Karanvir Singh

CUNY ID (Last 4 digits) – 5441

Class Section: CSCI 331-35 8:00-9:15PM

Submission Date: May 15th, 2019

Project 3

1. Identify reporters who received the most customer comments create in the last two months. Display two columns: Reporter name and number of comments. Display one distinct row for each reporter. Display the reporter with the most comments first.

select r.r\_Fname "First Name", r.r\_LName "Last Name", count(1) "Number of Comments"

from comments c, reporter\_article ra, reporter r

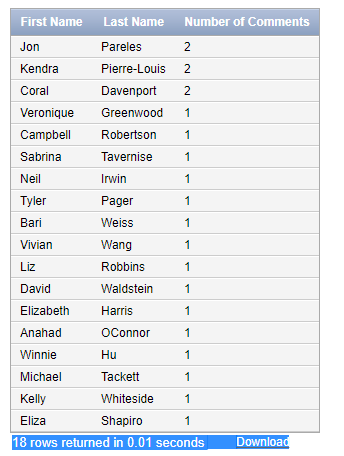
where c.comment\_date >= date '2019-03-06'

and r.r\_id = ra.r\_id

and ra.article\_id = c.article\_id

group by r.r\_Fname, r.r\_LName

order by 3 desc



2. Identify customers who have not written article comments in the last six months. Display the customer name. Use a nested select to answer this question.

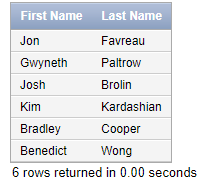
select c\_first "First Name", c\_last "Last Name"

from customer

where c\_id not in (select c\_id

from comments

where comment\_date >= date '2018-11-07');



3. Identify the articles in the last three months with the most customer reads. Display three columns: article name, topic and number of customer reads. Display one distinct row for each article. Display the article with the most reads first.

select article\_title "Article Title", topic\_type "Topic", sum(readnum) "Customer Reads"

from article, topic, articleread

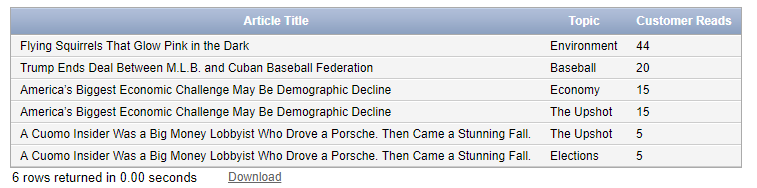
where article.article\_id = topic.article\_id

and articleread.article\_id = article.article\_id

and readdate >= date '2019-02-08'

group by article\_title, topic\_type

order by 3 desc



4. Identify customers who have not read any articles in the last quarter of a year. Display the customer name, account and email. Order the output the customer name.

select c\_first "First Name", c\_last "Last Name", c\_account "Account Number", c\_email "Email Address"

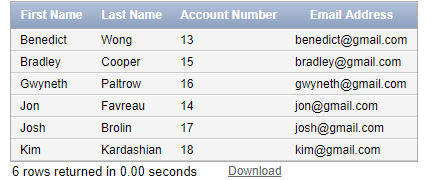
from customer

where c\_id not in (select c\_id

from articleread

where readdate >= date '2018-02-08')

order by c\_first asc



5. Identify topics in the last month with the most customer reads. Display two columns: Topic and number of customer reads. Display one distinct row for each topic. Display the topic with the most reads first.

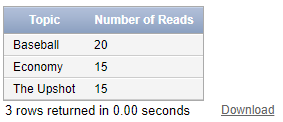
select topic\_type "Topic", readnum "Number of Reads"

from topic, articleread

where topic.article\_id = articleread.article\_id

and articleread.readdate >= date '2019-04-07'

order by readnum desc



6. Identify comments in articles that might include personally identifiable information (PII). This includes any data that could potentially be used to identify a person. For instance, examples of PII include email address, Social Security number, bank account number, home address, and full name. Display the article name, customer who created the comment, date of comment and the comment. Order the output by customer.

select article\_title "Title", c\_first "First", c\_last "Last", comment\_date "Date", comment\_text "Text"

from comments, customer, article

where comments.c\_id = customer.c\_id

and comments.article\_id = article.article\_id

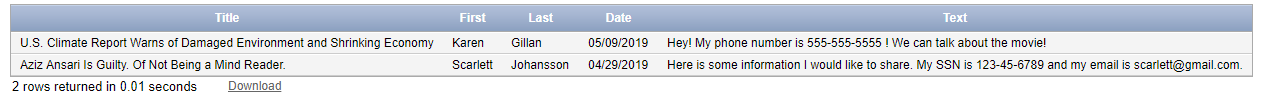
and (comment\_text like '% \_\_\_-\_\_\_-\_\_\_\_ %'

or comment\_text like '% \_\_\_-\_\_-\_\_\_\_%'

or comment\_text like '%@%'

or comment\_text like '% ST %')

order by c\_first



7. Identify customers with credit cards that will expire soon. Display the customer name, email address, credit card and expiration date. Order the output the customer name.

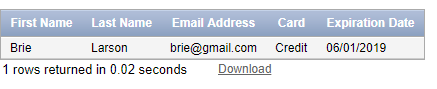
select c\_first "First Name", c\_last "Last Name", c\_email "Email Address", c\_payment "Card", c\_exp "Expiration Date"

from customer

where c\_exp <= date '2019-12-31'

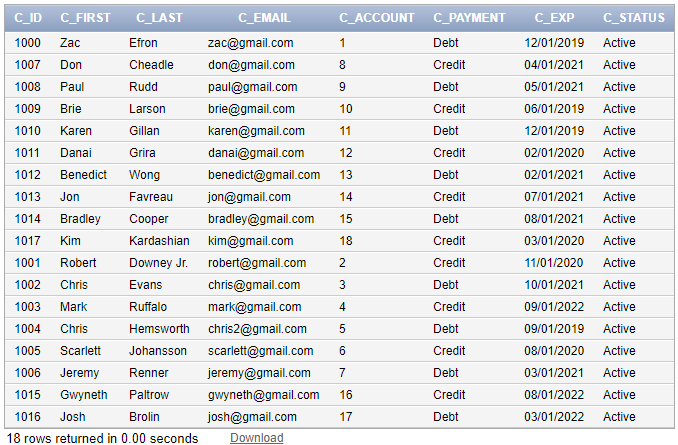
and c\_payment = 'Credit'

order by c\_first



8. Customer Robert Downey Jr. wants to cancel his subscription to the New York Times. Identify the SQL to implement.

Pre-Update

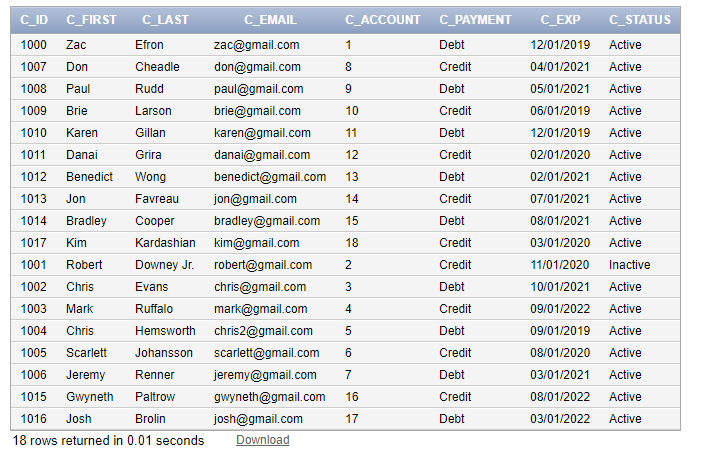


update customer

set c\_status = 'Inactive'

where c\_first = 'Robert'

Post - Update

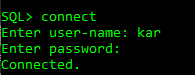


9. In one SQL window, delete all topics. Don’t commit. In another SQL window, create five new topics. Don’t commit. In each SQL window, identify the number of topics. Explain your results. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations.

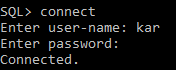
--SQL operations are in screenshots and some are typed--

For both SQL windows, I connect. Both windows are shown below:

First Window



Second Window



In the first SQL window, I deleted everything from the topic table. This was done using the command:

Delete from topic;

Output: 30 rows deleted

In the second SQL window, I inserted 5 new topics using insert command. This was done using the commands:

insert into topic

values(topic\_seq.nextval, 312, 'Lunch')

insert into topic

values(topic\_seq.nextval, 314, 'Celebrity')

insert into topic

values(topic\_seq.nextval, 310, 'Bills')

insert into topic

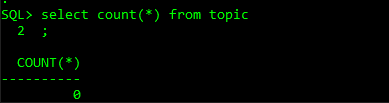
values(topic\_seq.nextval, 307, 'Sounds')

insert into topic

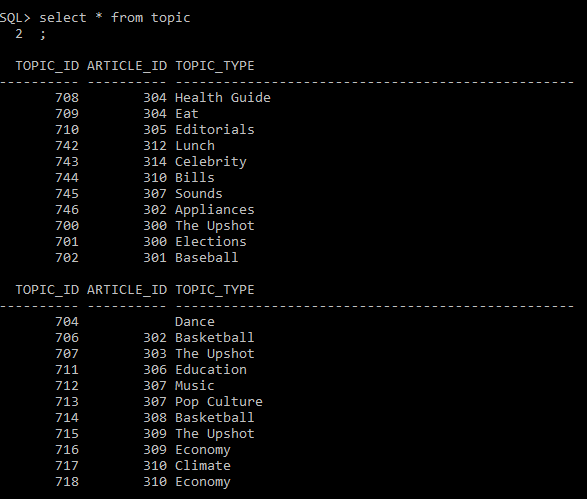
values(topic\_seq.nextval, 302, 'Appliances')

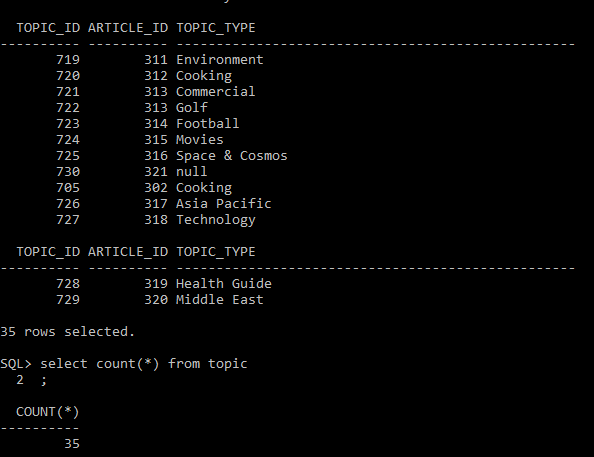
Now for both windows I used the count function to display the number of rows. The count was 0 for the first window. Now for the second SQL window, the output was 35.

First Window



Second Window





My results have showed that the counts of the two separate windows are different. In the first window, I have 0 rows for topics, although for my second window I have 35. From here, I can infer that the numbers are different because nothing was committed, therefore it was not saved. The operations were not committed in one window; therefore, it was not reflected in the other. Window 1 is showing true to its environment, and window 2 is showing true to its own environment.

10. In one SQL window, delete all articles written by reporter Tyler Pager. Don’t commit. In another SQL window, reassign all articles written by Tyler Pager to David Waldstein. Don’t commit. Explain your results. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT \* FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the window before performing this operation. Show all SQL to perform these operations.

--SQL operations are in some screenshots and some are typed--

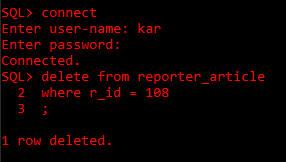
Creating the backup table for reporter\_article (Using SQL Command Line)

Create Table NewReporter\_Article

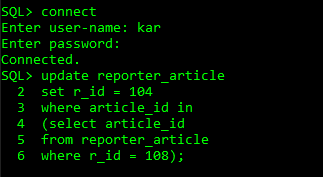
as Select \* from Reporter\_Article

commit;

First Window – Delete all articles written by Tyler Pager



Second Window – Reassign all articles from Tyler Pager to David Waldstein

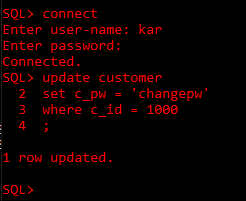


After I implemented the above code to update the reporter\_article table in the second window, I was not able to type in any more commands into the second window. There was no output because the reporter\_article table was locked by the first user in the preceding window; hence this is a table lock.

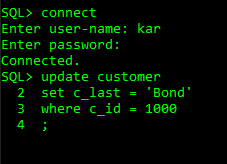
11. In one SQL window, change the password for the customer Zac Efron. Don’t commit. In another SQL window, change the last name of customer Zac Efron. Don’t commit. Quit both Oracle sessions. Login to Oracle again and display all columns for the customer Zac Efron. Explain your results. Disable the auto commit flag at the top of the windows before performing this operation. Show all SQL to perform these operations.

--SQL operations are in screenshots and some are typed--

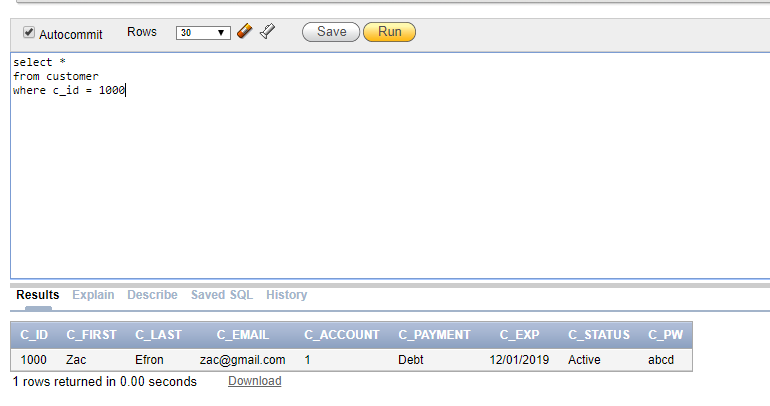
First Window: Change password for Zac Efron



Second Window: Change last name of Zac Efron to Zac Bond



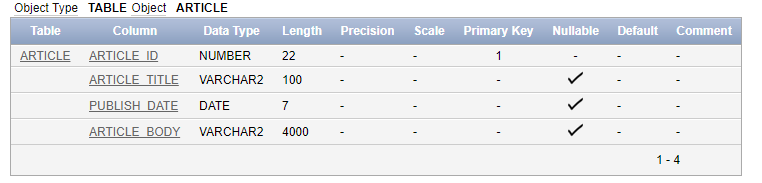
Logged back onto Oracle, saw that nothing was changed for all columns pertaining information about Zac Efron



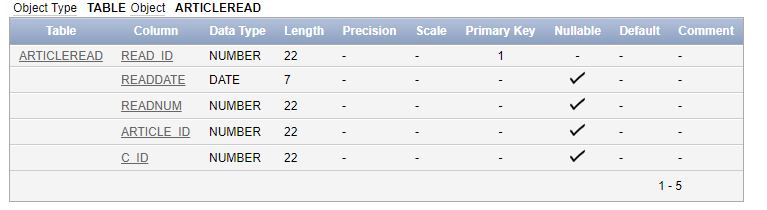
From the two windows that I performed commands in, and after closing out each window and login onto Oracle again, I noticed the value did not change. This was because nothing was committed. There is a row lock on the second window because the first window had a write lock on customer Zac Efron. I closed both windows without committing and opened the windows again to check if the values are changed, but they stayed the same, because I did not commit before exiting both windows.

12. Use the SQL DESCRIBE operation to display the structure for all tables.

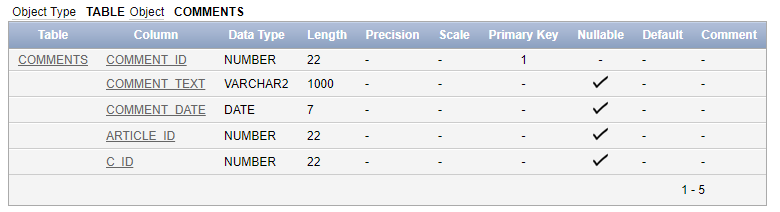
Describe Article



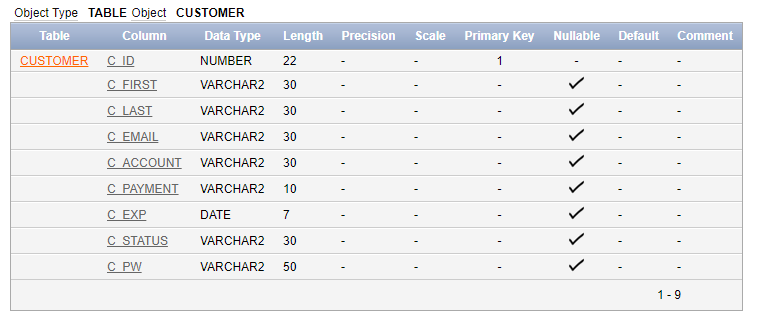
Describe ArticleRead



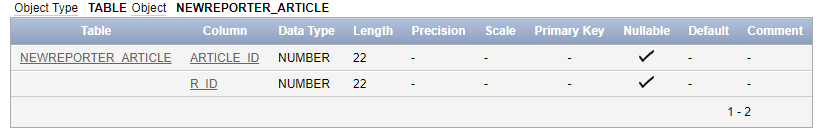
Describe Comments



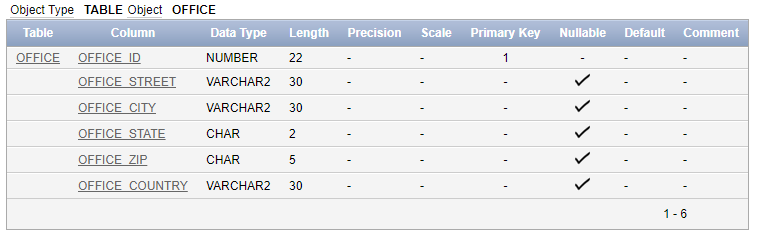
Describe Customer



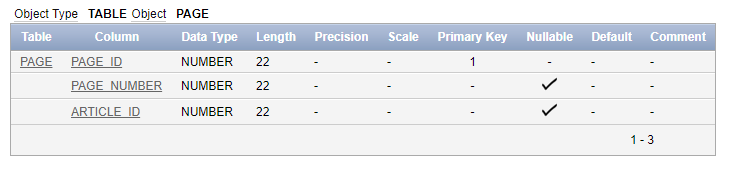
Describe NewReporterArticle – this table was required for Q10



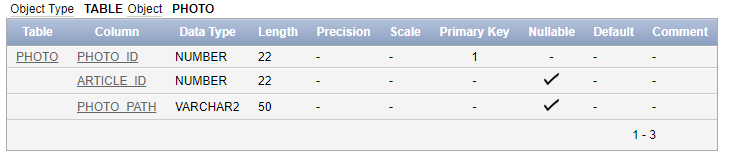
Describe Office



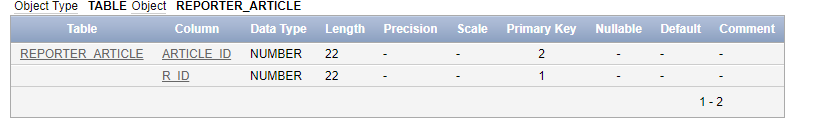
Describe Page



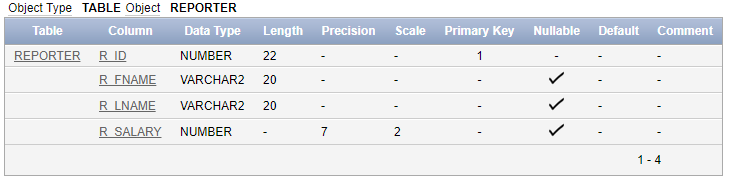
Describe Photo



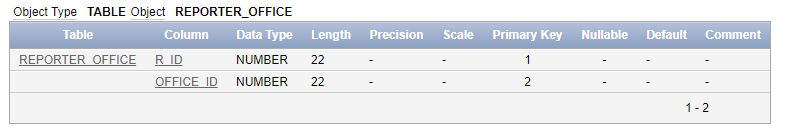
Describe Reporter\_Article



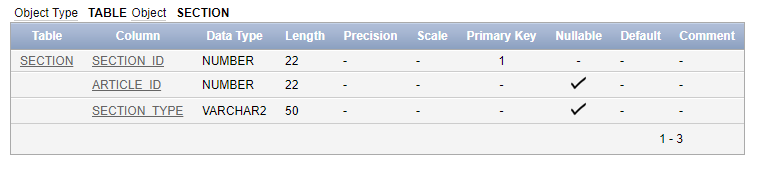
Describe Reporter



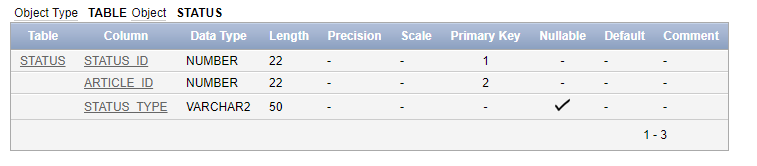
Describe Reporter\_Office



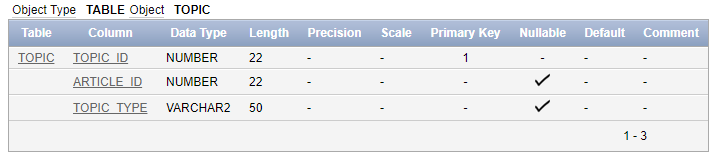
Describe Section



Describe Status



Describe Topic



**SECURITY COMMANDS**

**This command was done in order to let the system user grant all privileges to the user ‘kar’, where I did all my security questions**

grant all privileges to kar;

**In user ‘kar’ I created to a new view excluding expiration date and password.**

create view custview

as select c\_id, c\_first, c\_last, c\_email, c\_status, c\_account

from customer;

**I created a new user ‘Daniel’, and allowed the new user to view all comments.**

create user Daniel identified by d123;

grant select,insert on comments to Daniel;

**CREATE TABLE COMMANDS**

create table customer (

c\_id number Primary Key,

c\_first varchar(30),

c\_last varchar(30),

c\_email varchar(30),

c\_account varchar(30),

c\_payment varchar(10)

c\_exp DATE

)

create table comments (

comment\_id number Primary Key,

comment\_text varchar(1000),

comment\_date DATE,

article\_id number,

c\_id number,

foreign key (c\_id) references customer(c\_id),

foreign key (article\_id) references article(article\_id)

)

create table articleRead (

read\_id number Primary Key,

readDate DATE,

readNum number,

article\_id number,

c\_id number,

foreign key(c\_id) references customer(c\_id),

foreign key(article\_id) references article(article\_ID)

);

**CREATE SEQUENCE COMMANDS**

create sequence customer\_seq

start with 1000

increment by 1;

create sequence comment\_seq

start with 1100

increment by 1;

create sequence articleread\_seq

start with 1200

increment by 1;

**Customer Table Data**

insert into customer

values (customer\_seq.nextval, 'Zac', 'Efron', 'zac@gmail.com', '1', 'Debt', date '2019-12-01')

insert into customer

values (customer\_seq.nextval, 'Robert', 'Downey Jr.', 'robert@gmail.com', '2', 'Credit', date '2020-11-01')

insert into customer

values (customer\_seq.nextval, 'Chris', 'Evans', 'chris@gmail.com', '3', 'Debt', date '2021-10-01')

insert into customer

values (customer\_seq.nextval, 'Mark', 'Ruffalo', 'mark@gmail.com', '4', 'Credit', date '2022-09-01')

insert into customer

values (customer\_seq.nextval, 'Chris', 'Hemsworth', 'chris2@gmail.com', '5', 'Debt', date '2019-09-01')

insert into customer

values (customer\_seq.nextval, 'Scarlett', 'Johansson', 'scarlett@gmail.com', '6', 'Credit', date '2020-08-01')

insert into customer

values(customer\_seq.nextval, 'Jeremy', 'Renner', 'jeremy@gmail.com', '7', 'Debt', date '2021-03-01')

insert into customer

values(customer\_seq.nextval, 'Don', 'Cheadle', 'don@gmail.com', '8', 'Credit', date '2021-04-01')

insert into customer

values(customer\_seq.nextval, 'Paul', 'Rudd', 'paul@gmail.com', '9', 'Debt', date '2021-05-01')

insert into customer

values(customer\_seq.nextval, 'Brie', 'Larson', 'brie@gmail.com', '10', 'Credit', date '2019-06-01')

insert into customer

values(customer\_seq.nextval, 'Karen', 'Gillan', 'karen@gmail.com', '11', 'Debt', date '2019-12-01')

insert into customer

values(customer\_seq.nextval, 'Danai', 'Grira', 'danai@gmail.com', '12', 'Credit', date '2020-02-01')

insert into customer

values(customer\_seq.nextval, 'Benedict', 'Wong', 'benedict@gmail.com', '13', 'Debt', date '2021-02-01')

insert into customer

values(customer\_seq.nextval, 'Jon', 'Favreau', 'jon@gmail.com', '14', 'Credit', date '2021-07-01')

insert into customer

values(customer\_seq.nextval, 'Bradley', 'Cooper', 'bradley@gmail.com', '15', 'Debt', date '2021-08-01')

insert into customer

values(customer\_seq.nextval, 'Gwyneth', 'Paltrow', 'gwyneth@gmail.com', '16', 'Credit', date '2022-08-0')

insert into customer

values(customer\_seq.nextval, 'Josh', 'Brolin', 'josh@gmail.com', '17', 'Debt', date '2022-03-01')

insert into customer

values(customer\_seq.nextval, 'Kim', 'Kardashian', 'kim@gmail.com', '18', 'Credit', date '2020-03-01')

**ALTER AND UPDATE CUSTOMER BY CREATING A STATUS COLUMN AND VALUES FOR ALL CUSTOMERS – NECESSARY FOR QUESTION 8**

alter table customer

add c\_status varchar(30)

update customer

set c\_status = 'Active'

**ALTER AND UPDATE CUSTOMER BY CREATING A PASSWORD COLUMN AND VALUES FOR ALL CUSTOMERS – NECCESSARY FOR QUESTION 11**

alter table customer

add c\_pw varchar(50)

update customer

set c\_pw = 'abcd'

where c\_id = 1000

update customer

set c\_pw = 'pokemon123'

where c\_id = 1001

update customer

set c\_pw = 'sdhsudhs8'

where c\_id = 1002

update customer

set c\_pw = 'siey738'

where c\_id = 1004

update customer

set c\_pw = '9u9hqq'

where c\_id = 1003

update customer

set c\_pw = '633bf8y8'

where c\_id = 1005

update customer

set c\_pw = 'e9ur883bb'

where c\_id = 1006

update customer

set c\_pw = '98y3x'

where c\_id = 1007

update customer

set c\_pw = '8833bb0'

where c\_id = 1008

update customer

set c\_pw = 'dou98b3b'

where c\_id = 1009

update customer

set c\_pw = 'hdihf9333'

where c\_id = 1010

update customer

set c\_pw = 'h344nno'

where c\_id = 1011

update customer

set c\_pw = '973brb'

where c\_id = 1012

update customer

set c\_pw = '97844b'

where c\_id = 1013

update customer

set c\_pw = '1234qqx'

where c\_id = 1014

update customer

set c\_pw = 'bnjbr890'

where c\_id = 1015

update customer

set c\_pw = 't9494bbvwiw'

where c\_id = 1016

update customer

set c\_pw = '229bb'

where c\_id = 1017

**ARTICLEREAD DATA**

insert into articleread

values (articleread\_seq.nextval, date '2019-04-06', 5, 300, 1000)

insert into articleread

values (articleread\_seq.nextval, date '2019-04-09', 20, 301, 1001)

insert into articleread

values (articleread\_seq.nextval, date '2018-12-31', 1, 302, 1002)

insert into articleread

values (articleread\_seq.nextval, date '2018-12-30', 2, 303, 1003)

insert into articleread

values (articleread\_seq.nextval, date '2018-09-24', 7, 304, 1004)

insert into articleread

values (articleread\_seq.nextval, date '2018-12-21', 10, 305, 1005)

insert into articleread

values (articleread\_seq.nextval, date '2018-06-30', 3, 306, 1006)

insert into articleread

values (articleread\_seq.nextval, date '2018-07-04', 11, 307, 1007)

insert into articleread

values (articleread\_seq.nextval, date '2018-04-04', 19, 308, 1008)

insert into articleread

values (articleread\_seq.nextval, date '2019-05-06', 15, 309, 1009)

insert into articleread

values (articleread\_seq.nextval, date '2018-11-29', 44, 310, 1010)

insert into articleread

values (articleread\_seq.nextval, date '2019-03-29', 44, 311, 1011)

**COMMENTS DATA**

insert into comments

values(comment\_seq.nextval, 'Avengers is better', date '2019-04-06', 300, 1000)

insert into comments

values(comment\_seq.nextval, 'Iron Man dies in Avengers', date '2019-05-05', 301, 1001)

insert into comments

values(comment\_seq.nextval, 'Black Widow dies', date '2019-04-26', 302, 1002)

insert into comments

values(comment\_seq.nextval, 'Thor gets drunk', date '2019-04-27', 303, 1003)

insert into comments

values(comment\_seq.nextval, 'Captain America grows old', date '2019-04-28', 304, 1004)

insert into comments

values(comment\_seq.nextval, ‘Here is some information I would like to share. My SSN is 123-45-6789 and my email is scarlett@gmail.com.’, date '2019-04-29', 305, 1005)

insert into comments

values(comment\_seq.nextval, 'Time travel is how they beat Thanos.', date '2019-04-30', 306, 1006)

insert into comments

values(comment\_seq.nextval, 'Black Panther is back!', date '2019-05-01', 307, 1007)

insert into comments

values(comment\_seq.nextval, 'I think DC is better than Marvel', date '2019-05-02', 308, 1008)

insert into comments

values(comment\_seq.nextval, 'Hulk becomes a big bad smart guy', date '2019-05-03', 309, 1009)

insert into comments

values(comment\_seq.nextval, 'Iron Man uses the infinity stones to destroy Thanos.', date '2019-05-04', 310, 1010)

insert into comments

values(comment\_seq.nextval, 'There is a five-year time lapse.', date '2019-05-05', 311, 1011)

insert into comments

values(comment\_seq.nextval, 'Houston Rockets are gonna win the NBA Championship!', date '2019-05-07', 307, 1007)

insert into comments

values(comment\_seq.nextval, 'Hey! My phone number is 555-555-5555 ! We can talk about the movie!', date '2019-05-09', 310, 1010)