

You may use your COSC 640 project from the last semester or use the data from PROJECT.table_name. List of tables are:

1. office
2. office_timings
3. employee
4. office_day
5. employee_certificates
6. employee_certificate_dates
7. customer
8. car
9. accident
10. office_general_manager
11. client_customer_feedback
12. car_customer_rent
13. car_client_list
14. employee_customer
15. employee_client_manage
16. employee_car_list
17. customer_car_acceptance
18. office_employee
19. client

12. Create the following views:

a. **Employee_Info** with the following attributes:

Name of Employee	wp
Address	wp
Home Phone Number	
Cell Phone	
Zip code of Employee Address	
SSN	wp
Salary	wp
Birth Date	wp
Age of Employee	wp
Marital Status	wp
Name of Spouse	wp
Branch Name (Employee Works at)	wp
Last Degree	wp
Last Degree date	wp
Name of the Employee Manager	wp

b. **Employee_salary** with the following attributes:

First Name
Last Name
Address

SSN
Salary
Tax Deduction
Birth Date
Marital Status
Branch Employee Works at
Highest salary at his/her branch
Average salary at his/her branch

c. **Branch_Info** with the following attributes:

office ID
Name
Address
Main Phone Number
Fax Number
Regular Open Hours
Days Of Week Business is open
Manager Name
The Date Manager is Assigned

d. **Valued_Customers** with the following attributes:
(Customer rented cars more than three times last year)

Name
Age
Cell Phone
Address
Zip Code
Email
Driver's License Number
State the Driver's License is Issued
Total number of time this customer rented a car

e. **List_Of_Cars** with the following attributes:

Car ID
Car Plate No
Car State registration
Year
Owner name
Class (Compact, Economy, Luxury, Pickup, Van)
Number of time this car has been rented
Make
Color
Number of times this car has been rented
Number of times this car had accidents

Total cost of damage

f. **List_Of_Transaction** with the following attributes:

Trans ID	X
Number of days this car was used in this transaction	X
Total cast of this transaction	
Price/Day	
Price/Week	
Price/Month	
Location of the return	
Number of miles the car was used in this transaction	X
Mileage of car at the end of this transaction	-
Employee name who worked with this car	

g. **Statistics_by_Branch** year by year with the following attributes:

Office Id	
Owner name (Client)	
Customer name (Car renter)	
Year	(car table)
Total number of cars this client listed (in that year)	
Total number of transactions (in that year)	X
Total money client received (in that year)	

h. **Last_employee_Branch**: List of the last employee hired at each branch with the following attributes:

Name	
Office name	
Starting date	
Salary	
Age	

i. **Managers** with the following attributes:

Manager name	
Date s/he become a manager	
Title of the manager	X
How many employees works under the manager	

j, k: Create two more views that can be used by customers. (Make sure they are useful to the **customers**)

13. Create the following indices:

- a. Index on customer last and first name.
- b. Index on birthdate of customer.

} Duplicate

14. Create the following sequences:

- a. Create a sequence called **ID_generator** to be used for Customer ID.
Start with 1000
Generate only even numbers
Cache 40 numbers at a time
- b. Create a sequence to be used for the Transaction ID. (Make your own assumption).
- c. Create a sequence called **trackNumber**. Start with zero and increment by one.
- d. Create a table for inquiries. People request for rental cars. We need to keep track of the date, type of request, name of person, and their email. Use the **trackNumber** sequence as the primary key of this table. Insert five rows in to this table.

15. Subprograms & Packages with exception handling: (make sure your subprograms have appropriate exception handling)

a. Create a package called **EmpInsert_pkg** with the following subprograms:

1. Subprogram to insert data for a new employee. Remember the data will be inserted into several tables.
2. Subprogram to update the marital status of an employee.
3. Subprogram to update the last degree of an employee.
4. Subprogram to update the certificate of an employee.
5. Subprogram to change the manager of an employee. → office - 10

b. Create a package called **Insert_pkg** with the following subprograms:

1. Subprogram to insert a new customer.
2. Subprogram to insert a new car owner.
3. Subprogram to update a customer information.
4. Subprogram to update a new car owner information.
5. Subprogram to insert a new transaction.
6. Subprogram to insert a new accident.

c. Create a package called **Vehicle_pkg** with the following subprograms

- Subprogram called **Car_info** that accepts a vehicle ID and returns:
 1. Plate No
 2. State
 3. Year
 4. Current Mileage
 5. Class

6. Features
 7. Make
 8. Color
- Subprogram called **CarOwner** that accepts a SSN of an owner and print:
 1. Car ID(s)
 2. Plate No(s)
 3. SSN of the owner
 4. Name of owner
 5. Number of time each car has been rented.
 6. Number of time each car has been in an accident
 - Subprogram called **CarRented** that accepts a car ID and output the list of customers' name who rented this car.
 - Subprogram called **CarRented** that accepts the person SSN and returns the cars ID he/she rented last. (overload).
- d. Create a package called **Accident_pkg** with the following subprograms
1. Subprogram called **CarAccident** that accepts a car ID and returns the last accident info:
 1. Customer name ✓
 2. Customer phone ✓
 3. Customer address ✓
 4. Date ✓
 5. Time ✓
 6. Location ✓
 7. Extent of Damage ✓
 8. Cost of Damage
- Note: A car may/may not have an accident
2. Subprogram called **CustomerAccident** that will accept a person's SSN and return a Boolean value based on whether s/he have had an accident
- e. Create a subprogram called **BirthdaySub** that accepts today's date as default and writes the first name, last name, email, and address of a customer whose birth date (day and month only) is 15 or less days from today's date. Write them into a temp file (table called BirthdayFile). Every time we run this subprogram, the original data from this file will be erased.
- f. Create a subprogram to output branch ID's, name of branch's (office), manager's name, and the date the manager is hired.

16. Create the following triggers:

- g. Any deletion from employee file, trigger to write the record into an employee_History file.
- h. Any modification to the customer accounts, write who, date, and the nature of (old, new) modification into a Customer_Log file.
- i. Any accident must also be stored in the Accident_Log file. Name of employee, name of customer, and the date of accident.

17. All the errors generated by your run-time programs must be stored into an Error_log file. We would like to know the name of table, trigger name, the date, trigger is fired, and the trigger type.

18. Report writing

- a. Write a script to generate a report. The report will be triggered at log in. The data should be for all cars returned on the previous day. Your report heading should consist of Today's Date, with a title of "Car Rented for" Previous Date. Title each column appropriately. The report contains:
 - i. Owners Name
 - ii. Employee Name
 - iii. Customer SSN
 - iv. Customer Name
 - v. Car ID
 - vi. Rented Date
 - vii. Total Miles (End Miles – Start Miles)
 - viii. Total Amount of transaction (balance)

Consider:

- Calculate sum of Balance and Miles for each Branch and Employee. Title it appropriately.
- Include a Grand total on Balance and Miles for that day.
- Title and number on each page of your report
- Put the name of your script in the bottom of your report.
- Your lists should be sorted by Miles in ascending order.

Please display content of your tables with the tables' name.
For this semester I only need chapter 12 and after.

Note 1: Make sure your queries have numbers as described in this document. For example 18.a: I will not be able to grade your project if the numbers are not correct or missing.

Note 2: Make sure each part of your project has

- a- Question Number
- b- Query
- c- Output

Note 3: Make sure your report is readable. Insert spaces between and within your display data.

Good Luck