Assignment 10 Ras for all queries:

**-- Query 1:**

SELECT Customer\_Info

FROM Customers c

WHERE EXISTS (

SELECT 1

FROM Bookings b

WHERE b.Customer\_Info = c.Customer\_Info

);

**--RA for query 1:**

π Customer\_Info (σ EXISTS (π 1 (Bookings ⨝ Customer\_Info = Customer\_Info Customers c)))

**-- Query 2:**

SELECT Car\_VIN, Performance

FROM Cars

UNION

SELECT VehicleRented, BookingDate

FROM Bookings;

**--RA for query 2:**

π Car\_VIN, Performance ((π Car\_VIN, Performance Cars) ⋃ (π VehicleRented, BookingDate Bookings))

**-- Query 3:**

SELECT Department\_Name, Cars\_Types

FROM Departments d

WHERE NOT EXISTS (

SELECT 1

FROM Cars c

WHERE c.Departments\_Car\_Types = d.Cars\_Types

AND c.CarsAvailable = 'Yes'

);

**--RA for query 3:**

π Department\_Name, Cars\_Types (σ NOT EXISTS (π 1 (σ Departments\_Car\_Types = Cars\_Types AND CarsAvailable = 'Yes' Cars c) Departments d))

-- Query 4:

SELECT Departments\_Car\_Types, COUNT(\*) AS NumBookings

FROM Bookings

JOIN Cars ON Bookings.VehicleRented = Cars.Car\_VIN

GROUP BY Departments\_Car\_Types;

**--RA for query 4:**

π Departments\_Car\_Types, COUNT(\*) AS NumBookings (Bookings ⨝ VehicleRented = Car\_VIN Cars) (σ Departments\_Car\_Types = Departments\_Car\_Types (γ Departments\_Car\_Types, COUNT(\*) Bookings))

-- Query 5:

SELECT Departments\_Car\_Types, COUNT(\*) AS NumBookings

FROM Bookings

JOIN Cars ON Bookings.VehicleRented = Cars.Car\_VIN

GROUP BY Departments\_Car\_Types

HAVING COUNT(\*) > 10;

**--RA for query 5:**

π Departments\_Car\_Types, COUNT(\*) AS NumBookings (Bookings ⨝ VehicleRented = Car\_VIN Cars) (σ Departments\_Car\_Types = Departments\_Car\_Types (γ Departments\_Car\_Types, COUNT(\*) Bookings) (σ COUNT(\*) > 10))

-- Query 6:

SELECT PickUpInfo, DropOffInfo, StoreLocation, c.Customer\_Info

FROM Locations l

JOIN Customers c ON l.Customer\_Info = c.Customer\_Info

WHERE c.Rental\_Info IS NOT NULL

UNION

SELECT PickUpInfo, DropOffInfo, StoreLocation, c.Customer\_Info

FROM Locations l

JOIN Transactions t ON l.Customer\_Info = t.Customer\_Info

JOIN Customers c ON t.Customer\_Info = c.Customer\_Info

WHERE t.TotalCost > 1000;

**--RA for query 6:**

π PickUpInfo, DropOffInfo, StoreLocation, Customer\_Info ((Locations ⨝ Customer\_Info = Customer\_Info Customers c) ⋃ (Locations ⨝ Customer\_Info = Customer\_Info Transactions t ⨝ TotalCost > 1000 Customers c))

-- Query 7:

SELECT d.Department\_Name, d.Cars\_Types, COUNT(e.Employee\_ID) AS NumEmployees

FROM Departments d

LEFT JOIN Employees e ON d.Cars\_Types = e.Departments\_Car\_Types

GROUP BY d.Department\_Name, d.Cars\_Types

HAVING COUNT(e.Employee\_ID) > 0;

**--RA for query 7:**

π Department\_Name, Cars\_Types, COUNT(Employee\_ID) AS NumEmployees ((γ Department\_Name, Cars\_Types, COUNT(Employee\_ID) Employees e) ⨝ Cars\_Types = Departments\_Car\_Types (Departments d)) (σ COUNT(Employee\_ID) > 0)

-- Query 8:

SELECT Department\_Name

FROM Departments;

**--RA for query 8:**

π Department\_Name Departments d

-- Query 9:

UPDATE Cars

SET Performance = 'High'

WHERE Car\_VIN = '#F3E4NVE';

**--RA for query 9:**

Cars ← Cars ⨝ Car\_VIN = '#F3E4NVE'

Cars ← Cars {\_ Performance ← 'High'}

-- Query 10:

DELETE FROM Customers

WHERE Customer\_Info = 'John Doe: 433 Wish Wash street';

**--RA for query 10:**

Customers ← Customers - Customer\_Info = 'John Doe: 433 Wish Wash street'

-- Query 11:

SELECT AVG(TotalCost)

FROM Transactions;

**--RA for query 11:**

π AVG(TotalCost) (Transactions)

-- Query 12:

SELECT PickUpInfo, DropOffInfo

FROM Locations

WHERE Customer\_Info = 'customer123';

**--RA for query 12:**

π PickUpInfo, DropOffInfo (σ Customer\_Info = 'customer123' Locations)

-- Query 13:

SELECT DISTINCT Departments\_Car\_Types

FROM Employees

WHERE Employee\_ID IN (SELECT DISTINCT Employee\_ID FROM Transactions);

**--RA for query 13:**

π DISTINCT Departments\_Car\_Types (σ Employee\_ID IN (π DISTINCT Employee\_ID (Transactions)) Employees)

-- Query 14:

SELECT BookingDate, BookingTime, VehicleRented

FROM Bookings

WHERE BookingDate >= '2022-01-01'

ORDER BY BookingDate DESC, BookingTime DESC;

**--RA for query 14:**

π BookingDate, BookingTime, VehicleRented (σ BookingDate >= '2022-01-01' (γ BookingDate, BookingTime, VehicleRented (Bookings)))

-- Query 15:

SELECT Customer\_Info, SUM(Revenue) AS TotalRevenue

FROM Reports

GROUP BY Customer\_Info

HAVING TotalRevenue > 1000;

**--RA for query 15:**

π Customer\_Info, SUM(Revenue) AS TotalRevenue (γ Customer\_Info, SUM(Revenue) Reports) (σ SUM(Revenue) > 1000)