

PRINCESS NORA BINT ABDULRAHMAN UNIVERSITY

College of Computer and Information Sciences

Department of Information Systems



Yelo Car Rental (Management System)

Project Description:

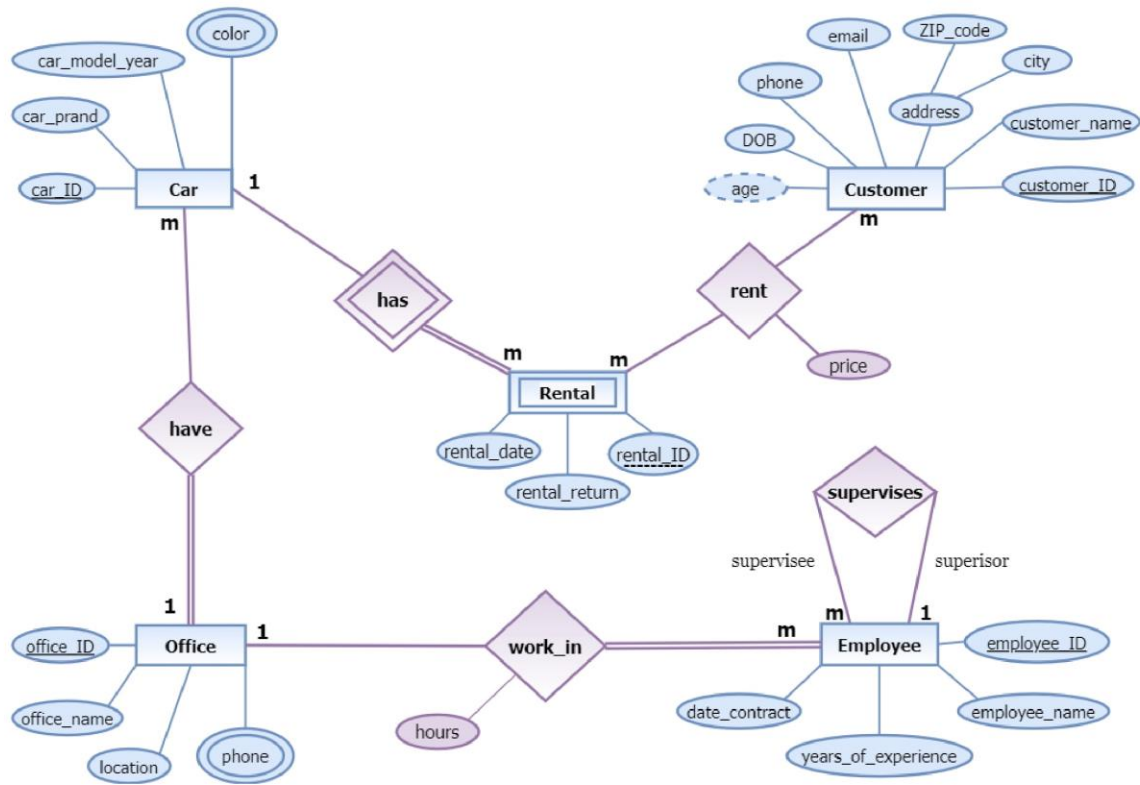
Yelo is a recognized brand with more than 80 branches in numerous cities and airports in the Kingdom of Saudi Arabia. Due to its vision and the excellence of its employees, they built car rental in Saudi Arabia. Their services are offered in 18 branches in airports and other branches located in the cities of the Kingdom of Saudi Arabia, with a fleet of more than 17,000 of the latest cars available in the market of various brands, and those who want to get the most Luxurious and finest types of cars for daily, monthly and annual rent can benefit from them. Customers can rent any car and complete the booking procedures electronically through the site and can choose any car, the duration of the contract and more. Beneficiaries of the company are tourists, traveling entrepreneurs, people who need a car temporarily and those who cannot afford buying a car.

Yelo rental company needs to establish a database in order to keep track of all data related to the company, such as employees, rentals, customers, cars and offices. The company has a number of rental offices in several cities. Each office identified by its office id, location, office name and many phone numbers, a rental office must have many cars and each car exists only in one office. Each car has a car id, car model year, car brand and a set of colors. A car has many rentals and each car is rented by many customers. If a car is rented, the price must be recorded. Rental has a rental date, rental id and rental return. If there is no car, a rental will be removed. A customer is identified by their id, name, phone, email, date, age and address (city, zip code). Each Employee is identified by their id, name, date of contract and year of experience. Many employees must work only in one office and all employees are supervised by one supervisor.

Entity Relationship Diagram (ERD):

12:57 2021/10/9

Untitled Diagram.drawio



<https://drive.google.com/drive/my-drive> 1/2

Relational Schema:

Office(office_ID, office_name, location)

Office_phone(office_ID , Phone)

FK1: office_ID references Office(office_ID)

Car(car_ID , office_ID, car_brand ,car_model_year)

FK1: office_ID references Office(office_ID)

Car_color(color , car_ID)

FK1: car_ID references Car(car_ID)

Customer(customer_ID,customer_name,city,ZIP_code,email,phone,DOB)

Rental(rental_ID , car_ID , rental_return,rental_date)

FK1:car_ID references Car(car_ID)

Rent(customer_ID, rental_ID , car_ID , price)

FK1: customer_ID references Customer(customer_ID)

FK2:rental_ID , car_ID references Rental(rental_ID , car_ID)

Employee(employee_ID

emp_ID,employee_name,year_of_experience,date_contract,hours ,office_ID)

FK1:emp_ID references Employee(employee_ID)

FK2: office_ID references Office(office_ID)

DDL Commands:

Submit a single hardcopy file (printout) of the COMMANDS and the resulted OUTPUT (snapshot) from the execution of the DB table's creation, data insertion and the data queries.

```
DROP TABLE Office CASCADE CONSTRAINT;  
CREATE TABLE Office (  
Office_ID CHAR(5) PRIMARY KEY,  
Office_Name VARCHAR2(40) NOT NULL,  
LOCATION VARCHAR2(30) DEFAULT 'RIYADH'  
);
```

```
DROP TABLE Office_phone CASCADE CONSTRAINT;  
CREATE TABLE Office_phone(  
Office_ID CHAR(5),  
Phone NUMBER(10) ,  
CONSTRAINT OfficePhone_PK PRIMARY KEY(Office_ID , Phone),  
CONSTRAINT OfficePhone_FK FOREIGN KEY(Office_ID) REFERENCES  
Office(Office_ID)  
);
```

```
DROP TABLE Car CASCADE CONSTRAINT;  
Create table Car (  
Car_ID char(5) primary key,  
Office_ID char(5),  
car_brand varchar2(20),  
Car_model_year NUMBER(4),  
Constraint Car_FK foreign key (Office_ID) References Office (office_ID)  
);
```

```
DROP TABLE Car_color CASCADE CONSTRAINT;  
Create table Car_color (  
Color char (5) ,  
Car_id char (5) ,  
CONSTRAINT CarColor_PK PRIMARY KEY(Color , Car_id),  
CONSTRAINT CarColor_FK FOREIGN KEY (Car_ID) REFERENCES car  
(Car_ID)  
)
```

DROP TABLE Customer CASCADE CONSTRAINT;

```
Create table Customer(  
Customer_ID char (5) primary key ,  
Customer_name varchar(55) not null,  
City varchar2(9),  
ZIP_code number(8),  
Email varchar2(19) NOT NULL ,  
Phone number(10),  
DOB date  
);
```

DROP TABLE Rental CASCADE CONSTRAINT;

```
CREATE TABLE Rental (  
rental_ID CHAR (5) ,  
car_ID CHAR (5) REFERENCES Car(car_ID),  
rental_return DATE ,  
rental_date DATE ,  
CONSTRAINT Rental_PK PRIMARY KEY(rental_ID ,car_ID)  
);
```

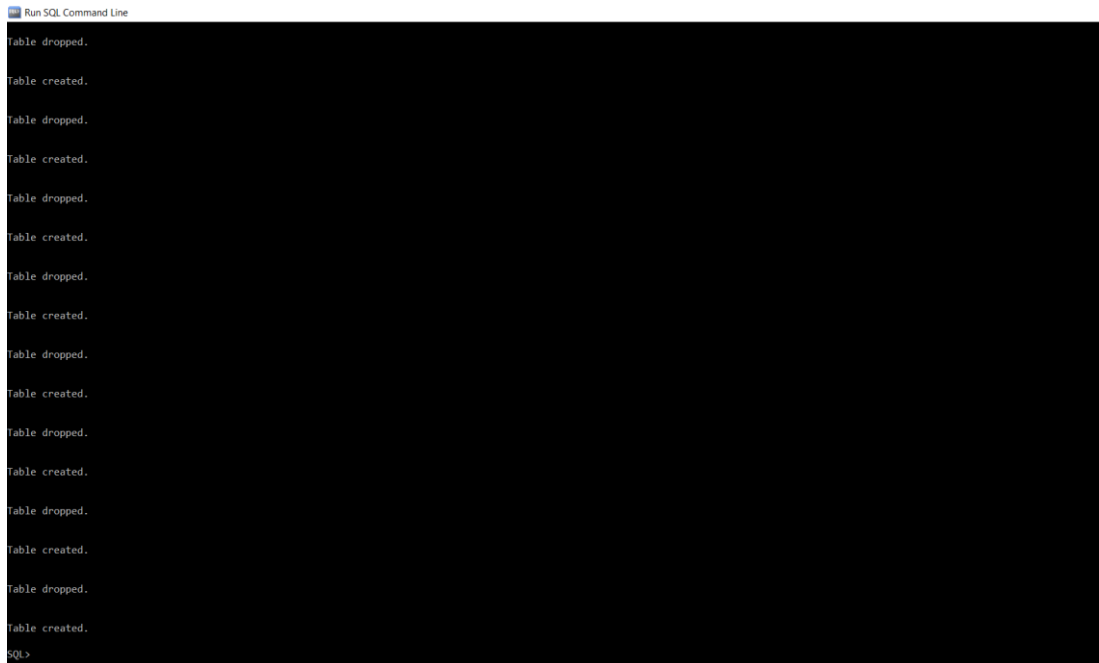
DROP TABLE Rent CASCADE CONSTRAINT;

```
Create table Rent (  
customer_id char (5),  
rental_ID char(5),  
car_ID char (5),  
Price number(10) CHECK (Price > 100 ),  
CONSTRAINT Rent_PK PRIMARY KEY(customer_id , Rental_id , Car_id),  
CONSTRAINT Rent_FK1 FOREIGN KEY (rental_ID , car_ID) REFERENCES Rental  
(rental_ID , car_ID),  
CONSTRAINT Rent_FK2 FOREIGN KEY (customer_id) REFERENCES  
Customer(Customer_ID)  
);
```

DROP TABLE Employee CASCADE CONSTRAINT;

```
CREATE TABLE Employee(  
employee_id CHAR(6) ,
```

```
emp_id CHAR(6) ,
employee_name VARCHAR2(15) ,
year_of_experience NUMBER(4) ,
date_contract DATE,
office_ID char (5),
hours NUMBER(4),
CONSTRAINT Employee_PK PRIMARY KEY(employee_id),
Constraint Employee_FK1 foreign key (emp_id) References Employee(employee_id),
Constraint Employee_FK2 foreign key (office_ID) References Office(office_ID)
);
```



The screenshot shows a 'Run SQL Command Line' window with a black background and white text. It displays a series of 16 messages alternating between 'Table dropped.' and 'Table created.', indicating a sequence of table operations. The messages are as follows:

- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.
- Table dropped.
- Table created.

The window ends with the prompt 'SQL>'.

INSERT COMMAND :

*Insert into office values('C1234' , 'yelo1' , 'Riyadh');
Insert into office values('C5678' , 'yelo2' , 'Jeddah');
Insert into office values('C9012' , 'yelo3' , 'Riyadh');*

*Insert into office_phone values('C1234' , 0556789435);
Insert into office_phone values('C5678' , 0556452349);
Insert into office_phone values('C9012' , 0598346702);*

*INSERT INTO car VALUES('F1234','C1234','Lincoln',2017);
INSERT INTO car VALUES('F5678','C5678','mercedes',2015);
INSERT INTO car VALUES('F9012','C9012','BMW',2019);*

*INSERT INTO Car_color VALUES('white', 'F1234');
INSERT INTO Car_color VALUES('Black', 'F5678');
INSERT INTO Car_color VALUES('Blue', 'F9012');*

*INSERT INTO customer
VALUES ('rc032', 'Ahmed', 'Jeddah', '11432567', 'ahmedali@gmail.com', '0532456798',
TO_DATE('11-01-1999','dd-mm-yyyy'));
INSERT INTO customer
VALUES ('rc044' , 'Ali', 'Riyadh', '11432997', 'ali2002x@gmail.com', '0532400792',
TO_DATE('19-02-1988','dd-mm-yyyy'));
INSERT INTO customer
VALUES ('rc009' , 'Rana', 'Dammam', '11430097', 'ranaomar@gmail.com',
'0532411654', TO_DATE('29-12-2000','dd-mm-yyyy'));*

*INSERT INTO Rental
VALUES ('MN990', 'F1234' , TO_DATE('01-01-2020','dd-mm-yyyy') , TO_DATE('02-01-2021','dd-mm-yyyy')) ;
INSERT INTO Rental
VALUES ('BH467', 'F5678' , TO_DATE('03-02-2021','dd-mm-yyyy') , TO_DATE('22-02-2021','dd-mm-yyyy'));
INSERT INTO Rental
VALUES ('GC473', 'F9012' , TO_DATE('02-12-2002','dd-mm-yyyy') , TO_DATE('05-02-2020','dd-mm-yyyy'));*

*Insert into rent values ('rc032','MN990','F1234',12345);
Insert into rent values ('rc044','BH467','F5678',10345);
Insert into rent values ('rc009','GC473','F9012',10245);*

*Insert into Employee values('ABC12' , 'ABC12' , 'bader' ,5, TO_DATE('01-11-2015','dd-mm-yyyy') , 'C1234' ,15);
Insert into Employee values('ABC13' , 'ABC13' , 'Ahmad' ,6, TO_DATE('01-06-2017','dd-mm-yyyy') , 'C9012' ,17);
Insert into Employee values('ABC14' , 'ABC14' , 'mona' , 5 , TO_DATE('06-01-2021','dd-mm-yyyy') , 'C9012' ,15);*


```
Run SQL Command Line
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
```

```
Run SQL Command Line
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
1 row created.
SQL> _
```

DML Commands:

-- part 1

```
SELECT * from office WHERE location= 'Riyadh' ;
```

```
UPDATE Customer SET city = 'Riyadh' WHERE customer_Id = 'rc032' ;
```

```
SELECT employee_id , employee_name  
FROM Employee  
WHERE date_contract>TO_DATE('01-01-2017','dd-mm-yyyy');
```

--part 2

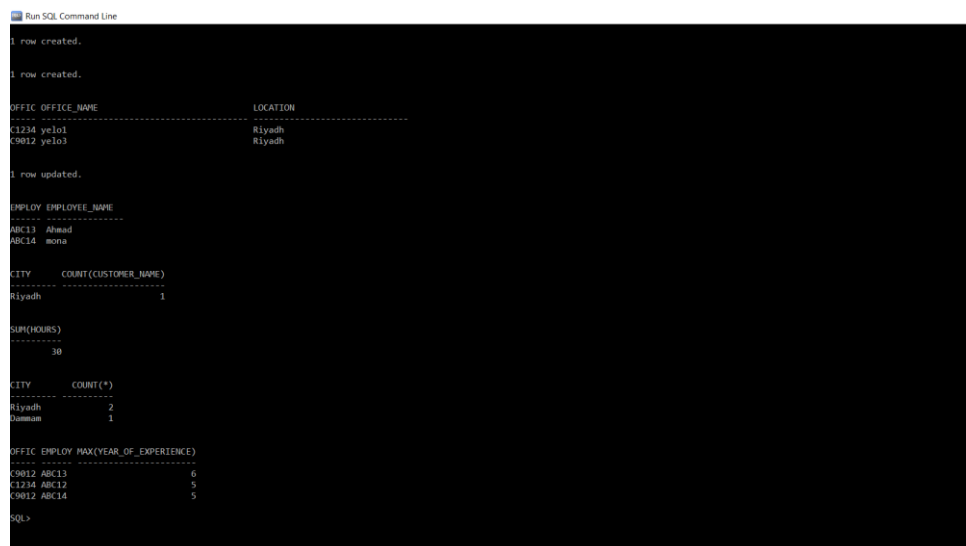
```
SELECT city, count (customer_name) from customer WHERE ZIP_code = '11432567'  
group by city ;
```

```
SELECT sum(hours) from Employee where hours < 17 ;
```

```
SELECT city , count(*) from customer group by city ;
```

--To select employees with more than 3 years of experience in each office

```
SELECT office_ID,employee_id, MAX (year_of_experience) FROM Employee  
GROUP BY office_ID, employee_id  
HAVING MAX(year_of_experience) > 3;
```



```
Run SQL Command Line  
1 row created.  
1 row created.  
OFFIC OFFICE_NAME LOCATION  
-----  
11234 ye101 Riyadh  
19012 ye101 Riyadh  
1 row updated.  
EMPLOY EMPLOYEE_NAME  
-----  
ABC13 Ahmad  
ABC14 Mona  
CITY COUNT(CUSTOMER_NAME)  
-----  
Riyadh 1  
SUM(HOURS)  
-----  
30  
CITY COUNT(*)  
-----  
Riyadh 2  
Dammam 1  
OFFIC EMPLOY MAX(YEAR_OF_EXPERIENCE)  
-----  
19012 ABC13 6  
11234 ABC12 5  
19012 ABC14 5  
SQL>
```