Table of Contents:

Overview & Objectives:

The goal of this data(base) model is to provide foundation for a basic website for blogging, Where blogs are associated with an author and a category. Posts and categories can have multiple tags. In this way, multiple tags can be grouped by category. The model has also option for adding multiple email to a user and one of them can be set as primary email address.
To verify an email account, a key is set to email addresses and only the owner of that email address can know the key. So when the user enters the key, it’s checked against the key associated with the email address user wants to verify; if it’s the same then the email address gets stored in database as verified email address for the user.

Handle	BlogText	Tag	Category	Email	Key
Tux	Helle 123	Linux, NSFW	OS	tanjim@amar.email	435f33
Audacious	Hello 3452	mac	OS	audacioustux@protonmail.com tangimhossain1@gmail.com	343f45 -
Tux	Lorem ipsum	Express, koa, NSFW	NodeJs	tanjim@amar.email	343f45

1NF:

Handle	**BlogText**	**Tag**	**Category**	**Email**	**Key**
Tux	Helle 123	linux	OS	tanjim@amar.email	435f33
Tux	Helle 123	NSFW	OS	tanjim@amar.email	435f33
Audacious	Hello 3452	mac	OS	audacioustux@protonmail.com	343f45
Audacious	Hello 3452	mac	OS	tangimhossain1@gmail.com	-
Tux	Lorem ipsum	express	NodeJs	tanjim@amar.email	435f33
Tux	Lorem ipsum	koa	NodeJs	tanjim@amar.email	435f33
Tux	Lorem ipsum	NSFW	NodeJs	tanjim@amar.email	435f33

2NF:

User		**Email**	
Id	**Handle**	**Email**	**UserId**
1	Tux	tanjim@amar.email	1
2	Audacious	audacioustux@protonmail.com	2
		tangimhossain1@gmail.com	2

Tag		**Category**	
Id	**Tag**	**Id**	**Category**
1	Mac	1	OS
2	Linux	2	NodeJs
3	express		
4	NSFW		

Blog			**cat_tag**		**blog_tag**	
Id	**content**	**userId**	**TagId**	**CatId**	**TagId**	**BlogId**
1	Helle 123	1	**1**	**1**	**4**	**1**
2	Hello 3452	2	**2**	**1**	**4**	**3**
3	Lorem ipsum	1	**3**	**2**		

UnverifiedEmails	
Email	**Key**
tanjim@amar.email	435f33
audacioustux@protonmail.com	343f45

3NF: No transitive dependency, already in 3rd normal form

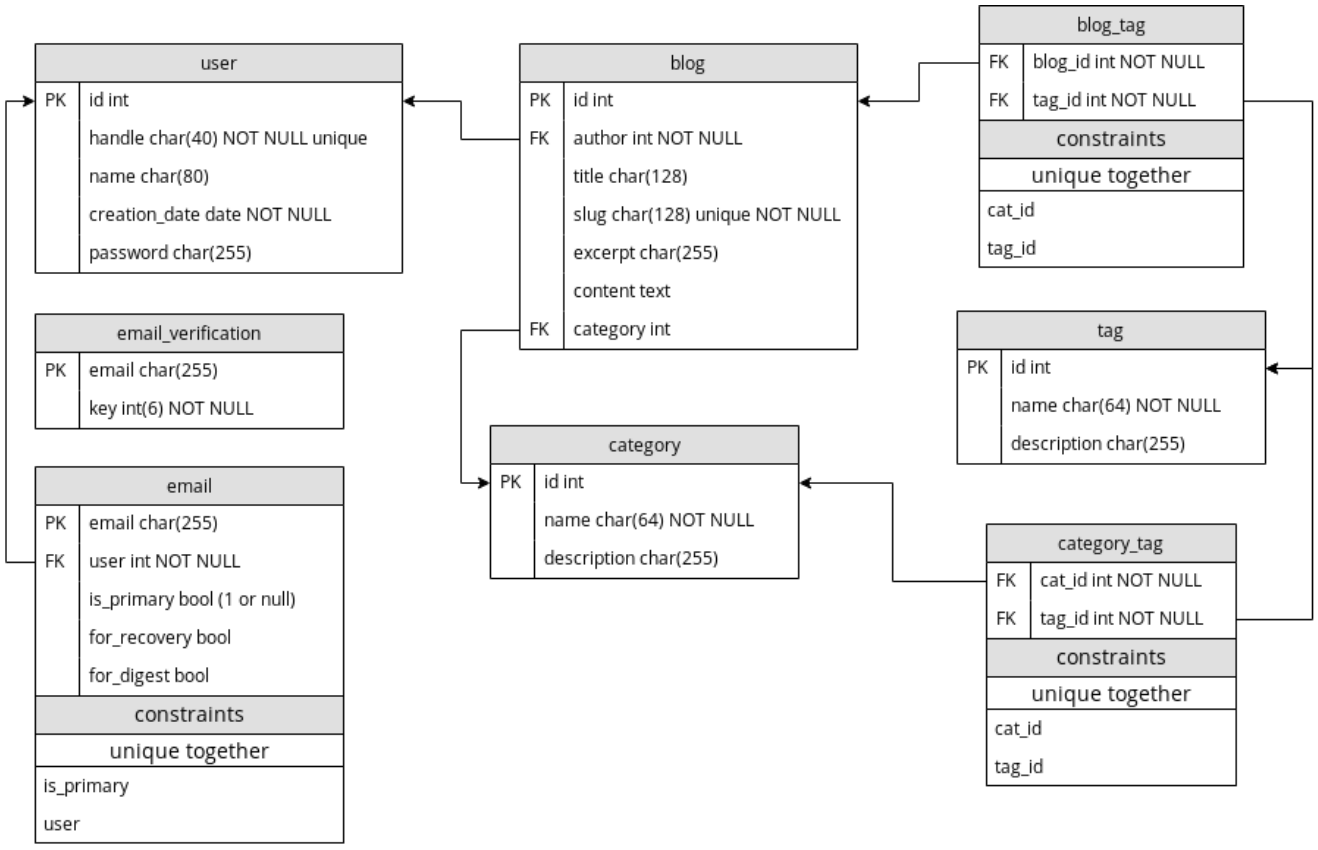

Figure 1: Entity Relation Diagram (ERD)

Table Descriptions:

user: for storing user information (handle/username with unique constraint)
email_verification: for storing unverified email addresses pending to be verified with a key
email: all verified email addresses, each of them belong to a user
blog: for storing blog data with author and taxonomy information
tag: all blogs can have multiple tags (many to many) and gives hint about the content
blog_tag: through model for constructing many to many relation between tag and blog model
category: a single category groups multiple tags and also gives hint about the content of a blog
cat_tag: through model for constructing many to many relation between tag and category model

Initialization & data insertion:

```
create table users (
id number,
handle varchar2(40) NOT NULL,
name varchar2(80),
creation_date TIMESTAMP WITH TIME ZONE NOT NULL,
password varchar2(255) NOT NULL,
CONSTRAINT pk_users PRIMARY KEY(id)
);

create table email_verification(
email varchar2(255),
key number(6) not null,
CONSTRAINT pk_email_verification PRIMARY KEY(email)
);

create table email(
email varchar2(255),
author number constraint user_email_id_fk references users on delete cascade,
is_primary char check (is_primary in (0,1)),
for_recovery char check (for_recovery in (0,1)),
for_digest char check (for_digest in (0,1)),
CONSTRAINT pk_email PRIMARY KEY(email)
);

create table tag(
id number,
name varchar2(64) NOT NULL,
description varchar2(255),
CONSTRAINT pk_tag PRIMARY KEY(id)
);

create table category(
id number,
name varchar2(64) NOT NULL,
description varchar2(255),
CONSTRAINT pk_category PRIMARY KEY(id)
);

create table cat_tag(
cat_id number constraint cat__cat_tag_id_fk references category on delete cascade,
tag_id number constraint tag__cat_tag_id_fk references tag on delete cascade,
CONSTRAINT unique__cat_tag UNIQUE(cat_id, tag_id)
);

create table blog(
id number,
author number constraint blog_author_id_fk references users on delete cascade,
title varchar2(128),
slug varchar2(128) not null,
excerpt varchar2(255),
content varchar2(4000),
category number constraint blog_category_id_fk references users on delete cascade,
CONSTRAINT pk_blog PRIMARY KEY(id)
);

create table blog_tag(
```

```
blog_id number constraint blog__blog_tag_id_fk references blog on delete cascade,
tag_id number constraint tag__blog_tag_id_fk references tag on delete cascade,
CONSTRAINT unique__blog_tag UNIQUE(blog_id, tag_id)
);

insert into users(id, handle, name, creation_date, password) values(1, 'tux', 'Tanjim hossain', sysdate, '1234');
insert into users(id, handle, name, creation_date, password) values(2, 'tux3', 'Tanjim hossain3', sysdate, '1234');
insert into users(id, handle, name, creation_date, password) values(3, 'tux2', 'Tanjim hossain4', sysdate, '1234');
insert into users(id, handle, name, creation_date, password) values(4, 'tuxq', 'Tanjim hossain7', sysdate, '1234');
insert into users(id, handle, name, creation_date, password) values(5, 'tuxa', 'Tanjim hossain9', sysdate, '1234');

insert into email(email, author) values('tangimhossain1@gmail.com', 1);
insert into email(email, author) values('tangimhossain12@gmail.com', 1);
insert into email(email, author) values('tangimhossain13@gmail.com', 2);
insert into email(email, author) values('tangimhossain14@gmail.com', 4);
insert into email(email, author) values('tangimhossain15@gmail.com', 1);

insert into email_verification(email, key) values('tangimhossain1@gmail.com', 43523);
insert into email_verification(email, key) values('tangimhossain12@gmail.com',43523);
insert into email_verification(email, key) values('tangimhossain13@gmail.com',43523);
insert into email_verification(email, key) values('tangimhossain14@gmail.com',43523);
insert into email_verification(email, key) values('tangimhossain15@gmail.com',43523);

insert into blog(id, author, slug, title, content) values(1, 2, 'sdf0e3d-dsd', 'abcdcedsdf', '1234');
insert into blog(id, author, slug, title, content) values(2, 2, 'sdf0e3d-dsd3', 'abcdcedsdf', '1234');
insert into blog(id, author, slug, title, content) values(3, 1, 'sdf0e3d-dsd4', 'abcdcedsdf', '1234');
insert into blog(id, author, slug, title, content) values(4, 3, 'sdf0e3d-dsd7', 'abcdcedsdf', '1234');
insert into blog(id, author, slug, title, content) values(5, 2, 'sdf0e3d-dsd9', 'abcdcedsdf', '1234');
insert into tag(id, name) values(1, 'dsfef3ds');
insert into tag(id, name) values(2, 'dsfef3ds4');
insert into tag(id, name) values(3, 'dsfef3ds4');
insert into tag(id, name) values(4, 'dsfef3ds4');
insert into tag(id, name) values(5, 'dsfef3ds4');

insert into category(id, name) values(1, 'dsfef3ds4');
insert into category(id, name) values(2, 'dsfef3ds4');
insert into category(id, name) values(3, 'dsfef3ds4');
insert into category(id, name) values(4, 'dsfef3ds4');
insert into category(id, name) values(5, 'dsfef3ds4');

insert into cat_tag values (1,1);
insert into cat_tag values (1,2);
insert into cat_tag values (2,2);
insert into cat_tag values (2,1);
insert into cat_tag values (1,3);

insert into blog_tag values (2,1);
insert into blog_tag values (1,2);
insert into blog_tag values (4,2);
insert into blog_tag values (3,1);
insert into blog_tag values (1,3);

update blog set category=1 where id=2;
```

BLOG_ID	TAG_ID
2	1
1	2
4	2
3	1
1	3

ID	AUTHOR	TITLE	SLUG	EXCERPT	CONTENT	CATEGORY
1	2	abcdcedsdf	sdf0e3d-dsd	–	1234	–
2	2	abcdcedsdf	sdf0e3d-dsd3	–	1234	1
3	1	abcdcedsdf	sdf0e3d-dsd4	–	1234	–
4	3	abcdcedsdf	sdf0e3d-dsd7	–	1234	–
5	2	abcdcedsdf	sdf0e3d-dsd9	–	1234	–

EMAIL	KEY
tangimhossain1@gmail.com	43523
tangimhossain12@gmail.com	43523
tangimhossain13@gmail.com	43523
tangimhossain14@gmail.com	43523
tangimhossain15@gmail.com	43523

ID	NAME	DESCRIPTION
1	dsfef3ds4	–
2	dsfef3ds4	–
3	dsfef3ds4	–
4	dsfef3ds4	–
5	dsfef3ds4	–

CAT_ID	TAG_ID
1	1
1	2
2	2
2	1
1	3

EMAIL	AUTHOR	IS_PRIMARY	FOR_RECOVERY	FOR_DIGEST
tangimhossain12@gmail.com	1	–	–	–
tangimhossain13@gmail.com	2	–	–	–
tangimhossain14@gmail.com	4	–	–	–
tangimhossain15@gmail.com	1	–	–	–

Queries based on scenarios:

ID	NAME	DESCRIPTION
1	dsfef3ds	-
2	dsfef3ds4	-
3	dsfef3ds4	-

1. all category->tags, of blog (id=2)

```
select tag.* from tag, cat_tag where cat_id in (select category from blog where id=2) and cat_tag.tag_id = tag.id
```

ID	AUTHOR	TITLE	SLUG	EXCERPT	CONTENT	CATEGORY
1	2	abcdcedsdf	sdf0e3d-dsd	-	1234	-
4	3	abcdcedsdf	sdf0e3d-dsd7	-	1234	-

2. blogs with tag=2

```
select blog.* from blog_tag, blog where blog_tag.tag_id = 2 and blog.id=blog_tag.blog_id
```

EMAIL
tangimhossain12@gmail.com
tangimhossain13@gmail.com
tangimhossain14@gmail.com
tangimhossain15@gmail.com
tangimhossain1@gmail.com

3. all email addresses that exist in email_verification, also in emails

```
select email_verification.email from email, email_verification where email.email = email_verification.email
```

ID	NAME	DESCRIPTION
1	dsfef3ds	-
2	dsfef3ds4	-
3	dsfef3ds4	-
5	dsfef3ds4	-

4. all tags of blog (id=2)
select tag.* from tag where tag.id in (select tag_id from blog_tag where blog_tag.blog_id=2) or tag.id in ((select tag.id from tag, cat_tag where cat_id in (select category from blog where id=2) and tag.id in cat_tag.tag_id))

```
no data found
```

5. users with primary email address
select email.*, users.* from users, email where email.is_primary=1 and email.author = users.id

ID	HANDLE	NAME	CREATION_DATE	PASSWORD
1	tux	Tanjim hossain	18-APR-19 01.41.57.000000 AM US/PACIFIC	1234
2	tux3	Tanjim hossain3	18-APR-19 01.41.57.000000 AM US/PACIFIC	1234
3	tux2	Tanjim hossain4	18-APR-19 01.41.57.000000 AM US/PACIFIC	1234
4	tuxq	Tanjim hossain7	18-APR-19 01.41.57.000000 AM US/PACIFIC	1234
5	tuxa	Tanjim hossain9	18-APR-19 01.41.57.000000 AM US/PACIFIC	1234

6. users joined after or on the same time with user.handle='tux'

```
select u.* from users t, users u where t.creation_date <= u.creation_date
and t.handle='tux'
```

ID	AUTHOR	TITLE	SLUG	EXCERPT	CONTENT	CATEGORY	ID	NAME	DESCRIPTION
2	2	abcdcedsdf	sdf0e3d-dsd3	-	1234	1	1	dsfef3ds4	-

7. list all blogs with category = 'dsfef3ds4'

```
select * from blog, category where blog.category=category.id and
category.name='dsfef3ds4'
```

learning experience:
to much repetitive things to do and figured out oracle doesn’t have a boolean type!

Problems faced:

1. varchar2 max capacity is limited to 4000 or something near the value. Need more capacity for blog post contents
2. setting auto incriminating primary key is too much complicated In oracle. Not familiar with sequence

future plan:
None! At-least no plan with oracle. But the practical experience of working without any ORM or RDBMS will definitely help me to optimize queries manually in postgres or other RDB

Thoughts: A night without SLEEP!
Lessons learned: working in a group full of ghosts was a bad idea :)