John Haney

November 21, 2020

COSC 640 – Database System I

Frostburg State University

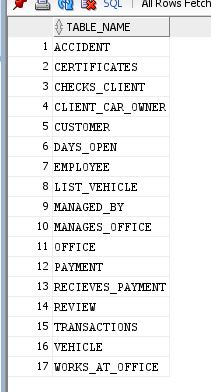
1. Write a query to list the name of your tables. (5 points)

SELECT TABLE\_NAME

FROM all\_tables

WHERE OWNER = 'CS640F2005'

ORDER BY table\_name;



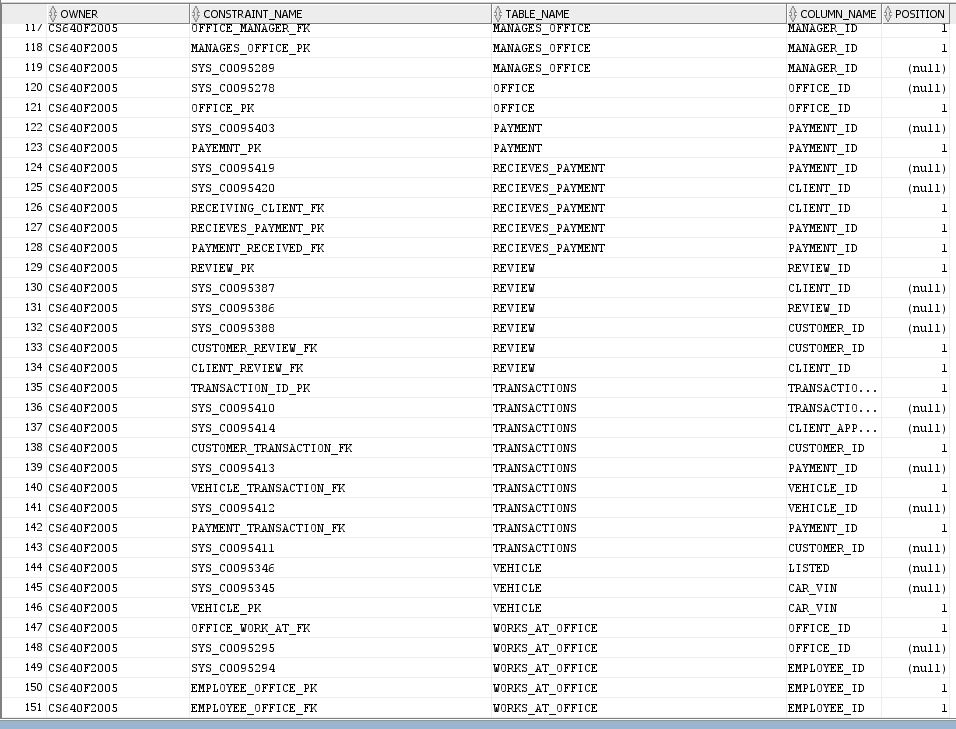
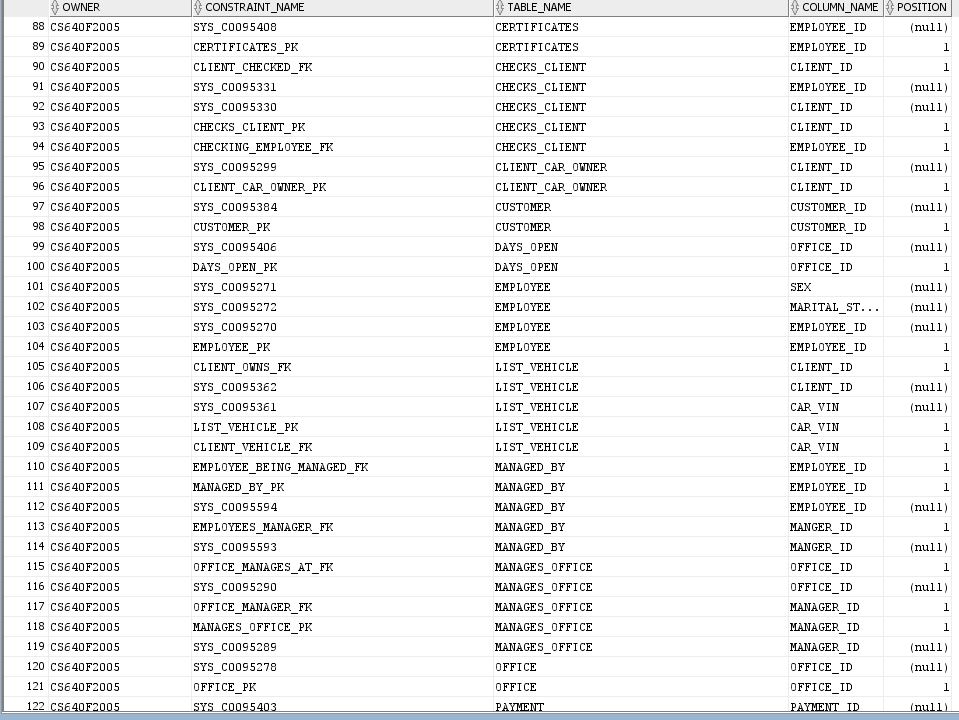
1. Write a query to list the table constraints (table by table). (5 points)

SELECT \*

FROM user\_cons\_columns

WHERE OWNER = 'CS640F2005'

ORDER BY table\_name;



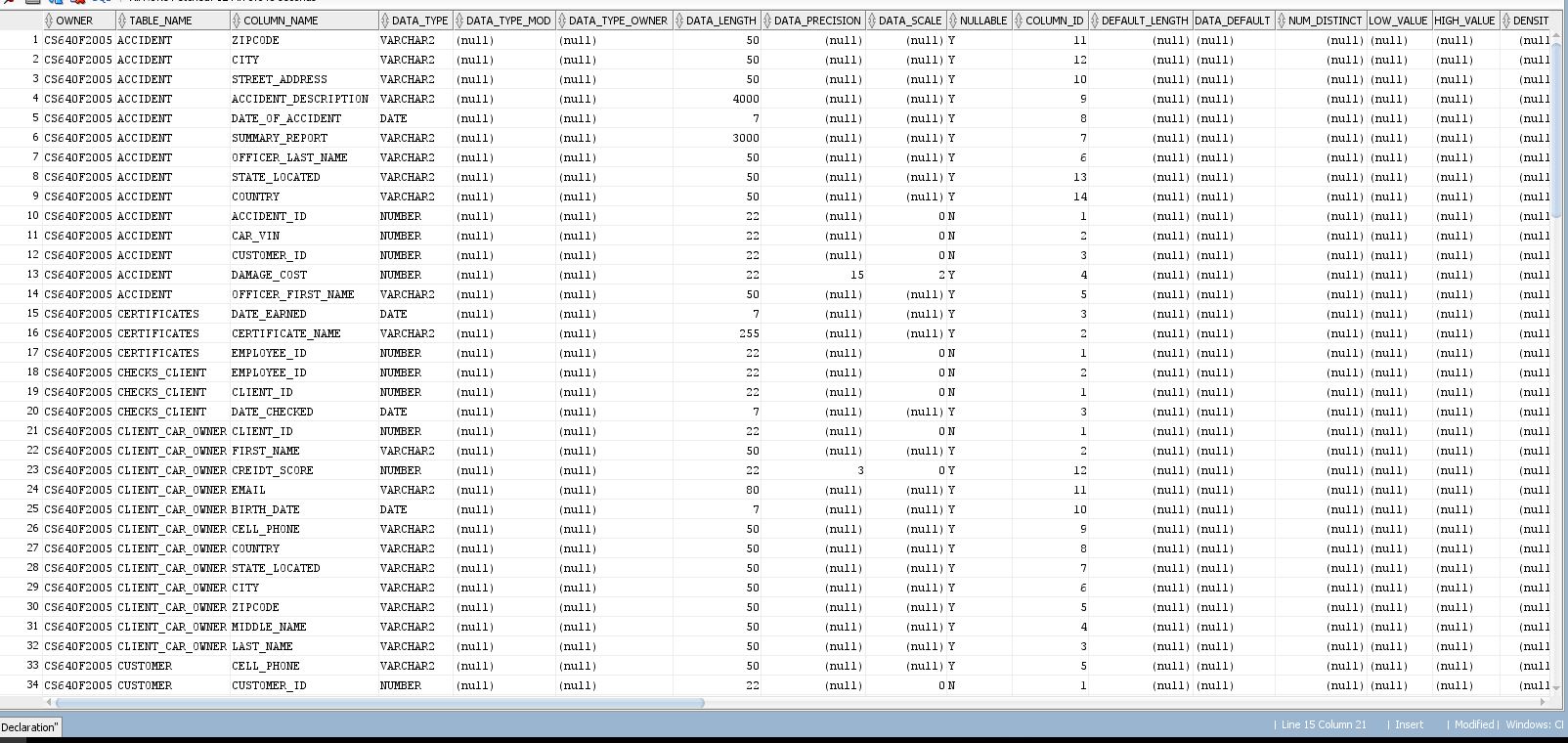
1. Write a query to show the values (data) on each table. (10 points)

SELECT \*

FROM all\_tab\_columns

WHERE OWNER = 'CS640F2005'

ORDER BY table\_name;



1. Create and run the following queries. Based on your data in your database, your query may/may not return any value. (each 5 points). Please make sure you:
2. State the problem,
3. Query,
4. The results you get after you run the query:
5. Create mailing labels for the employees (name & address).

For this query we need to find who the managers are from the two relationship tables and then use those results to find their names and addresses.

SELECT FIRST\_NAME || ' ' || LAST\_NAME, COUNTRY, state\_living\_in, CITY, ZIPCODE, STREET\_ADDRESS

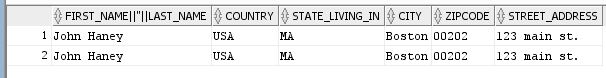
FROM EMPLOYEE

WHERE EMPLOYEE\_ID IN

(SELECT MANAGER\_ID FROM MANAGES\_OFFICE

left outer join managed\_by

on MANAGES\_OFFICE.MANAGER\_ID = managed\_by.manger\_id);



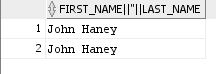
1. Display the name of employees that are office managers.

This problem requires us just to find the manager ID’s (employee ID’s) and then user those to filter out the data from the employee table we only see the names of the managers.

SELECT FIRST\_NAME || ' ' || LAST\_NAME

FROM EMPLOYEE

WHERE EMPLOYEE\_ID IN (SELECT MANAGER\_ID FROM MANAGES\_OFFICE);

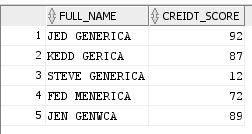


1. Display the first & last name of client with their credit score.

This problem requires taking data from the client table and filtering for only the first name, last name, and credit score.

SELECT FIRST\_NAME || ' ' || LAST\_NAME AS FULL\_NAME, CREIDT\_SCORE

FROM client\_car\_owner;



1. Display name of client, name of employee, the date a client was checked, and the office name the employee work at.

This problem requires us to join several tables together. We will have to travel from the client to the office joining each table and targeting the proper data along the way. The starting values for the statement will be the clients so there will be as many rows as there are clients.

SELECT

A.FIRST\_NAME || ' ' || A.LAST\_NAME AS CLIENT\_FULL\_NAME,

b.date\_checked,

C.FIRST\_NAME || ' ' || C.LAST\_NAME,

E.OFFICE\_NAME

FROM CLIENT\_CAR\_OWNER A

left OUTER JOIN checks\_client B

ON A.CLIENT\_ID = b.client\_id

left OUTER JOIN employee C

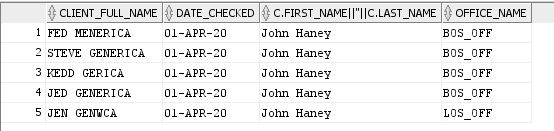
ON b.employee\_id = C.EMPLOYEE\_ID

left OUTER JOIN works\_at\_office D

ON C.EMPLOYEE\_ID = d.employee\_id

left OUTER JOIN OFFICE E

ON d.office\_id = E.OFFICE\_ID;



1. Display name of customers who had an accident, car Id, and cost of the damage.

This problem requires us to join the customer table from the accidents table. Since the accidents table is a relationship between the vehicle and customer tables, we will not need to join the vehicle as the VIN is already in the accident table.

SELECT

B.FIRST\_NAME || ' ' || B.LAST\_NAME,

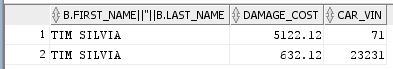
a.damage\_cost,

a.car\_vin

FROM ACCIDENT A

LEFT OUTER JOIN CUSTOMER B

ON A.CUSTOMER\_ID = b.customer\_id;



1. Display the total number of cars rented by each customer.

This problem requires us to look at the transactions and check how many of them have a request\_end date after today.

select count(client\_approval) as Total\_cars\_rented

from transactions

where request\_end > sysdate;

****

1. Display the number of employees for each office.

For this question we will select the office first and join the works at office table. We can then use an aggriate count function on the employees and group by the offices.

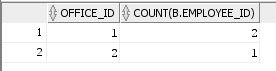
select a.office\_Id, count(b.employee\_id)

from office a

left outer join works\_at\_office b

on a.office\_id = b.office\_id

group by a.office\_id;



1. Display name, phone number of clients over 30 years old.

This problem requires us to take the birth date and subtract out that amount form the sysdate and them compare it to 30.

select \* from employee

where (birth\_date - sysdate) > 30;



1. Display SSN, address of female employees with more than one certificates.

This problem requires us to select the employee table and use the employee id in a subquery to count the number of certificate names. In the subquery we will join the employee table and count based on employee\_id

select a.ssn from employee a

where a.sex = 'F'

having 1 > (select count(certificate\_name)

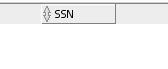
from certificates b

left outer join employee c

on b.employee\_id = c.employee\_id

where b.employee\_id = c.employee\_id

);



1. Display the total number of cars listed at any given day. User will input the date and time.

This problem requires starting with the vehicle and joining the transactions table. From here we constrain the results to show the cars that are either not out or will be back by the time the user inputs.

select a.\* from vehicle a

left outer join transactions b

on a.car\_vin = b.vehicle\_id

where b.request\_end < &date;



1. List the employees’ name, clients name with score higher than 50 in 2017.

This problem requires us to start at the client table and join the checks\_client and employee tables together. With all the tables as one we will put a constraint on the table to show us only rows where the client has a credit score higher than 50

select a.first\_name || ' ' || a.last\_name, c.first\_name || ' ' || c.last\_name

from client\_car\_owner a

left outer join checks\_client b

on a.client\_id = b.client\_id

left outer join employee c

on b.employee\_Id = c.employee\_id

where a.creidt\_score > 50 and b.date\_checked > '31-dec-16' and b.date\_checked < '01-jan-17';

1. List of feedback (client name, customer name, the rating number, who did the rating)

This problem requires us to join the review table with both the client and customer tables. We will then target the attributes we need to such as names and rating.

select b.first\_name || ' ' || b.last\_name, c.first\_name || ' ' || c.last\_name, rating

from review a

left outer join customer b

on a.customer\_id = b.customer\_id

left outer join client\_car\_owner c

on a.client\_id = c.client\_id;

1. The total number of times cars are rented, and the total cost of renting for a given year. User enter the year.

This question requires us to select the transactions and count the number of transactions that occurred. We can do this by extracting the date from the return of the transaction and seeing how many occurred if they match the user entered year

select count(vehicle\_id) from transactions

where client\_approval = 'yes' and extract(year from request\_end) < &year;



1. List employees’ name who is the manager of an office.

This problem will have us start at the manages office table and join it to the employees table so we can get the names of managers.

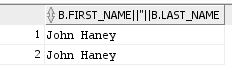
/\*n. List employees’ name who is the manager of an office.\*/

select b.first\_name || ' ' || b.last\_name

from manages\_office a

left outer join employee b

on a.manager\_id = b.employee\_id;



1. Search the database for a client. User input the data. Display client useful info.

This question requires us to select the client table and put a where constraint on the selection. We will then need to take a user’s input and check it against the where constraint.

select \* from client\_car\_owner

where first\_name || ' ' ||middle\_name || ' ' || last\_name like '%' || '&search' || '%' ;



1. Total salary and the number of employee in each office.

For this question we will have to join the three tables office, works at office, and employee together starting with office. We will then be able to count the employees and sum the salary while grouping by the offices.

select a.office\_Id, sum(c.annual\_sarary), count(b.employee\_id)

from office a

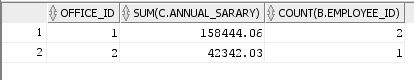
left outer join works\_at\_office b

on a.office\_id = b.office\_id

left outer join employee c

on b.employee\_id = c.employee\_id

group by a.office\_id;



1. List the certificates on an employee. User enter the name.

The question here is looking to list all certificates. Therefore we should start at the certificates table first and join the employee table. This will allow for multiple certificates to be shown for a single user. We will then need to use a where clause and a user input search   
  
select a.first\_name || ' ' || a.last\_name,

b.certificate\_name

from certificates b

left outer join employee a

on a.employee\_id = b.employee\_id

where a.first\_name || ' ' || a.middle\_name || ' ' || a.last\_name like '%' || '&search' || '%';

1. List of employees without a cell phone number.

This question is having us check for null values being present in the employee table. So, we will select the values from employee and use a where clause to check the cell\_phone attribute for a null. We will return if we find a null.

select \* from employee

where cell\_phone = null;

1. Display the total millage used in a given period. User input the start and end date.

I do not think with my current implementation I am able to create a query to handle this request. I do not have an attribute where the new milage is persisted. If I added a return attribute it could be possible to join several tables starting from the vehicle and getting the difference in milage from the returned vehicle.

1. Write a useful query that customer my need.

select a.\* from vehicle a

left outer join list\_vehicle B

ON a.car\_vin = b.car\_vin

left outer join client\_car\_owner c

on b.client\_id = c.client\_id

where c.city like '%' || '&search' || '%';

This query allows a user to search for a vehicle based on a city by joining the client table and using the user input in a where clause against the clients city attribute.