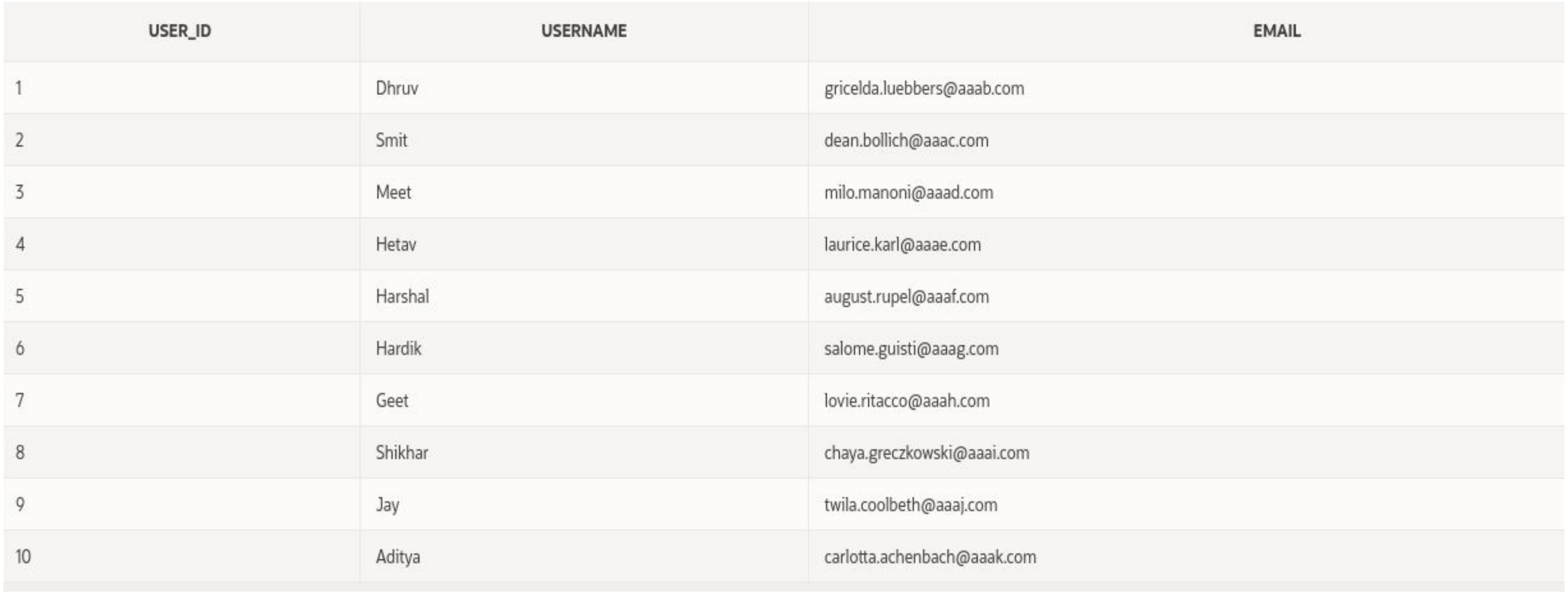
```
create table article_category(
    article_category_id  number primary key,
   article_id number references articles on delete cascade,
    category_id number references categories on delete cascade
)
;
```

Create Tables

Comments

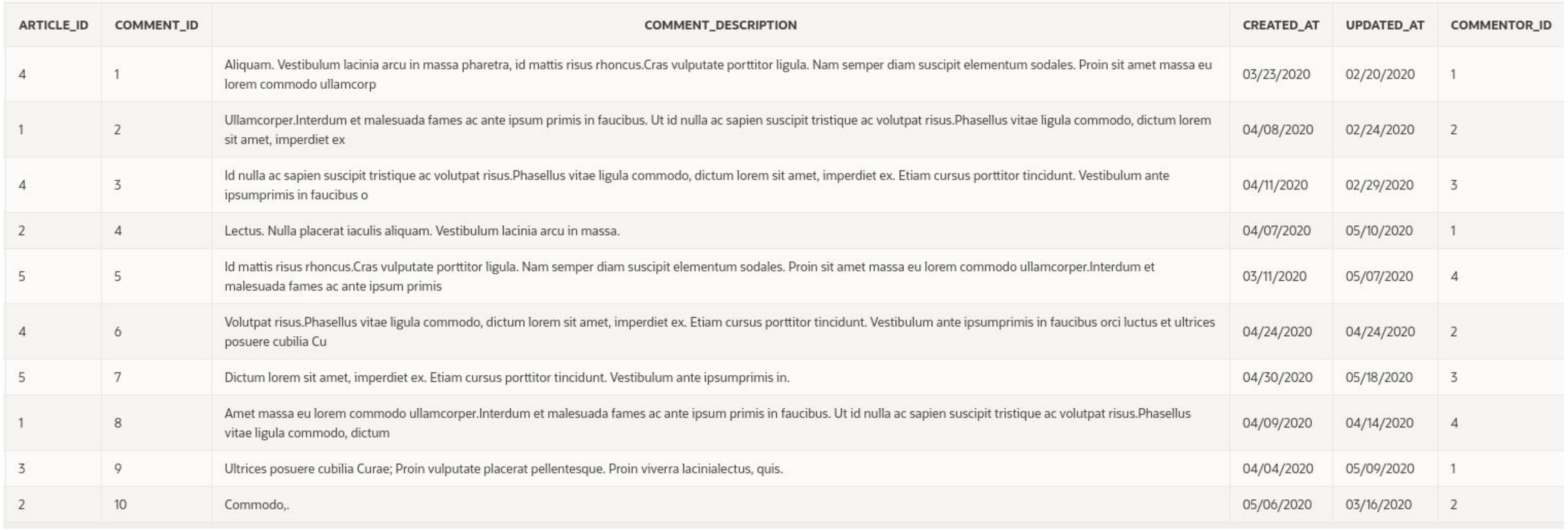
```
create table comments (
    comment_id                      number,
    article_id                      number references articles,
    comment_description             varchar2(200),
    created_at                      date,
    updated_at                      date,
    commentor_id                    number
                                    constraint comments_commentor_id_fk
                                    references users on delete cascade,
    primary key(comment_id, article_id)
)
;
```

Users Table



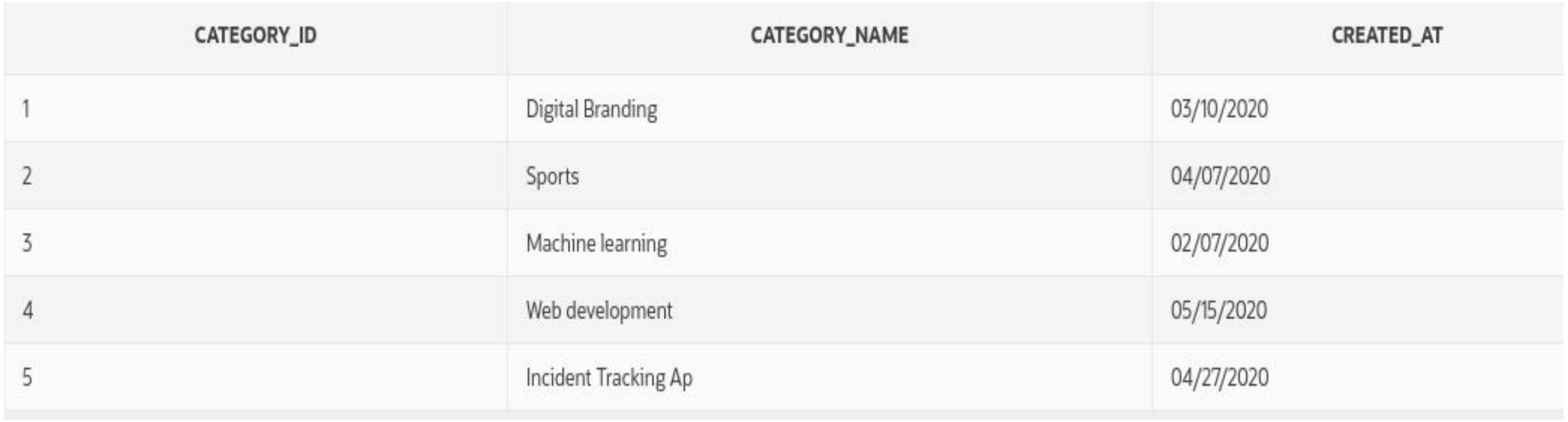
USER_ID	USERNAME	EMAIL
1	Dhruv	gricelda.luebbers@aaab.com
2	Smit	dean.bollich@aaac.com
3	Meet	milo.manoni@aaad.com
4	Hetav	laurice.karl@aaae.com
5	Harshal	august.rupel@aaaf.com
6	Hardik	salome.guisti@aaag.com
7	Geet	lovie.ritacco@aaah.com
8	Shikhar	chaya.greczkowski@aaai.com
9	Jay	twila.coolbeth@aaaj.com
10	Aditya	carlotta.achenbach@aaak.com

Comments Table



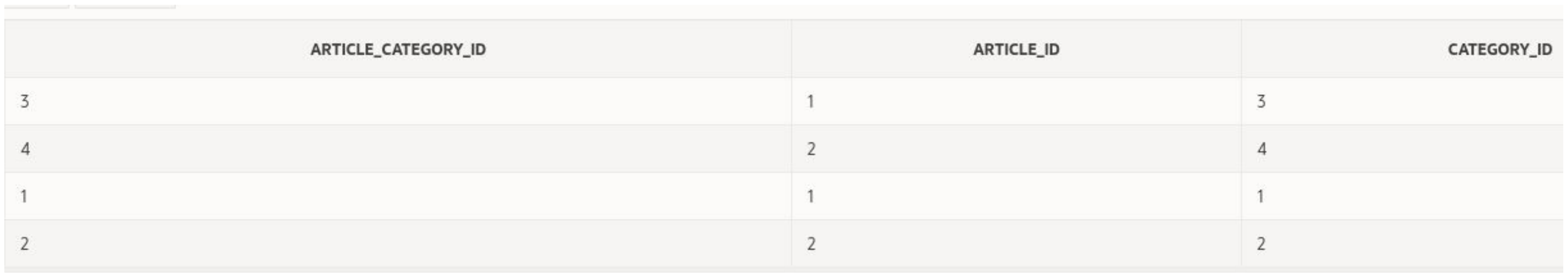
ARTICLE_ID	COMMENT_ID	COMMENT_DESCRIPTION	CREATED_AT	UPDATED_AT	COMMENTOR_ID
4	1	Aliquam. Vestibulum lacinia arcu in massa pharetra, id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorp	03/23/2020	02/20/2020	1
1	2	Ullamcorper.Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex	04/08/2020	02/24/2020	2
4	3	Id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus o	04/11/2020	02/29/2020	3
2	4	Lectus. Nulla placerat iaculis aliquam. Vestibulum lacinia arcu in massa.	04/07/2020	05/10/2020	1
5	5	Id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis	03/11/2020	05/07/2020	4
4	6	Volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus orci luctus et ultrices posuere cubilia Cu	04/24/2020	04/24/2020	2
5	7	Dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in.	04/30/2020	05/18/2020	3
1	8	Amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum	04/09/2020	04/14/2020	4
3	9	Ultrices posuere cubilia Curae; Proin vulputate placerat pellentesque. Proin viverra lacinialectus, quis.	04/04/2020	05/09/2020	1
2	10	Commodo,.	05/06/2020	03/16/2020	2

Categories Table



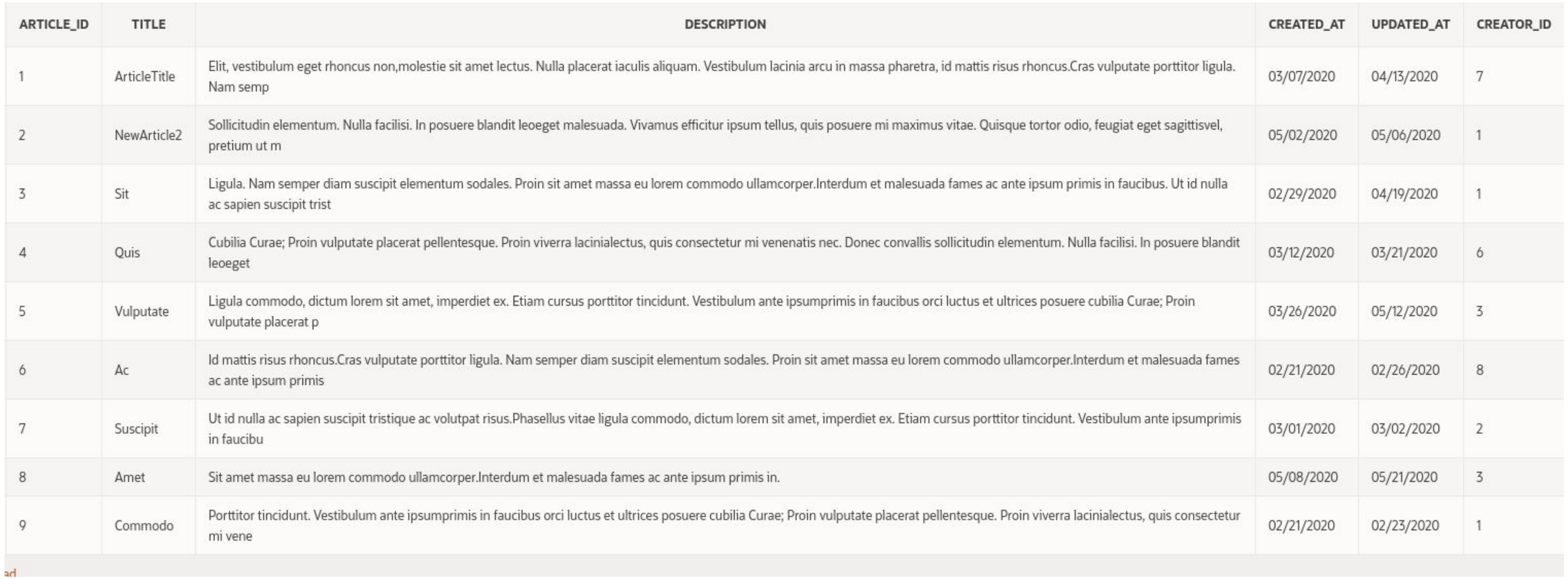
CATEGORY_ID	CATEGORY_NAME	CREATED_AT
1	Digital Branding	03/10/2020
2	Sports	04/07/2020
3	Machine learning	02/07/2020
4	Web development	05/15/2020
5	Incident Tracking Ap	04/27/2020

Article-category table



ARTICLE_CATEGORY_ID	ARTICLE_ID	CATEGORY_ID
3	1	3
4	2	4
1	1	1
2	2	2

Articles table



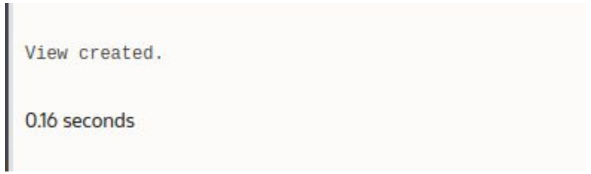
ARTICLE_ID	TITLE	DESCRIPTION	CREATED_AT	UPDATED_AT	CREATOR_ID
1	ArticleTitle	Elit, vestibulum eget rhoncus non,molestie sit amet lectus. Nulla placerat iaculis aliquam. Vestibulum lacinia arcu in massa pharetra, id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semp	03/07/2020	04/13/2020	7
2	NewArticle2	Sollicitudin elementum. Nulla facilisi. In posuere blandit leoeget malesuada. Vivamus efficitur ipsum tellus, quis posuere mi maximus vitae. Quisque tortor odio, feugiat eget sagittisvel, pretium ut m	05/02/2020	05/06/2020	1
3	Sit	Ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut id nulla ac sapien suscipit trist	02/29/2020	04/19/2020	1
4	Quis	Cubilia Curae; Proin vulputate placerat pellentesque. Proin viverra lacinialectus, quis consectetur mi venenatis nec. Donec convallis sollicitudin elementum. Nulla facilisi. In posuere blandit leoeget	03/12/2020	03/21/2020	6
5	Vulputate	Ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin vulputate placerat p	03/26/2020	05/12/2020	3
6	Ac	Id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis	02/21/2020	02/26/2020	8
7	Suscipit	Ut id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibu	03/01/2020	03/02/2020	2
8	Amet	Sit amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis in.	05/08/2020	05/21/2020	3
9	Commodo	Porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin vulputate placerat pellentesque. Proin viverra lacinialectus, quis consectetur mi vene	02/21/2020	02/23/2020	1

ad

Adding Views to Tables

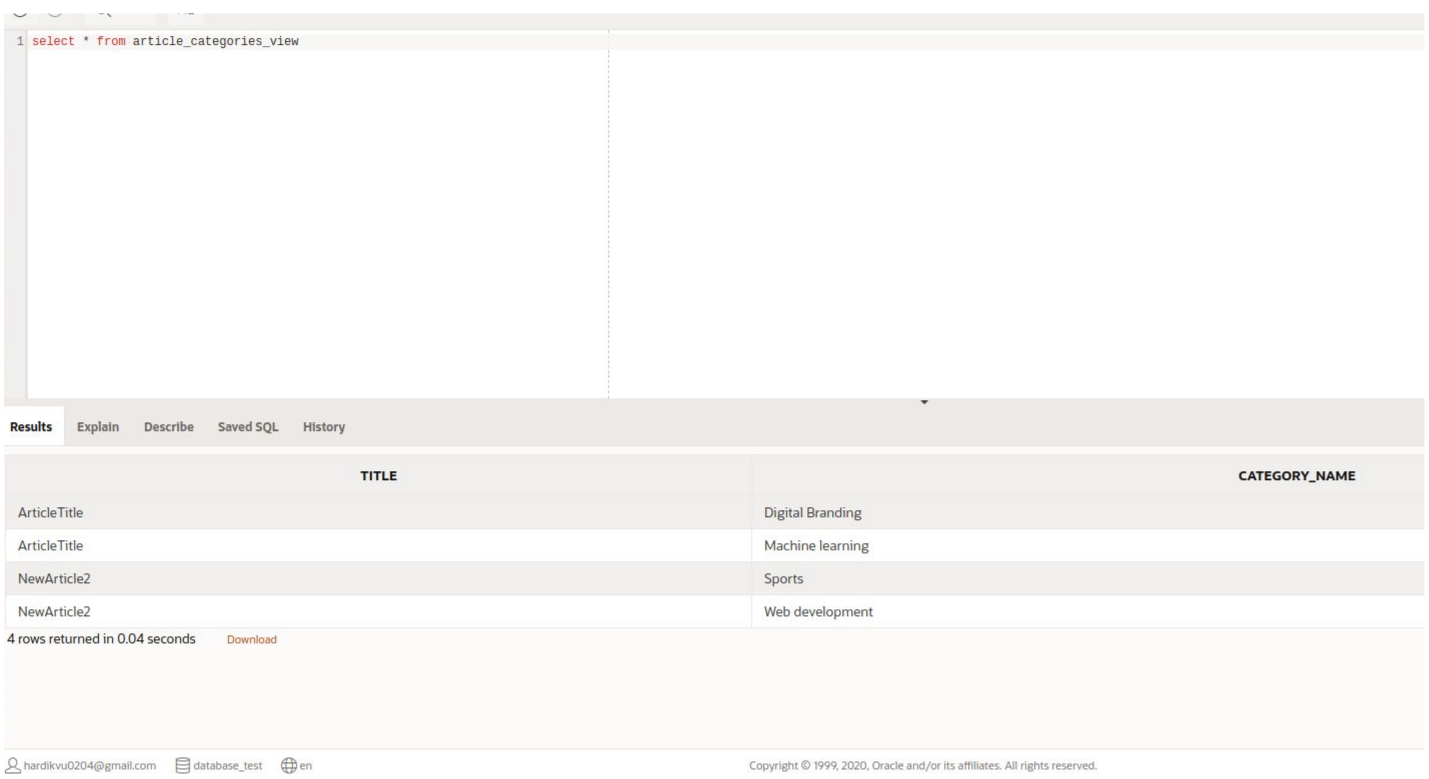
Article_categories_view: Many a times it is necessary to understand to which categories an article belongs, so this view will provide this detail.

```
create or replace view article_categories_view as
   select title, category_name
   from articles, categories,article_category
   where articles.article_id = article_category.article_id
   and categories.category_id = article_category.category_id
   order by titie;
```



```
View created.

0.16 seconds
```

Adding Views to tables



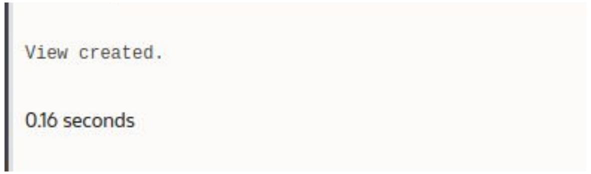
```
select * from article_categories_view
```

TITLE	CATEGORY_NAME
ArticleTitle	Digital Branding
ArticleTitle	Machine learning
NewArticle2	Sports
NewArticle2	Web development

4 rows returned in 0.04 seconds

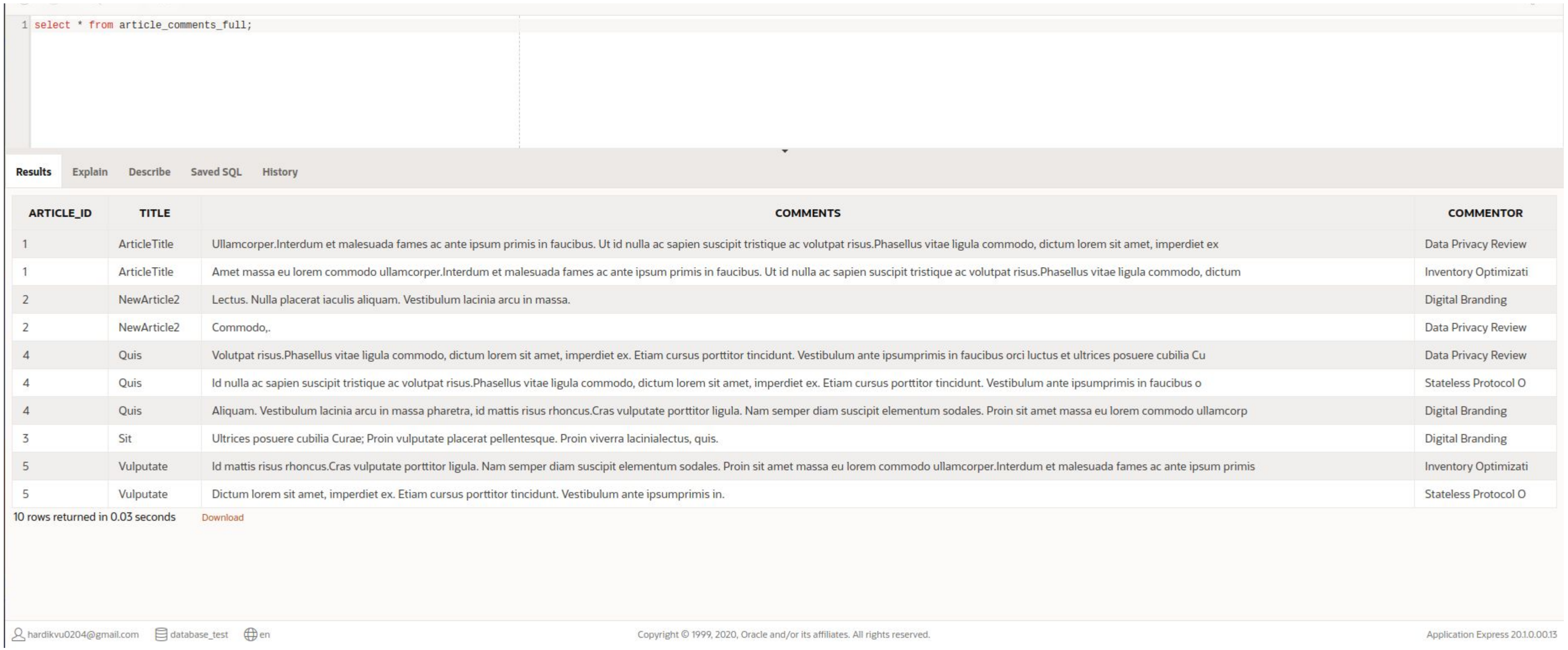
Adding Views to Tables

```
create or replace view article_comments_full as
    select articles.article_id, title, comment_description as comments, username as commentor
    from articles, comments, users
    where comments.article_id = articles.article_id
    and comments.commentor_id = users.user_id
    order by articles.title;
```



```
View created.

0.16 seconds
```

Adding views to Tables



```
select * from article_comments_full;
```

ARTICLE_ID	TITLE	COMMENTS	COMMENTOR
1	ArticleTitle	Ullamcorper.Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex	Data Privacy Review
1	ArticleTitle	Amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis in faucibus. Ut id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum	Inventory Optimizati
2	NewArticle2	Lectus. Nulla placerat iaculis aliquam. Vestibulum lacinia arcu in massa.	Digital Branding
2	NewArticle2	Commodo,.	Data Privacy Review
4	Quis	Volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus orci luctus et ultrices posuere cubilia Cu	Data Privacy Review
4	Quis	Id nulla ac sapien suscipit tristique ac volutpat risus.Phasellus vitae ligula commodo, dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in faucibus o	Stateless Protocol O
4	Quis	Aliquam. Vestibulum lacinia arcu in massa pharetra, id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorp	Digital Branding
3	Sit	Ultrices posuere cubilia Curae; Proin vulputate placerat pellentesque. Proin viverra lacinialectus, quis.	Digital Branding
5	Vulputate	Id mattis risus rhoncus.Cras vulputate porttitor ligula. Nam semper diam suscipit elementum sodales. Proin sit amet massa eu lorem commodo ullamcorper.Interdum et malesuada fames ac ante ipsum primis	Inventory Optimizati
5	Vulputate	Dictum lorem sit amet, imperdiet ex. Etiam cursus porttitor tincidunt. Vestibulum ante ipsumprimis in.	Stateless Protocol O

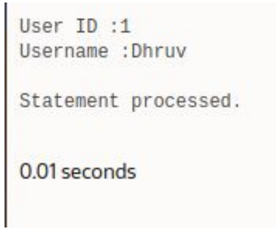
10 rows returned in 0.03 seconds   Download

hardikvu0204@gmail.com   database_test   en   Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.   Application Express 20.1.0.00.13

Cursor

Cursor to select article id 1 : This cursor fetches the first user with id = 1 and prints it else does nothing.

```
declare
cursor c is select * from users;
begin
for user in c loop
if user.user_id = 1 then
dbms_output.put_line('User ID :' || user.user_id);
dbms_output.put_line('Username :' || user.username);
end if;
end loop;
end;
```

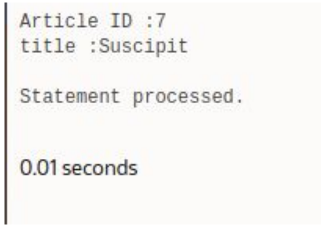


```
User ID :1
Username :Dhruv

Statement processed.


0.01 seconds
```

Cursor to get the article id and title of article created

by particular user.

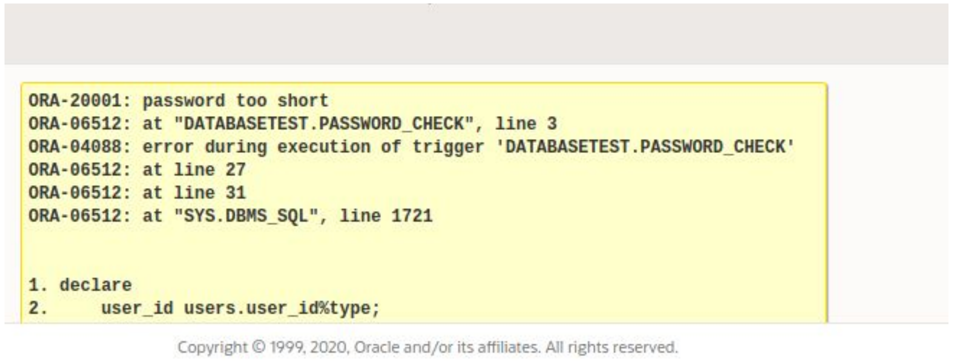
```
declare
cursor c is select * from articles;
u_id users.user_id%type;
begin
u_id := :u_id;
for article in c loop
if article.creator_id = u_id then
dbms_output.put_line('Article ID :' || article.article_id);
dbms_output.put_line('title :' || article.title);
end if;
end loop;
End;
```



```
Article ID :7
title :Suscipit

Statement processed.


0.01 seconds
```

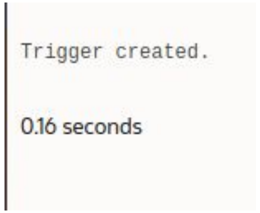
Trigger:

```
Trigger to check if password length is < 3.
create or replace trigger password_check
before insert or update on users
for each row
begin
    if length(:NEW.password) < 3 then
     RAISE_APPLICATION_ERROR (-20001,'password too short');
    end if;
end;
```

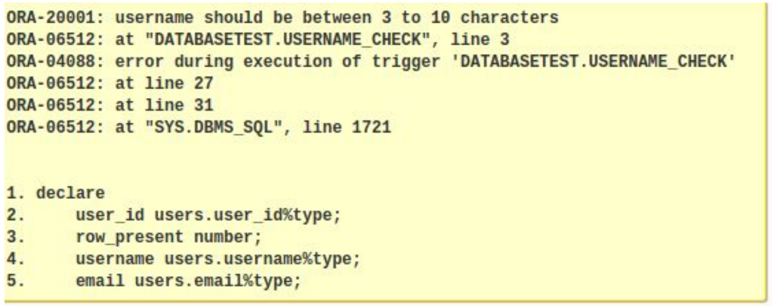


```
ORA-20001: password too short
ORA-06512: at "DATABASETEST.PASSWORD_CHECK", line 3
ORA-04088: error during execution of trigger 'DATABASETEST.PASSWORD_CHECK'
ORA-06512: at line 27
ORA-06512: at line 31
ORA-06512: at "SYS.DBMS_SQL", line 1721


1. declare
2.     user_id users.user_id%type;
```

```
Trigger to check if username is between 3 to 10 characters.
create or replace trigger username_check
before insert or update on users
for each row
begin
    if length(:NEW.username) > 10 or length(:NEW.username) < 3 then
     RAISE_APPLICATION_ERROR (-20001,'username should be between 3 to 10 characters');
    end if;
end;
```



```
Trigger created.

0.16 seconds
```

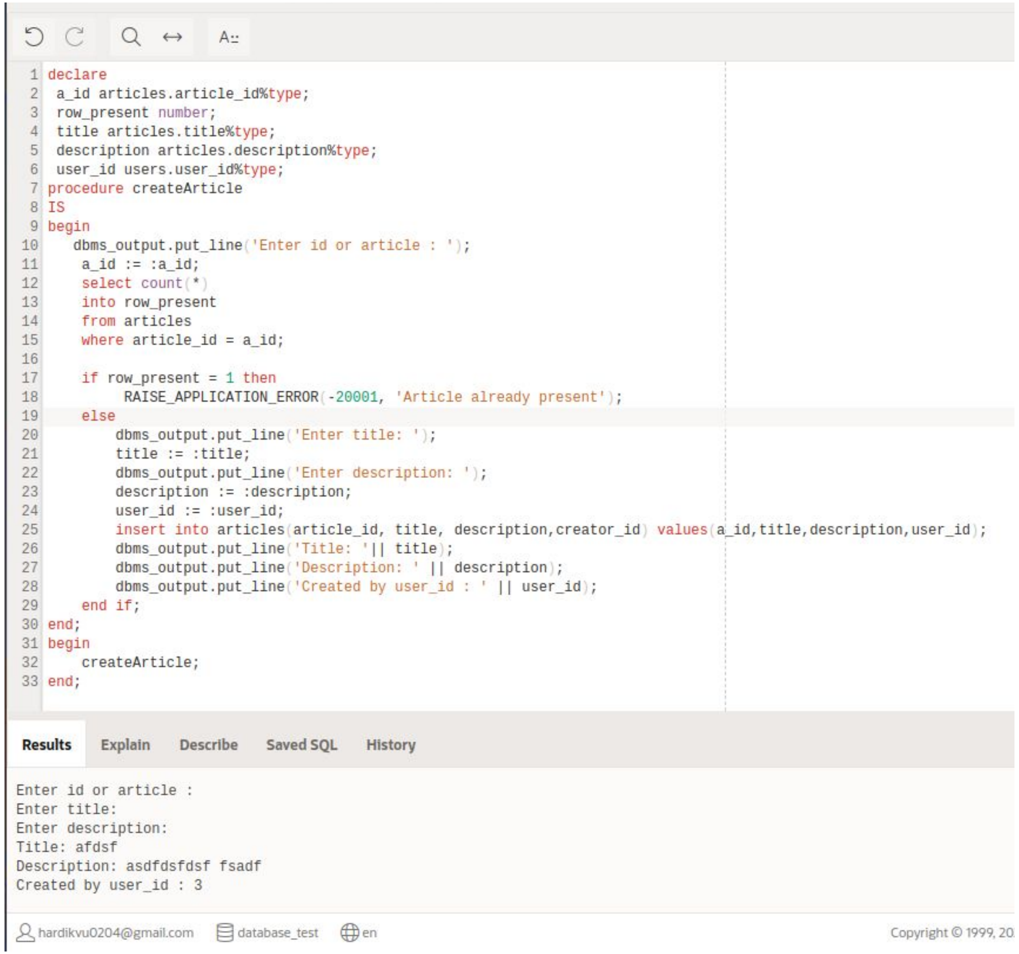


```
ORA-20001: username should be between 3 to 10 characters
ORA-06512: at "DATABASETEST.USERNAME_CHECK", line 3
ORA-04088: error during execution of trigger 'DATABASETEST.USERNAME_CHECK'
ORA-06512: at line 27
ORA-06512: at line 31
ORA-06512: at "SYS.DBMS_SQL", line 1721


1. declare
2.     user_id users.user_id%type;
3.     row_present number;
4.     username users.username%type;
5.     email users.email%type;
```

Procedures:

A procedure to create an article by taking input

From user



A:

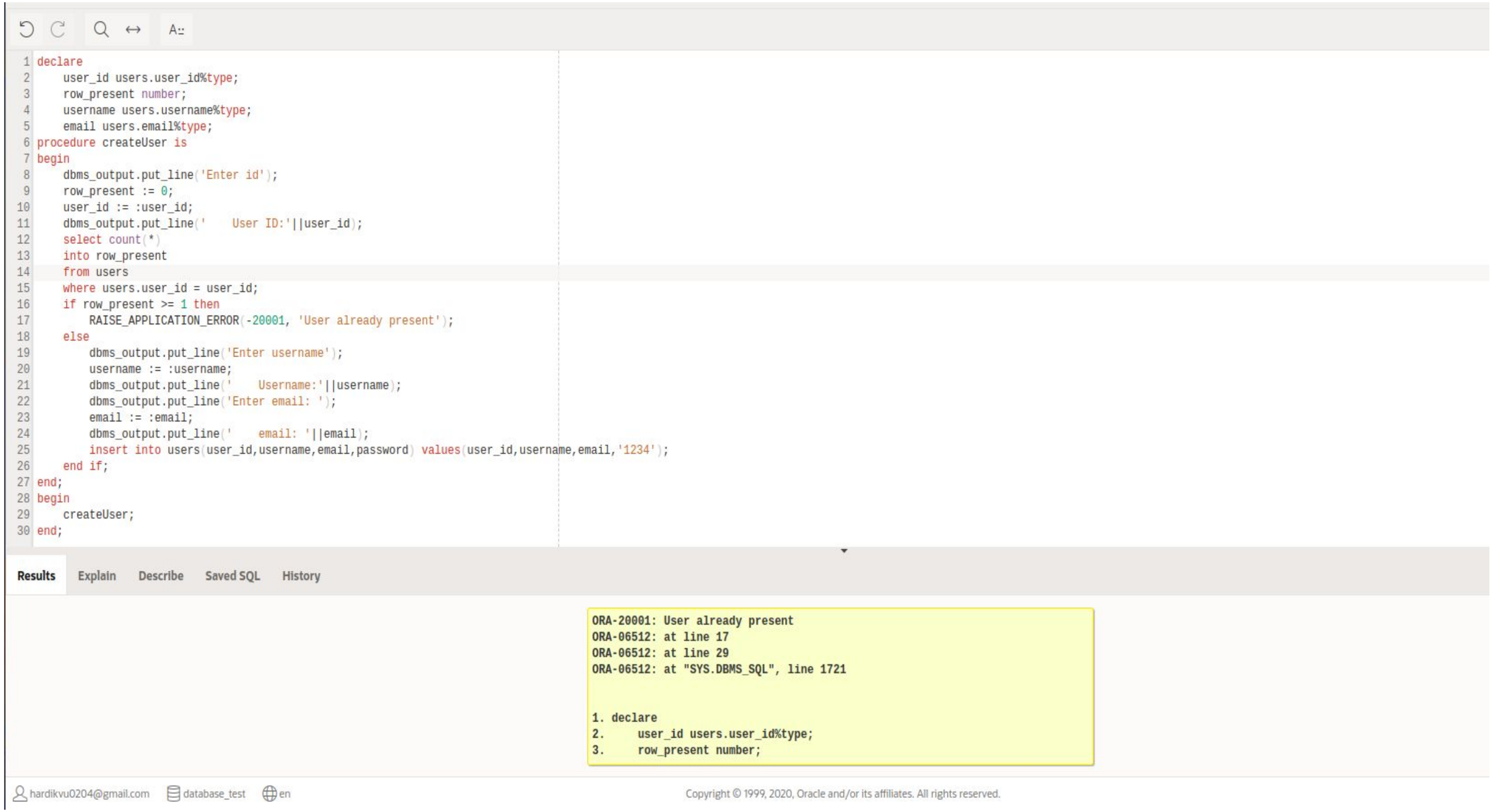
```
declare
 a_id articles.article_id%type;
 row_present number;
 title articles.title%type;
 description articles.description%type;
 user_id users.user_id%type;
procedure createArticle
IS
begin
   dbms_output.put_line('Enter id or article : ');
    a_id := :a_id;
    select count(*)
    into row_present
    from articles
    where article_id = a_id;

    if row_present = 1 then
         RAISE_APPLICATION_ERROR(-20001, 'Article already present');
    else
        dbms_output.put_line('Enter title: ');
        title := :title;
        dbms_output.put_line('Enter description: ');
        description := :description;
        user_id := :user_id;
        insert into articles(article_id, title, description,creator_id) values(a_id,title,description,user_id);
        dbms_output.put_line('Title: '|| title);
        dbms_output.put_line('Description: ' || description);
        dbms_output.put_line('Created by user_id : ' || user_id);
    end if;
end;
begin
    createArticle;
end;
```

Results Explain Describe Saved SQL History

```
Enter id or article :
Enter title:
Enter description:
Title: afdsf
Description: asdfdsfdsf fsadf
Created by user_id : 3
```

hardikvu0204@gmail.com database_test en Copyright © 1999, 20

Procedure to create users by taking input: raises application error if user already present



A::

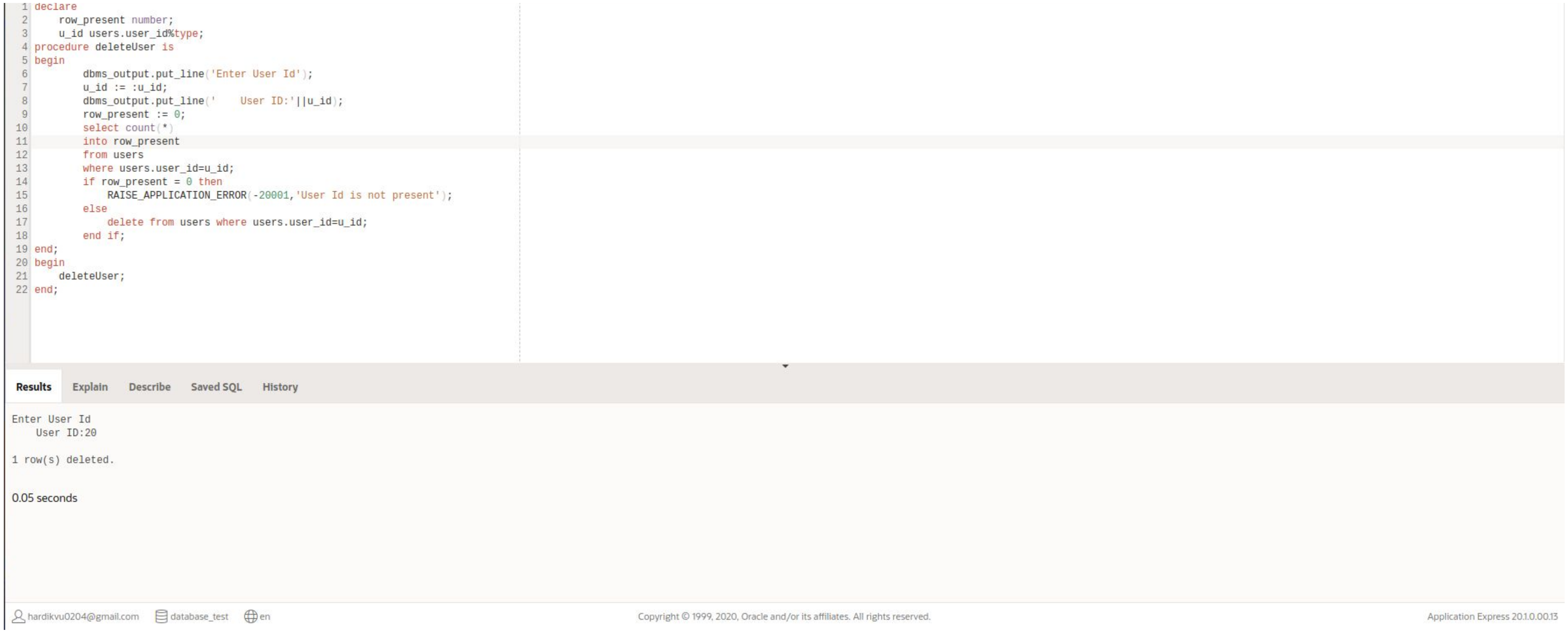
```
declare
    user_id users.user_id%type;
    row_present number;
    username users.username%type;
    email users.email%type;
procedure createUser is
begin
    dbms_output.put_line('Enter id');
    row_present := 0;
    user_id := :user_id;
    dbms_output.put_line('    User ID:'||user_id);
    select count(*)
    into row_present
    from users
    where users.user_id = user_id;
    if row_present >= 1 then
        RAISE_APPLICATION_ERROR(-20001, 'User already present');
    else
        dbms_output.put_line('Enter username');
        username := :username;
        dbms_output.put_line('    Username:'||username);
        dbms_output.put_line('Enter email: ');
        email := :email;
        dbms_output.put_line('    email: '||email);
        insert into users(user_id,username,email,password) values(user_id,username,email,'1234');
    end if;
end;
begin
    createUser;
end;
```

Results Explain Describe Saved SQL History

```
ORA-20001: User already present
ORA-06512: at line 17
ORA-06512: at line 29
ORA-06512: at "SYS.DBMS_SQL", line 1721


1. declare
2.     user_id users.user_id%type;
3.     row_present number;
```

hardikvu0204@gmail.com database_test en

Procedure to delete a user from users table



```
declare
    row_present number;
    u_id users.user_id%type;
procedure deleteUser is
begin
        dbms_output.put_line('Enter User Id');
        u_id := :u_id;
        dbms_output.put_line('    User ID:'||u_id);
        row_present := 0;
        select count(*)
        into row_present
        from users
        where users.user_id=u_id;
        if row_present = 0 then
            RAISE_APPLICATION_ERROR(-20001,'User Id is not present');
        else
            delete from users where users.user_id=u_id;
        end if;
end;
begin
    deleteUser;
end;
```

```
Enter User Id
    User ID:20

1 row(s) deleted.
```

0.05 seconds

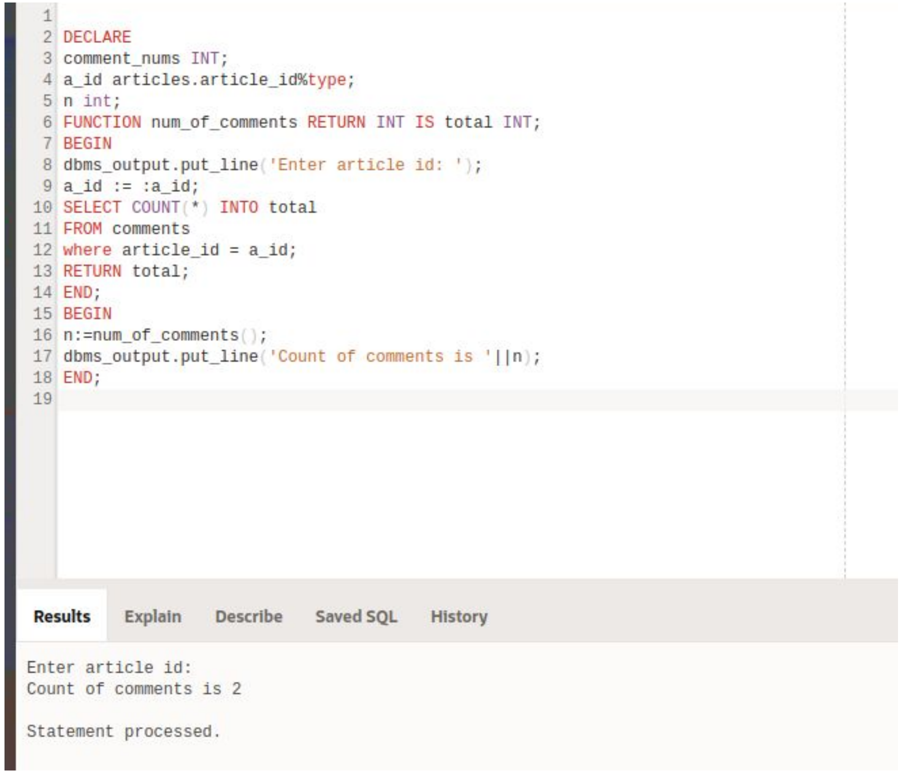
hardikvu0204@gmail.com
database_test
en
Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.
Application Express 20.1.0.00.13

Functions

Description of functions implemented:

- Enter the article id through prompt and the function will return the count of comments in that article.
- Function to return id of user with maximum number of posts.

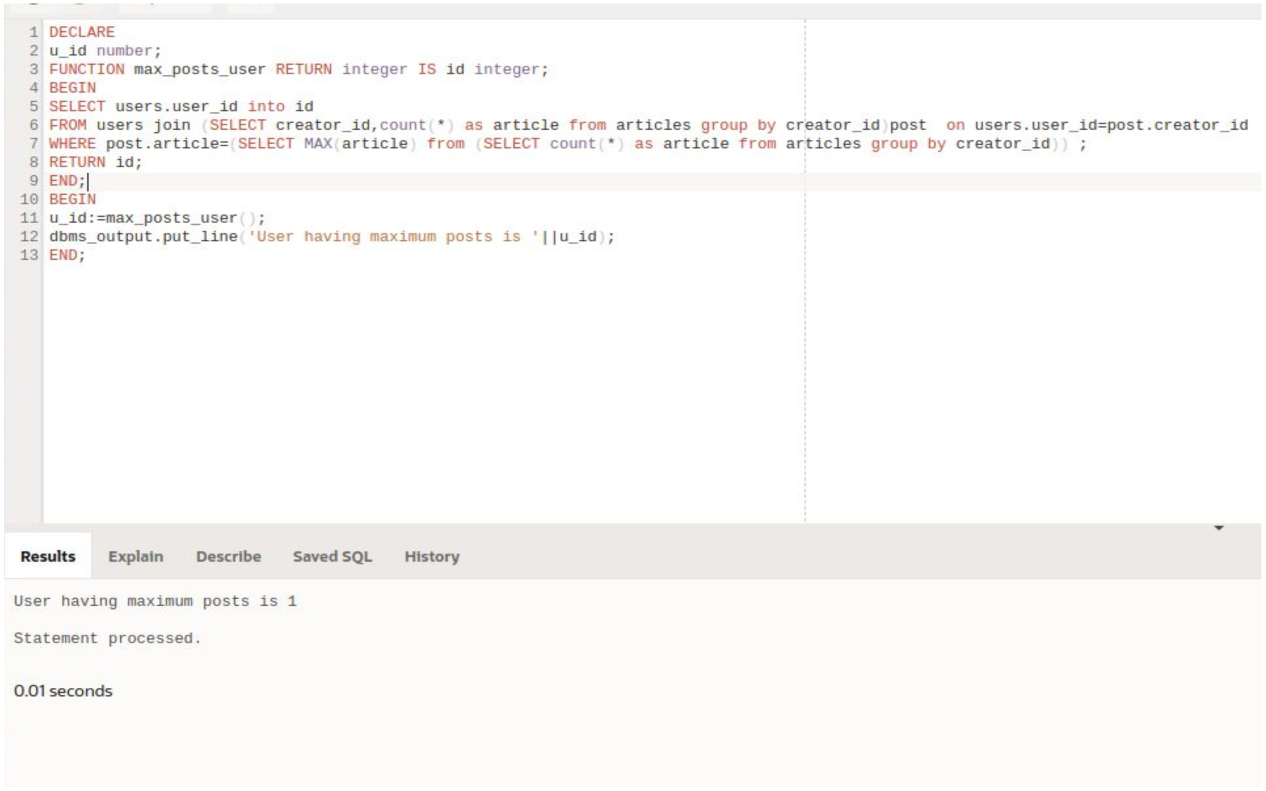
Function to get no. of comments of an article:



```

DECLARE
comment_nums INT;
a_id articles.article_id%type;
n int;
FUNCTION num_of_comments RETURN INT IS total INT;
BEGIN
dbms_output.put_line('Enter article id: ');
a_id := :a_id;
SELECT COUNT(*) INTO total
FROM comments
where article_id = a_id;
RETURN total;
END;
BEGIN
n:=num_of_comments();
dbms_output.put_line('Count of comments is '||n);
END;

```

```
Enter article id:
Count of comments is 2

Statement processed.
```

Function to get the user with maximum posts



```
DECLARE
u_id number;
FUNCTION max_posts_user RETURN integer IS id integer;
BEGIN
SELECT users.user_id into id
FROM users join (SELECT creator_id,count(*) as article from articles group by creator_id)post  on users.user_id=post.creator_id
WHERE post.article=(SELECT MAX(article) from (SELECT count(*) as article from articles group by creator_id)) ;
RETURN id;
END;
BEGIN
u_id:=max_posts_user();
dbms_output.put_line('User having maximum posts is '||u_id);
END;
```

```
User having maximum posts is 1

Statement processed.
```

0.01 seconds